MULTITASK DIFFUSION LMS WITH SPARSITY-BASED REGULARIZATION

Roula Nassif[(1)], *Cédric Richard*[(1)], *André Ferrari*[(1)], *Ali H. Sayed*[(2)]

[(1)] Université de Nice Sophia-Antipolis, France
{roula.nassif, cedric.richard, andre.ferrari}@unice.fr

[(2)]University of California, Los Angeles, USA
sayed@ee.ucla.edu

ABSTRACT

In this work, a diffusion-type algorithm is proposed to solve multitask estimation problems where each cluster of nodes is interested in estimating its own optimum parameter vector in a distributed manner. The approach relies on minimizing a global mean-square error criterion regularized by a term that promotes piecewise constant transitions in the parameter vector entries estimated by neighboring clusters. We provide some results on the mean and mean-square-error convergence. Simulations are conducted to illustrate the effectiveness of the strategy.

Index Terms— Distributed optimization, diffusion adaptation, multitask learning, cooperation, sparse regularization.

1. INTRODUCTION

Consider a distributed adaptive estimation problem where a connected network of N nodes is employed to simultaneously estimate a number of parameter vectors from noisy measurements using in-network processing. Depending on the number of parameter vectors, we distinguish between two types of networks. In a single-task network, all agents are interested in estimating the same parameter vector. In a multitask network, nodes are organized into clusters and agents within the same cluster are interested in estimating a common parameter vector (also called task). Different clusters will generally have different (though related) tasks. Diffusion strategies for single-task networks have been proposed and analyzed in the literature rather extensively (see, e.g., [1–4] and the references therein). These strategies are attractive since they are scalable, robust, and enable continuous learning and adaptation in response to concept drifts.

In comparison, diffusion multi-task strategies have been approached in two main ways. In a first scenario, no prior information on possible relationships between the tasks is assumed. In this case, it was argued in [5] that the diffusion iterates will converge to a Pareto optimal solution when confronted with multi-objective optimization problems consisting of a sum of individual costs with possibly different minimizers. In [6, 7], strategies are developed for selecting the combination weights adaptively in order to enable automatic network clustering and subsequent cooperation over the clustered agents. In a second scenario, diffusion strategies are derived by exploiting prior information about relationships among the tasks. A couple of works have addressed variations of this scenario. For example, in [8, 9], it is assumed that nodes are interested in estimating some parameters of global and others of local interest. In [10], a global regularized optimization problem is formulated where ℓ_2-norm co-regularizers are added to the mean-square error criterion in order to promote smoothness of the graph signal (which refers to an $N \times 1$ block vector whose k-th block is the optimum parameter vector at node k).

In some applications, such as in cognitive radio [8, 9] and remote sensing [10], it may happen that the optimum parameter vectors of neighboring clusters may have a large number of similar or identical entries, and a small number of different entries. It is therefore advantageous to develop a distributed strategy that involves cooperation among adjacent clusters in order to promote similarity between their tasks. This objective is the theme of this work.

Notation. We use normal font letters to denote scalars, boldface lowercase letters to denote column vectors, and boldface uppercase letters for matrices. The operator $(\cdot)^\top$ denotes matrix transposition, the operator $\otimes$ refers to the Kronecker product and col$\{\cdot\}$ stacks the column vectors entries on top of each other. The set $\mathcal{N}_k$ denotes the neighbors of node k, $\mathcal{C}(k)$ denotes the cluster to which node k belongs, and $\mathcal{C}_i$ is the i-th cluster.

