CHAPTER

Multitask learning over adaptive networks with grouping strategies

1

Jie Chen[1,a,*], **Cédric Richard**[2,**], **Shang Kee Ting**[†] **and Ali H. Sayed**[3,‡]

[*]*Northwestern Polytechnical University, School of Marine Science and Technology, Center of Intelligent Acoustics and Immersive Communications, Xi'an, China* [**]*Université Côte d'Azur, CNRS, OCA, Laboratoire Lagrange, Nice, France* [†]*DSO National Laboratories, Singapore* [‡]*Ecole Polytechnique Fédérale de Lausanne, Faculté des Sciences et Techniques de l'Ingénieur, Lausanne, Switzerland*

[a]*Corresponding:* dr.jie.chen@ieee.org

ABSTRACT

Considering groups of parameters, rather than parameters individually, can be beneficial for estimation accuracy if structural relationships between parameters exist (e.g., spatial, hierarchical or related to the physics of the problem). Group-sparsity inducing estimators are typical examples that benefit from such prior information. Building on this principle, we show that the diffusion LMS algorithm used for distributed inference over adaptive networks can be extended to deal with structured criteria built upon groups of variables, leading to a flexible framework that can encode various relationships in the parameters to estimate. We also introduce online strategies to group the parameters to estimate in an unsupervised manner, and to promote or inhibit collaborations between nodes depending if these groups are locally or globally applicable. Simulations illustrate the theoretical findings and the estimation strategies.

Keywords: Diffusion adaptation, distributed optimization, online learning, multi-task learning, group-based estimation

[1] The work of J. Chen was supported in part by NSFC grant 61671382.
[2] The work of C. Richard was supported in part by ANR and DGA under Grant ANR-13-ASTR-0030.
[3] The work of A. H. Sayed was supported in part by NSF grant CCF-1524250.

1.1 INTRODUCTION

A large variety of applications are network-structured and require adaptation to time-varying dynamics. Sensor networks, vehicular networks, communication networks, and power grids are some typical examples.

While centralized strategies can extract information from aggregated data more accurately, they nevertheless become prohibitive in large data scenarios and rely on a risky fusion-based architecture where failure of the single central processor can turn this solution unreliable. Distributed strategies are more robust and can be designed to process data in an online streaming fashion, thus avoiding the need to steer large amounts of raw information. Signal processing over networks has provided a powerful and convenient set of tools for such scenarios, allowing for efficient in-network learning and adaptation. Several strategies have been proposed in the literature, including incremental [1, 2, 3, 4], consensus [5, 6, 7], and diffusion strategies [8, 9, 10, 11, 12, 13, 14, 15]. Diffusion strategies are particularly attractive since they are scalable, robust, and enable continuous learning and adaptation in response to data drifts [16, 17, 18].

The working hypothesis for these earlier studies is that the nodes cooperate with each other to monitor a single process or to estimate a common parameter vector. We shall refer to problems of this type as single-task problems. Reaching consensus among the agents is critical for successful inference in these problems. Due to the increased heterogeneity in models and data types, there has been been growing interest in multi-task problems. Over multi-task networks, rather than promote consensus among all agents, the agents are allowed to track node-specific interests that happen to share some dependency relation with the interest of other agents. In this way, even though the objectives may be different, the agents can still benefit from cooperation. In [19, 20], the authors describe distributed node-specific estimation algorithms over fully connected networks or tree networks. In [21], the authors formalize the problem of adaptation and learning over multi-task networks. They devise a set of distributed online algorithms based on diffusion adaptation strategy. Extensions to asynchronous networks are considered in [22]. In [23], the performance of a single-task diffusion implementation is analyzed when it operates in a multi-task environment. An unsupervised clustering strategy that allows each agent to automatically select the neighboring agents with which it can collaborate is also introduced. In this scenario, the only available information is that clusters of nodes with common interests may exist in the network but nodes do not know which other nodes share the same interest. Other useful works have also addressed variations of this scenario in [24, 25, 26, 27]. In [28], the authors use multi-task diffusion adaptation as described in [21] with a node clustering strategy for studying the relation between the tremor intensity and the brain connectivity of Parkinson's patients. In [29], the authors derive a distributed strategy that allows each node in the network to locally adapt the intensity of cooperation with other nodes. The authors in [30] promote cooperation between clusters with ℓ_1-norm co-regularizers. The authors in [31, 32]

examine an alternative way to model relations between tasks by assuming that they all share a common latent feature representation. Variations of this scenario are addressed in [33]. In another scenario, it is assumed that there are parameters of global interest to all nodes in the network, a collection of parameters of common interest within sub-groups of nodes, and a set of parameters of local interest at each node [34, 35, 36]. In [37, 38], the optimum parameter vectors to be estimated by agents are related according to a set of constraints.

An inspection of the literature on diffusion adaptation over networks shows that, in most existing works, single-task and multi-task oriented algorithms fuse information from neighboring agents via weighted combinations of estimated parameter vectors. These combinations assign the same scaling weight to all entries in the combined iterates.There are situations, however, where some groups of entries within the iterate vectors should be weighted differently than other groups of entries within the same iterates. Consider an example where the top half of the entries of the parameter vectors to estimate are common across all agents, while the bottom half entries are randomly distributed without obvious relationship. Uniformly combining estimates may cause large estimation error due to the presence of significantly different entries.

Considering groups of variables, rather than variables individually, can be beneficial for estimation accuracy if structural relationships between variables exist (e.g., spatial, hierarchical or related to the physics of the problem). Group-sparsity inducing estimators are typical examples that benefit from such prior information. In this chapter, we build on this principle to show how diffusion LMS can be extended to deal with structured criteria involving groups of variables.

This chapter is organized as follows. Section 1.2 presents the network model and provides a brief review of diffusion LMS. The group diffusion LMS algorithm is devised in Section 1.3. Its stochastic behavior is analyzed for known groups of variables and fixed combination coefficients. Section 1.4 introduces unsupervised strategies for grouping the variables and setting the combination coefficients of the group diffusion LMS. In Section 1.5, experiments are conducted to validate the algorithms and theoretical findings. Section 1.6 concludes this chapter.

Notation. Normal font x denotes scalars. Boldface small letters $\boldsymbol{x}$ denote vectors. All vectors are column vectors. Boldface capital letters $\boldsymbol{X}$ denote matrices. The (k,ℓ)-th entry of a matrix is denoted by $(\cdot)_{k\ell}$, and the (k,ℓ)-th block of a block matrix is denoted by $[\,\cdot\,]_{k\ell}$. The superscript $(\cdot)^\top$ represents transpose of a matrix or a vector. The notation $\|\cdot\|$ denotes the ℓ_2-norm of its matrix or vector argument, while $\|\cdot\|_{\mathrm{b},\infty}$ denotes the block maximum norm of its block vector or matrix argument. Spectral radius of a square matrix is denoted by $\rho(\cdot)$. Matrix trace is denoted by trace$(\cdot)$. The operator col$\{\cdot\}$ stacks its vector arguments on the top of each other to generate a connected vector. The operator diag$\{\cdot\}$ formulates a (block) diagonal matrix with its arguments. Identity matrix of size $N\times N$ is denoted by $\boldsymbol{I}_N$. Kronecker product is denoted by $\otimes$, and expectation is denoted by $\mathbb{E}\{\cdot\}$. We denote by $\mathcal{N}_k$ the set of node indices in the neighborhood of node k, including k itself, and $|\mathcal{N}_k|$ its set cardinality.

1.2 NETWORK MODEL AND DIFFUSION LMS

1.2.1 NETWORK MODEL

Let us consider a connected network $G = (\mathcal{V}, \mathcal{E})$ defined by a set $\mathcal{V} = \{1, 2, \ldots, N\}$ of N agents, along with a set $\mathcal{E}$ of edges that are 2-element subsets of $\mathcal{V}$. We address the problem of estimating an $L \times 1$ unknown vector at each node k from streaming data collected over the network. At each time instant n, node k has access to time sequences $\{d_k(n), \boldsymbol{x}_{k,n}\}$, where $d_k(n)$ denotes the reference signal, and $\boldsymbol{x}_{k,n}$ represents an $L \times 1$ regression vector with covariance matrix $\boldsymbol{R}_{x,k} = \mathbb{E}\{\boldsymbol{x}_{k,n}\boldsymbol{x}_{k,n}^{\top}\} > 0$. We assume that the data are related via the linear model:

$$d_k(n) = \boldsymbol{w}_k^{\star\top}\boldsymbol{x}_{k,n} + z_k(n) \tag{1.1}$$

for all k, with $\boldsymbol{w}_k^{\star}$ an unknown parameter vector at node k, and $z_k(n)$ a zero-mean i.i.d. noise of variance $\sigma_{z,k}^2$ that is independent of every other signal. For determining the parameter vectors $\boldsymbol{w}_k^{\star}$, we consider the mean-square error criterion at each node k defined as:

$$J_k(\boldsymbol{w}_k) = \mathbb{E}\{|d_k(n) - \boldsymbol{x}_{k,n}^{\top}\boldsymbol{w}_k|^2\} \tag{1.2}$$

We shall refer to scenarios where all nodes estimate the same parameter vector, that is, $\boldsymbol{w}_1^{\star} = \cdots = \boldsymbol{w}_N^{\star}$, as *single-task* problems. Collaboration among nodes with standard distributed strategies can enhance the estimation performance over the network. On the contrary, we shall refer to cases where nodes may estimate distinct parameter vectors, namely, cases where the $\{\boldsymbol{w}_k^{\star}\}_{k=1}^N$ are not necessarily the same, as *multi-task* problems. Still, we assume that similarities exist in some sense among these parameter vectors. Otherwise the estimation problem would be node-independent and would reduce to the non-cooperative setting.

