ADAPTIVE CLUSTERING FOR MULTITASK DIFFUSION NETWORKS

Jie Chen ⋆ *Cédric Richard* † *Ali H. Sayed* ‡

⋆ Center of Intelligent Acoustics and Immersive Communications
School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China
† Laboratoire Lagrange, Université de Nice Sophia-Antipolis, CNRS, France
‡ Electrical Engineering Department, University of California, Los Angeles, USA
dr.jie.chen@ieee.org cedric.richard@unice.fr sayed@ee.ucla.edu

ABSTRACT

Diffusion LMS was originally conceived for online distributed parameter estimation in single-task environments where agents pursue a common objective. However, estimating distinct but correlated objects (multitask problems) is useful in many applications. To address multitask problems with combine-then-adapt diffusion LMS strategies, we derive an unsupervised strategy that allows each node to continuously select the neighboring nodes with which it should exchange information to improve its estimation accuracy. Simulation experiments illustrate the efficiency of this clustering strategy. In particular, nodes do not know which other nodes share similar objectives.

***Index Terms*—** Diffusion LMS, combine-then-adapt, multitask problems, adaptive network, online learning, distributed learning.

1. INTRODUCTION

Distributed adaptive estimation allows a collection of interconnected nodes to perform preassigned tasks from streaming measurements. For online parameter estimation, among various strategies [1–7], diffusion LMS [8, 9] is an efficient algorithm that is particularly attractive due to its enhanced adaptation performance and wider stability ranges [10]. Its variants and performance have been extensively studied in the literature, under various scenarios [11–19].

The working hypothesis for several earlier studies on diffusion LMS strategies is that the nodes cooperate with each other to estimate a single parameter vector. We shall refer to problems of this type as *single-task* problems. However, many problems of interest happen to be *multitask*-oriented in the sense that there are multiple optimum parameter vectors to be inferred simultaneously and in a collaborative manner. The multitask learning problem is relevant in several machine learning formulations [20–22]. In the distributed estimation context, which is the focus of this work, there exist many applications where either agents are subject to data measurements arising from different models or they are sensing data that varies over the spatial domain. When this is the case, it is important for cooperation to occur only among agents with similar or related objectives to avoid biased solutions.

A handful of works have considered before problem formulations that deal with multitask scenarios [23–28]. It is generally assumed in these studies that the nodes have some prior knowledge about clustering or about the parameter space. In this work, we propose an unsupervised clustering strategy that allows each node to select, via adaptive adjustments of combination weights, the neighboring nodes with which it should collaborate to improve its estimation accuracy. In the related work [23], we formulated the multitask problem directly over networks with connected clusters of nodes. In that work, the clusters were assumed to be known beforehand and no clustering strategy was proposed. In [28], the parameter vectors were assumed to lie in a common subspace. In the current work, building up on our recent results from [29], neither clusters nor other latent structures are assumed to be known. It then becomes necessary to examine how this lack of information influences performance. It also becomes necessary to endow the nodes with the ability to identify and form appropriate clusters to enhance performance. We do so by considering the combine-then-adapt diffusion strategy, extending and enhancing the earlier procedure proposed in [26], and also complementing the result in [29] which focused on studying the alternative adapt-then-combine formulation.

Notation. Boldface small letters $\boldsymbol{x}$ denote column vectors. The superscript $(\cdot)^\top$ represents the transpose of a matrix or a vector. Identity matrix of size $N \times N$ is denoted by $\boldsymbol{I}_N$, and the all-one vector of length N is denoted by $\mathbf{1}_N$. We denote by $\mathcal{N}_k$ the set of node indices in the neighborhood of node k, including k itself.

2. MULTITASK PROBLEMS AND DIFFUSION LMS

2.1. Modeling assumptions

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

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

where $z_k(n)$ is a zero-mean i.i.d. additive noise at node k and time instant n. Noise $z_k(n)$ is assumed to be independent of other signals and has variance $\sigma_{z,k}^2$. Let $J_k(\boldsymbol{w})$ be the mean-square-error criterion at node k, namely,

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

It is clear that each $J_k(\boldsymbol{w})$ is minimized at $\boldsymbol{w}_k^\star$. Depending on whether the minima of all the $J_k(\boldsymbol{w})$ are achieved at the same location or not, referred to as tasks, the distributed learning problem can be single-task or multitask oriented [23]. In a single-task network, all nodes have to estimate the same parameter vector $\boldsymbol{w}^\star$, namely:

$$\boldsymbol{w}_k^\star = \boldsymbol{w}^\star, \quad \forall k \in \{1, ..., N\}. \tag{3}$$

