Proximal Multitask Learning Over Networks With Sparsity-Inducing Coregularization

Roula Nassif, Cédric Richard, *Senior Member, IEEE*, André Ferrari, *Member, IEEE*, and Ali H. Sayed, *Fellow, IEEE*

***Abstract*—In this work, we consider multitask learning problems where clusters of nodes are interested in estimating their own parameter vector. Cooperation among clusters is beneficial when the optimal models of adjacent clusters have a good number of similar entries. We propose a fully distributed algorithm for solving this problem. The approach relies on minimizing a global mean-square error criterion regularized by nondifferentiable terms to promote cooperation among neighboring clusters. A general diffusion forward–backward splitting strategy is introduced. Then, it is specialized to the case of sparsity promoting regularizers. A closed-form expression for the proximal operator of a weighted sum of ℓ_1-norms is derived to achieve higher efficiency. We also provide conditions on the step-sizes that ensure convergence of the algorithm in the mean and mean-square error sense. Simulations are conducted to illustrate the effectiveness of the strategy.**

***Index Terms*—Distributed processing, multitask networks, diffusion LMS, forward-backward splitting approach, sparsity-inducing coregularizers, adaptive regularization factors.**

I. Introduction

We consider the problem of distributed adaptive learning over networks to simultaneously estimate several parameter vectors from noisy measurements using in-network processing. Depending on the number of parameter vectors to estimate, we distinguish between single-task networks and multitask networks. In a single-task scenario, the entire network aims to estimate a common parameter vector for all nodes. The nodes are allowed to exchange information with their neighbors to improve their own estimates. Then, the estimates are combined in order to achieve the solution of the problem. Different cooperation rules have been proposed and studied in the literature [1]–[17]. Diffusion strategies [4]–[11] are particularly attractive since they are scalable, robust, and enable continuous learning and adaptation in response to concept drifts. They have also been shown to outperform consensus implementations over adaptive networks when constant step-sizes are employed to enable continuous adaptation [4], [5], [18].

Manuscript received September 4, 2015; revised April 4, 2016 and May 26, 2016; accepted July 26, 2016. Date of publication August 18, 2016; date of current version October 6, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Brian Sadler. The work of C. Richard and A. Ferrari was supported in part by ANR and DGA under Grant ANR-13-ASTR-0030 (ODISSEE project). The work of A. H. Sayed was supported in part by NSF under Grants CIF-1524250 and ECCS-1407712.

R. Nassif and A. Ferrari are with the Université Côte d'Azur, OCA, CNRS 06000 Nice, France (e-mail: roula.nassif@oca.eu; andre.ferrari@unice.fr).

C. Richard is with the Université Côte d'Azur, OCA, CNRS 06000 Nice, France and on leave from INRIA Sophia Antipolis—Méditerranée, Valbonne 06902, France (e-mail: cedric.richard@unice.fr).

A. H. Sayed is with the Department of Electrical Engineering, University of California, Los Angeles, CA 90095 USA (e-mail: sayed@ee.ucla.edu).

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

Digital Object Identifier 10.1109/TSP.2016.2601282

In this work, we are interested in distributed estimation over multitask networks: nodes are grouped into clusters, and each cluster is interested in estimating its own parameter vector (i.e., each cluster has its own task). Although clusters may generally have distinct though related tasks to perform, the nodes may still be able to capitalize on inductive transfer between clusters to improve their estimation accuracy. Such situations occur when the tasks of nearby clusters are correlated, which happens, for instance, in monitoring applications where agents in a network need to track multiple targets moving along correlated trajectories. Multitask diffusion estimation problems of this type have been addressed before in two main ways.

In a first scenario, no prior information on possible relationships between tasks is assumed and nodes do not know which other nodes share the same task. In this case, all nodes cooperate with each other as dictated by the network topology. It was shown in [11] that the diffusion iterates will end up converging to a Pareto optimal solution corresponding to a multi-objective optimization problem. If, on the other hand, the only available information is that clusters may exist in the network (but their structures are not known), then extended diffusion strategies can be developed [19]–[22] for setting the combination weights in an online manner in order to enable automatic network clustering and, subsequently, to limit cooperation between clustered agents. In a second scenario, it is assumed that nodes know which clusters they belong to. In this case, multitask diffusion strategies can be derived by exploiting this information on the relationships between tasks. A couple of useful works have addressed variations of this scenario. For example, in [23], a diffusion LMS strategy estimates spatially-varying parameters by exploiting the spatio-temporal correlations of the measurements at neighboring nodes. In [24], it is assumed that there are three types of parameters: parameters of global interest to all nodes in the network, parameters of common interest to a subset of nodes, and a collection of parameters of local interest. A diffusion strategy was developed to perform estimation under these conditions. A similar work dealing with incremental strategies instead of diffusion strategies appears in [25]. Likewise, in the works [26], [27], distributed algorithms are developed to estimate node-specific parameter vectors that lie in a common latent signal subspace. In another work [28], the parameter space is decomposed into two orthogonal subspaces, with one of the subspaces being common to all nodes. There is yet another useful way to exploit and model relationships among tasks, namely, to formulate optimization problems with appropriate co-regularizers between nodes. The strategy developed in [29] adds squared ℓ_2-norm co-regularizers to the mean-square-error criterion in order to promote smoothness of the graph signal. Its convergence behavior is studied over asynchronous networks in [30].

In some applications, however, such as cognitive radio [24], [28] and remote sensing [29], it may happen that the optimum parameter vectors of neighboring clusters have a large number

of similar entries and a relatively small number of distinct components. In this work, we build on the second scenario where the composition of the clusters is assumed to be known and where nodes know which cluster they belong to. It is then advantageous to develop distributed strategies that involve cooperation among adjacent clusters in order to promote and exploit such similarity. Although the current problem seems to be related to the problem studied in [29], it should be noted that the differentiable regularizers used in [29] are not effective when sparsity promoting regularization is required. Moreover, when neighboring nodes belonging to different clusters are aware of the indices of common and distinct entries, and when these indices are fixed over time, one may appeal to the multitask diffusion strategies developed in [24], [28]. However, in the current work, we are interested in solutions that are able to handle situations where the only available information is that the optimum parameter vectors of neighboring clusters have a large number of similar entries. A multitask diffusion algorithm with ℓ_1-norm co-regularizers is proposed in [31] to address this problem leading to a subgradient descent method distributed among the agents. The aim of this work is to introduce a more general approach for solving such convex but *non-differentiable* problems by employing instead a diffusion forward-backward splitting strategy based on the proximal projection operator. Before proceeding, we recall the forward-backward splitting approach in a single-agent deterministic environment [32]–[34].

Consider the problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^M} \; f(\boldsymbol{x}) + g(\boldsymbol{x}) \tag{1}$$

with f a real-valued differentiable convex function whose gradient is β-Lipschitz continuous, and g a real-valued convex function. The *proximal gradient* method or the *forward-backward splitting* approach for solving (1) is given by the iteration [32], [34]:

$$\boldsymbol{x}(i+1) = \text{prox}_{\mu g}(\boldsymbol{x}(i) - \mu\nabla f(\boldsymbol{x}(i))), \tag{2}$$

where μ is a constant step-size chosen such that $\mu \in (0, 2\beta^{-1}]$ to ensure convergence to the minimizer of (1). The gradient-descent step is the forward step (explicit step) and the proximal step is the backward step (implicit step). The proximal operator of $\mu g(\boldsymbol{x})$ at a given point $\boldsymbol{v} \in \mathbb{R}^M$ is a real-valued map given by [34]:

$$\text{prox}_{\mu g}(\boldsymbol{v}) = \underset{\boldsymbol{x}\in\mathbb{R}^M}{\text{argmin}} \; g(\boldsymbol{x}) + \frac{1}{2\mu}\|\boldsymbol{x}-\boldsymbol{v}\|^2. \tag{3}$$

Since the proximal operator needs to be calculated at each iteration in (2), it is important to have a closed form expression for evaluating it. In this work, we derive a *multitask diffusion adaptation strategy* where each node employs this approach for minimizing a cost function with sparsity based co-regularizers. Instead of using iterative algorithms for evaluating the proximal operator of a weighted sum of ℓ_1-norms at each iteration [33], we shall instead derive a closed form expression that allows us to compute it exactly. We shall also examine under which conditions on the step-sizes the proposed multitask diffusion strategy is mean and mean-square stable. Simulations are conducted to show the effectiveness of the proposed strategy. An adaptive rule to guarantee an appropriate cooperation between clusters is also introduced.

Notation: In what follows, normal font letters denote scalars, boldface lowercase letters denote column vectors, and boldface uppercase letters denote matrices. We use the symbol $(\cdot)^\top$ to denote matrix transpose, the symbol $(\cdot)^{-1}$ to denote matrix inverse, and the symbol $\text{Tr}(\cdot)$ to denote the trace operator. The operator $\text{col}\{\cdot\}$ stacks the column vectors entries on top of each other. The symbol $\otimes$ denotes the Kronecker product operation. The identity matrix of size $N \times N$ is denoted by $\boldsymbol{I}_N$. The $N \times M$ matrices of zeros and ones are denoted by $\boldsymbol{0}_{N\times M}$ and $\boldsymbol{1}_{N\times M}$, respectively. The set $\mathcal{N}_k$ denotes the neighbors of node k including k. The set $\mathcal{N}_k^-$ denotes the neighbors of node k excluding k. Finally, $\mathcal{C}_i$ denotes the set of nodes in the i-th cluster and $\mathcal{C}(k)$ denotes the cluster to which node k belongs.

II. Multitask Diffusion LMS With Forward-Backward Splitting

A. Network Model and Problem Formulation

We consider a network of N nodes grouped into Q connected clusters in a predefined topology. Clusters are assumed to be connected, i.e., there exists a path between any pair of nodes in the cluster. At every time instant i, every node k has access to a zero-mean measurement $d_k(i)$ and a zero-mean $M \times 1$ regression vector $\boldsymbol{x}_k(i)$ with positive covariance matrix $\boldsymbol{R}_{\boldsymbol{x},k} = \mathbb{E}\{\boldsymbol{x}_k(i)\,\boldsymbol{x}_k^\top(i)\} > 0$. We assume the data to be related via the linear model:

$$d_k(i) = \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k^o + z_k(i), \tag{4}$$

where $\boldsymbol{w}_k^o$ is the $M \times 1$ unknown parameter vector, also called task, we wish to estimate at node k, and $z_k(i)$ is a zero-mean measurement noise of variance $\sigma_{z,k}^2$, independent of $\boldsymbol{x}_\ell(j)$ for all ℓ and j, and independent of $z_\ell(j)$ for $\ell \neq k$ or $i \neq j$. We assume that all nodes in a cluster are interested in estimating the same parameter vector, namely, $\boldsymbol{w}_k^o = \boldsymbol{w}_{\mathcal{C}_q}^o$ whenever k belongs to cluster $\mathcal{C}_q$. However, if cluster $\mathcal{C}_p$ is connected to cluster $\mathcal{C}_q$, that is, there exists at least one link connecting a node from $\mathcal{C}_p$ to a node from $\mathcal{C}_q$, vectors $\boldsymbol{w}_{\mathcal{C}_p}^o$ and $\boldsymbol{w}_{\mathcal{C}_q}^o$ are assumed to have a large number of similar entries and only a relatively small number of distinct entries. Cooperation across these clusters can therefore be beneficial to infer $\boldsymbol{w}_{\mathcal{C}_p}^o$ and $\boldsymbol{w}_{\mathcal{C}_q}^o$.

Considerable interest has been shown in the literature about estimating an optimum parameter vector $\boldsymbol{w}^o$ subject to the property of being sparse. Motivated by the well-known LASSO problem [35] and compressed sensing framework [36], different techniques for sparse adaptation have been proposed. For example, the authors in [37], [38] promote sparsity within an LMS framework by considering regularizers based on the ℓ_1-norm, reweighed ℓ_1-norm, and convex approximation of ℓ_0-norm. In [39], projections of streaming data onto hyperslabs and weighted ℓ_1 balls are used instead of minimizing regularized costs recursively. Proximal forward-backward splitting is considered in an adaptive scenario in [40]. In the context of distributed learning over *single-task* networks, diffusion LMS methods promoting sparsity have been proposed. Sparse diffusion LMS strategies using subgradient methods are proposed in [41]–[43] and using proximal methods are proposed in [44]–[46]. In [47], the authors employ projection-based techniques [39] to derive distributed diffusion algorithms promoting sparsity, and in [48] a diffusion LMS algorithm for estimating an s-sparse vector is proposed based on adaptive greedy techniques similar to [49]. These techniques estimate the positions of non-zero entries in the target vector, and then perform

computations on this subset. More generally, diffusion strategies based on proximal gradient for minimizing general costs (not necessarily mean-square error costs) and subject to a broader class of constraints on the parameter vector to be estimated (including sparsity) are derived in [46].

Our purpose is to derive an adaptive learning algorithm over *multitask* networks where optimum parameter vectors of neighboring clusters share a large number of similar entries and a relatively small number of distinct entries. Consider nodes k and ℓ of neighboring clusters $\mathcal{C}(k)$ and $\mathcal{C}(\ell)$, and let $\boldsymbol{\delta}_{k,\ell}$ denote the vector difference $\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}$. Promoting the sparsity of $\boldsymbol{\delta}_{k,\ell}$ can be performed by considering the pseudo ℓ_0-norm of $\boldsymbol{\delta}_{k,\ell}$ as it denotes the number of nonzero entries. Nevertheless, $\|\boldsymbol{\delta}_{k,\ell}\|_0$ is a non-convex co-regularizer that leads to computational challenges. A common alternative is to use the ℓ_1-norm regularization function defined as

$$f_1(\boldsymbol{\delta}_{k,\ell}) = \|\boldsymbol{\delta}_{k,\ell}\|_1 = \sum_{m=1}^{M} |[\boldsymbol{\delta}_{k,\ell}]_m|. \tag{5}$$

Since the ℓ_1-norm uniformly shrinks all the components of a vector and does not distinguish between zero and non-zero entries [50], it is common in the sparse adaptive filtering framework [37], [39]–[42], [44], [45], [47], [51] to consider a weighted formulation of the ℓ_1-norm. Weighted ℓ_1-norm was designed to reduce the bias induced by the ℓ_1-norm and enhance the penalization of the non-zero entries of a vector [39], [50], [52]. Given the weight vector $\boldsymbol{\alpha}_{k\ell} = [\alpha_{k\ell}^1, \ldots, \alpha_{k\ell}^M]^\top$, with $\alpha_{k\ell}^m > 0$ for all m, the weighted ℓ_1-norm is defined as:

$$f_2(\boldsymbol{\delta}_{k,\ell}) = \sum_{m=1}^{M} \alpha_{k\ell}^m \big|[\boldsymbol{\delta}_{k,\ell}]_m\big|. \tag{6}$$

The weights are usually chosen as:

$$\alpha_{k\ell}^m = \frac{1}{\epsilon + |[\boldsymbol{\delta}_{k,\ell}^o]_m|}, \quad m = 1, \ldots, M, \tag{7}$$

where $\boldsymbol{\delta}_{k,\ell}^o = \boldsymbol{w}_k^o - \boldsymbol{w}_\ell^o$. Since the optimum parameter vectors are not available beforehand, we set

$$\alpha_{k\ell}^m(i) = \frac{1}{\epsilon + |[\boldsymbol{\delta}_{k,\ell}(i-1)]_m|}, \quad m = 1, \ldots, M, \tag{8}$$

at each iteration i, where ϵ is a small constant to prevent the denominator from vanishing and $\boldsymbol{\delta}_{k,\ell}(i)$ is the estimate of $\boldsymbol{\delta}_{k,\ell}^o$ at nodes k and ℓ and iteration i. This technique, also known as reweighted ℓ_1 minimization [50], is performed at each iteration of the stochastic optimization process. It has been shown in [50] that, by minimizing (6) with the weights (8), one minimizes the log-sum penalty function, $\sum_{m=1}^{M} \log(\epsilon + |[\boldsymbol{\delta}_{k,\ell}]_m|)$, which acts like the ℓ_0-norm by allowing a relatively large penalty to be placed on small nonzero coefficients and more strongly encourages them to be set to zero. In the sequel, we shall use $f(\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)})$ to refer to the unweighted or reweighted ℓ_1-norm promoting the sparsity of $\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}$.