1.2.2 A BRIEF REVIEW OF DIFFUSION LMS

Before introducing the diffusion strategy at the group level, we provide a brief review of standard diffusion LMS derived for single-task scenarios. The goal of this algorithm is to minimize the following global cost function in a distributed manner for an enhanced estimation performance over a non-cooperative strategy:

$$J^{\text{glob}}(\boldsymbol{w}) = \sum_{k=1}^{N} J_k(\boldsymbol{w}) \tag{1.3}$$

We denote the minimizer of (1.3) by $\boldsymbol{w}^{\star}$. Minimizing (1.3) over $\boldsymbol{w}$ with J_k defined by the mean-square error (1.2) is equivalent to minimizing the following alternative cost [12, 13]:

$$J^{\text{glob}'}(\boldsymbol{w}) = J_k(\boldsymbol{w}) + \sum_{\ell \neq k} \|\boldsymbol{w} - \boldsymbol{w}^{\star}\|_{\boldsymbol{R}_{x,\ell}}^2 \tag{1.4}$$

To bypass the unknown second-order statistics $\boldsymbol{R}_{x,\ell}$, one can rely on the Rayleigh-Ritz characterization of eigenvalues to approximate the weighted norm in (1.4) by a scaled unweighted norm [12, 13], say as,

$$\|\boldsymbol{w}-\boldsymbol{w}^{\star}\|^2_{\boldsymbol{R}_{x,\ell}} \approx b_{\ell k}\,\|\boldsymbol{w}-\boldsymbol{w}^{\star}\|^2 \tag{1.5}$$

for some nonnegative coefficients $b_{\ell k}$. This leads to the following modified cost function at node k:

$$J^{\text{glob}''}(\boldsymbol{w}) = J_k(\boldsymbol{w}) + \sum_{\ell\neq k} b_{\ell k}\,\|\boldsymbol{w}-\boldsymbol{w}^{\star}\|^2 \tag{1.6}$$

Calculating the gradient vector of (1.6), restricting communication to immediate neighbors, and using approximation (1.5) along with the arguments from [13], we arrive at the adapt-then-combine (ATC) strategy without raw data exchange [9]:

$$\boldsymbol{\psi}_{k,n} = \boldsymbol{w}_{k,n-1} + \mu\,\boldsymbol{x}_{k,n}\,[d_k(n) - \boldsymbol{w}_{k,n-1}^{\top}\boldsymbol{x}_{k,n}] \tag{1.7a}$$

$$\boldsymbol{w}_{k,n} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\,\boldsymbol{\psi}_{\ell,n} \tag{1.7b}$$

where μ is a small positive step-size. The combine-then-adapt (CTA) form can be derived in a similar way; it is sufficient for our purposes to continue with the ATC form (1.7). The coefficients $\{a_{\ell k}\}$ in the above algorithm are given by:

$$a_{kk} = 1-\mu \sum_{\ell\in\mathcal{N}_k\backslash\{k\}} b_{\ell k} \tag{1.8}$$

$$a_{\ell k} = \mu\, b_{\ell k}, \qquad \ell\in\mathcal{N}_k\backslash\{k\} \tag{1.9}$$

$$a_{\ell k} = 0, \qquad \ell\notin\mathcal{N}_k \tag{1.10}$$

In practice, the coefficients $\{a_{\ell k}\}$ are usually treated as free weighting parameters to be chosen by the designer. That is, it is not necessary to worry about selecting the coefficients $\{b_{\ell k}\}$. It is sufficient to select the $\{a_{\ell k}\}$ as nonnegative convex combination coefficients satisfying:

$$a_{\ell k}\geq 0, \quad \sum_{\ell\in\mathcal{N}_k} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell\notin\mathcal{N}_k \tag{1.11}$$

The selection of the $\{a_{\ell k}\}$ has a significant impact on the performance of the algorithm for both single and multi-task scenarios [13, 14, 15, 23, 25].

1.3 GROUP DIFFUSION LMS

1.3.1 MOTIVATION

It is explained in [13] how (1.5) leads to the fusion (1.7b) of local estimates in the neighborhood of each node. Note now that all the entries of the intermediate estimate $\boldsymbol{\psi}_{\ell,n}$ are scaled by the same weight $a_{\ell k}$. Figure 1.1 illustrates one possible limitation

of uniform scaling of the entries and why grouping can be useful in some important situations. For example, in the figure, adjacent nodes k and ℓ are estimating parameter vectors $\boldsymbol{w}_k^\star$ and $\boldsymbol{w}_\ell^\star$ whose entries are grouped into three separate sets: both vectors have the same entries in the first group, they significantly differ in the second group due to sensor failure for instance, and only differ slightly in the third group due to sensor drift. It is not suitable to view this scenario neither as a single-task problem, nor as a multi-task problem, with a single set of combination weights $a_{\ell k}$. A small combination weight may not be sufficient to promote the closeness of entries in the first and third groups, whereas a large combination weight may lead to a large estimation bias caused by the second group.

This example motivates us to introduce a grouping strategy. More generally, let M be a positive integer less than or equal to L, and let $\{\mathcal{G}_m\}_{m=1}^M$ be a partition of the set of indexes $\mathcal{G} = \{1, \ldots, L\}$, namely,

$$\bigcup_{m=1}^{M} \mathcal{G}_m = \mathcal{G}, \quad \mathcal{G}_m \cap \mathcal{G}_{m'} = \varnothing \ \text{ if } \ m \neq m'. \tag{1.12}$$

We also let $\boldsymbol{w}_{\mathcal{G}_m}$ or $[\boldsymbol{w}]_{\mathcal{G}_m}$ denote a sub-vector of $\boldsymbol{w}$ indexed by $\mathcal{G}_m$. In the case of Fig. 1.1(b), these are the sub-vectors that correspond to the groups $\mathcal{G}_1, \mathcal{G}_2$ and $\mathcal{G}_3$. We can then assign larger combination weights to the first group, smaller or even null-valued weights to the second group, and medium-value weights to the third group. Such grouping strategy ends up exploiting the structure of the parameter vectors more fully. However, since information on the internal group structures may not be available beforehand, one possible strategy is to split parameter vectors into a number of groups of preset lengths and assign a combination coefficient to each group, as illustrated in Fig. 1.1(c). In the sequel, we shall describe an unsupervised adaptive strategy to estimate the parameter vectors in these scenarios in an online manner. Note that the parameter vector entries within each group need not be necessarily contiguous. In the scope of this chapter, we shall only focus on homogeneous groups of entries across the network, namely, we shall assume that the parameter vectors at all nodes possess the same grouping structure across the network. While heterogeneous group models are able to represent more complex application scenarios, it will require further notation and a more complex algorithm development.

1.3.2 GROUP DIFFUSION LMS ALGORITHM

We now motivate the group diffusion LMS from the single-task derivation by approximating the second-order statistics $\boldsymbol{R}_{x,\ell}$ in an alternative manner. Inspecting (1.5), we now assign a scaling factor to each group of entries instead of using a single factor, i.e., we now use

$$\|\boldsymbol{w} - \boldsymbol{w}^\star\|_{\boldsymbol{R}_{x,\ell}}^2 \approx \sum_{m=1}^{M} b_{\ell k,m} \|\boldsymbol{w}_{\mathcal{G}_m} - \boldsymbol{w}_{\mathcal{G}_m}^\star\|^2 \tag{1.13}$$

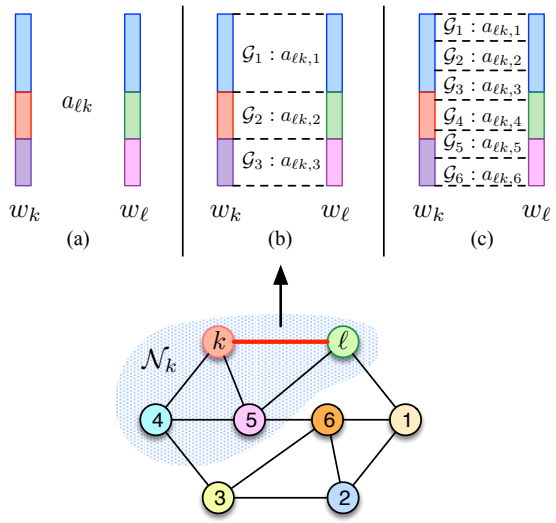


FIGURE 1.1

(a) Parameter vector structures for nodes k and ℓ: 3 sets of entries have different levels of similarity, encoded by colors. (b) A scenario with 3 groups; (c) a second scenario with 6 groups.

where $b_{\ell k,m}$ is the nonnegative weight for group m. The global cost (1.6) is then relaxed as follows:

$$J^{\text{glob}''}(\boldsymbol{w}) = J_k(\boldsymbol{w}) + \sum_{\ell \neq k} \sum_{m=1}^{M} b_{\ell k,m} \|\boldsymbol{w}_{\mathcal{G}_m} - \boldsymbol{w}^{\star}_{\mathcal{G}_m}\|^2 \tag{1.14}$$

Calculating the gradient vector of (1.14), following the same steps as for diffusion LMS, and introducing the following combination weights $a_{\ell k,m}$ for each group m:

$$a_{kk,m} = 1 - \mu \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} b_{\ell k,m} \tag{1.15}$$

$$a_{\ell k,m} = \mu\, b_{\ell k,m}, \qquad \ell \in \mathcal{N}_k \backslash \{k\} \tag{1.16}$$

$$a_{\ell k,m} = 0, \qquad \ell \notin \mathcal{N}_k, \tag{1.17}$$

we arrive at the group diffusion LMS algorithm:

$$\boldsymbol{\psi}_{k,n} = \boldsymbol{w}_{k,n-1} + \mu\, \boldsymbol{x}_{k,n}\, (d_k(n) - \boldsymbol{x}_{k,n}^{\top} \boldsymbol{w}_{k,n-1}) \tag{1.18a}$$

$$[\boldsymbol{w}_{k,n}]_{\mathcal{G}_m} = \sum_{\ell \in \mathcal{N}_k} a_{\ell k,m} [\boldsymbol{\psi}_{\ell,n}]_{\mathcal{G}_m}, \;\; \text{for } m = 1, \ldots, M. \tag{1.18b}$$

Parameters $\bigcup_{m=1}^{M}\{a_{\ell k,m}\}$ can be adjusted by the users. Each subset $\{a_{\ell k,m}\}$ now forms a left-stochastic matrix $\boldsymbol{A}_m$, i.e., for $m = 1, \ldots, M$

$$a_{\ell k,m} \geq 0, \quad \sum_{\ell \in \mathcal{N}_k} a_{\ell k,m} = 1, \quad a_{\ell k,m} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{1.19}$$

Appropriate selection of these coefficients can enhance the performance of diffusion LMS, especially for scenarios with structural relationships within groups. In Section 1.4.3, we shall introduce an unsupervised strategy to adjust these weights in an online manner. One earlier version of the group diffusion strategy (1.18a)–(1.18b) was introduced in [39] and applied there to the problem A/D converters tuning. In that application, the combination weights $\{a_{\ell k,m}\}$ were selected proportionally to the SNR conditions within relevant frequency bands.