Diffusion LMS strategies for the distributed estimation of $\boldsymbol{w}^\star$ under this scenario were derived in [6–9] by seeking the minimizer of the following aggregate cost function:

The work of C. Richard was partly supported by the Agence Nationale pour la Recherche, France, (ODISSEE project, ANR-13-ASTR-0030). The work of A. H. Sayed was supported in part by NSF grant CCF-1011918 and ECCS-1407712.

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

$$J^{\text{glob}}(\boldsymbol{w}) = \sum_{k=1}^{N} J_k(\boldsymbol{w}) \tag{4}$$

in a cooperative manner in order to improve estimation accuracy.

In a multitask network, on the other hand, each node needs to determine its own parameter vector $\boldsymbol{w}_k^\star$. The parameter vectors at two connected nodes can be related in various ways depending on the application [23,28,30]. In this work, we do not assume the availability of any prior information, and nodes do not know which other nodes share similar objectives.

2.2. Diffusion LMS in multitask environments

The diffusion LMS strategies can be subdivided into two forms: the adapt-then-combine (ATC) and the combine-then-adapt (CTA) strategies, depending in which order the adaptation and the consultation steps are performed. The related work [29] studied the case of ATC implementations. Here we extend diffusion LMS for multitask environment based on CTA variants. The CTA diffusion LMS algorithm was designed for minimizing the cost function (4) by employing the following recursive construction [6, 8, 9, 31]:

$$\begin{cases} \boldsymbol{\psi}_k(n) = \sum\limits_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_\ell(n) \\ \boldsymbol{w}_k(n+1) = \boldsymbol{\psi}_k(n) + \mu_k \boldsymbol{x}_k(n)\big[d_k(n) - \boldsymbol{x}_k^\top(n)\boldsymbol{\psi}_k(n)\big] \end{cases} \tag{5}$$

where the non-negative coefficients $a_{\ell k}$ are the (ℓ, k)-th entries of a left-stochastic matrices $\boldsymbol{A}$ such that:

$$\boldsymbol{A}^\top \mathbf{1}_N = \mathbf{1}_N \quad \text{and} \quad a_{\ell k} = 0 \quad \text{if} \quad \ell \notin \mathcal{N}_k. \tag{6}$$

There are several ways to select these coefficients such as using the averaging rule or the Metropolis rule for single-task environments. When operating in multitask environments, the bias and mean-square deviation (MSD) performance of diffusion LMS using these rules with fixed combination coefficients have been analyzed in [29, 32]. The main conclusion is that cooperation by means of diffusion LMS is still beneficial when the contrasts among the tasks are small enough, otherwise, as expected, degradation in performance occurs by leading to an estimation bias. In this work, we continue using the updating structure (5), but derive a time variant combination matrix to endow CTA diffusion LMS with the ability to work in multitask environments.

3. NODE CLUSTERING VIA ADAPTIVE COMBINATION

In order to reduce the bias that may arise in multitask environments, we now derive a clustering strategy where each node k can adjust the combination weights $a_{\ell k}$ in an online manner, for $\ell \in \mathcal{N}_k$.

3.1. Clustering via matrix $\boldsymbol{A}$ adjustments

Motivated by the construction in [26], we suggest to adjust $\boldsymbol{A}$ in an online manner via MSD optimization. Let the weight error vector after the combination step be denoted by $\boldsymbol{v}_k(n) = \boldsymbol{w}_k^\star - \boldsymbol{\psi}_k(n)$. Considering matrix $\boldsymbol{A}$ is left-stochastic, at each instant n the instantaneous MSD at node k is given by

$$\begin{aligned} \mathbb{E}\{\|\boldsymbol{v}_k(n)\|^2\} &= \mathbb{E}\big\{\big\|\boldsymbol{w}_k^\star - \sum_{\ell\in\mathcal{N}_k} a_{\ell k}\, \boldsymbol{w}_\ell(n)\big\|^2\big\} \\ &= \sum_{\ell\in\mathcal{N}_k}\sum_{p\in\mathcal{N}_k} a_{\ell k}\, a_{pk}\, [\boldsymbol{\Psi}_k]_{\ell p} \end{aligned} \tag{7}$$

where $\boldsymbol{\Psi}_k$ is the matrix at node k with (ℓ, p)-th entry defined as

$$[\boldsymbol{\Psi}_k]_{\ell p} = \begin{cases} \mathbb{E}\big\{[\boldsymbol{w}_k^\star - \boldsymbol{w}_\ell(n)]^\top [\boldsymbol{w}_k^\star - \boldsymbol{w}_p(n)]\big\}, & \ell, p \in \mathcal{N}_k \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