It is sufficient for this work to derive a distributed learning algorithm of the LMS type. We shall therefore assume that the local cost function $J_k(\boldsymbol{w}_{\mathcal{C}(k)})$ at node k is the mean-square error criterion defined by:

$$J_k(\boldsymbol{w}_{\mathcal{C}(k)}) = \mathbb{E}\big\{|d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_{\mathcal{C}(k)}|^2\big\}. \tag{9}$$

Combining local mean-square-error cost functions and regularization functions, the cooperative multitask estimation problem is formulated as the problem of seeking a fully distributed solution for solving:

$$\min_{\boldsymbol{w}_{\mathcal{C}_1}, \ldots, \boldsymbol{w}_{\mathcal{C}_Q}} \overline{J}^{\,\text{glob}}(\boldsymbol{w}_{\mathcal{C}_1}, \ldots, \boldsymbol{w}_{\mathcal{C}_Q}) = \min_{\boldsymbol{w}_{\mathcal{C}_1}, \ldots, \boldsymbol{w}_{\mathcal{C}_Q}} \sum_{k=1}^{N} J_k(\boldsymbol{w}_{\mathcal{C}(k)}) + \eta \sum_{k=1}^{N} \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} \rho_{k\ell} f(\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}), \tag{10}$$

where $\eta > 0$ is the regularization strength used to enforce sparsity. It ensures a tradeoff between fidelity to the measurements and prior information on the relationships between tasks. The weights $\rho_{k\ell} \geq 0$ aim at locally adjusting the regularization strength. The notation $\mathcal{N}_k \setminus \mathcal{C}(k)$ denotes the set of neighboring nodes of k that are not in the same cluster as k.

Note that the regularization terms (5) and (6) are symmetric with respect to the weight vectors $\boldsymbol{w}_{\mathcal{C}(k)}$ and $\boldsymbol{w}_{\mathcal{C}(\ell)}$, that is, $f(\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}) = f(\boldsymbol{w}_{\mathcal{C}(\ell)} - \boldsymbol{w}_{\mathcal{C}(k)})$. Due to the summation over the N nodes, each term $f(\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)})$ can be viewed as weighted by $\frac{(\rho_{k\ell} + \rho_{\ell k})}{2}$ in (10). Problem (10) can therefore be written in an alternative way as:

$$\min_{\boldsymbol{w}_{\mathcal{C}_1}, \ldots, \boldsymbol{w}_{\mathcal{C}_Q}} \overline{J}^{\,\text{glob}}(\boldsymbol{w}_{\mathcal{C}_1}, \ldots, \boldsymbol{w}_{\mathcal{C}_Q}) = \min_{\boldsymbol{w}_{\mathcal{C}_1}, \ldots, \boldsymbol{w}_{\mathcal{C}_Q}} \sum_{k=1}^{N} J_k(\boldsymbol{w}_{\mathcal{C}(k)}) + \eta \sum_{k=1}^{N} \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} p_{k\ell} f(\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}) \tag{11}$$

where the factors $\{p_{k\ell}\}$ are symmetric, i.e., $p_{k\ell} = p_{\ell k}$, and are given by:

$$p_{k\ell} \triangleq \frac{(\rho_{k\ell} + \rho_{\ell k})}{2}. \tag{12}$$

One way to avoid symmetrical regularization is to consider an alternative problem formulation defined in terms of Q Nash equilibrium problems as done in [29] with ℓ_2-norm co-regularizers. In this paper, we shall focus on problem (10).

Let us consider the variable $\boldsymbol{w}_{\mathcal{C}_j}$ of the j-th cluster. Given $\boldsymbol{w}_{\mathcal{C}(\ell)}$ with $\ell \in \mathcal{N}_k \setminus \mathcal{C}_j$ and $k \in \mathcal{C}_j$, the subdifferential of $\overline{J}^{\,\text{glob}}(\boldsymbol{w}_{\mathcal{C}_1}, \ldots, \boldsymbol{w}_{\mathcal{C}_Q})$ in (11) with respect to $\boldsymbol{w}_{\mathcal{C}_j}$ is given by:

$$\partial_{\boldsymbol{w}_{\mathcal{C}_j}} \overline{J}^{\,\text{glob}}(\boldsymbol{w}_{\mathcal{C}_1}, \ldots, \boldsymbol{w}_{\mathcal{C}_Q}) = \sum_{k \in \mathcal{C}_j} \nabla_{\boldsymbol{w}_{\mathcal{C}_j}} J_k(\boldsymbol{w}_{\mathcal{C}_j}) + 2\eta \sum_{k \in \mathcal{C}_j} \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}_j} p_{k\ell} \partial_{\boldsymbol{w}_{\mathcal{C}_j}} f(\boldsymbol{w}_{\mathcal{C}_j} - \boldsymbol{w}_{\mathcal{C}(\ell)}), \tag{13}$$

where we used the fact that the regularization terms (5), (6), and the regularization factors $\{p_{k\ell}\}$ are symmetric. Since we are interested in a distributed strategy for solving (10) that relies only on in-network processing, we associate the following regularized problem $(\mathcal{P}_j)$ with each cluster $\mathcal{C}_j$:

$$\min_{\boldsymbol{w}_{\mathcal{C}_j}} \overline{J}_{\mathcal{C}_j}(\boldsymbol{w}_{\mathcal{C}_j}) = \min_{\boldsymbol{w}_{\mathcal{C}_j}} \sum_{k \in \mathcal{C}_j} \mathbb{E}\big\{|d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_{\mathcal{C}_j}|^2\big\} + 2\eta \sum_{k \in \mathcal{C}_j} \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}_j} p_{k\ell} f(\boldsymbol{w}_{\mathcal{C}_j} - \boldsymbol{w}_{\mathcal{C}(\ell)}). \tag{14}$$

Given $\boldsymbol{w}_{\mathcal{C}(\ell)}$ with $\ell \in \mathcal{N}_k \setminus \mathcal{C}_j$, note that the costs in problems (10) and (14) have the same subdifferential relative to $\boldsymbol{w}_{\mathcal{C}_j}$. In order that each node can solve the problem in an autonomous and adaptive manner using only local interactions, we shall derive a distributed iterative algorithm for solving (10) by considering (14) since both costs have the same subdifferential information.

B. Problem Relaxation

We shall now extend the derivations in [7], [9], [53] to handle multitask estimation problems with nondifferentiable functions. In the sequel, we write $\boldsymbol{w}_k$ instead of $\boldsymbol{w}_{\mathcal{C}(k)}$ for simplicity of notation. First, we associate with each node k an unregularized local cost function $J_k^{\rm loc}(\cdot)$ and a regularized local cost function $\overline{J}_k^{\rm loc}(\cdot)$ of the form:

$$J_k^{\rm loc}(\boldsymbol{w}_k) = \sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k} \mathbb{E}\big\{|d_\ell(i) - \boldsymbol{x}_\ell^\top(i)\boldsymbol{w}_k|^2\big\}, \quad (15)$$

$$\overline{J}_k^{\rm loc}(\boldsymbol{w}_k) = \sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k} \mathbb{E}\big\{|d_\ell(i) - \boldsymbol{x}_\ell^\top(i)\boldsymbol{w}_k|^2\big\} + 2\eta \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} p_{k\ell} f(\boldsymbol{w}_k - \boldsymbol{w}_\ell), \quad (16)$$

where $\mathcal{N}_k \cap \mathcal{C}(k)$ denotes the set of nodes in the neighborhood of node k that belongs to its cluster, and $\{c_{\ell k}\}$ are non-negative weights satisfying

$$\sum_{k=1}^{N} c_{\ell k} = 1, \ \text{ and } \ c_{\ell k} = 0 \ \text{ if } \ k \notin \mathcal{N}_\ell \cap \mathcal{C}(\ell). \quad (17)$$

Note that $\boldsymbol{w}_k = \boldsymbol{w}_\ell$ whenever $\ell \in \mathcal{N}_k \cap \mathcal{C}(k)$. Both costs (15) and (16) consist of a convex combination of mean-square errors in the neighborhood of node k but limited to its cluster. In addition, expression (16) takes interactions among neighboring clusters into account. Let us consider node k belonging to cluster $\mathcal{C}_j$, i.e., $\mathcal{C}_j = \mathcal{C}(k)$. It can be checked that $\overline{J}_{\mathcal{C}_j}(\boldsymbol{w}_{\mathcal{C}_j})$ in (14) can be written as:

$$\overline{J}_{\mathcal{C}_j}(\boldsymbol{w}_{\mathcal{C}_j}) = \sum_{\ell \in \mathcal{C}_j} \overline{J}_\ell^{\rm loc}(\boldsymbol{w}_\ell) = \overline{J}_k^{\rm loc}(\boldsymbol{w}_k) + \sum_{\ell \in \mathcal{C}_j \setminus \{k\}} \overline{J}_\ell^{\rm loc}(\boldsymbol{w}_\ell), \quad (18)$$

The term $\sum_{\ell \in \mathcal{C}_j \setminus \{k\}} \overline{J}_\ell^{\rm loc}(\boldsymbol{w}_\ell)$ contains terms promoting relationships between nodes $\ell \in \mathcal{C}_j \setminus \{k\}$ and their neighbors that are outside $\mathcal{C}_j$ but not necessarily in the neighborhood of node k. To limit these inter-cluster information exchanges to node k and its extra-cluster neighbors, we relax $\sum_{\ell \in \mathcal{C}_j \setminus \{k\}} \overline{J}_\ell^{\rm loc}(\boldsymbol{w}_\ell)$ to $\sum_{\ell \in \mathcal{C}_j \setminus \{k\}} J_\ell^{\rm loc}(\boldsymbol{w}_\ell)$. Since (15) is second-order differentiable, a completion-of-squares argument shows that each $J_\ell^{\rm loc}(\boldsymbol{w}_\ell)$ can be expressed as [7]:

$$J_\ell^{\rm loc}(\boldsymbol{w}_\ell) = J_\ell^{\rm loc}(\boldsymbol{w}_\ell^{\rm loc}) + \|\boldsymbol{w}_\ell - \boldsymbol{w}_\ell^{\rm loc}\|_{\boldsymbol{R}_\ell}^2, \quad (19)$$

where the notation $\|\boldsymbol{x}\|_{\boldsymbol{\Sigma}}^2$ denotes $\boldsymbol{x}^\top \boldsymbol{\Sigma}\, \boldsymbol{x}$ for any nonnegative definite matrix $\boldsymbol{\Sigma}$, $\boldsymbol{w}_\ell^{\rm loc}$ is the minimizer of $J_\ell^{\rm loc}(\boldsymbol{w}_\ell)$, and $\boldsymbol{R}_\ell$ is given by:

$$\boldsymbol{R}_\ell = \sum_{k \in \mathcal{N}_\ell \cap \mathcal{C}(\ell)} c_{k\ell} \boldsymbol{R}_{\boldsymbol{x},k}. \quad (20)$$

Thus, using (16), (18), and (19) and dropping the constant term $J_\ell^{\rm loc}(\boldsymbol{w}_\ell^{\rm loc})$, we can replace the original cluster cost (14) by the following cost function for cluster $\mathcal{C}(k)$ at node k:

$$\overline{J}'_{\mathcal{C}(k)}(\boldsymbol{w}_k) = \sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k} \mathbb{E}\big\{|d_\ell(i) - \boldsymbol{x}_\ell^\top(i)\boldsymbol{w}_k|^2\big\} + 2\eta \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} p_{k\ell} f(\boldsymbol{w}_k - \boldsymbol{w}_\ell) + \sum_{\ell \in \mathcal{C}(k) \setminus \{k\}} \|\boldsymbol{w}_\ell - \boldsymbol{w}_\ell^{\rm loc}\|_{\boldsymbol{R}_\ell}^2. \quad (21)$$

Equation (21) is an approximation relating the local cost function $\overline{J}_k^{\rm loc}(\boldsymbol{w}_k)$ at node k to the global cost function (14) associated with the cluster $\mathcal{C}(k)$. Node k cannot minimize (21) directly since this cost still requires global information that may not be available in its neighborhood. To avoid access to information via multihop, we relax $\overline{J}'_{\mathcal{C}(k)}(\boldsymbol{w}_k)$ by limiting the sum in the third term on the RHS of (21) over the neighbors of node k. In addition, since the covariance matrices $\boldsymbol{R}_{\boldsymbol{x},\ell}$ may not be known beforehand within the context of online learning, a useful strategy proposed in [7] is to substitute the covariance matrices $\boldsymbol{R}_\ell$ by diagonal matrices of the form $b_{\ell k}\boldsymbol{I}_M$, where $b_{\ell k}$ are nonnegative coefficients that allow to assign different weights to different neighbors. Later, these coefficients will be incorporated into a left stochastic matrix and the designer does not need to worry about their selection. Based on the arguments presented so far, the cluster cost function at each node k can be relaxed as follows:

$$\overline{J}''_{\mathcal{C}(k)}(\boldsymbol{w}_k) = \sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k} \mathbb{E}\big\{|d_\ell(i) - \boldsymbol{x}_\ell^\top(i)\boldsymbol{w}_k|^2\big\} + 2\eta \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} p_{k\ell} f(\boldsymbol{w}_k - \boldsymbol{w}_\ell) + \sum_{\ell \in \mathcal{N}_k^- \cap \mathcal{C}(k)} b_{\ell k} \left\|\boldsymbol{w}_k - \boldsymbol{w}_\ell^{\rm loc}\right\|^2. \quad (22)$$

Since this cost function only relies on data available in the neighborhood of each node k, we can now proceed to derive distributed strategies.

The first and third terms on the RHS of (22) are second-order differentiable and strictly convex. The second term is convex but not continuously differentiable. In [31], a multitask Adapt-then-Combine (ATC) diffusion algorithm was derived using subgradient techniques. The purpose of this work is to obtain an iterative algorithm for solving the convex minimization problem (22) using a forward-backward splitting approach.