1.3.3 NETWORK BEHAVIOR

We now study the behavior of the group diffusion LMS algorithm (1.18) with constant combination weights $a_{\ell k,m}$ that satisfy conditions (1.19). To proceed, we collect the information from across the network into block vectors and matrices. In particular, we denote by $\boldsymbol{w}_n$ and $\boldsymbol{w}^\star$ the stacked weight estimate vector and the stacked optimum weight vector, respectively:

$$\boldsymbol{w}_n = \text{col}\{\boldsymbol{w}_{1,n}, \ldots, \boldsymbol{w}_{N,n}\} \tag{1.20}$$

$$\boldsymbol{w}^\star = \text{col}\{\boldsymbol{w}_1^\star, \ldots, \boldsymbol{w}_N^\star\} \tag{1.21}$$

We consider the case where the $\boldsymbol{w}_k^\star$ are distinct. The weight error vector $\widetilde{\boldsymbol{w}}_{k,n}$ for each node k at iteration n is defined by:

$$\widetilde{\boldsymbol{w}}_{k,n} = \boldsymbol{w}_{k,n} - \boldsymbol{w}_k^\star \tag{1.22}$$

These error vectors $\widetilde{\boldsymbol{w}}_{k,n}$ are also stacked on top of each other to get the vector:

$$\widetilde{\boldsymbol{w}}_n = \text{col}\{\widetilde{\boldsymbol{w}}_{1,n}, \ldots, \widetilde{\boldsymbol{w}}_{N,n}\} \tag{1.23}$$

We assume that the regression vectors $\boldsymbol{x}_{k,n}$ arise from a zero-mean random process that is temporally (over n) stationary, white, and independent over space (over k) with $\boldsymbol{R}_{x,k} = \mathbb{E}\{\boldsymbol{x}_k(n)\,\boldsymbol{x}_k^\top(n)\} > 0$. This independence assumption is widely used in the analysis of adaptive learning systems [40, App. 24.A], [14, Chs. 10-11].

1.3.3.1 Mean weight behavior analysis

Subtracting optimum vectors $\boldsymbol{w}_k^\star$ from both sides of the adaptation equation (1.18a), and using

$$d_k(n) - \boldsymbol{x}_{k,n}^\top \boldsymbol{w}_{k,n-1} = z_k(n) - \boldsymbol{x}_{k,n}^\top \widetilde{\boldsymbol{w}}_{k,n-1} \tag{1.24}$$

gives

$$\boldsymbol{\psi}_{k,n} - \boldsymbol{w}_k^\star = \widetilde{\boldsymbol{w}}_{k,n-1} - \mu\,\boldsymbol{x}_{k,n}\boldsymbol{x}_{k,n}^\top \widetilde{\boldsymbol{w}}_{k,n-1} + \mu\,\boldsymbol{x}_{k,n} z_k(n) \tag{1.25}$$

Before establishing the relation between the weight error vectors $\widetilde{\boldsymbol{w}}_n$ and $\widetilde{\boldsymbol{w}}_{n-1}$, it is convenient to introduce the $N \times N$ block matrix

$$\mathcal{A} = \begin{pmatrix} \mathcal{A}_{11} & \cdots & \mathcal{A}_{1N} \\ \vdots & \ddots & \vdots \\ \mathcal{A}_{N1} & \cdots & \mathcal{A}_{NN} \end{pmatrix} \tag{1.26}$$

Each block $\mathcal{A}_{\ell k}$ is an $L \times L$ diagonal matrix whose i-th diagonal entry is $a_{\ell k,m}$, where m refers to the subset of indexes $\mathcal{G}_m$ to which index i belongs. In the single task case, expression (1.26) reduces to matrix $\mathcal{A} = \boldsymbol{A} \otimes \boldsymbol{I}_N$ considered in [14, Ch. 8] for analyzing the convergence behavior of diffusion LMS, with $(\boldsymbol{A})_{\ell k} = a_{\ell k}$.

Matrix $\mathcal{A}$ can also be expressed as follows:

$$\mathcal{A} = (\boldsymbol{A}_1 \otimes \boldsymbol{J}_1) + \cdots + (\boldsymbol{A}_M \otimes \boldsymbol{J}_M) \tag{1.27}$$

where $\boldsymbol{J}_m$ is an $L \times L$ diagonal matrix with diagonal entries defined as:

$$(\boldsymbol{J}_m)_{ii} = 1, \quad \text{if } i \in \mathcal{G}_m \tag{1.28}$$

$$(\boldsymbol{J}_m)_{ii} = 0, \quad \text{otherwise} \tag{1.29}$$

Since the weights $a_{\ell k,m}$ satisfy condition (1.19), i.e., each matrix $\boldsymbol{A}_m$ is left-stochastic, matrix $\mathcal{A}$ is also left-stochastic.

With the above matrix $\mathcal{A}$, it can be verified that:

$$\widetilde{\boldsymbol{w}}_n = \mathcal{A}^{\top} (\boldsymbol{\psi}_n - \boldsymbol{w}^{\star}) + (\mathcal{A}^{\top} - \boldsymbol{I})\, \boldsymbol{w}^{\star} \tag{1.30}$$

where $\boldsymbol{\psi}_n = \text{col}\{\boldsymbol{\psi}_{1,n}, \cdots, \boldsymbol{\psi}_{N,n}\}$. Using (1.25), we write:

$$\widetilde{\boldsymbol{w}}_n = \mathcal{A}^{\top} (\boldsymbol{I} - \mu\, \mathcal{R}_{x,n}) \widetilde{\boldsymbol{w}}_{n-1} + \mu\, \mathcal{A}^{\top} \boldsymbol{p}_{xz,n} + (\mathcal{A}^{\top} - \boldsymbol{I}) \boldsymbol{w}^{\star} \tag{1.31}$$

with $\mathcal{R}_{x,n} = \text{diag}\{\boldsymbol{x}_{1,n}\boldsymbol{x}_{1,n}^{\top}, \ldots, \boldsymbol{x}_{N,n}\boldsymbol{x}_{N,n}^{\top}\}$ and $\boldsymbol{p}_{xz,n} = \{\boldsymbol{x}_{1,n} z_1(n), \ldots, \boldsymbol{x}_{N,n} z_N(n)\}$. Taking the expectation of both sides of (1.31) and using the independence assumption, we arrive at the mean behavior equation of the group diffusion LMS algorithm:

$$\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\} = \mathcal{A}^{\top} (\boldsymbol{I} - \mu\, \mathcal{R}_x)\, \mathbb{E}\{\widetilde{\boldsymbol{w}}_{n-1}\} + (\mathcal{A}^{\top} - \boldsymbol{I})\, \boldsymbol{w}^{\star} \tag{1.32}$$

with $\mathcal{R}_x = \text{diag}\{\boldsymbol{R}_{x,1}, \ldots, \boldsymbol{R}_{x,N}\}$. We shall now provide a condition on μ to guarantee the stability of (1.32).

The convergence of (1.32) is determined by the stability of $\mathcal{A}^{\top} (\boldsymbol{I} - \mu\, \mathcal{R}_x)$. Algorithm parameters should be chosen to satisfy the mean stability condition:

$$\rho(\mathcal{A}^{\top}(\boldsymbol{I} - \mu\, \mathcal{R}_x)) < 1. \tag{1.33}$$

where $\rho(\cdot)$ denotes spectral radius of its matrix argument. Let us first focus on matrix $\mathcal{A}$. Let $\boldsymbol{x} = \text{col}\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ be an arbitrary $L \times 1$ block vector whose individual entries $\{\boldsymbol{x}_k\}$ are vectors of size $L \times 1$ each. Considering (1.27), using that $\sum_{i=1}^{N} a_{ji,m} = 1$

with $a_{ji,m} \geq 0$, and Jensen's inequality, we have:

$$\left\| \left(\sum_{m=1}^{M} \boldsymbol{A}_m^{\top} \otimes \boldsymbol{J}_m \right) \boldsymbol{x} \right\|^2 = \sum_{i=1}^{N} \left\| \sum_{m=1}^{M} \sum_{j=1}^{N} a_{ji,m} \boldsymbol{J}_m \boldsymbol{x}_j \right\|^2$$
$$\leq \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{j=1}^{N} a_{ji,m} \left\| \boldsymbol{J}_m \boldsymbol{x}_j \right\|^2$$
$$= \sum_{m=1}^{M} \sum_{j=1}^{N} \left\| \boldsymbol{J}_m \boldsymbol{x}_j \right\|^2 \tag{1.34}$$