However, we cannot determine $a_{\ell k}$ by minimizing objective (7) since, at each node k, the optimum $\boldsymbol{w}_k^\star$ and the covariance $\boldsymbol{\Psi}_k$ are unknown. We suggest to use an approximation $\widehat{\boldsymbol{w}}_k^\star$ instead of $\boldsymbol{w}_k^\star$, to approximate matrix $\boldsymbol{\Psi}_k$ by an instantaneous value, and to drop its off-diagonal entries in order to make the problem tractable. The resulting problem is then as follows:

$$\begin{aligned} \min_{\boldsymbol{a}_k} \quad & \sum_{\ell=1}^{N} a_{\ell k}^2 \, \|\widehat{\boldsymbol{w}}_k^\star - \boldsymbol{w}_\ell(n))\|^2 \\ \text{subject to} \quad & \mathbf{1}_N^\top \boldsymbol{a}_k = 1, \quad a_{\ell k} \geq 0, \\ & a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \end{aligned} \tag{9}$$

where the notation $\boldsymbol{a}_k$ refers to a column vector containing the entries $\{a_{\ell k}\}$ for $\ell = 1, \ldots, N$. A direct consequence of this objective is that large combination weight $a_{\ell k}$ will be penalized if the local estimate at node ℓ is far from the objective at node k. We now discuss the solution of (9) before dwelling on the selection of $\widehat{\boldsymbol{w}}_k^\star$. We omit the non-negativity constraints $a_{\ell k} \geq 0$ for the moment, and consider the simplified optimization problem only with the sum-to-one equality constraint. The Lagrangian associated with this simplified problem is given by

$$\mathcal{L}(\boldsymbol{a}_k, \lambda) = \sum_{\ell=1}^{N} a_{\ell k}^2 \, \|\widehat{\boldsymbol{w}}_k^\star - \boldsymbol{w}_\ell(n))\|^2 + \lambda\,(\mathbf{1}_N^\top \boldsymbol{a}_k - 1),$$

with λ the Lagrange multiplier for the equality constraint. Equating the gradient of $\mathcal{L}(\boldsymbol{a}_k, \lambda)$ with respect to $\boldsymbol{a}_k$ and λ to 0, we get the solution at time $n+1$

$$a_{\ell k}(n+1) = \frac{\|\widehat{\boldsymbol{w}}_k^\star - \boldsymbol{w}_\ell(n))\|^{-2}}{\sum_{j\in\mathcal{N}_k} \|\widehat{\boldsymbol{w}}_k^\star - \boldsymbol{w}_j(n))\|^{-2}}, \text{ for } \ell \in \mathcal{N}_k. \tag{10}$$

Observe that this solution is always non-negative and, consequently, (10) is also the solution to problem (9). Let us now discuss the approximation for $\boldsymbol{w}_k^\star$ to be used in (10). Since $\boldsymbol{w}_k^\star$ is unknown and needs to be estimated iteratively, we assign node k with a time variant approximation $\widehat{\boldsymbol{w}}_k^\star(n)$ at each instant n. In order to reduce the MSD bias that results from the inappropriate cooperation of nodes performing distinct estimation tasks, one strategy is to use the local one-step unbiased approximation:

$$\widehat{\boldsymbol{w}}_k^\star(n) = \boldsymbol{w}_k(n) - \mu_k \, \nabla J_k(\boldsymbol{w})\big|_{\boldsymbol{w}=\boldsymbol{w}_k(n)}. \tag{11}$$