C. Multitask Diffusion With Forward-Backward Splitting Approach

Let $\boldsymbol{w}_k(i)$ denote the estimate of $\boldsymbol{w}_k^o$ at node k and iteration i. Considering a forward-backward splitting strategy for solving (22), we have:

$$\boldsymbol{w}_k(i+1) = \mathrm{prox}_{2\eta\nu_k \tilde{g}_{k,i}}\Big(\boldsymbol{w}_k(i) - \nu_k \nabla_{\boldsymbol{w}_k} J''_{\mathcal{C}(k)}(\boldsymbol{w}_k(i))\Big), \quad (23)$$

with ν_k a positive step-size parameter,

$$\tilde{g}_{k,i}(\boldsymbol{w}_k) = \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} p_{k\ell} f(\boldsymbol{w}_k - \boldsymbol{w}_\ell(i)), \quad (24)$$

and $J''_{\mathcal{C}(k)}(\boldsymbol{w}_k)$ denoting the unregularized part of $\overline{J}''_{\mathcal{C}(k)}(\boldsymbol{w}_k)$ limited to the first and third terms on the RHS of (22). Let

$$\boldsymbol{\phi}_k(i+1) = \boldsymbol{w}_k(i) - \nu_k \nabla_{\boldsymbol{w}_k} J''_{\mathcal{C}(k)}(\boldsymbol{w}_k(i)). \tag{25}$$

Node k can run the Adapt-then-Combine (ATC) form of diffusion [7] for evaluating $\boldsymbol{\phi}_k(i+1)$. Thus, we arrive at the following Adapt-then-Combine (ATC) diffusion strategy with forward-backward splitting for solving problem (10) in a fully distributed adaptive manner:

$$\begin{cases} \boldsymbol{\psi}_k(i+1) = \boldsymbol{w}_k(i) \\ \qquad\qquad + \mu_k \displaystyle\sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k}\, \boldsymbol{x}_\ell(i)[d_\ell(i) - \boldsymbol{x}_\ell^\top(i)\boldsymbol{w}_k(i)], \\ \boldsymbol{\phi}_k(i+1) = \displaystyle\sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} a_{\ell k} \boldsymbol{\psi}_\ell(i+1), \\ \boldsymbol{w}_k(i+1) = \text{prox}_{\eta \mu_k g_{k,i+1}}(\boldsymbol{\phi}_k(i+1)), \end{cases} \tag{26}$$

where $\mu_k = 2\nu_k$ is introduced to avoid an extra factor of 2 multiplying ν_k and coming from evaluating the gradient of squared quantities in $J''_{\mathcal{C}(k)}(\boldsymbol{w}_k)$, $\{a_{\ell k}\}$ are nonnegative combination coefficients satisfying:

$$\sum_{\ell=1}^{N} a_{\ell k} = 1, \quad \text{and} \quad a_{\ell k} = 0 \quad \text{if} \quad \ell \notin \mathcal{N}_k \cap \mathcal{C}(k), \tag{27}$$

and

$$g_{k,i+1}(\boldsymbol{w}_k) \triangleq \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} p_{k\ell} f(\boldsymbol{w}_k - \boldsymbol{\phi}_\ell(i+1)). \tag{28}$$

Functions $\tilde{g}_{k,i}(\cdot)$ in (24) and $g_{k,i+1}(\cdot)$ in (28) are iteration dependent through $\boldsymbol{w}_\ell(i)$ and $\boldsymbol{\phi}_\ell(i+1)$. Note that we have substituted $\boldsymbol{w}_\ell(i)$ in (24) by $\boldsymbol{\phi}_\ell(i+1)$ in (28) since $\boldsymbol{\phi}_\ell(i+1)$ is an updated estimate of $\boldsymbol{w}_\ell(i)$ at node ℓ. The proximal operator of $\eta\mu_k g_{k,i+1}(\cdot)$ in the third step of (26) needs to be evaluated at each iteration $i+1$ and for all nodes k in the network. A closed-form expression is recommended to achieve higher computational efficiency. We shall derive such closed-form expression when f in (28) is selected either as the ℓ_1-norm or the reweighted ℓ_1-norm — see Section II-D for details.

The multitask diffusion LMS (26) with forward-backward splitting starts with an initial estimate $\boldsymbol{w}_k(0)$ for all k, and repeats (26) at each instant $i \geq 0$ and for all k. In the first step of (26), which corresponds to the adaptation step, node k receives from its intra-cluster neighbors their raw data $\{d_\ell(i), \boldsymbol{x}_\ell(i)\}$, combines this information through the coefficients $\{c_{\ell k}\}$, and uses it to update its estimate $\boldsymbol{w}_k(i)$ to an intermediate estimate $\boldsymbol{\psi}_k(i+1)$. The second step in (26) is a combination step where node k receives the intermediate estimates $\{\boldsymbol{\psi}_\ell(i+1)\}$ from its intra-cluster neighbors and combines them through the coefficients $\{a_{\ell k}\}$ to obtain the intermediate value $\boldsymbol{\phi}_k(i+1)$. Finally, in the third step in (26), node k receives the intermediate estimates $\{\boldsymbol{\phi}_\ell(i+1)\}$ from its neighbors that are outside its cluster and evaluates the proximal operator of the function in (28) at $\boldsymbol{\phi}_k(i+1)$ to obtain $\boldsymbol{w}_k(i+1)$. To run the algorithm, each node k only needs to know the step-size μ_k, the regularization strength η, the regularization weights $\{p_{k\ell}\}_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)}$, and the coefficients $\{a_{\ell k}, c_{\ell k}\}_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)}$ satisfying conditions (17) and (27). The scalars $\{a_{\ell k}, c_{\ell k}\}$ and $\{\rho_{k\ell}\}$ correspond to weighting coefficients over the edges linking node k to its neighbors ℓ according to whether these neighbors lie inside or outside its cluster. There are several ways to select these coefficients [4], [5], [7], [29]. In Section IV, we propose an adaptive rule for selecting each regularization weight $p_{k\ell}$ based on a measure of the sparsity level of $\boldsymbol{w}_k^o - \boldsymbol{w}_\ell^o$ at node k. Finally, note that alternative implementations of (26) may be considered. In particular, the adaptation step can be followed by the proximal step, before or after aggregation as in the possible Adapt-then-Combine and Combine-then-Adapt diffusion strategies.

Algorithm (26) may be applied to multitask problems involving any type of coregularizers $f(\cdot)$ provided that the proximal operator of a weighted sum of these regularizers can be assessed in closed form. In the next section, we shall focus on the particular case of sparsity promoting regularizers.

D. Proximal Operator of Weighted Sum of ℓ_1-Norms

We shall now derive a closed form expression for the proximal operator of the convex function $g_{k,i+1}(\boldsymbol{w}_k)$ in (28). Considering both regularizations addressed in this work, that is, the ℓ_1-norm (5) and the reweighted ℓ_1-norm (6), we write:

$$\begin{aligned} g_{k,i+1}(\boldsymbol{w}_k) &= \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} p_{k\ell} \sum_{m=1}^{M} \alpha_{k\ell}^m(i) \big|[\boldsymbol{w}_k]_m - [\boldsymbol{\phi}_\ell(i+1)]_m\big| \\ &= \sum_{m=1}^{M} \Phi_{k,m,i+1}([\boldsymbol{w}_k]_m) \end{aligned} \tag{29}$$

where $\Phi_{k,m,i+1}([\boldsymbol{w}_k]_m)$ is the iteration-dependent function given by:

$$\Phi_{k,m,i+1}([\boldsymbol{w}_k]_m) = \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} p_{k\ell}\, \alpha_{k\ell}^m(i) \big|[\boldsymbol{w}_k]_m - [\boldsymbol{\phi}_\ell(i+1)]_m\big|. \tag{30}$$

Since $g_{k,i+1}(\boldsymbol{w}_k)$ is fully separable, its proximal operator can be evaluated component-wise [34]:

$$\begin{aligned} &[\text{prox}_{\eta\mu_k g_{k,i+1}}(\boldsymbol{\phi}_k(i+1))]_m \\ &\quad = \text{prox}_{\eta\mu_k \Phi_{k,m,i+1}}([\boldsymbol{\phi}_k(i+1)]_m), \quad \forall m = 1, \ldots, M. \end{aligned} \tag{31}$$

For clarity of presentation, we shall now derive the proximal operator of a function $h(\cdot)$ similar to $\Phi_{k,m,i+1}$. Next, we shall establish the closed-form expression for $\text{prox}_{\eta\mu_k \Phi_{k,m,i+1}}(\cdot)$ by identification.

Let $h : \mathbb{R} \to \mathbb{R}$ be a combination of absolute value functions defined as:

$$h(x) \triangleq \sum_{j=1}^{J} c_j\, h_j(x) = \sum_{j=1}^{J} c_j\, |x - b_j|, \tag{32}$$

with $c_j > 0$ for all j and $b_1 < b_2 < \ldots < b_J$. Note that this ordering is assumed for convenience of derivation and does not affect the final result. Iterative algorithms have been proposed in the literature for evaluating the proximal operator of sums of composite functions [32], [33]. We are, however, able to derive a closed-form expression for (32) as detailed in the sequel. From the optimality condition for (3), namely that zero belongs to the

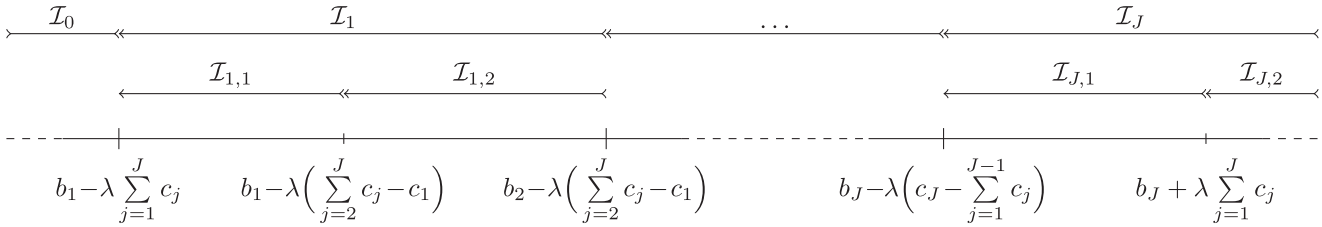


Fig. 1. Decomposition of $\mathbb{R}$ into $J+1$ intervals given by (37)–(40). The width of the intervals depends on the weights $\{c_j\}_{j=1}^J$ and on the coefficients $\{b_j\}_{j=1}^J$.

subgradient set at the minimizer $\text{prox}_{\lambda h}(v)$, we have,

$$\begin{aligned} 0 &\in \partial h(\text{prox}_{\lambda h}(v)) + \frac{1}{\lambda}(\text{prox}_{\lambda h}(v) - v) \\ &\Rightarrow v - \text{prox}_{\lambda h}(v) \in \lambda \partial h(\text{prox}_{\lambda h}(v)). \end{aligned} \tag{33}$$

Since $x \in \mathbb{R}$ and c_j are non-negative, we have [54, Chapter 5: Lemma 10]:

$$\partial\left(\sum_{j=1}^{J} c_j h_j(x)\right) = \sum_{j=1}^{J} c_j \partial h_j(x) = \sum_{j=1}^{J} c_j \partial|x - b_j|. \tag{34}$$

Hence, the subdifferential of the real valued convex function $h(x)$ in (32) is:

$$\partial h(x) = \begin{cases} -\sum_{j=1}^{J} c_j, & \text{if} \quad x < b_1, \\ c_1 \cdot [-1,1] - \sum_{j=2}^{J} c_j, & \text{if} \quad x = b_1, \\ c_1 - \sum_{j=2}^{J} c_j, & \text{if} \quad b_1 < x < b_2, \\ \quad \vdots & \\ \sum_{j=1}^{J-1} c_j + c_J \cdot [-1,1], & \text{if} \quad x = b_J, \\ \sum_{j=1}^{J} c_j, & \text{if} \quad x > b_J. \end{cases} \tag{35}$$

From (33) and (35), extensive but routine calculations lead to the following implementation for evaluating the proximal operator of h in (32). Let us decompose $\mathbb{R}$ into $J+1$ intervals such that $\mathbb{R} = \bigcup_{n=0}^{J} \mathcal{I}_n$ where, as illustrated in Fig. 1:

$$\mathcal{I}_0 \triangleq \left]-\infty,\, b_1 - \lambda \sum_{j=1}^{J} c_j\right[, \tag{36}$$

$$\mathcal{I}_n \triangleq \mathcal{I}_{n,1} \cup \mathcal{I}_{n,2}, \qquad n = 1, \ldots, J, \tag{37}$$

with

$$\mathcal{I}_{n,1} \triangleq \left[b_n - \lambda\left(\sum_{j=n}^{J} c_j - \sum_{j=1}^{n-1} c_j\right), b_n - \lambda\left(\sum_{j=n+1}^{J} c_j - \sum_{j=1}^{n} c_j\right)\right[, \quad n = 1, \ldots, J, \tag{38}$$

$$\mathcal{I}_{n,2} \triangleq \left[b_n - \lambda\left(\sum_{j=n+1}^{J} c_j - \sum_{j=1}^{n} c_j\right), b_{n+1} - \lambda\left(\sum_{j=n+1}^{J} c_j - \sum_{j=1}^{n} c_j\right)\right[, \quad n = 1, \ldots, J-1, \tag{39}$$

$$\mathcal{I}_{J,2} \triangleq \left[b_J + \lambda \sum_{j=1}^{J} c_j, +\infty\right[. \tag{40}$$

Depending on the interval to which v belongs, we evaluate the proximal operator according to:

$$\text{prox}_{\lambda h}(v) = \begin{cases} v + \lambda \sum_{j=1}^{J} c_j, & \text{if } v \in \mathcal{I}_0 \\ b_n, & \text{if } v \in \mathcal{I}_{n,1} \\ v + \lambda\left(\sum_{j=n+1}^{J} c_j - \sum_{j=1}^{n} c_j\right), & \text{if } v \in \mathcal{I}_{n,2}. \end{cases} \tag{41}$$

In order to make clearer how the operator in (41) works, we plot $\text{prox}_h(v)$ for three expressions of h in Fig. 2.