Matrix $\boldsymbol{J}_m$ is actually an orthogonal projection matrix that sets to 0 in (1.34) the entries of $\boldsymbol{x}_j$ that are not indexed by $\mathcal{G}_m$. Since $\{\mathcal{G}_m\}_{m=1}^{M}$ is a partition of the set of indexes, we have:

$$\sum_{m=1}^{M} \sum_{j=1}^{N} \left\| \boldsymbol{J}_m \boldsymbol{x}_j \right\|^2 = \sum_{j=1}^{N} \left\| \boldsymbol{x}_j \right\|^2 = \|\boldsymbol{x}\|^2 \tag{1.35}$$

We conclude that

$$\left\| \mathcal{A}^{\top} \right\| = \left\| \sum_{m=1}^{M} \boldsymbol{A}_m^{\top} \otimes \boldsymbol{J}_m \right\|^2 \leq 1 \tag{1.36}$$

We know that the spectral radius of any matrix $\boldsymbol{X}$ satisfies $\rho(\boldsymbol{X}) \leq \|\boldsymbol{X}\|$, for any induced norm. Applying this to $\mathcal{A}^{\top}(\boldsymbol{I} - \mu \mathcal{R}_x)$, we have:

$$\rho(\mathcal{A}^{\top}(\boldsymbol{I} - \mu \mathcal{R}_x)) \leq \left\| \mathcal{A}^{\top} \right\| \left\| \boldsymbol{I} - \mu \mathcal{R}_x \right\| \tag{1.37}$$
$$\leq \left\| \boldsymbol{I} - \mu \mathcal{R}_x \right\| \tag{1.38}$$

It then follows that the group diffusion LMS asymptotically converges in the mean, for any initial condition, if the step-size satisfies:

$$0 < \mu < \frac{2}{\max_k \lambda_{\max}(\boldsymbol{R}_{x,k})} \tag{1.39}$$

Setting $n \longrightarrow \infty$ in (1.32) leads to the the asymptotic mean bias expression:

$$\widetilde{\boldsymbol{w}}_{\infty} = \left[\boldsymbol{I} - \mathcal{A}^{\top}(\boldsymbol{I} - \mu \mathcal{R}_x) \right]^{-1} (\mathcal{A}^{\top} - \boldsymbol{I}) \, \boldsymbol{w}^{\star}. \tag{1.40}$$

1.3.3.2 Mean-square error behavior analysis

We shall now perform a mean-square error analysis of the group diffusion LMS. The purpose of this analysis is to evaluate how the variance $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_n\|^2\}$ evolves with time. This analysis is based on the energy conservation framework used in [13, 23, 32], which starts from the weight-error vector recursion in the compact form:

$$\widetilde{\boldsymbol{w}}_n = \boldsymbol{B}_n \, \widetilde{\boldsymbol{w}}_{n-1} - \boldsymbol{g}_n - \boldsymbol{r} \tag{1.41}$$

with the transition matrix:

$$\boldsymbol{B}_n = \mathcal{A}^\top(\boldsymbol{I} - \mu\,\mathcal{R}_{x,n}) \tag{1.42}$$

the stochastic driving term:

$$\boldsymbol{g}_n = \mu\mathcal{A}^\top \boldsymbol{p}_{xz,n} \tag{1.43}$$

and the constant driving term:

$$\boldsymbol{r} = (\mathcal{A}^\top - \boldsymbol{I})\,\boldsymbol{w}^\star \tag{1.44}$$

The expected values of the stochastic quantities (1.42)–(1.43) are given by:

$$\boldsymbol{B} = \mathcal{A}^\top(\boldsymbol{I} - \mu\,\mathcal{R}_x) \tag{1.45}$$

$$\boldsymbol{g} = \boldsymbol{0}_{NL} \tag{1.46}$$

We define the matrix

$$\boldsymbol{K} = \mathbb{E}\{\boldsymbol{B}_n^\top \otimes \boldsymbol{B}_n^\top\} \tag{1.47}$$

and approximate it by $\boldsymbol{K} \approx \boldsymbol{B}^\top \otimes \boldsymbol{B}^\top$ for sufficiently small step sizes. We also define:

$$\boldsymbol{G} = \mathbb{E}\{\boldsymbol{g}_n\boldsymbol{g}_n^\top\} = \mu^2\mathcal{A}^\top \operatorname{diag}\{\sigma_{z,1}^2\boldsymbol{R}_{x,1},\ldots,\sigma_{z,n}^2\boldsymbol{R}_{x,N}\}\,\mathcal{A}. \tag{1.48}$$

We skip the derivations here and refer instead to [24,33]. Following similar arguments, the following statements can be justified.

Theorem 1. *(**Mean-square stability**) Consider the data model* (1.1) *and assume the independence assumption holds. The group diffusion strategy* (1.18) *is mean-square stable when the matrix* $\boldsymbol{K}$ *defined by* (1.47)*, or its approximation, is stable. This condition is satisfied by sufficiently small step-sizes.*

□

Theorem 2. *(**Network learning curve**) Consider the same setting of Theorem 1 and let* $\zeta_n \triangleq \mathbb{E}\{\|\widetilde{\boldsymbol{w}}_n\|^2/N\}$ *denote the average network mean-square deviation (MSD) at time n. Then, the learning curve of the network corresponds to the evolution of* ζ_n *with time and is described by the following recursion over* $n \geq 0$*:*

$$\begin{aligned}\zeta_{n+1} = \zeta_n + \Big[&(vec\{\boldsymbol{G}^\top\})^\top\boldsymbol{K}^n\boldsymbol{\sigma}_I + \|\boldsymbol{r}\|^2_{\boldsymbol{K}^n\boldsymbol{\sigma}_I} - \|\widetilde{\boldsymbol{w}}_0\|^2_{(\boldsymbol{I}_{(NL)^2}-\boldsymbol{K})\boldsymbol{K}^n\boldsymbol{\sigma}_I}\\ &- 2\left[\boldsymbol{\gamma}_n^\top + (\boldsymbol{B}\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\})^\top \otimes \boldsymbol{r}^\top\right]\boldsymbol{\sigma}_I\Big]\end{aligned} \tag{1.49}$$

$$\boldsymbol{\gamma}_{n+1} = \boldsymbol{K}^\top\boldsymbol{\gamma}_n + (\boldsymbol{K} - \boldsymbol{I}_{(NL)^2})^\top(\boldsymbol{B}\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\} \otimes \boldsymbol{r}) \tag{1.50}$$

with $\boldsymbol{\sigma}_I = vec\{\frac{1}{N}\boldsymbol{I}_{NL}\}$, $\zeta_0 = \frac{1}{N}\|\widetilde{\boldsymbol{w}}_0\|^2$, $\boldsymbol{\gamma}_0 = \boldsymbol{0}_{(NL)^2\times 1}$.

□

Theorem 3. *(**Steady-state MSD**) Consider the same setting of Theorem 1. The*

steady-state MSD of the group diffusion strategy (1.18) *is given by:*

$$\zeta_\infty = (vec\{\boldsymbol{G}^\top\})^\top \boldsymbol{\sigma} + \boldsymbol{r}^\top \boldsymbol{\Sigma}(\boldsymbol{r} - 2\boldsymbol{B}\,\widetilde{\boldsymbol{w}}_\infty\}) \tag{1.51}$$

with $\widetilde{\boldsymbol{w}}_\infty$ *determined by* (1.40)*, and* $vec\{\boldsymbol{\Sigma}\} = \boldsymbol{\sigma} = \frac{1}{N}(\boldsymbol{I}_{(NL)^2} - \boldsymbol{K})^{-1} vec\{\boldsymbol{I}_{NL}\}$.

□

Although Theorems 2 and 3 provide closed-form expressions for the network MSD and steady-state MSD, it may not be practical to evaluate (1.49)–(1.51) due to the size of the matrices involved, which have dimensions $(NL)^2 \times (NL)^2$. In what follows, we derive equivalent but more compact expressions with matrices of size $NL \times NL$ (see the proof in Appendix 1).

Corollary 1. ***(Alternative transient MSD expression)*** *Consider the same setting of Theorem 1. The MSD learning curve of the group diffusion strategy* (1.18)*, provided by Theorem 2, can be equivalently expressed as follows:*

$$\begin{aligned}\zeta_{n+1} = \zeta_n + \frac{1}{N} trace\Big(&[\boldsymbol{G} + \boldsymbol{r}\boldsymbol{r}^\top]\boldsymbol{B}^{n\top}\boldsymbol{B}^n \\ &- \widetilde{\boldsymbol{w}}_0\widetilde{\boldsymbol{w}}_0^\top\,[\boldsymbol{B}^{n\top}\boldsymbol{B}^n - \boldsymbol{B}^{n+1\top}\boldsymbol{B}^{n+1}] - 2\boldsymbol{\Gamma}_n - 2\boldsymbol{B}\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\}\boldsymbol{r}^\top\Big)\end{aligned} \tag{1.52}$$

$$\boldsymbol{\Gamma}_{n+1} = \boldsymbol{B}\boldsymbol{\Gamma}_n\boldsymbol{B}^\top + \boldsymbol{B}\boldsymbol{r}\,(\boldsymbol{B}^2\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\})^\top - \boldsymbol{r}(\boldsymbol{B}\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\})^\top \tag{1.53}$$

with $\zeta_0 = \frac{1}{N}\|\widetilde{\boldsymbol{w}}_0\|^2$ *and* $\boldsymbol{\Gamma}_0 = \boldsymbol{0}_{NL}$.

□

Corollary 2. ***(Alternative steady-state MSD expression)*** *Consider the same setting of Theorem 1. The steady-state MSD of the group diffusion strategy* (1.18)*, provided by Theorem 3, can be equivalently expressed as follows:*

$$\zeta_\infty = \sum_{n=0}^{\infty} \boldsymbol{B}^n\,(\boldsymbol{G} + (\boldsymbol{r} - 2\boldsymbol{B}\widetilde{\boldsymbol{w}}_\infty)\,\boldsymbol{r}^\top)\,(\boldsymbol{B}^n)^\top. \tag{1.54}$$

Expression (1.54) is obtained by performing a series expansion of (1.51).