Since the true negative gradient $-\nabla J_k(\boldsymbol{w}) = \mathbb{E}\{e_k(n)\boldsymbol{x}_k(n)\}$ with $e_k(n) = [d_k(n) - \boldsymbol{x}_k^\top(n)\boldsymbol{w}_k(n)]$ at $\boldsymbol{w}_k(n)$ is not available in an adaptive implementation, we can approximate it by using its instantaneous value $\boldsymbol{q}_k = e_k(n)\boldsymbol{x}_k(n)$. This yields the following approximation:

$$\widehat{\boldsymbol{w}}_k^\star(n) = \boldsymbol{w}_k(n) + \mu_k \, \boldsymbol{q}_k(n). \tag{12}$$

Substituting this expression into (10), we get the combination rule

$$\begin{aligned} a_{\ell k}(n+1) = & \frac{\|\boldsymbol{w}_k(n) + \mu_k \, \boldsymbol{q}_k(n) - \boldsymbol{w}_\ell(n)\|^{-2}}{\sum_{j\in\mathcal{N}_k} \|\boldsymbol{w}_k(n) + \mu_k \, \boldsymbol{q}_k(n) - \boldsymbol{w}_j(n)\|^{-2}} \\ & \text{for } \ell \in \mathcal{N}_k. \end{aligned} \tag{13}$$

This rule uses the local unbiased estimate (11) as a reference in order to reduce the MSD bias caused by cooperation among neighboring nodes that estimate distinct parameter vectors. Besides, consider the inverse of the numerator of rule (13):

$$\begin{aligned} \|\boldsymbol{w}_k(n) + \mu_k \, \boldsymbol{q}_k(n) - \boldsymbol{w}_\ell(n)\|^2 = & \\ \mu_k^2 \, \|\boldsymbol{q}_k(n)\|^2 + \|\boldsymbol{w}_\ell(n) - \boldsymbol{w}_k(n)\|^2 + & \\ 2[\boldsymbol{w}_\ell(n) - \boldsymbol{w}_k(n)]^\top [-\mu_k \, \boldsymbol{q}_k(n)] & \end{aligned} \tag{14}$$

The second term $\|\boldsymbol{w}_\ell(n) - \boldsymbol{w}_k(n)\|^2$ on the RHS accounts for the distance between the current estimates at nodes k and ℓ; this term

tends to decrease the combination weight $a_{\ell k}(n+1)$ if the distance is large, and to limit information exchange. Now, consider the approximated first-order Taylor series expansion of $J_k(\boldsymbol{w})$ at $\boldsymbol{w}_k(n)$:

$$J_k(\boldsymbol{w}) \approx J_k(\boldsymbol{w}_k(n)) - [\boldsymbol{w} - \boldsymbol{w}_k(n)]^\top \boldsymbol{q}_k(n). \quad (15)$$

The third term $[\boldsymbol{w}_\ell(n) - \boldsymbol{w}_k(n)]^\top [-\mu_k \, \boldsymbol{q}_k(n)]$ on the RHS of (14) is proportional to $J_k(\boldsymbol{w}_\ell(n)) - J_k(\boldsymbol{w}_k(n))$. This term also tends to decrease $a_{\ell k}(n)$ if $J_k(\boldsymbol{w}_\ell(n)) > J_k(\boldsymbol{w}_k(n))$. Indeed, in this case, it is not recommended to promote the combination of $\boldsymbol{w}_k(n)$ and $\boldsymbol{w}_\ell(n)$ because the latter induces an increase of the cost function.

3.2. Algorithm

The CTA diffusion algorithm with adaptive clustering defined by time-variant combination matrices $\boldsymbol{A}(n)$ is summarized below. Note that we use the normalized gradient $\boldsymbol{q}_k(n)/(\|\boldsymbol{q}_k(n)\| + \xi)$ instead of $\boldsymbol{q}_k(n)$, with ξ a small positive number, since it improves the robustness of the algorithm. The matrix $\boldsymbol{A}(0)$ is initialized with $\boldsymbol{I}_N$, considering that no prior information on clusters is available.

Algorithm 1: CTA Diffusion LMS with adaptive clustering for multitask problems

Initialization: Set $\boldsymbol{A}(0) = \boldsymbol{I}_N$.
Set $\boldsymbol{w}_k(0) = 0$ for all $k = 1, ..., N$.