It can be checked that the proximal operator in (41) can be written more compactly as:

$$\text{prox}_{\lambda h}(v) = v - \lambda \Gamma(v), \tag{42}$$

where

$$\Gamma(v) = \frac{1}{2}\sum_{n=1}^{J}\left\{\left|\frac{v - b_n}{\lambda} - \sum_{j=1}^{n-1} c_j + \sum_{j=n}^{J} c_j\right| - \left|\frac{v - b_n}{\lambda} - \sum_{j=1}^{n} c_j + \sum_{j=n+1}^{J} c_j\right|\right\}. \tag{43}$$

Comparing (33) and (42), we remark that $\Gamma(v)$ is a subgradient of h at $\text{prox}_{\lambda h}(v)$. Based on equation (41), $\Gamma(v)$ is bounded as follows:

$$|\Gamma(v)| \leq \sum_{j=1}^{J} c_j \tag{44}$$

for all v. In fact, equality holds when v belongs to $\mathcal{I}_0$ in (36) or $\mathcal{I}_{J,2}$ in (40). When v belongs to an interval of the form of $\mathcal{I}_{n,1}$

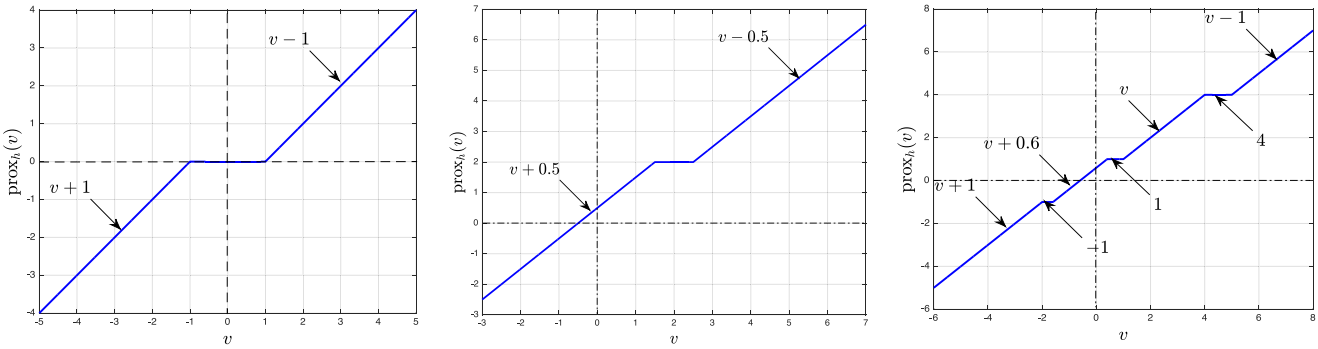


Fig. 2. Proximal operator $\text{prox}_{\lambda h}(v)$ versus $v \in \mathbb{R}$ with $\lambda = 1$ and $h : \mathbb{R} \to \mathbb{R}, h(x) = \sum_{j=1}^{J} c_j |x - b_j|$. (a) $h(x) = |x|$. (b) $h(x) = \frac{1}{2}|x - 2|$. (c) $h(x) = \frac{1}{5}|x + 1| + \frac{3}{10}|x - 1| + \frac{1}{2}|x - 4|$.

in (38), we have:

$$\Gamma(v) = \frac{v - b_n}{\lambda} \in \left[\sum_{j=1}^{n-1} c_j - \sum_{j=n}^{J} c_j, \sum_{j=1}^{n} c_j - \sum_{j=n+1}^{J} c_j\right] \subset \left[-\sum_{j=1}^{J} c_j, \sum_{j=1}^{J} c_j\right], \tag{45}$$

and when it belongs to an interval of the form of $\mathcal{I}_{n,2}$ in (39), we have:

$$\Gamma(v) = \sum_{j=1}^{n} c_j - \sum_{j=n+1}^{J} c_j \in \left[-\sum_{j=1}^{J} c_j, \sum_{j=1}^{J} c_j\right]. \tag{46}$$

We note that the upper bound in (44) is independent of λ. Using (42), the m-th entry of $\text{prox}_{\eta\mu_k g_{k,i+1}}(\boldsymbol{\phi}_k(i+1))$ in (31) can be written as:

$$[\text{prox}_{\eta\mu_k g_{k,i+1}}(\boldsymbol{\phi}_k(i+1))]_m = [\boldsymbol{\phi}_k(i+1)]_m - \eta\mu_k \Gamma_{k,m,i+1}([\boldsymbol{\phi}_k(i+1)]_m). \tag{47}$$

Note that $\Gamma_{k,m,i+1}([\boldsymbol{\phi}_k(i+1)]_m)]$ is a function of the form (43) where, based on (30), the coefficients b_j and c_j are given by $[\boldsymbol{\phi}_\ell(i+1)]_m$ and $p_{k\ell}\,\alpha_{k\ell}^m(i)$, respectively, and the scalar v corresponds to the m-th component of the vector $\boldsymbol{\phi}_k(i+1)$. Using the boundedness of $\Gamma_{k,m,i+1}(\cdot)$ in (44), we obtain:

$$|\Gamma_{k,m,i+1}([\boldsymbol{\phi}_k(i+1)]_m)| \le \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} p_{k\ell}\,\alpha_{k\ell}^m(i) \triangleq s_k^m(i) \tag{48}$$

for all $[\boldsymbol{\phi}_k(i+1)]_m$. For the ℓ_1-norm (5), we have:

$$s_k^m(i) = s_k \triangleq \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} p_{k\ell}, \tag{49}$$

for all i and $m = 1, \ldots, M$. For the reweighted ℓ_1-norm (6), we have:

$$s_k^m(i) = \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} \frac{p_{k\ell}}{\epsilon + |[\boldsymbol{\delta}_{k,\ell}(i-1)]_m|} = \frac{1}{\epsilon} \sum_{\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)} \frac{p_{k\ell}}{1 + \frac{|[\boldsymbol{\delta}_{k,\ell}(i-1)]_m|}{\epsilon}} \le \frac{s_k}{\epsilon} \tag{50}$$

for all i and $m = 1, \ldots, M$. Using (47), the proximal operator of $\eta\mu_k g_{k,i+1}$ can be written as:

$$\text{prox}_{\eta\mu_k g_{k,i+1}}(\boldsymbol{\phi}_k(i+1)) = \boldsymbol{\phi}_k(i+1) - \eta\mu_k \boldsymbol{\Gamma}_{k,i+1}(\boldsymbol{\phi}_k(i+1)), \tag{51}$$

where $\boldsymbol{\Gamma}_{k,i+1}(\boldsymbol{\phi}_k(i+1))$ is the $M \times 1$ vector given by:

$$\boldsymbol{\Gamma}_{k,i+1}(\boldsymbol{\phi}_k(i+1)) = \text{col}\Big\{\Gamma_{k,1,i+1}([\boldsymbol{\phi}_k(i+1)]_1), \ldots, \Gamma_{k,M,i+1}([\boldsymbol{\phi}_k(i+1)]_M)\Big\}. \tag{52}$$

As a consequence, the ℓ_2-norm of the vector $\boldsymbol{\Gamma}_{k,i+1}(\cdot)$ can be bounded as:

$$\|\boldsymbol{\Gamma}_{k,i+1}(\cdot)\|_2 \le s_k\sqrt{M}, \text{for the } \ell_1\text{-norm}, \tag{53}$$

$$\|\boldsymbol{\Gamma}_{k,i+1}(\cdot)\|_2 \le \frac{s_k\sqrt{M}}{\epsilon}, \text{for the reweighted } \ell_1\text{-norm}. \tag{54}$$

III. Stability Analysis

A. Error Vector Recursion

We shall now analyze the stability of the multitask diffusion algorithm (26) in the mean and mean-square-error sense. We first define at node k and iteration i the weight error vector $\widetilde{\boldsymbol{w}}_k(i) \triangleq \boldsymbol{w}_k^o - \boldsymbol{w}_k(i)$ and the intermediate error vector $\widetilde{\boldsymbol{\phi}}_k(i) \triangleq \boldsymbol{w}_k^o -$

$\boldsymbol{\phi}_k(i)$. Furthermore, we introduce the network vectors:

$$\widetilde{\boldsymbol{w}}(i) \triangleq \text{col}\,\{\widetilde{\boldsymbol{w}}_1(i),\ldots,\widetilde{\boldsymbol{w}}_N(i)\} \tag{55}$$

$$\boldsymbol{\phi}(i) \triangleq \text{col}\,\{\boldsymbol{\phi}_1(i),\ldots,\boldsymbol{\phi}_N(i)\} \tag{56}$$

$$\widetilde{\boldsymbol{\phi}}(i) \triangleq \text{col}\left\{\widetilde{\boldsymbol{\phi}}_1(i),\ldots,\widetilde{\boldsymbol{\phi}}_N(i)\right\}. \tag{57}$$

Let $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{R}}_{\boldsymbol{x}}(i)$ be the $MN\times MN$ block diagonal matrices defined as:

$$\boldsymbol{\mathcal{M}} \triangleq \text{diag}\,\{\mu_k \boldsymbol{I}_M\}_{k=1}^N \tag{58}$$

$$\boldsymbol{\mathcal{R}}_{\boldsymbol{x}}(i) \triangleq \text{diag}\left\{\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\,\boldsymbol{x}_\ell(i)\,\boldsymbol{x}_\ell^\top(i)\right\}_{k=1}^N \tag{59}$$

and $\boldsymbol{p}_{zx}(i)$ be the $MN\times 1$ block vector defined as:

$$\boldsymbol{p}_{zx}(i) \triangleq \boldsymbol{\mathcal{C}}^\top \text{col}\big\{\boldsymbol{x}_k(i)\,z_k(i)\big\}_{k=1}^N, \tag{60}$$

where $\boldsymbol{\mathcal{C}} \triangleq \boldsymbol{C}\otimes\boldsymbol{I}_M$ and $\boldsymbol{C}$ is the $N\times N$ right-stochastic matrix whose ℓk-th entry is $c_{\ell k}$. Let $\boldsymbol{\mathcal{A}} \triangleq \boldsymbol{A}\otimes\boldsymbol{I}_M$ where $\boldsymbol{A}$ is the $N\times N$ left-stochastic matrix whose ℓk-th entry is $a_{\ell k}$. Subtracting $\boldsymbol{w}_k^o$ from both sides of the first and second step in (26), and using the linear data model (4), we obtain:

$$\widetilde{\boldsymbol{\phi}}(i+1) = \boldsymbol{\mathcal{A}}^\top[\boldsymbol{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{\boldsymbol{x}}(i)]\widetilde{\boldsymbol{w}}(i) - \boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}\boldsymbol{p}_{z\boldsymbol{x}}(i). \tag{61}$$

Subtracting $\boldsymbol{w}_k^o$ from both sides of the third step in (26), and using result (51), we get:

$$\widetilde{\boldsymbol{w}}_k(i+1) = \widetilde{\boldsymbol{\phi}}_k(i+1) + \eta\mu_k\,\boldsymbol{\Gamma}_{k,i+1}(\boldsymbol{\phi}_k(i+1)). \tag{62}$$

Hence, the network error vector for the diffusion strategy (26) evolves according to the following recursion:

$$\boxed{\begin{aligned}\widetilde{\boldsymbol{w}}(i+1) = \boldsymbol{\mathcal{A}}^\top[\boldsymbol{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{\boldsymbol{x}}(i)]\,\widetilde{\boldsymbol{w}}(i) - \boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}\boldsymbol{p}_{z\boldsymbol{x}}(i)\\ +\eta\,\boldsymbol{\mathcal{M}}\boldsymbol{\Gamma}_{i+1}(\boldsymbol{\phi}(i+1)),\end{aligned}} \tag{63}$$

where $\boldsymbol{\Gamma}_{i+1}(\boldsymbol{\phi}(i+1))$ is the $N\times 1$ block vector whose k-th block is given by (52), namely,

$$\boldsymbol{\Gamma}_{i+1}(\boldsymbol{\phi}(i+1)) \triangleq \text{col}\Big\{\boldsymbol{\Gamma}_{k,i+1}(\boldsymbol{\phi}_k(i+1))\Big\}_{k=1}^N. \tag{64}$$

In order to make the presentation clearer, we shall use the following notation for terms in recursion (63):

$$\boldsymbol{\mathcal{B}}(i) \triangleq \boldsymbol{\mathcal{A}}^\top[\boldsymbol{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{\boldsymbol{x}}(i)], \tag{65}$$

$$\boldsymbol{g}(i) \triangleq \boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}\boldsymbol{p}_{z\boldsymbol{x}}(i), \tag{66}$$

$$\boldsymbol{r}(i+1) \triangleq \eta\,\boldsymbol{\mathcal{M}}\boldsymbol{\Gamma}_{i+1}(\boldsymbol{\phi}(i+1)). \tag{67}$$

Hence, recursion (63) can be rewritten as follows:

$$\widetilde{\boldsymbol{w}}(i+1) = \boldsymbol{\mathcal{B}}(i)\widetilde{\boldsymbol{w}}(i) - \boldsymbol{g}(i) + \boldsymbol{r}(i+1). \tag{68}$$

Before proceeding, let us introduce the following assumptions on the regression data and step-sizes.

Assumption 1: (Independent regressors) The regression vectors $\boldsymbol{x}_k(i)$ arise from a zero-mean random process that is temporally white and spatially independent.

It follows that $\boldsymbol{x}_k(i)$ is independent of $\boldsymbol{w}_\ell(j)$ for $i\geq j$ and for all ℓ. This assumption is commonly used in adaptive filtering since it helps simplify the analysis. Furthermore, performance results obtained under this assumption match well the actual performance of stand alone filters for sufficiently small step-sizes [55].

Assumption 2: (Small step-sizes) The step-sizes μ_k are sufficiently small so that terms that depend on higher order powers of the step-sizes can be ignored.

B. Mean Behavior Analysis

Taking the expectation of both sides of (68), using Assumption 1, and $\mathbb{E}\{\boldsymbol{p}_{z\boldsymbol{x}}(i)\}=0$, we obtain that the mean error vector evolves according to the following recursion:

$$\mathbb{E}\{\widetilde{\boldsymbol{w}}(i+1)\} = \boldsymbol{\mathcal{B}}\,\mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\} + \mathbb{E}\{\boldsymbol{r}(i+1)\}, \tag{69}$$

where

$$\boldsymbol{\mathcal{B}} \triangleq \boldsymbol{\mathcal{A}}^\top(\boldsymbol{I}_{MN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{\boldsymbol{x}}), \tag{70}$$

$$\boldsymbol{\mathcal{R}}_{\boldsymbol{x}} \triangleq \mathbb{E}\{\boldsymbol{\mathcal{R}}_{\boldsymbol{x}}(i)\} = \text{diag}\left\{\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\boldsymbol{R}_{\boldsymbol{x},\ell}\right\}_{k=1}^N \tag{71}$$

$$\mathbb{E}\{\boldsymbol{r}(i+1)\} \triangleq \eta\boldsymbol{\mathcal{M}}\mathbb{E}\{\boldsymbol{\Gamma}_{i+1}(\boldsymbol{\phi}(i+1))\}. \tag{72}$$

The following theorem guarantees the mean stability of the multitask diffusion LMS (26) with forward-backward splitting.

Recall that the block maximum norm of an $N\times 1$ block vector $\boldsymbol{x}=\text{col}\{\boldsymbol{x}_k\}_{k=1}^N$ and the induced block maximum norm of an $N\times N$ block matrix $\boldsymbol{\mathcal{X}}$ are defined as [7]:

$$\|\boldsymbol{x}\|_{b,\infty} = \max_{1\leq k\leq N}\|\boldsymbol{x}_k\|_2,$$

$$\|\boldsymbol{\mathcal{X}}\|_{b,\infty} = \max_{\boldsymbol{x}}\frac{\|\boldsymbol{\mathcal{X}}\boldsymbol{x}\|_{b,\infty}}{\|\boldsymbol{x}\|_{b,\infty}}, \tag{73}$$

Theorem 1: (Stability in the mean) Assume data model (4) and Assumption 1 hold. Then, for any initial conditions, the multitask diffusion strategy (26) converges in the mean to a small bounded region of the order of $\mu_{\max}$, i.e., $\lim_{i\to\infty}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|_{b,\infty}\} = O(\mu_{\max})$, if the step-sizes are chosen such that:

$$0<\mu_k<\frac{2}{\lambda_{\max}(\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\boldsymbol{R}_{\boldsymbol{x},\ell})},\quad k=1,\ldots,N, \tag{74}$$

where $\mu_{\max}\triangleq\max_{1\leq k\leq N}\mu_k$ and $\lambda_{\max}(\cdot)$ is the maximum eigenvalue of its matrix argument. The block maximum norm of the bias can be upper bounded as:

$$\lim_{i\to\infty}\|\mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}\|_{b,\infty} \leq \frac{\eta\,\mu_{\max}\,s_{\max}\sqrt{M}}{1-\|\boldsymbol{\mathcal{B}}\|_{b,\infty}}, \tag{75}$$

$$\lim_{i\to\infty}\|\mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}\|_{b,\infty} \leq \frac{1}{\epsilon}\cdot\frac{\eta\,\mu_{\max}\,s_{\max}\sqrt{M}}{1-\|\boldsymbol{\mathcal{B}}\|_{b,\infty}}, \tag{76}$$

for the ℓ_1-norm and the reweighted ℓ_1-norm, respectively.