1.4 GROUPING STRATEGIES

In many practical cases, information about the group structure is not available beforehand. It is thus necessary to devise grouping strategies to endow agents with ability to partition the estimated parameter vectors and to associate appropriate combination weights to each group.

1.4.1 FIXED GROUPING STRATEGY

A simple strategy is to split parameter vectors into a number of contiguous groups with preset lengths, possibly equal, and then assign a combination coefficient to each group, as illustrated in Fig. 1.1(c). Splitting the parameter vector into subvectors can improve the performance for some applications, especially when there exist correlations among adjacent entries. However, this strategy may fail with particular configurations. For instance, consider the case where only the odd entries of the parameter vectors show some correlation. No matter how the group length is set, the algorithm will not be able to benefit from a uniform grouping strategy except perhaps if the group size is set to one. This motivates us to derive smart adaptive grouping strategies.

1.4.2 ADAPTIVE GROUPING STRATEGY

Adaptive grouping can be viewed as a clustering problem where we need to assign a label to each entry of a parameter vector. Before proceeding with the derivation, it is important to keep in mind that, since we are considering algorithms with linear complexity (LMS-type algorithms) within the context of online learning and distributed adaptation, grouping/clustering should neither be performed in a centralized manner nor significantly increase the computational complexity. In other words, deriving a grouping strategy with quadratic complexity would not make much sense in this context. This constraint rules out most clustering algorithms used in machine learning and data analysis, e.g., hierarchical clustering, k-means or spectral clustering. In what follows, we introduce a simple but efficient strategy. As this strategy is time independent, we shall omit the time index in notation for the sake of simplicity.

We start by introducing the following quantity that characterizes the deviation between the intermediate estimates defined in (1.18a) at nodes k and ℓ:

$$\delta_{k\ell,i} = \left|(\boldsymbol{\psi}_k)_i - (\boldsymbol{\psi}_\ell)_i\right| \quad \text{for} \quad \ell \in \mathcal{N}_k. \tag{1.55}$$

for $i = 1, \ldots, L$. By averaging pairwise quantities $\delta_{k\ell,i}$ within the neighborhood, we then associate with each node k the following L quantities:

$$\delta_{k,i} = \frac{1}{|\mathcal{N}_k|} \sum_{\ell \in \mathcal{N}_k} \delta_{k\ell,i} \quad \text{for} \quad i = 1, \ldots, L. \tag{1.56}$$

Each $\delta_{k,i}$ shows how the i-th entry of $\boldsymbol{\psi}_k$ deviates from those of its neighbors. Entries with similar $\delta_{k,i}$ can be assigned to the same group since they have a similar average contrast level with respect to their neighbors. In this way, groups of entries with small (resp., large) contrast will lead the multi-task diffusion algorithm to adopt a consensus (resp., non-cooperative) strategy over these entries. Note that each node k can calculate $\delta_{k,i}$ in (1.56) after collecting the estimates from its neighbors, after the adaptation step (1.18a). We now propose the following steps to generate the groups:

1. Sort $\{\delta_{k,i}\}_{i=1}^{L}$ in ascending order to obtain the ordered sequence $\{\overline{\delta}_{k,i}\}_{i=1}^{L}$;

2. Generate the difference sequence $\Delta_{k,i} = \overline{\delta}_{k,i+1} - \overline{\delta}_{k,i}$, and determine the $K-1$ largest values $\Delta_{k,i}$ to split $\{\overline{\delta}_{k,i}\}_{i=1}^{L}$ into K sections where the largest changes occur;
3. Form a group with the original entries $\delta_{k,i}$ that are in a same section. Repeat this operation with the K sections determined in Step 2.

The computational complexity of this procedure is dominated by the sorting operation in Step 1. A complexity of $O(L \log L)$ can be achieved with an efficient sorting algorithm. This grouping strategy derives from a vertex clustering algorithm commonly used in the literature [41]. Indeed, consider a fully connected graph with L vertices associated to the L entries of the local estimate $\boldsymbol{\psi}_k$. The edge between vertices (i.e., entries) i and j is assigned a weight equal to $|\delta_{k,i} - \delta_{k,j}|$. The above grouping procedure is then equivalent to generating a minimum spanning tree (MST) with Prim's algorithm on this graph [42], and then grouping the vertices into K clusters by cutting the most significant edges.

Before concluding this section, observe that Step 2. does not take relative differences into consideration for small $\overline{\delta}_{k,i}$. We suggest to determine the cutting positions by considering the $K-1$ largest values of the normalized sequence $\xi_{k,i}$, with $\xi_{k,i} = 0$ if $\overline{\delta}_{k,i} < \tau$ and $\xi_{k,i} = (\overline{\delta}_{k,i+1} - \overline{\delta}_{k,i})/\overline{\delta}_{k,i}$ if $\overline{\delta}_{k,i} \geq \tau$, where τ denotes a given threshold.

1.4.3 ADAPTIVE COMBINATION STRATEGY

We now derive an adaptive combination strategy for group diffusion LMS. Motivated by [23, 43], it consists of adjusting the combination weights $a_{\ell k,m}$ in an online manner via instantaneous MSD minimization. Let us denote by $\widetilde{\boldsymbol{w}}_{k,n}$ the weight error vector $\boldsymbol{w}_{k,n} - \boldsymbol{w}_k^{\star}$ after the combination step (1.18b). Considering groups $\mathcal{G}_m$, the instantaneous MSD at each agent k can be expressed as a function of $a_{\ell k,m}$ as follows:

$$\begin{aligned}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}_{k,n}\|^2\} &= \sum_{m=1}^{M} \mathbb{E}\Big\{\Big\|[\boldsymbol{w}_k^{\star}]_{\mathcal{G}_m} - \sum_{\ell\in\mathcal{N}_k} a_{\ell k,m}\,[\boldsymbol{\psi}_{\ell,n}]_{\mathcal{G}_m}\Big\|^2\Big\} \\ &= \sum_{m=1}^{M} \sum_{\ell\in\mathcal{N}_k} \sum_{p\in\mathcal{N}_k} a_{\ell k,m}\, a_{pk,m} \Big(\boldsymbol{\Psi}_{k,n}^{(m)}\Big)_{\ell p} \end{aligned} \tag{1.57}$$

where matrix $\boldsymbol{\Psi}_{k,n}^{(m)}$ is the covariance matrix of the weight error for group m at node k and time instant n, with (ℓ, p)-th entry given by:

$$\Big(\boldsymbol{\Psi}_{k,n}^{(m)}\Big)_{\ell p} = \begin{cases} \mathbb{E}\Big\{[\boldsymbol{w}_k^{\star} - \boldsymbol{\psi}_{\ell,n}]_{\mathcal{G}_m}^{\mathsf{T}}[\boldsymbol{w}_k^{\star} - \boldsymbol{\psi}_{p,n}]_{\mathcal{G}_m}\Big\}, & \ell, p \in \mathcal{N}_k \\ 0, & \text{otherwise.} \end{cases} \tag{1.58}$$

To make the problem tractable, we approximate $\boldsymbol{\Psi}_{k,n}^{(m)}$ by an instantaneous value and we drop its off-diagonal entries. In addition, since $\boldsymbol{w}_k^{\star}$ is unknown, we approximate it by $\widehat{\boldsymbol{w}}_k^{\star}$ as shown in (1.61). The instantaneous MSD minimization then leads to the

optimization problem:

$$\begin{aligned} \min_{\boldsymbol{a}_{k,m}} \quad & \sum_{\ell=1}^{N}\sum_{m=1}^{M} a_{\ell k,m}^2 \left\| [\widehat{\boldsymbol{w}}_k^{\star} - \boldsymbol{\psi}_{\ell,n}]_{\mathcal{G}_m} \right\|^2 \\ \text{subject to} \quad & \mathbf{1}_N^{\top}\boldsymbol{a}_{k,m} = 1, \quad a_{\ell k,m} \geq 0 \\ & a_{\ell k,m} = 0 \qquad \text{if } \ell \notin \mathcal{N}_k \end{aligned} \tag{1.59}$$

where $\boldsymbol{a}_{k,m} = [a_{1k,m}, \ldots, a_{Nk,m}]^{\top}$. The above objective function promotes weak information exchange via small $a_{\ell k,m}$ if the estimate of group $\mathcal{G}_m$ at node ℓ is far from its counterpart at node k. The solution of (1.59) is given by:

$$a_{\ell k,m} = \frac{\|[\widehat{\boldsymbol{w}}_k^{\star} - \boldsymbol{\psi}_{\ell,n}]_{\mathcal{G}_m}\|^{-2}}{\sum_{j\in\mathcal{N}_k} \|[\widehat{\boldsymbol{w}}_k^{\star} - \boldsymbol{\psi}_{j,n}]_{\mathcal{G}_m}\|^{-2}}, \text{ for } \ell \in \mathcal{N}_k. \tag{1.60}$$

We now introduce an instantaneous approximation $\widehat{\boldsymbol{w}}_{k,n}^{\star}$ for $\boldsymbol{w}_k^{\star}$ at each node k and time instant n. In order to reduce the MSD bias that may result from an inappropriate cooperation between nodes performing distinct estimation tasks, a possible strategy is to use the local one-step ahead approximation:

$$\widehat{\boldsymbol{w}}_{k,n}^{\star} = \boldsymbol{\psi}_{k,n} + \mu_k' \, \boldsymbol{q}_{k,n} \tag{1.61}$$

where $\boldsymbol{q}_{k,n} = [d_k(n) - \boldsymbol{x}_{k,n}^{\top}\boldsymbol{\psi}_{k,n}]\,\boldsymbol{x}_{k,n}$ is the instantaneous approximation of the negative gradient of $J_k(\boldsymbol{w})$ at $\boldsymbol{\psi}_{k,n}$. Substituting this expression into (1.60) leads to the combination rule:

$$a_{\ell k,m}(n) = \frac{\|[\boldsymbol{\psi}_{k,n} + \mu_k' \, \boldsymbol{q}_{k,n} - \boldsymbol{\psi}_{\ell,n}]_{\mathcal{G}_m}\|^{-2}}{\sum_{j\in\mathcal{N}_k} \|[\boldsymbol{\psi}_{k,n} + \mu_k' \, \boldsymbol{q}_{k,n} - \boldsymbol{\psi}_{j,n}]_{\mathcal{G}_m}\|^{-2}} \tag{1.62}$$

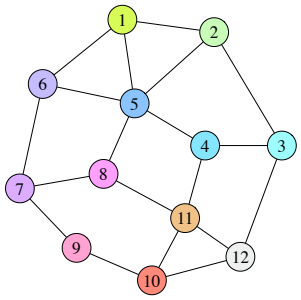
for $\ell \in \mathcal{N}_k$ and $m = 1, \ldots, M$. Furthermore, we observed in our experiments that the normalized gradient $\boldsymbol{q}_{k,n} \leftarrow \boldsymbol{q}_{k,n}/(\|\boldsymbol{q}_{k,n}\| + \epsilon)$ with ϵ a small positive constant can increase the robustness of the resulting strategy.