Algorithm:
At each time $n \geq 1$, and for each node k, update by:

Update the combination coefficients:

$$\boldsymbol{q}_k(n) = [d_k(n) - \boldsymbol{x}_k^\top(n)\boldsymbol{w}_k(n)]\, \boldsymbol{x}_k(n) \quad (16)$$

$$\text{Normalize } \boldsymbol{q}_k(n) : \boldsymbol{q}_k(n)/(\|\boldsymbol{q}_k(n)\| + \xi) \quad (17)$$

$$a_{\ell k}(n) = \frac{\|\boldsymbol{w}_k(n) + \mu_k \boldsymbol{q}_k(n) - \boldsymbol{w}_\ell(n))\|^{-2}}{\sum_{j \in \mathcal{N}_k} \|\boldsymbol{w}_k(n) + \mu_k \boldsymbol{q}_k(n) - \boldsymbol{w}_j(n))\|^{-2}} \quad (18)$$

Combine:

$$\boldsymbol{\psi}_k(n) = \sum_{\ell \in \mathcal{N}_k} a_{\ell k}(n)\, \boldsymbol{w}_k(n) \quad (19)$$

Adapt:

$$\boldsymbol{w}_k(n+1) = \boldsymbol{\psi}_k(n) + \mu_k \left[d_\ell(n) - \boldsymbol{x}_\ell^\top(n)\boldsymbol{\psi}_k(n)\right] \boldsymbol{x}_\ell(n) \quad (20)$$

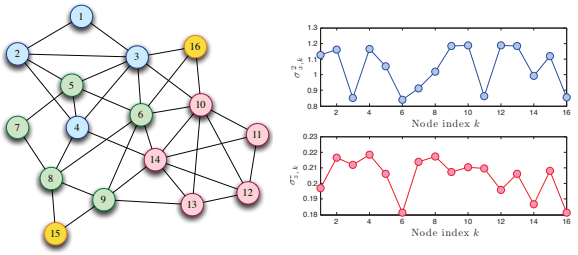
4. SIMULATIONS

We now report simulation results to illustrate the operation of the proposed algorithm in a manner similar to what was done in [29] for the ATC implementation. All nodes were initialized with zero parameter vectors $\boldsymbol{w}_k(0)$. Simulation curves were obtained by averaging over 100 runs.

4.1. Stationary environment

Consider the network of 16 agents depicted in Fig. 1(a). The regression inputs $\boldsymbol{x}_k(n)$ were zero-mean 2×1 random vectors governed by a Gaussian distribution with covariance matrices $\boldsymbol{R}_{x,k} = \sigma_{x,k}^2 \, \boldsymbol{I}_L$. The background noises $z_k(n)$ were i.i.d. zero-mean Gaussian random variables, independent of any other signals. The variances $\sigma_{x,k}^2$ and $\sigma_{z,k}^2$ are depicted in Fig. 1(b). The parameter vectors to be estimated are as follows:

$$\boldsymbol{w}_k^\star = \begin{cases} [0.5 \;\; -0.4]^\top & k = 1, \ldots, 4 \quad \text{Cluster 1} \\ [0.6 \;\; -0.2]^\top & k = 5, \ldots, 9 \quad \text{Cluster 2} \\ [0.3 \;\; -0.3]^\top & k = 10, \ldots, 14 \quad \text{Cluster 3} \\ [-0.8 \;\; 0.5]^\top & k = 15, 16 \quad \text{Cluster 4} \end{cases} \quad (21)$$



(a) Network topology (b) Input variances (top) and noise variances (bottom)

Fig. 1. Network topology in Section 4.1 and associated input variances and noise variances.

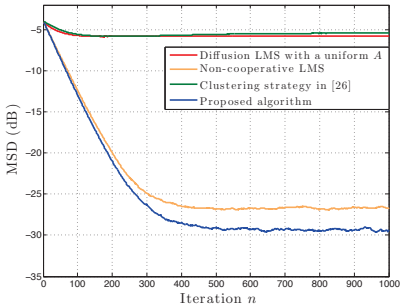


Fig. 2. Network MSD comparison in a stationary multitask environment

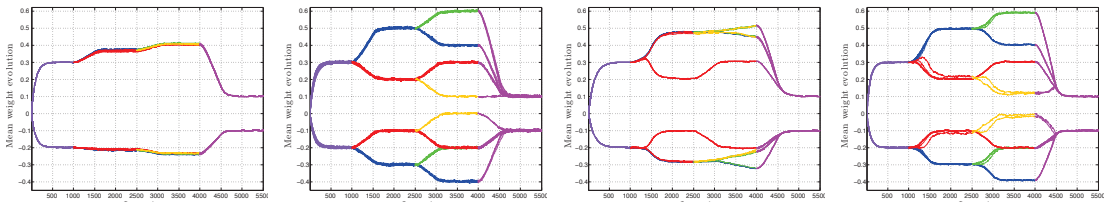
The following algorithms were considered for estimating the four optimum parameter vectors: 1) diffusion LMS with a uniform combination matrix $\boldsymbol{A}$, 2) non-cooperative LMS, 3) diffusion LMS with the clustering strategy introduced in [26], 4) diffusion LMS with our clustering strategy. The step size was set to $\mu = 0.01$ for all nodes.