Proof: Iterating (69) starting from $i=0$, we arrive to the following expression:

$$\mathbb{E}\{\widetilde{\boldsymbol{w}}(i+1)\} = \boldsymbol{\mathcal{B}}^{i+1}\mathbb{E}\{\widetilde{\boldsymbol{w}}(0)\} + \sum_{j=0}^{i}\boldsymbol{\mathcal{B}}^j\,\mathbb{E}\{\boldsymbol{r}(i+1-j)\}, \tag{77}$$

where $\mathbb{E}\{\widetilde{\boldsymbol{w}}(0)\}$ is the initial condition. $\mathbb{E}\{\widetilde{\boldsymbol{w}}(i+1)\}$ converges when $i\to\infty$ if, and only if, both terms on the RHS of (77)

converges to finite values. The first term converges to zero as $i \to \infty$ if the matrix $\boldsymbol{\mathcal{B}}$ is stable. A sufficient condition to ensure the stability of $\boldsymbol{\mathcal{B}}$ is to choose the step-sizes according to (74) (the proof can be obtained using the same arguments as [7, Theorem 5.1]). We shall now prove the convergence of the second term on the RHS of (77). To prove the convergence of the series $\sum_{j=0}^{+\infty} \boldsymbol{\mathcal{B}}^j \, \mathbb{E}\{\boldsymbol{r}(i+1-j)\}$, it is sufficient to prove that the series $\sum_{j=0}^{+\infty} [\boldsymbol{\mathcal{B}}^j \, \mathbb{E}\{\boldsymbol{r}(i+1-j)\}]_k$ converges for $k = 1, \ldots, MN$. A series is absolutely convergent if each term of the series can be bounded by a term of an absolutely convergent series [42]. Since the block maximum norm of a block vector is greater than or equal to the largest absolute value of its entries, each term $\left|[\boldsymbol{\mathcal{B}}^j \, \mathbb{E}\{\boldsymbol{r}(i+1-j)\}]_k\right|$ can be bounded as:

$$\begin{aligned}\left|[\boldsymbol{\mathcal{B}}^j \, \mathbb{E}\{\boldsymbol{r}(i+1-j)\}]_k\right| &\leq \|\boldsymbol{\mathcal{B}}\|_{b,\infty}^j \cdot \|\mathbb{E}\{\boldsymbol{r}(i+1-j)\}\|_{b,\infty} \\ &\leq \|\boldsymbol{\mathcal{B}}\|_{b,\infty}^j r_{\max}. \end{aligned} \tag{78}$$

The quantity $\|\mathbb{E}\{\boldsymbol{r}(i+1-j)\}\|_{b,\infty}$ is finite for all i and j and bounded by some constant $r_{\max} = \mathcal{O}(\mu_{\max})$. In fact, from (72), we have:

$$\|\mathbb{E}\{\boldsymbol{r}(i+1)\}\|_{b,\infty} \leq \eta \mu_{\max} \|\mathbb{E}\{\boldsymbol{\Gamma}_{i+1}(\boldsymbol{\phi}(i+1))\}\|_{b,\infty} \tag{79}$$

since $\|\boldsymbol{\mathcal{M}}\|_{b,\infty} = \mu_{\max}$. Using (53)–(54), the block maximum norm of $\boldsymbol{\Gamma}_{i+1}(\boldsymbol{\phi}(i+1))$ in (64) can be bounded as:

$$\|\boldsymbol{\Gamma}_{i+1}(\boldsymbol{\phi}(i+1))\|_{b,\infty} \leq \; s_{\max}\sqrt{M}, (\ell_1\text{-norm}) \tag{80}$$

$$\|\boldsymbol{\Gamma}_{i+1}(\boldsymbol{\phi}(i+1))\|_{b,\infty} \leq \; \frac{s_{\max}\sqrt{M}}{\epsilon}, (\text{rew. } \ell_1\text{-norm}) \tag{81}$$

for all i, where $s_{\max} = \max_{1 \leq k \leq N} s_k$. If the step-sizes are chosen according to (74), the series $\sum_{j=0}^{+\infty} \|\boldsymbol{\mathcal{B}}\|_{b,\infty}^j r_{\max}$ is absolutely convergent. Therefore, the series $\sum_{j=0}^{+\infty} [\boldsymbol{\mathcal{B}}^j \, \mathbb{E}\{\boldsymbol{r}(i+1-j)\}]_k$ is an absolutely convergent series.

Note that when $i \to \infty$, the block maximum norm of the bias can be bounded as

$$\begin{aligned}\lim_{i\to\infty} \|\mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}\|_{b,\infty} &= \lim_{i\to\infty} \left\| \sum_{j=0}^{i} \boldsymbol{\mathcal{B}}^j \, \mathbb{E}\{\boldsymbol{r}(i+1-j)\} \right\|_{b,\infty} \\ &\leq \lim_{i\to\infty} \sum_{j=0}^{\infty} \|\boldsymbol{\mathcal{B}}^j \, \mathbb{E}\{\boldsymbol{r}(i+1-j)\}\|_{b,\infty} \\ &\leq \lim_{i\to\infty} \sum_{j=0}^{\infty} \|\boldsymbol{\mathcal{B}}\|_{b,\infty}^j r_{\max} = \frac{r_{\max}}{1 - \|\boldsymbol{\mathcal{B}}\|_{b,\infty}}, \end{aligned} \tag{82}$$

■

C. Mean-Square-Error Stability

We examine the mean-square-error stability by studying the convergence of the weighted variance $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|_{\boldsymbol{\Sigma}}^2\}$, where $\boldsymbol{\Sigma}$ is a positive semi-definite matrix that we are free to choose. Evaluating the variance, we obtain:

$$\begin{aligned}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|_{\boldsymbol{\Sigma}}^2\} &= \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|_{\boldsymbol{\Sigma}'}^2\} + \mathbb{E}\{\|\boldsymbol{g}(i)\|_{\boldsymbol{\Sigma}}^2\} \\ &\quad + \varphi(\boldsymbol{r}(i+1), \boldsymbol{\Sigma}, \boldsymbol{\mathcal{B}}(i), \widetilde{\boldsymbol{w}}(i), \boldsymbol{g}(i)), \end{aligned} \tag{83}$$

where $\boldsymbol{\Sigma}' \triangleq \mathbb{E}\{\boldsymbol{\mathcal{B}}^\top(i) \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}(i)\}$ and

$$\begin{aligned}\varphi(\boldsymbol{r}(i+1), \boldsymbol{\mathcal{B}}(i), \widetilde{\boldsymbol{w}}(i), \boldsymbol{g}(i)) &= \mathbb{E}\{\|\boldsymbol{r}(i+1)\|_{\boldsymbol{\Sigma}}^2\} \\ &\quad + 2\mathbb{E}\{\boldsymbol{r}^\top(i+1) \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}}(i) \widetilde{\boldsymbol{w}}(i)\} - 2\mathbb{E}\{\boldsymbol{r}^\top(i+1) \boldsymbol{\Sigma} \boldsymbol{g}(i)\} \end{aligned} \tag{84}$$

is a term coming from promoting relationships between clusters. The last two terms on the RHS of (84) contain higher-order powers of the step-sizes. Using Assumption 2, we get the following approximation:

$$\varphi(\boldsymbol{r}(i+1), \widetilde{\boldsymbol{w}}(i)) \approx \mathbb{E}\{\|\boldsymbol{r}(i+1)\|_{\boldsymbol{\Sigma}}^2\} + 2\mathbb{E}\{\boldsymbol{r}^\top(i+1) \boldsymbol{\Sigma} \boldsymbol{\mathcal{B}} \widetilde{\boldsymbol{w}}(i)\} \tag{85}$$

Let $\boldsymbol{\sigma} \triangleq \text{vec}(\boldsymbol{\Sigma})$ and $\boldsymbol{\sigma}' \triangleq \text{vec}(\boldsymbol{\Sigma}')$ where the $\text{vec}(\cdot)$ operator stacks the columns of a matrix on top of each other. We will use the notation $\|\widetilde{\boldsymbol{w}}\|_{\boldsymbol{\sigma}}^2$ and $\|\widetilde{\boldsymbol{w}}\|_{\boldsymbol{\Sigma}}^2$ interchangeably to denote the same quantity $\widetilde{\boldsymbol{w}}^\top \boldsymbol{\Sigma} \widetilde{\boldsymbol{w}}$. Using the property $\text{vec}(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}) = (\boldsymbol{W}^\top \otimes \boldsymbol{U})\text{vec}(\boldsymbol{\Sigma})$, the relation between $\boldsymbol{\sigma}'$ and $\boldsymbol{\sigma}$ can be expressed in the following form:

$$\boldsymbol{\sigma}' = \boldsymbol{\mathcal{F}} \boldsymbol{\sigma}, \tag{86}$$

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

$$\boldsymbol{\mathcal{F}} \triangleq \mathbb{E}\{\boldsymbol{\mathcal{B}}^\top(i) \otimes \boldsymbol{\mathcal{B}}^\top(i)\} \approx \boldsymbol{\mathcal{B}}^\top \otimes \boldsymbol{\mathcal{B}}^\top. \tag{87}$$

The approximation in (87) is reasonable under Assumption 2 [7]. Introducing the matrix $\boldsymbol{G}$:

$$\boldsymbol{G} \triangleq \mathbb{E}\{\boldsymbol{g}(i)\boldsymbol{g}^\top(i)\} = \boldsymbol{\mathcal{A}}^\top \boldsymbol{\mathcal{M}} \boldsymbol{\mathcal{C}}^\top \text{diag}\{\boldsymbol{R}_{\boldsymbol{x},k} \sigma_{z,k}^2\}_{k=1}^N \boldsymbol{\mathcal{C}} \boldsymbol{\mathcal{M}} \boldsymbol{\mathcal{A}} \tag{88}$$

and using the property $\text{tr}(\boldsymbol{\Sigma}\boldsymbol{X}) = [\text{vec}(\boldsymbol{X}^\top)]^\top \text{vec}(\boldsymbol{\Sigma})$, the second term on the RHS of (83) can be written as:

$$\mathbb{E}\{\|\boldsymbol{g}(i)\|_{\boldsymbol{\Sigma}}^2\} = [\text{vec}(\boldsymbol{G}^\top)]^\top \boldsymbol{\sigma}. \tag{89}$$

Hence, the variance recursion (83) can be expressed as

$$\begin{aligned}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|_{\boldsymbol{\sigma}}^2\} &= \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|_{\boldsymbol{\mathcal{F}}\boldsymbol{\sigma}}^2\} + [\text{vec}(\boldsymbol{G}^\top)]^\top \boldsymbol{\sigma} \\ &\quad + \varphi(\boldsymbol{r}(i+1), \boldsymbol{\sigma}, \widetilde{\boldsymbol{w}}(i)). \end{aligned} \tag{90}$$

Theorem 2: (Mean-square-error Stability) Assume data model (4) and Assumptions 1 and 2 hold. Then, for any initial conditions, the multitask diffusion strategy (26) is mean-square stable if the error recursion (63) is mean stable and the matrix $\boldsymbol{\mathcal{F}}$ is stable. Using the approximation (87), the matrix $\boldsymbol{\mathcal{F}}$ is stable if the step-sizes satisfy (74).

Proof. Since $\boldsymbol{\Sigma}$ is a positive semi-definite matrix and the vector $\boldsymbol{r}(i+1)$ is uniformly bounded for all i, $\mathbb{E}\{\|\boldsymbol{r}(i+1)\|_{\boldsymbol{\Sigma}}^2\}$ can be bounded as

$$0 \leq \mathbb{E}\{\|\boldsymbol{r}(i+1)\|_{\boldsymbol{\Sigma}}^2\} \leq \kappa_1 \tag{91}$$

for all i, where κ_1 is a positive constant. Since $\boldsymbol{r}(i+1)$ is uniformly bounded for all i, the vector $2\boldsymbol{r}^\top(i+1)\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{B}}$ is also bounded for all i. Let $\gamma_{\max}$ be a bound on the largest component of $2\boldsymbol{r}^\top(i+1)\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{B}}$ in absolute value for all i. We obtain

$$\begin{aligned}2|\mathbb{E}\{\boldsymbol{r}^\top(i+1)\boldsymbol{\Sigma}\boldsymbol{\mathcal{B}}\widetilde{\boldsymbol{w}}(i)\}| &\leq \gamma_{\max} \sum_{\ell=1}^{MN} \left|\mathbb{E}\left\{\widetilde{\boldsymbol{w}}_\ell(i)\right\}\right| \\ &= \gamma_{\max} \cdot \|\mathbb{E}\left\{\widetilde{\boldsymbol{w}}(i)\right\}\|_1. \end{aligned} \tag{92}$$

Under condition (74) on the step-sizes, the mean error vector $\mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}$ converges to a small bounded region as $i \to \infty$. Hence,

$\|\mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}\|_1$ can be upper bounded by some positive constant scalar κ_2 for all i, and using the approximation (85), $|\varphi(\boldsymbol{r}(i+1),\boldsymbol{\sigma},\widetilde{\boldsymbol{w}}(i))|$ satisfies:

$$|\varphi(\boldsymbol{r}(i+1),\boldsymbol{\sigma},\widetilde{\boldsymbol{w}}(i))| \leq \kappa_1 + \gamma_{\max}\kappa_2 \tag{93}$$

for all i. The positive constant $\kappa_3 \triangleq \kappa_1 + \gamma_{\max}\kappa_2$ can be written as a scaled multiple of the positive quantity $[\mathrm{vec}(\boldsymbol{G}^\top)]^\top\boldsymbol{\sigma}$ as $\kappa_3 = t[\mathrm{vec}(\boldsymbol{G}^\top)]^\top\boldsymbol{\sigma}$ where $t \geq 0$ [42]. We arrive at the following inequality for (90):

$$\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\} \leq \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|^2_{\boldsymbol{\mathcal{F}}\boldsymbol{\sigma}}\} + (1+t)\cdot[\mathrm{vec}(\boldsymbol{G}^\top)]^\top\boldsymbol{\sigma}. \tag{94}$$

Iterating (94) starting from $i=0$, we obtain

$$\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\} \leq \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(0)\|^2_{\boldsymbol{\mathcal{F}}^{i+1}\boldsymbol{\sigma}}\} + (1+t)[\mathrm{vec}(\boldsymbol{G}^\top)]^\top\sum_{j=0}^{i}\boldsymbol{\mathcal{F}}^j\boldsymbol{\sigma}, \tag{95}$$

where $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(0)\|^2\}$ is the initial condition. If we show that the RHS of (95) converges, then $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\}$ is stable. The first term on the RHS of (95) vanishes as $i\to\infty$ if the matrix $\boldsymbol{\mathcal{F}}$ is stable. Consider now the second term on the RHS of (95). The series $\sum_{j=0}^{\infty}\boldsymbol{\mathcal{F}}^j\boldsymbol{\sigma}$ converges if $\sum_{j=0}^{\infty}[\boldsymbol{\mathcal{F}}^j\boldsymbol{\sigma}]_k$ converges for $k=1,\ldots,(MN)^2$. Each term of the series can be bounded as

$$[\boldsymbol{\mathcal{F}}^j\boldsymbol{\sigma}]_k \leq |[\boldsymbol{\mathcal{F}}^j\boldsymbol{\sigma}]_k| \leq \|\boldsymbol{\mathcal{F}}^j\boldsymbol{\sigma}\|_{b,\infty} \leq \|\boldsymbol{\mathcal{F}}^j\|_{b,\infty}\cdot\|\boldsymbol{\sigma}\|_{b,\infty}. \tag{96}$$