1.5 SIMULATIONS

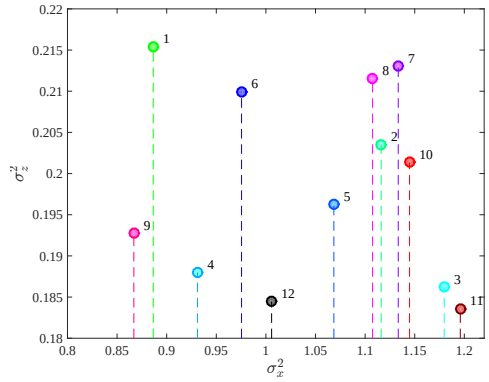
In this section, we shall first report simulation results that illustrate the theoretical findings, and then simulate the adaptive grouping and combination weight adjustment algorithms. All agents were initialized with zero parameter vector $\boldsymbol{w}_{k,0} = \mathbf{0}_L$ for all k. Simulated curves were obtained by averaging over 100 Monte-Carlo runs.

1.5.1 MODEL VALIDATION

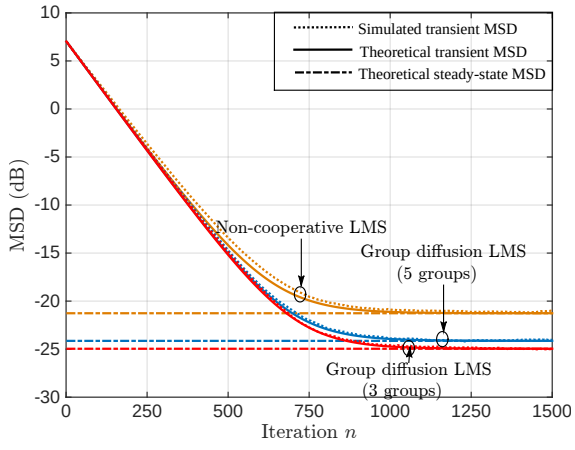
We considered the network with $N = 12$ nodes shown in Fig. 1.2(a). The optimum parameter vectors $\{\boldsymbol{w}_k^{\star}\}_{k=1}^N$ consisted of $L = 15$ entries. The first 6 entries were com-



(a) Network.



(b) Variance of regressors and noise at each node.



(c) MSD learning curves.

Figure 1.2 Validation of the mean-square error behavior analysis.

mon across all nodes, that is,

$$[\boldsymbol{w}_1^\star]_{\mathcal{G}_1} = \ldots = [\boldsymbol{w}_N^\star]_{\mathcal{G}_1} \quad \text{with} \quad \mathcal{G}_1 = \{1, \ldots, 6\}. \tag{1.63}$$

These entries were sampled from a uniform distribution $\mathcal{U}(-1, 1)$. The next 4 entries were uniformly sampled from $\mathcal{U}(-1, 1)$ for each node, so that there was no

relationship between the entries of this group, that is,

$$[\boldsymbol{w}_k^\star]_i = u_{ki} \quad \text{for} \quad i \in \mathcal{G}_2 = \{7, \ldots, 10\}, \tag{1.64}$$

with independent u_{ki} sampled from $\mathcal{U}(-1, 1)$. The last 5 entries were set to:

$$[\boldsymbol{w}_k^\star]_i = [\boldsymbol{w}^o]_i + u_{ki} \quad \text{for} \quad i \in \mathcal{G}_3 = \{11, \ldots, 15\}, \tag{1.65}$$

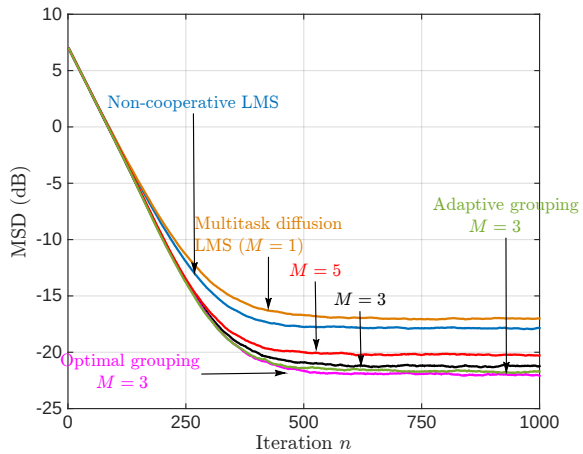
with $[\boldsymbol{w}^o]_i$ uniformly sampled from $\mathcal{U}(-1, 1)$, identical for all nodes, and i.i.d. perturbations u_{ki} sampled from $\mathcal{U}(-0.1, 0.1)$. In the sequel, we shall refer to the generative model (1.63)–(1.65) by $\mathcal{S}_1$ for short. Observe that nodes did not know this setup beforehand. Input vectors $\boldsymbol{x}_n$ were zero-mean $L \times 1$ random vectors governed by a Gaussian distribution with covariance matrix $\boldsymbol{R}_{x,k} = \sigma_{x,k}^2 \boldsymbol{I}_L$. The noises $z_k(n)$ were i.i.d. zero-mean Gaussian random variables, independent of any other signal with variances $\sigma_{z,k}^2$. Variances $\sigma_{x,k}^2$ and $\sigma_{z,k}^2$ used in this experiment were sampled from $\mathcal{U}(0.8, 1.2)$ and $\mathcal{U}(0.18, 0.22)$, respectively. Their values are depicted on the signal-noise variance plot shown in Fig. 1.2(b).

First, we illustrate the theoretical model with constant combination coefficients as characterized in Sec. 1.3.3. The following algorithm settings were considered:

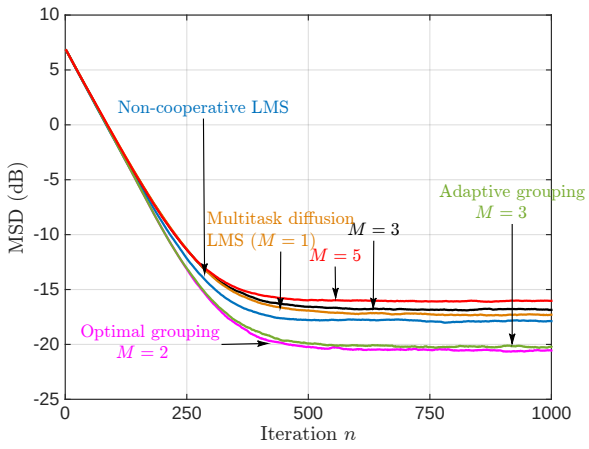
- Non-cooperative LMS. This is the limit case where the combination coefficient matrix $\boldsymbol{A}$ is set to $\boldsymbol{I}$.
- Diffusion LMS. This is the limit case where there exists only one group. A uniform combination matrix $\boldsymbol{A}$ with $a_{\ell k} = |\mathcal{N}_k|^{-1}$ was used for this experiment. As for the non-cooperative LMS, this setting was used as a baseline.
- Group diffusion LMS with 3 groups. In this setting, the groups were set according to the generative model. For the first group, $\boldsymbol{A}_1$ was set to a uniform combination matrix with $a_{\ell k,1} = |\mathcal{N}_k|^{-1}$. For the second group, the combination matrix was set to $\boldsymbol{A}_2 = \boldsymbol{I}$. For the third group, the combination matrix was set to an intermediate version $\boldsymbol{A}_3 = 0.5\boldsymbol{I} + 0.5\boldsymbol{A}_1$.
- Group diffusion LMS with 5 groups. We split the L entries into five groups of $L/5$ consecutive entries. A uniform combination matrix $\boldsymbol{A}_1$ with $a_{\ell k,1} = |\mathcal{N}_k|^{-1}$ was used for the first group. Matrix $\boldsymbol{A}_2$ was generated with the Metropolis rule, that is, $a_{\ell k,2} = \max\{|\mathcal{N}_k|, |\mathcal{N}_\ell|\}^{-1}$ for $k \in \mathcal{N}_k \backslash \{k\}$, $a_{kk,2} = 1 - \sum_{\ell \in \mathcal{N}_k \backslash \{k\}} a_{\ell k,2}$, otherwise $a_{\ell k,2} = 0$, for the second group. The identity matrix was used for the third and fourth groups, namely, $\boldsymbol{A}_3 = \boldsymbol{A}_4 = \boldsymbol{I}$. For the fifth group, the combination matrix was set to $\boldsymbol{A}_5 = 0.5\boldsymbol{I} + 0.5\boldsymbol{A}_1$.