Fig. 2 illustrates the MSD convergence behavior for these algorithms. Due to large bias of the estimated weights, diffusion LMS with a uniform combination matrix had large MSD. Non-cooperative LMS performs better since its leads to unbiased estimates due to the lack of cooperation among nodes with different tasks. The proposed algorithm achieved the best performance in this context.

4.2. Non-stationary environment

Consider now a dynamic environment. Properties of input signals and noise were the same as those in the above case. From instant $n = 1$ to 1000, the network consisted of one cluster with a unique optimum parameter vector. From $n = 1501$ to 2500, nodes were split into two clusters with two different optimums. From $n = 3001$ to 4000, nodes were split again to give four clusters. Finally, from instant $n = 4501$, nodes were aggregated into one cluster with another unique parameter vector. Cluster structures and optimum parameter vectors are illustrated in Fig. 4 and 5, respectively.

The same four algorithms as before were considered for comparison. Transient stages can be clearly observed on both weight behavior curves Fig. 3 and MSD behavior curves Fig. 6. Diffusion LMS enforced the weight vectors estimated by each agent to converge to the same solution at each stage. As a consequence, the MSD learning curve shows poor performance due to large bias. Non-cooperative LMS converged without bias towards the optimum parameter vectors. The algorithm introduced by [26] showed some ability to conduct clustering but did not provide satisfactory results during transient episodes. During stages 1 and 4, it worked as well as diffusion LMS. However, during stages 2 and 3, it only performed slightly better than diffusion LMS. The proposed algorithm was able



(a) Diffusion LMS. (b) Non-cooperative LMS. (c) Algorithm in [26]. (d) Proposed.

Fig. 3. Mean weight behavior of various algorithms in the non-stationary environment. Colors are consistent with cluster colors in Fig. 1(a).

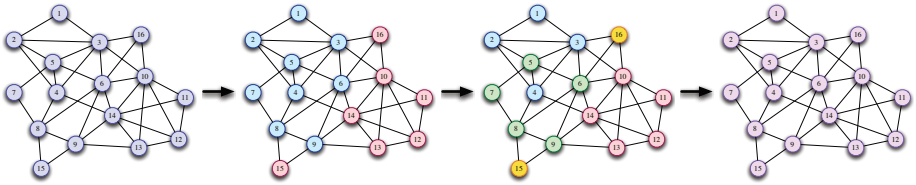
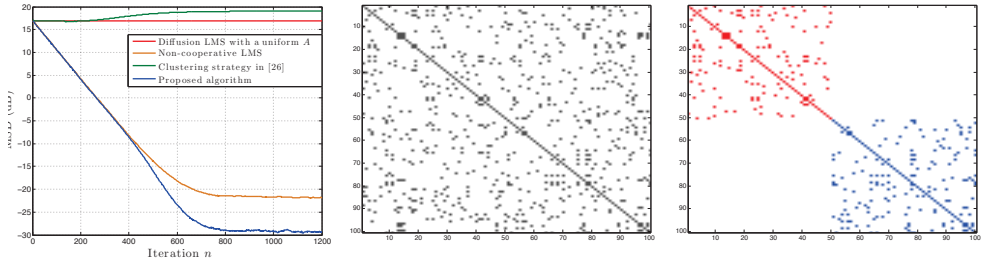


Fig. 4. Evolution of cluster structures of the network (1 cluster → 2 clusters → 4 clusters → 1 cluster).



(a) MSD learning curves. (b) Network physical connection. (c) Selected connection.

Fig. 7. Simulation results of Sec. 4.3.