Since $\boldsymbol{\mathcal{F}}$ is stable, there exists a submultiplicative norm[1] $\|\cdot\|_\rho$ such that $\|\boldsymbol{\mathcal{F}}\|_\rho = \zeta < 1$. All norms are equivalent in finite dimensional vector spaces. Thus, we have:

$$\|\boldsymbol{\mathcal{F}}^j\|_{b,\infty} \leq \tau\|\boldsymbol{\mathcal{F}}^j\|_\rho \leq \tau\|\boldsymbol{\mathcal{F}}\|^j_\rho = \tau\zeta^j, \tag{97}$$

for some positive constant τ. Considering this bound with (96) yields:

$$\sum_{j=0}^{\infty}|[\boldsymbol{\mathcal{F}}^j\boldsymbol{\sigma}]_k| \leq \sum_{j=0}^{\infty}\|\boldsymbol{\mathcal{F}}^j\|_{b,\infty}\cdot\|\boldsymbol{\sigma}\|_{b,\infty} \leq \tau\sum_{j=0}^{\infty}\zeta^j\|\boldsymbol{\sigma}\|_{b,\infty} = \frac{\tau\cdot\|\boldsymbol{\sigma}\|_{b,\infty}}{1-\zeta}. \tag{98}$$

As a consequence, since the second term on the RHS of (95) converges to a bounded region when $\boldsymbol{\mathcal{F}}$ is stable, $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{\boldsymbol{\sigma}}\}$ also converges. ■

IV. Simulation Results

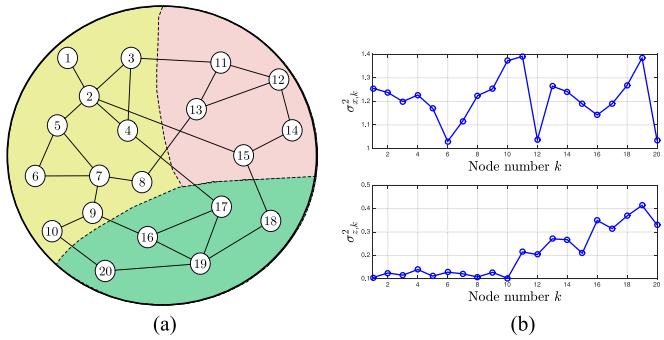
Before proceeding, we present a new rule for selecting the regularization weight $p_{k\ell}$ based on a measure of sparsity of the vector $\boldsymbol{w}^o_k - \boldsymbol{w}^o_\ell$. The intuition behind this rule is to employ a large weight $p_{k\ell}$ when the objectives at nodes k and ℓ have few distinct entries, i.e., $\boldsymbol{w}^o_k - \boldsymbol{w}^o_\ell$ is sparse, and a small weight $p_{k\ell}$ when the objectives have few similar entries, i.e., $\boldsymbol{w}^o_k - \boldsymbol{w}^o_\ell$ is not sparse. Among other possible choices for the sparsity measure, we select a popular one based on a relationship between the ℓ_1-norm and ℓ_2-norm [56]:

$$\xi(\boldsymbol{w}^o_k - \boldsymbol{w}^o_\ell) = \frac{M}{M-\sqrt{M}}\left(1 - \frac{\|\boldsymbol{w}^o_k - \boldsymbol{w}^o_\ell\|_1}{\sqrt{M}\cdot\|\boldsymbol{w}^o_k - \boldsymbol{w}^o_\ell\|_2}\right) \in [0,1]. \tag{99}$$

The quantity $\xi(\boldsymbol{w}^o_k - \boldsymbol{w}^o_\ell)$ is equal to one when only a single component of $\boldsymbol{w}^o_k - \boldsymbol{w}^o_\ell$ is non-zero, and zero when all elements of $\boldsymbol{w}^o_k - \boldsymbol{w}^o_\ell$ are relatively large [56]. Since the nodes do not know the true objectives $\boldsymbol{w}^o_k$ and $\boldsymbol{w}^o_\ell$, we propose to replace these quantities by the available estimates at each time instant i and allow the regularization factors to vary with time according to:

$$p_{k\ell}(i) \propto \begin{cases} \dfrac{M}{M-\sqrt{M}}\left(1 - \dfrac{\|\boldsymbol{\phi}_k(i+1) - \boldsymbol{\phi}_\ell(i+1)\|_1}{\sqrt{M}\cdot\|\boldsymbol{\phi}_k(i+1) - \boldsymbol{\phi}_\ell(i+1)\|_2}\right), & \text{if } \ell \in \mathcal{N}_k\backslash\mathcal{C}(k) \\ 0, & \text{otherwise} \end{cases} \tag{100}$$

where the symbol $\propto$ denotes proportionality. As we shall see in the simulations, this rule improves the performance of the algorithm and allows agent k to adapt the regularization strength $p_{k\ell}$ with respect to the sparsity level of the vector $\boldsymbol{w}^o_k - \boldsymbol{w}^o_\ell$ at time instant i.



(a) (b)

Fig. 3. Experimental setup. (a) Network topology. (b) Regression and noise variances.

A. Illustrative Example

We consider a clustered network with the topology shown in Fig. 3(a), consisting of 20 nodes divided into 3 clusters: $\mathcal{C}_1 = \{1,\ldots,10\}$, $\mathcal{C}_2 = \{11,\ldots,15\}$, and $\mathcal{C}_3 = \{16,\ldots,20\}$. The regression vectors $\boldsymbol{x}_k(i)$ are 18×1 zero-mean Gaussian distributed vectors with covariance matrices $\boldsymbol{R}_{\boldsymbol{x},k} = \sigma^2_{x,k}\boldsymbol{I}_{18}$. The variances $\sigma^2_{x,k}$ are shown in Fig. 3(b). The noises $z_k(i)$ are zero-mean i.i.d. Gaussian random variables, independent of any other signal, with variances $\sigma^2_{z,k}$ shown in Fig. 3(b). Let $\mathrm{card}\{\cdot\}$ denote the cardinal of its entry. We run the diffusion algorithm (26) by setting $c_{\ell k} = \frac{1}{\mathrm{card}\{\mathcal{N}_\ell\cap\mathcal{C}(\ell)\}}$ for $k\in\mathcal{N}_\ell\cap\mathcal{C}(\ell)$ and $a_{\ell k} = \frac{1}{\mathrm{card}\{\mathcal{N}_k\cap\mathcal{C}(k)\}}$ for $\ell\in\mathcal{N}_k\cap\mathcal{C}(k)$. The regularization weights are set to $\rho_{k\ell} = \frac{1}{\mathrm{card}\{\mathcal{N}_k\backslash\mathcal{C}(k)\}}$ for $\ell\in\mathcal{N}_k\backslash\mathcal{C}(k)$. We use a constant step-size $\mu = 0.02$ for all nodes, a sparsity strength $\eta = 0.06$ for the ℓ_1-norm regularizer, and $\eta = 0.04$ for the reweighted ℓ_1-norm regularizer with $\epsilon = 0.1$. The results are averaged over 200 Monte-Carlo runs.

[1]The norm $\|\cdot\|_\rho$ is called submultiplicative if for any square matrices $\boldsymbol{U}$ and $\boldsymbol{W}$ of compatible dimensions we have: $\|\boldsymbol{UW}\|_\rho \leq \|\boldsymbol{U}\|_\rho\cdot\|\boldsymbol{W}\|_\rho$.

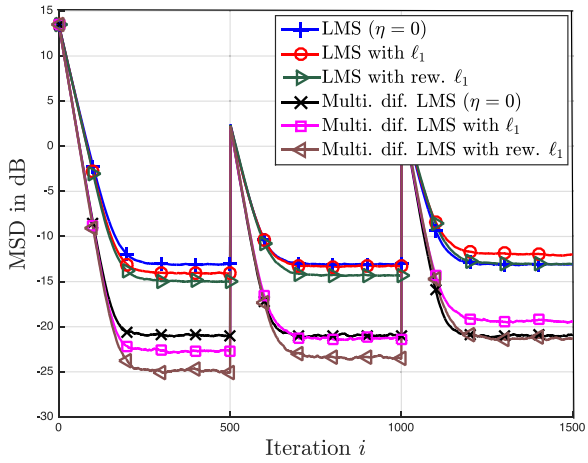


Fig. 4. Network MSD comparison for 6 different strategies: non-cooperative LMS (algorithm (26) with $\boldsymbol{A} = \boldsymbol{C} = \boldsymbol{I}_N$, $\eta = 0$), spatially regularized LMS (algorithm (26) with $\boldsymbol{A} = \boldsymbol{C} = \boldsymbol{I}_N$ with ℓ_1-norm and reweighted ℓ_1-norm, standard diffusion without cooperation between clusters (algorithm (26) with $\eta = 0$), and our proximal diffusion (26) with ℓ_1-norm and reweighted ℓ_1-norm.

The optimum vectors are set to $\boldsymbol{w}_{\mathcal{C}_j}^o = \boldsymbol{w}^o + \boldsymbol{\delta}_{\mathcal{C}_j}$ at each cluster with $\boldsymbol{w}^o$ an 18×1 vector whose entries are generated from the Gaussian distribution $\mathcal{N}(0,1)$. First, we set $\boldsymbol{\delta}_{\mathcal{C}_1}$ to $\boldsymbol{0}_{1\times 18}^\top$, $\boldsymbol{\delta}_{\mathcal{C}_2}$ to $[-1\ \boldsymbol{0}_{1\times 17}]^\top$, and $\boldsymbol{\delta}_{\mathcal{C}_3}$ to $[\boldsymbol{0}_{1\times 6}\ -1\ \boldsymbol{0}_{1\times 11}]^\top$. Observe that at most two entries differ between clusters. After 500 iterations, we set $\boldsymbol{\delta}_{\mathcal{C}_2}$ to $[-\boldsymbol{1}_{1\times 3}\ 1\ \boldsymbol{0}_{1\times 14}]^\top$ and $\boldsymbol{\delta}_{\mathcal{C}_3}$ to $[\boldsymbol{0}_{1\times 12}\ -\boldsymbol{1}_{1\times 3}\ \boldsymbol{0}_{1\times 3}]^\top$. In this way, at most 7 entries differ between clusters. After 1000 iterations, we set $\boldsymbol{\delta}_{\mathcal{C}_2}$ to $[-\boldsymbol{1}_{1\times 3}\ \boldsymbol{1}_{1\times 3}\ -\boldsymbol{1}_{1\times 3}\ \boldsymbol{0}_{1\times 9}]^\top$ and $\boldsymbol{\delta}_{\mathcal{C}_3}$ to $[\boldsymbol{0}_{1\times 9}\ \boldsymbol{1}_{1\times 3}\ -\boldsymbol{1}_{1\times 3}\ \boldsymbol{1}_{1\times 3}]^\top$. Thus, at most 18 entries now differ between clusters.

In Fig. 4, we compare 6 algorithms: the non-cooperative LMS (algorithm (26) with $\boldsymbol{A} = \boldsymbol{C} = \boldsymbol{I}_N$ and $\eta = 0$), the regularized LMS (algorithm (26) with $\boldsymbol{A} = \boldsymbol{C} = \boldsymbol{I}_N$) with ℓ_1-norm and reweighted ℓ_1-norm, the multitask diffusion LMS without regularization (algorithm (26) with $\eta = 0$), and the multitask diffusion LMS (26) with ℓ_1-norm and reweighted ℓ_1-norm regularization. As observed in this figure, when the tasks share a sufficient number of components, cooperation between clusters enhances the network MSD performance. When the number of common entries decreases, the cooperation between clusters becomes less effective. The use of the ℓ_1-norm can lead to a degradation of the MSD relative to the absence of cooperation among clusters. However, the use of the reweighted ℓ_1-norm allows to improve the performance.

In order to better understand the behavior of the algorithm (26) in the clusters, we report in Fig. 5 the learning curves for $i \in [0, 1000]$ of the common and distinct entries among clusters given by

$$\frac{1}{\text{card}\{\mathcal{C}_j\}} \sum_{k\in\mathcal{C}_j} \mathbb{E}\left\{ \sum_{m\in\Omega(i)} ([\boldsymbol{w}_k^o(i) - \boldsymbol{w}_k(i)]_m)^2 \right\}, \qquad (101)$$

for $j = 1$, 3, where $\Omega(i)$ is the set of identical (or distinct) components among all clusters at iteration i and $\boldsymbol{w}_k^o(i)$ is the optimum parameter vector at node k and iteration i. For example, for $i \in [0, 500]$, the set of distinct components is $\{1, 7\}$. As shown in this figure, cluster $\mathcal{C}_3$ benefits considerably from cooperation with other clusters in the estimation of the common entries. Nevertheless, cluster $\mathcal{C}_1$ benefits slightly from cooperation. This is due to the fact that the performance of $\mathcal{C}_3$ is low relatively to that of $\mathcal{C}_1$ since the SNR in $\mathcal{C}_3$ is small and the number of nodes employed in this cluster is 5.

We shall now illustrate the effect of the regularization strength η over the performance of the algorithm for different numbers of common entries between the optimum vectors $\boldsymbol{w}_k^o$. We consider the same settings as above, which means that the number of common entries among clusters is successively set to 16, 11, and 0 over 18. Parameter η is uniformly sampled over $[0, 0.14]$. Fig. 6 shows the gain in steady-state MSD versus the unregularized algorithm obtained for $\eta = 0$, as a function of η. For each η, the results are averaged over 50 Monte-Carlo runs and over 50 samples after convergence of the algorithm. It can be observed in Fig. 6 that the interval for η over which the network benefits from cooperation between clusters becomes smaller as the number of common entries decreases. In addition, the reweighted ℓ_1-norm regularizer provides better performance than the ℓ_1-norm regularizer.

In order to guarantee a correct cooperation among clusters, we repeat the same experiment as Fig. 4 using the adaptive rule in (100) for adjusting the regularization factors $p_{k\ell}$. The proportionality coefficient in (100) is set equal to one. As shown in Fig. 7, when the number of distinct components is small, both ℓ_1 and reweighted ℓ_1-norms yield better performance than the diffusion LMS with $\eta = 0$. When the number of distinct components increases ($i \in (1000, 1500]$), the performance of strategy (26) with ℓ_1-norm gets closer to diffusion LMS with $\eta = 0$, while the reweighted ℓ_1-norm still guarantees a gain. Thus, the mechanism proposed in (100) for the selection of the regularization factors improves the cooperation between nodes belonging to distinct clusters.