This experimental setup was considered to test the theoretical models rather than reveal the performance gain using a grouping strategy. The step size was set to $\mu = 0.005$. The resulting MSD curves are illustrated in Fig. 1.2(c). The theoretical transient and steady-state MSD were evaluated using (1.52)–(1.54). The theoretical curves are generally consistent with the Monte Carlo simulated curves. It can be observed that single-task diffusion LMS algorithm had a large MSD due to the bias caused by the averaging over the entries of the group $\mathcal{G}_2$. Group diffusion LMS with 3 groups performed the best, since its 3 groups correspond to the generative

model and we associated reasonable combination coefficients to each group. Group diffusion LMS with 5 groups performed slightly worse than with the 3 group setting, because the fourth group overlaps $\mathcal{G}_2$ and $\mathcal{G}_3$ of the generative model. This simulation confirms that a grouping strategy should improve the performance, and also suggests that it should be adaptive.



(a) $\mathcal{S}_1$



(b) $\mathcal{S}_2$

Figure 1.3 Comparison of MSD learning curves.

1.5.2 PERFORMANCE OF THE ADAPTIVE GROUPING STRATEGY

The aim of this section is to compare the static and adaptive grouping strategies, along with the adaptive method for setting the combination coefficients. The following algorithms and settings were considered:

- Non-cooperative LMS. This non-cooperative algorithm was used as a reference for the performance comparison.
- Multi-task diffusion LMS in [23]. The number of groups considered with this algorithm is $M = 1$. Nodes are, however, endowed with the adaptive combination function that allows them to adapt the combination weights $a_{\ell k}$ in an online way. This algorithm was also used as a baseline for performance comparison to illustrate the need for an adaptive grouping strategy.
- Group diffusion LMS with preset groups. First, we considered the same groups as the generative model. Next we uniformly split the parameter vectors into M contiguous groups of the same size. With this algorithm, nodes are endowed with the adaptive combination function only.
- Group diffusion LMS with adaptive grouping strategy. With this algorithm, nodes are endowed with the adaptive variables grouping function and the adaptive combination function.

First, we considered the generative model $\mathcal{S}_1$ used for model validation in section 1.5.1, namely, (1.63)–(1.65). The second generative model we considered, denoted by $\mathcal{S}_2$, consisted of a partition into two groups of the parameter vectors entries. The first group involved all the odd entries as follows:

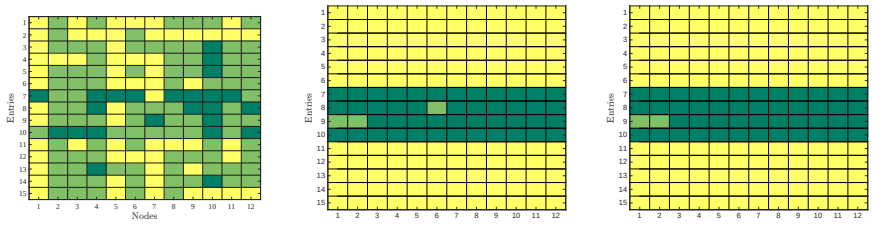
$$[\boldsymbol{w}_1^\star]_{\mathcal{G}_1} = \ldots = [\boldsymbol{w}_N^\star]_{\mathcal{G}_1} \quad \text{with} \quad \mathcal{G}_1 = \{1, 3, \ldots, 15\}, \text{ for } \forall k \tag{1.66}$$

and the second group involved all even entries as follows:

$$[\boldsymbol{w}_k^\star]_i = u_{ki} \quad \text{for} \quad i \in \mathcal{G}_2 = \{2, 4, \ldots, 14\}, \text{ for } \forall k \tag{1.67}$$

with u_{ki} randomly drawn from $\mathcal{U}(-1, 1)$.

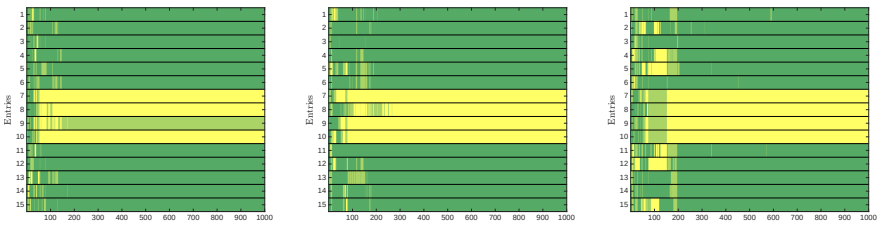
Figure 1.3 illustrates the MSD convergence behavior of the algorithms enumerated above. The non-cooperative LMS algorithm can be considered as a baseline for this comparative test since it does not rely on any cooperation. The multi-task diffusion LMS considered in [23] reached a slightly larger MSD than the non-cooperative LMS. This algorithm is able to adjust the combination weights $a_{\ell k}$ in an adaptive manner, but it cannot take possible group structures into account. It processed the parameter vectors as if they were significantly different and inhibited cooperation between nodes. This result reveals the need for a grouping strategy. By setting the group structure in accordance with the generative model, the group diffusion LMS with preset groups achieved the lowest MSD for both $\mathcal{S}_1$ and $\mathcal{S}_2$. With $M = 3$ preset uniform groups, the group diffusion LMS also led to a significant performance improvement over the non-cooperative LMS for $\mathcal{S}_1$, showing that preset groups can be beneficial. With $M = 5$ groups, this algorithm still outperformed



(a) $n = 50$ (b) $n = 250$ (c) $n = 1000$

FIGURE 1.4

Estimated group structures for the setup $\mathcal{S}_1$, at different time instants n.



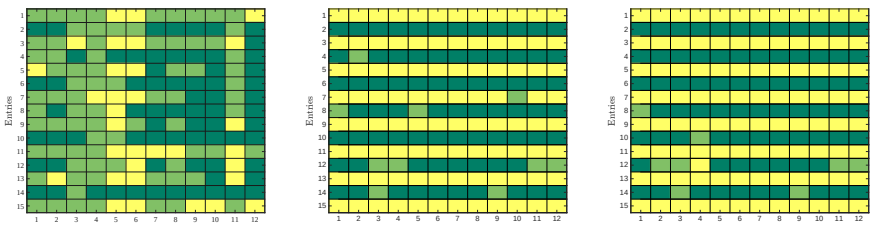
(a) $k = 1$ (b) $k = 6$ (c) $k = 9$

Figure 1.5 Estimated group structures for the setup $\mathcal{S}_1$, at different nodes k.

the non-cooperative algorithm. A larger MSD than in the case $M = 3$ was however observed, which shows that increasing the number of groups may not always be beneficial. It is worth noting that the group diffusion LMS with preset groups of sizes $M = 3$ and $M = 5$ led to unfavorable performance with $\mathcal{S}_2$. The limits of this strategy involving preset uniform groups of entries have already been discussed in Sec. 1.4.1. Finally, the proposed group diffusion LMS with adaptive grouping and adaptive combination coefficients was run with $M = 3$ groups. Note that M was thus voluntarily over estimated for $\mathcal{S}_2$. For experimental setups $\mathcal{S}_1$ and $\mathcal{S}_2$, it performed almost as well as when using the ground truth groups. Figures 1.4 and 1.6 show the group structures at time instants $n = 50$, 250 and 1000 for group settings $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. The entries encoded with the same color belong to the same group. Figure 1.5 and 1.7 show the evolution over time of the group structures at nodes $k = 1, 6$ and 9 for the group settings $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. All these results are consistent with the generative models.

1.6 CONCLUSION AND PERSPECTIVES

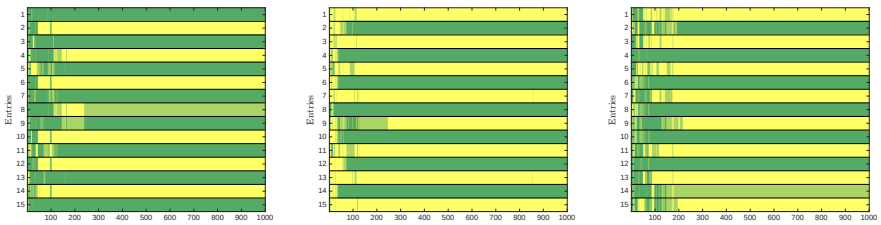
In this paper, we introduced an adaptive grouping procedure into diffusion adaptation to take advantage of structural similarities among parameter vectors to estimate.



(a) $n = 50$ (b) $n = 250$ (c) $n = 1000$

FIGURE 1.6

Estimated group structures for the setup $\mathcal{S}_2$, at different time instants n.



(a) $k = 1$ (b) $k = 6$ (c) $k = 9$

Figure 1.7 Estimated group structures for the setup $\mathcal{S}_2$, at different nodes k.

Simulation results illustrated the effectiveness of the grouping strategy and of the adaptive information fusion rule.