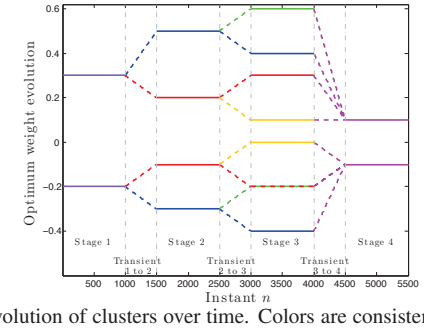


Fig. 5. Evolution of clusters over time. Colors are consistent with those of clusters in Fig. 4. Dashed lines represent optimums during transient episodes.

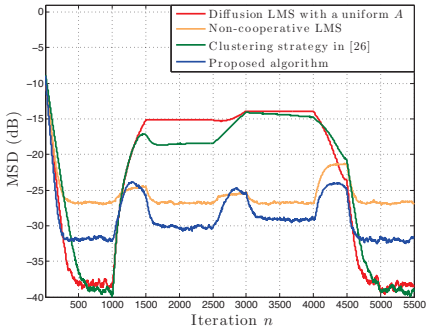


Fig. 6. Network MSD behavior comparison in the time variant multitask environment.

to track the system dynamic with correct clustering.

4.3. Large network and high-dimensional regressors

For the sake of simplicity, previous experiments were conducted with relatively small networks and low-dimensional optimum parameter vectors. A network consisting of two clusters with 50 nodes in each cluster was randomly deployed in a given area, with physical connections defined by the connectivity matrix in Fig. 7(b). The optimum parameter vectors were set as follows: $\boldsymbol{w}_k^\star = \mathbf{1}_{50}$ for $k = 1, \ldots, 50$, and $\boldsymbol{w}_k^\star = -\mathbf{1}_{50}$ for $k = 51, \ldots, 100$. The regression inputs $\boldsymbol{x}_k(n)$ were zero-mean 50×1 random vectors governed by a Gaussian distribution with covariance matrices $\boldsymbol{R}_{x,k} = \sigma_{x,k}^2 \boldsymbol{I}_L$. The background noises $z_k(n)$ were i.i.d. zero-mean Gaussian random variables, and independent of any other signal. The variances $\sigma_{x,k}^2$ and $\sigma_{z,k}^2$ were uniformly sampled in $[0.8, 1.2]$ and $[0.018, 0.022]$, respectively. For all nodes, the step-sizes were set to $\mu_k = 0.01$. The same four algorithms as before were considered. Our algorithm was used with the normalized gradient $\boldsymbol{q}_k(n)/(\|\boldsymbol{q}_k(n)\| + \xi)$ and $\xi = 0.01$. MSD learning curves are shown in Fig. 7(a), and the connectivity matrix determined by our algorithm is represented in Fig. 7(c). From these experiments the advantage of the proposed algorithm can clearly be observed.

5. CONCLUSION AND PERSPECTIVE

Many practical problems of interest happen to be multitask-oriented in the sense that there are multiple optimum parameter vectors to be inferred simultaneously. In this paper, we proposed an unsupervised clustering strategy of the combine-then-adapt (CTA) type that allows each node to select the neighboring nodes with which it can collaborate to address a given task. Reference [29] studies the alternative adapt-then-combine (ATC) strategy and carries out the necessary analysis and derivations. Simulations were presented to illustrate the efficiency of the proposed clustering strategy.

6. REFERENCES

[1] P. Braca, S. Marano, and V. Matta, "Enforcing consensus while monitoring the environment in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3375–3380, Jul. 2008.

[2] 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.

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

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

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

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

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

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

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

[11] 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.

[12] 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.

[13] 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. 1–16, Oct. 2013.

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

[15] 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.

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

[17] 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.

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

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

[20] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for leaning shared structures from muliple tasks," in *Proc. Ann. Int. Conf. Machine Learning (ICML)*, Montreal, Canada, Jun. 2009, pp. 137–144.

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

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

[23] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion LMS over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129–4144, Aug. 2014. Extended report available as arXiv: 1311.4894 [cs.MA], Nov. 2013.

[24] 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.

[25] 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.

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

[27] 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.

[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 on Machine Learn. for Signal Process. (MLSP)*, Reims, France, Sept. 2014, pp. 1–6.

[29] 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. Extended report available as arXiv:1404.6813 [cs.MA], Apr. 2014.

[30] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 2014, pp. 7223–7227.

[31] 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.

[32] J. Chen and C. Richard, "Performance analysis of diffusion LMS in multitask networks," in *Proc. IEEE Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Saint Martin, France, Dec. 2013, pp. 137–140.