Finally, we compare the current multitask diffusion strategy (26) with two other useful strategies existing in the literature [24], [29]. We consider a stationary environment where the optimum parameter vectors $\{\boldsymbol{w}_{\mathcal{C}_j}^o\}_{j=1}^3$ consist of a sub-vector $\boldsymbol{\xi}^o$ of 16 parameters of global interest to the whole network and a 2×1 sub-vector $\{\boldsymbol{\varsigma}_{\mathcal{C}_j}^o\}$ of common interest to nodes belonging to cluster $\mathcal{C}_j$, namely, $\boldsymbol{w}_{\mathcal{C}_j}^o = \text{col}\{\boldsymbol{\xi}^o, \boldsymbol{\varsigma}_{\mathcal{C}_j}^o\}$. The entries of $\boldsymbol{\xi}^o$, $\boldsymbol{\varsigma}_{\mathcal{C}_1}^o$, $\boldsymbol{\varsigma}_{\mathcal{C}_2}^o$, and $\boldsymbol{\varsigma}_{\mathcal{C}_3}^o$ are uniformly sampled from a uniform distribution $\mathcal{U}(-3, 3)$. Except for these changes, we consider the same experimental setup described in the first paragraph of the current section. When applying the strategy developed in [24], we assume that node k belonging to cluster $\mathcal{C}_j$ is aware that the first 16 parameters of $\boldsymbol{w}_{\mathcal{C}_j}^o$ are of global interest to the whole network while the remaining parameters are of common interest to nodes in cluster $\mathcal{C}_j$. However, the current method (26) and the algorithm in [29] do not require such assumption. We run the ATC D-NSPE strategy developed in [24] using uniform combination weights $a_{\ell k}^w = 1/\text{card}\{\mathcal{N}_k\}$ for $\ell \in \mathcal{N}_k$ and $a_{\ell k}^{\varsigma_{\mathcal{C}(k)}} = 1/\text{card}\{\mathcal{N}_k \cap \mathcal{C}(k)\}$ for $\ell \in \mathcal{N}_k \cap \mathcal{C}(k)$, and uniform step-sizes $\mu_k = 0.02\ \forall k$. We run the multitask diffusion strategy developed in [29] by setting $\{c_{\ell k}, a_{\ell k}, \rho_{k\ell}\}$ in the same manner described in the first paragraph of the current section, $\mu_k = 0.02\ \forall k$, and $\eta = 0.06$. The learning curves of the algorithms are reported in Fig. 8. As expected, it can be observed that the cooperation between clusters based on the ℓ_2-norm [29] degrades the performance relative to the case of non-cooperative clusters, i.e., $\eta = 0$. Indeed, the multitask diffusion strategy developed in [29] considers squared ℓ_2-norm co-regularizers to promote the smoothness of the graph signal, whereas, in the current simulation we need to promote the sparsity of the vector

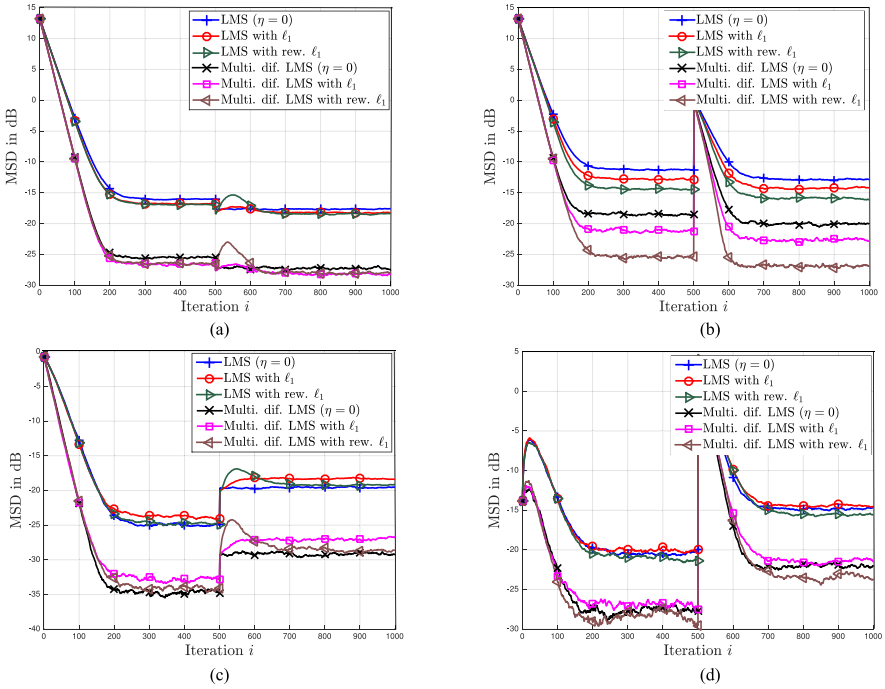


Fig. 5. Clusters MSD over identical and distinct components. Comparison for the same 6 different strategies considered in Fig. 4. (a) Cluster 1 MSD over identical entries. (b) Cluster 3 MSD over identical entries. (c) Cluster 1 MSD over distinct entries. (d) Cluster 3 MSD over distinct entries.

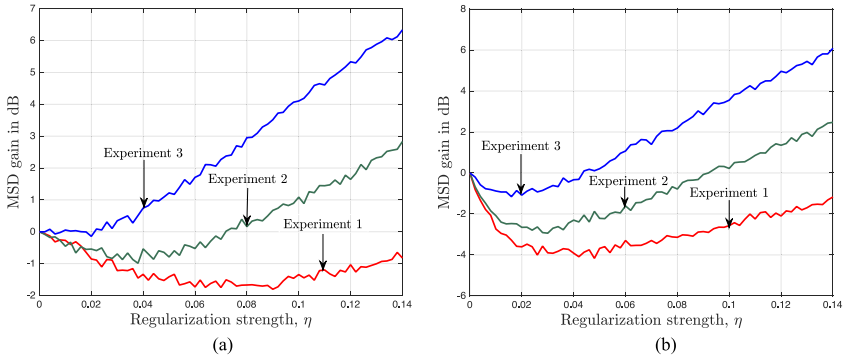


Fig. 6. Differential network MSD (MSD(η) − MSD($\eta = 0$)) in dB with respect to the regularization strength η for the multitask diffusion LMS (26) with ℓ_1-norm (left) and reweighted ℓ_1-norm (right) for 3 different degrees of similarity between tasks. Experiment 1: at most 2 entries differ between clusters. Experiment 2: at most 7 entries differ between clusters. Experiment 3: at most 18 entries differ between clusters. (a) ℓ_1-norm. (b) Reweighted ℓ_1-norm.

$\boldsymbol{w}_k^o - \boldsymbol{w}_\ell^o$. Furthermore, when the reweighted ℓ_1-norm is used, our method is able to perform well compared to the strategy developed in [24] that requires the knowledge of the indices of common and distinct entries in the parameter vectors. We note that recent unsupervised strategies [57], [58] dealing with group of variables rather than variables propose to add a step in order to adapt the cooperation between neighboring nodes based on the group at hand. It is shown in [57] that the performance depends heavily on the group decomposition of the parameter vectors.

B. Distributed Spectrum Sensing

Consider a cognitive radio network composed of N_P primary users (PU) and N_S secondary users (SU). To avoid causing harmful interference to the primary users, each secondary user has to detect the frequency bands used by all primary users, even under low signal to noise ratio conditions [7], [24], [59].

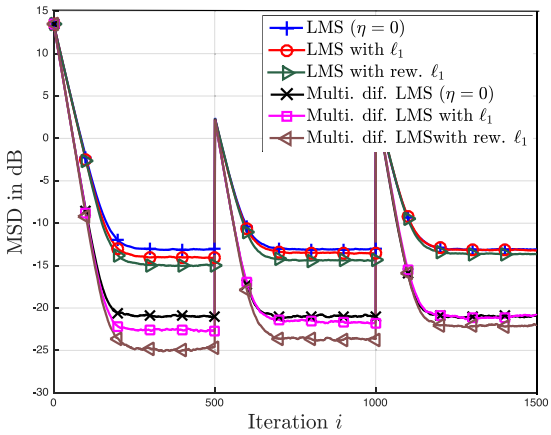


Fig. 7. Network MSD comparison for the same 6 different strategies considered in Fig. 4 using adaptive regularization factors $p_{k\ell}(i)$.

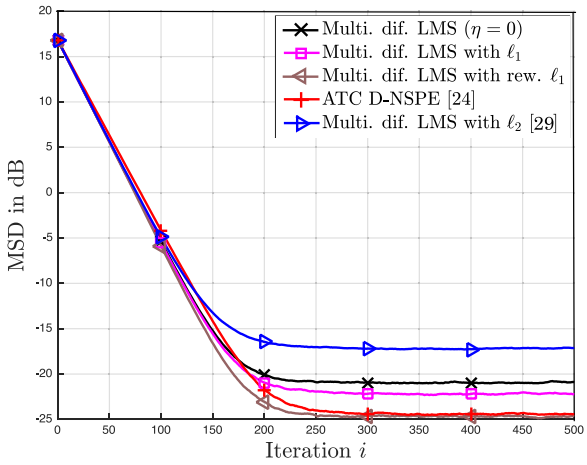


Fig. 8. Network MSD comparison for 5 different strategies: standard diffusion without cooperation between clusters (algorithm (26) with $\eta = 0$), our proximal diffusion (26) with ℓ_1-norm and reweighted ℓ_1-norm, the ATC D-NSPE algorithm developed in [24], and the multitask diffusion strategy with squared ℓ_2-norm coregularizers [29].

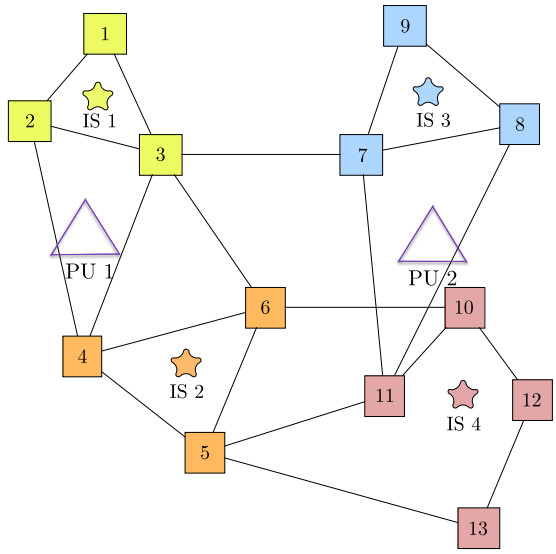


Fig. 9. A cognitive radio network consisting of 2 primary users and 13 secondary users grouped into 4 clusters containing each an interference source IS.

We assume that the secondary users are grouped into Q clusters and that there exists within each cluster a low power interference source (IS). The goal of each secondary user is to estimate the aggregated spectrum transmitted by all active primary users, as well as the spectrum of the interference source present in its cluster.

In order to facilitate the estimation task of the secondary users, we assume that the power spectrum of the signal transmitted by the primary user p and the interference source q can be represented by a linear combination of N_B basis functions $\phi_m(f)$:

$$S_p(f) = \sum_{m=1}^{N_B} \alpha_{pm}\,\phi_m(f), \quad p = 1, \ldots, N_P, \tag{102}$$

$$S_q(f) = \sum_{m=1}^{N_B} \beta_{qm}\,\phi_m(f), \quad q = 1, \ldots, Q, \tag{103}$$

where α_{pm}, β_{qm} are the combination weights, and f is the normalized frequency. Each secondary user $k \in \mathcal{C}_q$ has to estimate the $N_B \times (N_P + 1)$ vector $\boldsymbol{\Upsilon}_k^o = \text{col}\{\boldsymbol{\alpha}_1^o, \ldots, \boldsymbol{\alpha}_{N_P}^o, \boldsymbol{\beta}_q^o\}$ where $\boldsymbol{\alpha}_p^o = [\alpha_{p1}, \ldots, \alpha_{pN_B}]^\top$ and $\boldsymbol{\beta}_q^o = [\beta_{q1}, \ldots, \beta_{qN_B}]^\top$. Let $\ell_{p,k}(i)$ denote the path loss factor between the primary user p and the secondary user k at time i. Let also $\ell'_{q,k}(i)$ denote the path loss factor between the interference source q and the secondary user k at time i. Then, the power spectrum sensed by node $k \in \mathcal{C}_q$ at time i and frequency f_j can be expressed as follows:

$$r_{k,j}(i) = \sum_{p=1}^{N_P} \ell_{p,k}(i) S_p(f_j) + \ell'_{q,k}(i) S_q(f_j) + z_{k,j}(i), \tag{104}$$

where $z_{k,j}(i)$ is the sampling noise at the j-th frequency assumed to be zero-mean Gaussian with variance $\sigma^2_{z_{k,j}}$. At each time instant i, node k observes the power spectrum over N_F frequency samples. Let $\boldsymbol{r}_k(i)$ and $\boldsymbol{z}_k(i)$ be the $N_F \times 1$ vectors whose j-th entries are $r_{k,j}(i)$ and $z_{k,j}(i)$, respectively. Using (104), we can establish the following linear data model for node $k \in \mathcal{C}_q$:

$$\boldsymbol{r}_k(i) = \boldsymbol{\Phi}_k(i)\boldsymbol{\Upsilon}_k^o + \boldsymbol{z}_k(i), \tag{105}$$

where $\boldsymbol{\Phi}_k(i) \triangleq [\ell_{1,k}(i), \ldots, \ell_{N_P,k}(i), \ell'_{q,k}(i)] \otimes \boldsymbol{\Phi}$ with $\boldsymbol{\Phi}$ the $N_F \times N_B$ matrix whose j-th row contains the magnitudes of the N_B basis functions at the frequency sample f_j.

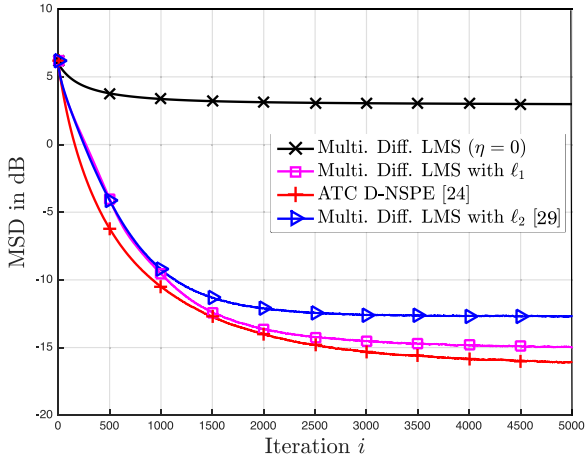


Fig. 10. Network MSD comparison for 4 different algorithms: standard diffusion LMS without cooperation between clusters ($\eta = 0$), our proximal diffusion (26) with ℓ_1-norm regularizer, the ATC D-NSPE algorithm developed in [24], and the multitask diffusion strategy [29].