PROOF OF COROLLARY 1

The results below are based on the following properties of the Kronecker product:

$$\text{vec}(\boldsymbol{XYZ}) = (\boldsymbol{Z}^\top \otimes \boldsymbol{X})\text{vec}(\boldsymbol{Y}) \quad \text{and} \quad \text{trace}(\boldsymbol{XY}) = (\text{vec}(\boldsymbol{Y}^\top))^\top \text{vec}(\boldsymbol{X}),$$

Then, the first term in (1.49) can be rewritten as follows:

$$(\text{vec}\{\boldsymbol{G}^\top\})^\top \boldsymbol{K}^n \boldsymbol{\sigma}_I = \frac{1}{N}\text{trace}\Big(\boldsymbol{G}\,[\boldsymbol{B}^{n\top}\boldsymbol{B}^n]\Big)$$

The second term is given by:

$$\|\boldsymbol{r}\|^2_{\boldsymbol{K}^n\boldsymbol{\sigma}_I} = \frac{1}{N}\text{trace}\Big(\boldsymbol{r}\boldsymbol{r}^\top\,[\boldsymbol{B}^{n\top}\boldsymbol{B}^n]\Big)$$

The third term can be expressed as follows:

$$\|\widetilde{\boldsymbol{w}}_0\|^2_{(\boldsymbol{I}_{(NL)^2}-\boldsymbol{K})\boldsymbol{K}^n\boldsymbol{\sigma}_I} = \frac{1}{N}\text{trace}\Big(\widetilde{\boldsymbol{w}}_0\widetilde{\boldsymbol{w}}_0^\top\,[\boldsymbol{B}^{n\top}\boldsymbol{B}^n - \boldsymbol{B}^{n+1\top}\boldsymbol{B}^{n+1}]\Big)$$

The last term can be rewritten as:

$$\left(\boldsymbol{\gamma}_n^\top + [\boldsymbol{B}\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\}]^\top \otimes \boldsymbol{r}^\top\right)\boldsymbol{\sigma}_I = \text{trace}(\boldsymbol{\Gamma}_n^\top) + \boldsymbol{r}^\top \boldsymbol{B}\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\}$$

where $\boldsymbol{\Gamma}_n$ is defined hereafter. With these expressions, we obtain the following update equation that now depends on $\boldsymbol{B}$ rather than $\boldsymbol{K}$:

$$\begin{aligned}\zeta_{n+1} = \zeta_n + \frac{1}{N}\text{trace}\Big([\boldsymbol{G} + \boldsymbol{r}\boldsymbol{r}^\top]\boldsymbol{B}^{n\top}\boldsymbol{B}^n \\ - \widetilde{\boldsymbol{w}}_0\widetilde{\boldsymbol{w}}_0^\top\,[\boldsymbol{B}^{n\top}\boldsymbol{B}^n - \boldsymbol{B}^{n+1\top}\boldsymbol{B}^{n+1}] - 2\boldsymbol{\Gamma}_n - 2\boldsymbol{B}\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\}\boldsymbol{r}^\top\Big)\end{aligned}$$

The matrix form $\boldsymbol{\Gamma}_{n+1}$ of $\boldsymbol{\gamma}_{n+1}$ is updated as follows:

$$\boldsymbol{\Gamma}_{n+1} = \boldsymbol{B}\boldsymbol{\Gamma}_n\boldsymbol{B}^\top + \boldsymbol{B}\boldsymbol{r}\,(\boldsymbol{B}^2\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\})^\top - \boldsymbol{r}(\boldsymbol{B}\,\mathbb{E}\{\widetilde{\boldsymbol{w}}_n\})^\top$$

REFERENCE

1. Bertsekas DP, A new class of incremental gradient methods for least squares problems. SIAM J Optimiz 1997; 7(4):913–926.
2. Rabbat MG, Nowak RD, Quantized incremental algorithms for distributed optimization. IEEE J of Sel Topics Areas Commun 2005; 23(4):798–808.
3. Blatt D, Hero AO, Gauchman H, A convergent incremental gradient method with constant step size. SIAM J Optimiz 2007; 18(1):29–51.
4. Lopes CG, Sayed AH, Incremental adaptive strategies over distributed networks. IEEE Trans Signal Process 2007; 55(8):4064–4077.
5. Nedic A, Ozdaglar A, Distributed subgradient methods for multi-agent optimization. IEEE Trans Autom Control 2009; 54(1):48–61.
6. Kar S, Moura JMF, Distributed consensus algorithms in sensor networks: Link failures and channel noise. IEEE Trans Signal Process 2009; 57(1):355–369.
7. Srivastava K, Nedic A, Distributed asynchronous constrained stochastic optimization. IEEE J Sel Topics Signal Process 2011; 5(4):772–790.
8. Lopes CG, Sayed AH, Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. IEEE Trans Signal Process 2008; 56(7):3122–3136.
9. Cattivelli FS, Sayed AH, Diffusion LMS strategies for distributed estimation. IEEE Trans Signal Process 2010; 58(3):1035–1048.
10. Chen J, Sayed AH, Diffusion adaptation strategies for distributed optimization and learning over networks. IEEE Trans Signal Process 2012; 60(8):4289–4305.
11. Chen J, Sayed AH, Distributed Pareto optimization via diffusion strategies. IEEE J Sel Topics Signal Process 2013; 7(2):205–220.
12. Sayed AH, Tu SY, Chen J, Zhao X, Towfic ZJ, Diffusion strategies for adaptation and learning over networks. IEEE Sig Process Mag 2013; 30(3):155–171.
13. Sayed AH, Diffusion adaptation over networks. In: Chellapa R, Theodoridis S, editors, E-Reference Signal Processing, vol. 3, vol. 3, Elsevier, 2014; pp. 323–454.
14. Sayed AH, Adaptation, learning, and optimization over networks. In: Foundations and Trends in Machine Learning, vol. 7, vol. 7, Boston-Delft: NOW Publishers, Jul 2014; pp. 311–801.
15. Sayed AH, Adaptive networks. Proc IEEE 2014; 102(4):460–497.
16. Chen J, Sayed AH, On the learning behavior of adaptive networks — Part I: Transient analysis.

IEEE Trans Inf Theory 2015; 61(6):3487–3517.
17. Chen J, Sayed AH, On the learning behavior of adaptive networks — Part II: Performance analysis. IEEE Trans Inf Theory 2015; 61(6):3518–3548.
18. Tu SY, Sayed AH, Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. IEEE Trans Signal Process 2012; 60(12):6217–6234.
19. Bertrand A, Moonen M, Distributed adaptive node-specific signal estimation in fully connected sensor networks – Part I: sequential node updating. IEEE Trans Signal Process 2010; 58(10):5277–5291.
20. Bertrand A, Moonen M, Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology. IEEE Trans Signal Process 2011; 59(5):2196–2210.
21. Chen J, Richard C, Sayed AH, Multitask diffusion adaptation over networks. IEEE Trans Signal Process 2014; 62(16):4129–4144.
22. Nassif R, Richard C, Ferrari A, Sayed AH, Multitask diffusion adaptation over asynchronous networks. IEEE Trans Signal Process 2016; 64(11):2835–2850.
23. Chen J, Richard C, Sayed AH, Diffusion LMS over multitask networks. IEEE Trans Signal Process 2015; 63(11):2733–2748.
24. Zhao X, Sayed AH, Distributed clustering and learning over networks. IEEE Trans Signal Process 2015; 63(13):3285–3300.
25. Chen J, Richard C, Sayed AH, Adaptive clustering for multitask diffusion networks. In: Proc. European Signal Process. Conf. (EUSIPCO), Nice, France, 2015, pp. 200–204.
26. Monajemi S, Sanei S, Ong SH, Sayed AH, Adaptive regularized diffusion adaptation over multitask networks. In: Proc. IEEE Int. Workshop on Machine Learn. for Signal Process. (MLSP), Boston, USA, 2015, pp. 1–5.
27. Khawatmi S, Zoubir AM, Sayed AH, Decentralized clustering over adaptive networks. In: Proc. European Signal Process. Conf. (EUSIPCO), Nice, France, 2015, pp. 2696–2700.
28. Monajemi S, Eftaxias K, Sanei S, Ong SH, An informed multitask diffusion adaptation approach to study tremor in Parkinson's disease. IEEE J Sel Top Signal Process 2016; 10(7):1306–1314.
29. Wang Y, Tay WP, Hu W, Multitask diffusion LMS with optimized inter-cluster cooperation. In: Proc. IEEE Stat. Signal Process. Workshop (SSP), Palma de Mallorca, Spain, 2016, pp. 1–5.
30. Nassif R, Richard C, Ferrari A, Sayed AH, Proximal multitask learning over networks with sparsity-inducing coregularization. IEEE Trans Signal Process 2016; 64(23):6329–6344.
31. Chen J, Richard C, Hero AO, Sayed AH, Diffusion LMS for multitask problems with overlapping hypothesis subspaces. In: Proc. IEEE Int. Workshop on Machine Learn. for Signal Process. (MLSP), Reims, France, 2014, pp. 1–6.
32. Chen J, Richard C, Sayed AH, Multitask diffusion adaptation over networks with common latent representations. IEEE J Sel Top Signal Process 2017; 11(3):563–579.
33. Hua J, Li C, Shen H, Distributed learning of predictive structures from multiple tasks over networks. IEEE Trans Ind Elect 2017; 64(5):4246–4256.
34. Bogdanović N, Plata-Chaves J, Berberidis K, Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Florence, Italy, 2014, pp. 7223–7227.
35. Plata-Chaves J, Bogdanović N, Berberidis K, Distributed diffusion-based LMS for node-specific adaptive parameter estimation. IEEE Trans Signal Process 2015; 63(13):3448–3460.
36. Plata-Chaves J, Bahari HH, Moonen M, Bertrand A, Unsupervised diffusion-based LMS for node-specific parameter estimation over wireless sensor networks. In: Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Shanghai, China, 2016, pp. 4159–4163.

37. Nassif R, Richard C, Ferrari A, Sayed AH, Diffusion LMS for multitask problems with local linear equality constraints. IEEE Trans Signal Process 2017; 65(19):4979 – 4993.
38. Hua F, Nassif R, Richard C, Wang H, Penalty-based multitask estimation with non-local linear equality constraints. In: Proc. IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Curaçao, 2017, pp. 433–437.
39. Ting SK, Adaptive Techniques for Mitigating Circuit Imperfections in High Performance A/D Converters. Ph.D. thesis, Electrical Engineering Department, UCLA, 2014.
40. Sayed AH, Adaptive Filters. John Wiley & Sons, 2008.
41. Asano T, Bhattacharya B, Keil M, Yao F, Clustering algorithms based on minimum and maximum spanning trees. In: Proc. 4th Annual Symposium on Computational Geometry (SCG), Urbana-Champaign, USA, 1988, pp. 252–257.
42. Rosen K, Discrete Mathematics and Its Applications. 7th ed., McGraw-Hill Science, 2011.
43. Zhao X, Sayed AH, Clustering via diffusion adaptation over networks. In: Proc. International Workshop on Cognitive Information Processing (CIP), Baiona, Spain, 2012, pp. 1–6.