To show the effect of multitask learning with sparsity-based regularization, we consider a cognitive radio network consisting of $N_P = 2$ primary users and $N_S = 13$ secondary users decomposed into 4 clusters as shown in Fig. 9. The power spectrum is represented by a combination of $N_B = 20$ Gaussian basis functions centered at the normalized frequency f_m with variance $\sigma_m^2 = 0.001$ for all m:

$$\phi_m(f) = \exp^{-\frac{(f - f_m)^2}{2\sigma_m^2}}, \tag{106}$$

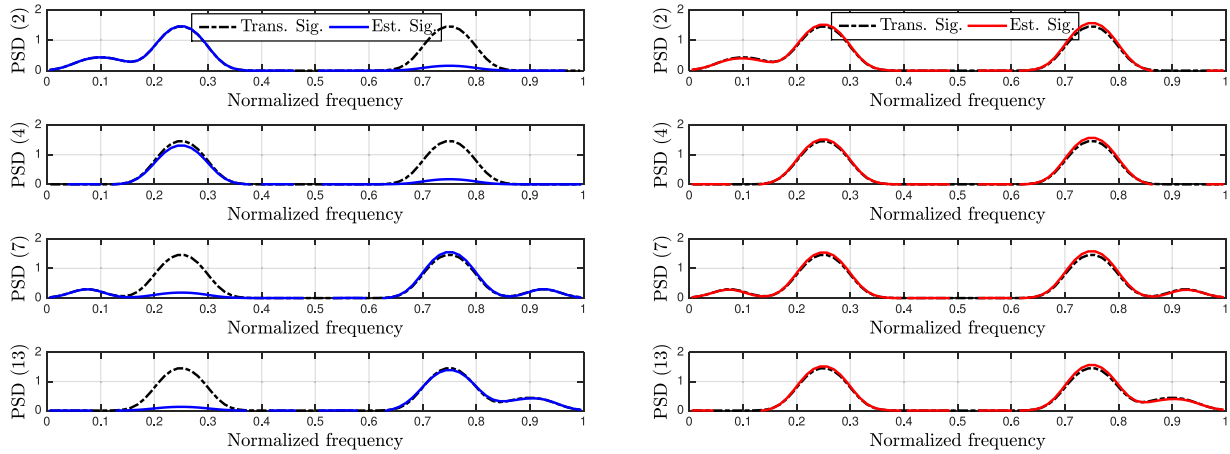


Fig. 11. PSD estimation for nodes 2 ($\mathcal{C}_1$), 4 ($\mathcal{C}_2$), 7 ($\mathcal{C}_3$), and 13 ($\mathcal{C}_4$). Left: noncooperating clusters (multitask strategy (26) with $\eta = 0$). Right: cooperating clusters (multitask strategy (26) with $\eta \neq 0$).

where the central frequencies f_m are uniformly distributed. The combination vectors are set to:

$$\begin{aligned}
\boldsymbol{\Upsilon}^o_{\mathcal{C}_1} &= [\mathbf{0}_{1\times 4}\ 1\ 1\ \mathbf{0}_{1\times 14}, \mathbf{0}_{1\times 14}\ 1\ 1\ \mathbf{0}_{1\times 4}, 0\ 0.3\ 0.3\ \mathbf{0}_{1\times 17}]^\top \\
\boldsymbol{\Upsilon}^o_{\mathcal{C}_2} &= [\mathbf{0}_{1\times 4}\ 1\ 1\ \mathbf{0}_{1\times 14}, \mathbf{0}_{1\times 14}\ 1\ 1\ \mathbf{0}_{1\times 4}, \mathbf{0}_{1\times 20}]^\top \\
\boldsymbol{\Upsilon}^o_{\mathcal{C}_3} &= [\mathbf{0}_{1\times 4}\ 1\ 1\ \mathbf{0}_{1\times 14}, \mathbf{0}_{1\times 14}\ 1\ 1\ \mathbf{0}_{1\times 4}, 0\ 0.3\ \mathbf{0}_{1\times 16}\ 0.3\ 0]^\top \\
\boldsymbol{\Upsilon}^o_{\mathcal{C}_4} &= [\mathbf{0}_{1\times 4}\ 1\ 1\ \mathbf{0}_{1\times 14}, \mathbf{0}_{1\times 14}\ 1\ 1\ \mathbf{0}_{1\times 4}, \mathbf{0}_{1\times 17}\ 0.3\ 0.3\ 0]^\top.
\end{aligned} \tag{107}$$

We consider $N_F = 80$ frequency samples. Based on the free propagation theory, we set the deterministic path loss factor $\overline{\ell}_{p,k}$ to the inverse of the squared distance between the transmitter p and the receiver k. At time instant i, we set $\ell_{p,k}(i) = \overline{\ell}_{p,k} + \delta\ell_{p,k}(i)$ with $\delta\ell_{p,k}(i)$ a zero-mean random Gaussian variable with standard deviation $0.1\overline{\ell}_{p,k}$. The secondary user k estimates $\ell_{p,k}(i)$ according to the following model:

$$\hat{\ell}_{p,k}(i) = \begin{cases} \overline{\ell}_{p,k}, & \text{if } \ell_{p,k}(i) > \ell_0, \\ 0, & \text{otherwise} \end{cases} \tag{108}$$

with ℓ_0 a threshold value. The same rule is used to set the path loss factor between the interference sources and the secondary users. We run the ATC diffusion algorithm (26) with the following adaptation step:

$$\boldsymbol{\psi}_k(i+1) = \boldsymbol{\Upsilon}_k(i) + \mu_k \hat{\boldsymbol{\Phi}}_k^\top(i)[\boldsymbol{r}_k(i) - \hat{\boldsymbol{\Phi}}_k(i)\boldsymbol{\Upsilon}_k(i)], \tag{109}$$

with $\boldsymbol{\Upsilon}_k(i)$ the estimate of $\boldsymbol{\Upsilon}^o_k$ at time instant i. The sampling noise $z_{k\ell,j}(i)$ is assumed to be a zero-mean random Gaussian variable with standard deviation 0.01. The combination coefficients $\{a_{\ell k}\}$ and regularization factors $\{\rho_{k\ell}\}$ are set in the same way as in the previous experimentation.

The MSD learning curves are averaged over 50 Monte-Carlo runs. We run the multitask diffusion LMS (26) in two different situations. In the first scenario, we do not allow any cooperation between clusters by setting $\eta = 0$. In the second scenario, we set the regularization strength η to 0.01 and we use the ℓ_1-norm as co-regularizing function. As can be seen in Fig. 10, the network MSD performance is significantly improved by cooperation among clusters. For comparison purposes, we also run the ATC D-NSPE strategy developed in [24] and the multitask diffusion strategy with ℓ_2-norm developed in [29]. For the ATC D-NSPE strategy we assume that nodes are aware that the first $N_P \times N_B$ components of the vector $\boldsymbol{\Upsilon}^o_k$ are of global interest to the whole network and that the remaining components are of common interest to the cluster $\mathcal{C}(k)$. The link weights $\{a_{\ell k}, c_{\ell k}, \rho_{k\ell}, a^w_{\ell k}, a^{\mathcal{C}(k)}_{\ell k}\}$ are set in the same manner as the experiment in Fig. 8. It can be observed from Fig. 10 that our strategy performs well without the need to know the parameters of global interest and the parameters of common interest during the learning process. Fig. 11 shows the estimated power spectrum density for nodes 2, 4, 7, and 13 when running the multitask diffusion strategy (26) with $\eta = 0$ (left) and $\eta = 0.01$ (right). In the left plot, we observe that the clusters are able to estimate their interference source. However, depending on the distance to the primary users, the secondary users do not always succeed in estimating the power spectrum transmitted by all active primary users. For example, clusters 1 and 2 are not able to estimate the power spectrum transmitted by PU2. As shown in the right plot, regardless of the distance between primary and secondary users, each secondary user is able to estimate the aggregated power spectrum transmitted by all the primary users and its own interference source by cooperating with nodes belonging to neighboring clusters.

V. Conclusion and Perspectives

In this work, we considered multitask learning problems over networks where the optimum parameter vectors to be estimated by neighboring clusters have a large number of similar entries and a relatively small number of distinct entries. It then becomes advantageous to develop distributed strategies that involve cooperation among adjacent clusters in order to exploit these similarities. A diffusion forward-backward splitting algorithm with ℓ_1-norm and reweighed ℓ_1-norm co-regularizers was derived to address this problem. A closed-form expression for the proximal operator was derived to achieve higher efficiency. Conditions on the step-sizes to ensure convergence of the algorithm in the mean and mean-square sense were derived. Finally, simulation results were presented to illustrate the benefit of cooperating to promote similarities between estimates. Future research efforts will be focused on exploiting other sparsity promoting

co-regularizers. Perspectives also include the derivation of other forms of cooperation depending on prior information.

REFERENCES

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

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

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

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

[5] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Found. Trends Mach. Learn.*, vol. 7, no. 4–5, pp. 311–801, 2014.

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

[7] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, vol. 3. Amsterdam, The Netherlands: Elsevier, 2014, pp. 322–454.

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

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

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

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

[12] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

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

[14] S. Kar and J. M. F. Moura, "Convergence rate analysis of distributed gossip (linear parameter) estimation: Fundamental limits and tradeoffs," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 674–690, Aug. 2011.

[15] S. Lee and A. Nedic, "Distributed random projection algorithm for convex optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 221–229, Apr. 2013.

[16] O. N. Gharehshiran, V. Krishnamurthy, and G. Yin, "Distributed energy-aware diffusion least mean squares: Game-theoretic learning," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 5, pp. 821–836, Oct. 2013.

[17] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *J. Optim. Theory Appl.*, vol. 147, no. 3, pp. 516–545, 2010.

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

[19] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *Proc. Int. Workshop Cogn. Inf. Process.*, Parador de Baiona, Spain, May 2012, pp. 1–6.

[20] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2733–2748, Jun. 2015.

[21] X. Zhao and A. H. Sayed, "Distributed clustering and learning over networks," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3285–3300, Jul. 2015.

[22] S. Khawatmi, A. M. Zoubir, and A. H. Sayed, "Decentralized clustering over adaptive networks," in *Proc. 23rd Eur. Signal Process. Conf.*, Nice, France, Aug. 2015, pp. 2696–2700.

[23] R. Abdolee, B. Champagne, and A. H. Sayed, "Estimation of space-time varying parameters using a diffusion LMS algorithm," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 403–418, Jan. 2014.

[24] J. Plata-Chaves, N. Bogdanović, and K. Berberidis, "Distributed diffusion-based LMS for node-specific adaptive parameter estimation," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3448–3460, Jul. 2015.

[25] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific adaptive parameter estimation," *IEEE Trans. Signal Process.*, vol. 62, no. 20, pp. 5382–5397, Oct. 2014.

[26] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks-Part I: Sequential node updating," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5277–5291, Oct. 2010.

[27] A. Bertrand and M. Moonen, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2196–2210, May 2011.

[28] J. Chen, C. Richard, A. O. Hero, and A. H. Sayed, "Diffusion LMS for multitask problems with overlapping hypothesis subspaces," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, Reims, France, Sep. 2014, pp. 1–6.

[29] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129–4144, Aug. 2014.

[30] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion adaptation over asynchronous networks," *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2835–2850, Jun. 2016.

[31] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion LMS with sparsity-based regularization," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Brisbane, Australia, Apr. 2015, pp. 3516–3520.

[32] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in H. H. Bauschke *et al.* (eds.) *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* (vol. 49 of Springer Optimization and Its Applications), Springer, New York, 2011, pp. 185–212.

[33] P. L. Combettes, D. Dũng, and B. C. Vũ, "Proximity for sums of composite functions," *J. Math. Anal. Appl.*, vol. 380, no. 2, pp. 680–688, 2011.

[34] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2013.

[35] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Statist. Soc.*, vol. 58, pp. 267–288, 1996.

[36] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2007.

[37] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Taipei, Taiwan, Apr. 2009, pp. 3125–3128.

[38] Y. Gu, J. Jin, and S. Mei, "ℓ_0-norm constraint LMS algorithm for sparse system identification," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 774–777, Sep. 2009.

[39] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted ℓ_1-balls," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 936–952, Mar. 2011.

[40] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Dallas, USA, Mar. 2010, pp. 3734–3737.

[41] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Sparse diffusion LMS for distributed adaptive estimation," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 3281–3284.

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

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

[44] M. W. Wee and I. Yamada, "A proximal splitting approach to regularized distributed adaptive estimation in diffusion networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, Canada, May 2013, pp. 5420–5424.

[45] P. Di Lorenzo, "Diffusion adaptation strategies for distributed estimation over Gaussian Markov random fields," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5748–5760, Nov. 2014.

[46] S. Vlaski and A. H. Sayed, "Proximal diffusion for stochastic costs with non-differentiable regularizers," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Brisbane, Australia, Apr. 2015, pp. 3352–3356.

[47] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "Sparsity-promoting adaptive algorithm for distributed learning in diffusion networks," in *Proc. 20th Eur. Signal Process. Conf.*, Bucharest, Romania, Aug. 2012, pp. 1084–1088.

[48] S. Chouvardas, G. Mileounis, N. Kalouptsidis, and S. Theodoridis, "A greedy sparsity-promoting LMS for distributed adaptive learning in diffusion networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, Canada, May 2013, pp. 5415–5419.

[49] G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh, "An adaptive greedy algorithm with application to nonlinear communications," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 2998–3007, Jun. 2010.
[50] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, 2008.
[51] W. Gao, J. Chen, C. Richard, J. Huang, and R. Flamary, "Kernel LMS algorithm with forward-backward splitting for dictionary learning," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, Canada, May 2013, pp. 5735–5739.
[52] H. Zou, "The adaptive lasso and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006.
[53] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS for clustered multitask networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 2014, pp. 5487–5491.
[54] B. T. Polyak, *Introduction to Optimization* (Optimization Software). New York, NY, USA: Springer, 1987.
[55] A. H. Sayed, *Adaptive Filters*. New York, NY, USA: Wiley, 2008.
[56] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, 2004.
[57] J. Chen, S. K. Ting, C. Richard, and A. H. Sayed, "Group diffusion LMS," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 4925–4929.
[58] J. Plata-Chaves, M. H. Bahari, M. Moonen, and A. Bertrand, "Unsupervised diffusion-based LMS for node-specific parameter estimation over wireless sensor networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016.
[59] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks," *IEEE Trans. Signal Process.*, vol. 61, no. 12, pp. 3183–3197, Jun. 2013.

Roula Nassif was born in Beirut, Lebanon. She received the bachelor's degree in electrical engineering from the Lebanese University, Lebanon, in 2013. She received the M.S. degrees in industrial control and intelligent systems for transport from the Lebanese University, Lebanon, and from Compiègne University of Technology, France, in 2013. Since October 2013, she is working toward the Ph.D. degree at the Lagrange Laboratory, University of Nice Sophia Antipolis, CNRS, Observatoire de la Côte d'Azur. Her research interest is focused on distributed optimization over multitask networks.

Cédric Richard (S'98–M'01–SM'07) received the Dipl.-Ing. and the M.S. degrees in 1994, and the Ph.D. degree in 1998, from Compiègne University of Technology, France, all in electrical and computer engineering. He is a Full Professor at the Université Côte d'Azur, France. He was a Junior Member of the Institut Universitaire de France in 2010–2015. His current research interests include statistical signal processing and machine learning. He is the author of more than 250 papers. He was the General Co-Chair of the IEEE SSP Workshop that was held in Nice, France, in 2011. He was the Technical Co-Chair of EUSIPCO 2015 that was held in Nice, France, and of the IEEE CAMSAP Workshop 2015 that was held in Cancun, Mexico. He serves as a Senior Area Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS since 2015. He is also an Associate Editor of *Signal Processing* Elsevier since 2009. He is a Member of the Machine Learning for Signal Processing (MLSP TC) Technical Committee, and served as a Member of the Signal Processing Theory and Methods (SPTM TC) Technical Committee in 2009–2014.

André Ferrari (SM'91–M'93) received the Ingénieur degree from École Centrale de Lyon, Lyon, France, in 1988 and the M.Sc. and Ph.D. degrees from the University of Nice Sophia Antipolis (UNS), France, in 1989 and 1992, respectively, all in electrical and computer engineering.

He is currently a Professor at UNS. He is a Member of the Joseph–Louis Lagrange Laboratory (CNRS, OCA), where his research activity is centered around statistical signal processing and modeling, with a particular interest in applications to astrophysics.

Ali H. Sayed (S'90–M'92–SM'99–F'01) is a Distinguished Professor and Past Chairman of electrical engineering at UCLA where he directs the UCLA Adaptive Systems Laboratory (www.ee.ucla.edu/asl). He an author or co-author of more than 480 scholarly publications and six books, his research involves several areas including adaptation and learning, system theory, statistical signal processing, network science, and information processing theories. His work has been recognized with several awards including the 2015 Education Award from the IEEE Signal Processing Society, the 2014 Papoulis Award from the European Association for Signal Processing, the 2013 Meritorious Service Award and the 2012 Technical Achievement Award from the IEEE Signal Processing Society, the 2005 Terman Award from the American Society for Engineering Education, the 2003 Kuwait Prize, and the 1996 IEEE Donald G. Fink Prize. He received several Best Paper Awards from the IEEE (2002, 2005, 2012, and 2014) and EURASIP (2015), and is a Fellow of the American Association for the Advancement of Science. He is recognized as a Highly Cited Researcher by Thomson Reuters. He is the President-Elect of the IEEE Signal Processing Society (2016–2017).