2. MULTITASK DIFFUSION ADAPTATION

2.1. Network model and problem formulation

We consider a connected network consisting of N nodes grouped into Q clusters. At each time instant i, node k observes a zero-mean scalar measurement $d_k(i)$ and a zero-mean $L \times 1$ regression vector $\boldsymbol{x}_k(i)$ with positive covariance matrix $\boldsymbol{R}_{\boldsymbol{x},k}$. The data are assumed to be related by the linear model:

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

The work of C. Richard and A. Ferrari was partly supported by the ANR and the DGA, 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.

where $\boldsymbol{w}_k^\star$ is the $L \times 1$ unknown parameter vector (also called task) sought by node k, and $z_k(i)$ is a zero-mean measurement noise of variance $\sigma_{z,k}^2$. The noise process is assumed to be temporally white and spatially independent. Nodes in the same cluster are interested in the same estimation task, namely, $\boldsymbol{w}_k^\star = \boldsymbol{w}_{\mathcal{C}_q}^\star$ whenever node k belongs to cluster $\mathcal{C}_q$. A link between two nodes belonging to two different clusters means that their tasks have a large number of similar components and only a relatively small number of different components. To promote such relationships between optimum parameter vectors, appropriate sparsity-based co-regularizers can be used. Several works exist in the literature for solving sparse single-task estimation problems using diffusion strategies [11–13]. We shall use the notation $f(\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)})$ to refer to the real-valued convex function used to promote the sparsity of $\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}$. Combining local mean-square error functions and the regularization functions, the multitask estimation problem is formulated as the problem of seeking a fully distributed solution for solving the following regularized problem $(\mathcal{P})$:

$$\begin{aligned}\min_{\boldsymbol{w}_{\mathcal{C}_1},\ldots,\boldsymbol{w}_{\mathcal{C}_Q}} & J^{\text{glob}}(\boldsymbol{w}_{\mathcal{C}_1},\ldots,\boldsymbol{w}_{\mathcal{C}_Q}) \\ & = \sum_{k=1}^{N} \mathbb{E}\{|d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_{\mathcal{C}(k)}|^2\} + \\ & \eta \sum_{k=1}^{N} \sum_{\ell \in \mathcal{N}_k \backslash \mathcal{C}(k)} \rho_{k\ell} f(\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}),\end{aligned} \tag{2}$$

where $\eta > 0$ is the regularization strength used to enforce sparsity, and $\rho_{k\ell} \geq 0$ are weights for locally adjusting the regularization strength. The notation $\mathcal{N}_k \backslash \mathcal{C}(k)$ denotes the set of neighboring nodes of k that are not in the same cluster as k. Let us now describe the regularization functions considered in this work. Since the ℓ_0-norm is non-convex, two alternative convex regularization functions are considered. First, we use the ℓ_1-norm, namely, $f_1(\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}) = \|\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}\|_1$ whose subgradient vector with respect to $\boldsymbol{w}_{\mathcal{C}(k)}$ given $\boldsymbol{w}_{\mathcal{C}(\ell)}$ is taken as:

$$\partial_{\boldsymbol{w}_{\mathcal{C}(k)}} f_1 = \text{sign}(\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}), \tag{3}$$

where the entries of the vector $\text{sign}(\boldsymbol{w})$ are obtained by applying the following function to each entry of $\boldsymbol{w}$:

$$\text{sign}([\boldsymbol{w}]_m) = \begin{cases} [\boldsymbol{w}]_m / |[\boldsymbol{w}]_m|, \text{ if } [\boldsymbol{w}]_m \neq 0 \\ 0, \text{ otherwise.} \end{cases} \tag{4}$$

The ℓ_1-regularizer is known to uniformly shrink all the components of a vector and does not distinguish between zero and non-zero elements [14, 15]. To address this imbalance, we also consider a weighted formulation of the ℓ_1-norm regularization designed to enhance the penalization of the non-zero components of a vector [15]:

$$f_2(\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}) = \sum_{m=1}^{L} \alpha_m |[\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}]_m| \tag{5}$$

where the α_m are positive weights to be dynamically adjusted. To reduce the bias induced by the ℓ_1-norm and better approximate the ℓ_0-norm, the weights α_m are usually chosen as $\alpha_m(i) = 1/[\epsilon + |\boldsymbol{w}_{\mathcal{C}(k)}(i-1) - \boldsymbol{w}_{\mathcal{C}(\ell)}(i-1)]_m|]$ at each iteration i, with ϵ a small positive number to avoid division by zero. In this case, we write:

$$\partial_{\boldsymbol{w}_{\mathcal{C}(k)}} f_2 = \text{diag}\left\{ \frac{1}{\epsilon + |[\boldsymbol{\delta w}_{k,\ell}]_m|} \right\}_{m=1}^{L} \text{sign}(\boldsymbol{\delta w}_{k,\ell}) \tag{6}$$

where $\boldsymbol{\delta w}_{k,\ell}$ refers to the difference $\boldsymbol{w}_{\mathcal{C}(k)} - \boldsymbol{w}_{\mathcal{C}(\ell)}$.

We are interested in a distributed strategy for solving (2) that relies only on in-network processing. For this reason, we associate with the j-th cluster the regularized problem $(\mathcal{P}_j)$:

$$\begin{aligned}\min_{\boldsymbol{w}_{\mathcal{C}_j}} J_{\mathcal{C}_j}(\boldsymbol{w}_{\mathcal{C}_j}) = & \sum_{k \in \mathcal{C}_j} \mathbb{E}\left\{|d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_{\mathcal{C}_j}|^2\right\} + \\ & \eta \sum_{k \in \mathcal{C}_j} \sum_{\ell \in \mathcal{N}_k \backslash \mathcal{C}_j} (\rho_{k\ell} + \rho_{\ell k}) f(\boldsymbol{w}_{\mathcal{C}_j} - \boldsymbol{w}_{\mathcal{C}(\ell)}).\end{aligned} \tag{7}$$

Note that the cost functions in $(\mathcal{P})$ and $(\mathcal{P}_j)$ have the same subgradient vector with respect to $\boldsymbol{w}_{\mathcal{C}_j}$. In order that each node can solve the problem autonomously and adaptively using only local interactions, we shall derive a distributed iterative algorithm for solving $(\mathcal{P})$ by considering $(\mathcal{P}_j)$ since both cost functions have the same subgradient information.

2.2. Multitask diffusion with sparsity regularization

Proceeding as in [3, 16], it is possible to derive several diffusion strategies for solving $(\mathcal{P}_j)$ and $(\mathcal{P})$ in a fully distributed and adaptive manner. In this work, we focus on the Adapt-then-Combine (ATC) strategy. Based on the subgradient method for non-differential convex functions, we arrive at the following multitask diffusion algorithm for solving $(\mathcal{P})$:

$$\begin{cases} \boldsymbol{\psi}_k(i+1) \\ \quad = \boldsymbol{w}_k(i) + \mu_k \displaystyle\sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} c_{\ell k}\, \boldsymbol{x}_\ell(i)[d_\ell(i) - \boldsymbol{x}_\ell^\top(i)\boldsymbol{w}_k(i)] \\ \quad - \mu_k \eta \displaystyle\sum_{\ell \in \mathcal{N}_k \backslash \mathcal{C}(k)} \frac{1}{2}(\rho_{k\ell} + \rho_{\ell k})\, \partial_{\boldsymbol{w}_k} f(\boldsymbol{w}_k(i) - \boldsymbol{w}_\ell(i)) \\ \boldsymbol{w}_k(i+1) = \displaystyle\sum_{\ell \in \mathcal{N}_k \cap \mathcal{C}(k)} a_{\ell k} \boldsymbol{\psi}_\ell(i+1), \end{cases} \tag{8}$$

for $k = 1, \ldots, N$, where $\boldsymbol{w}_k(i)$ denotes the local estimate of $\boldsymbol{w}_k^\star$ at node k and iteration i, μ_k is a positive step-size parameter and $\partial_{\boldsymbol{w}_k} f$ is the subgradient of f with respect to $\boldsymbol{w}_k$, given $\boldsymbol{w}_\ell$. In the first step, which corresponds to the adaptation stage, the coefficients $c_{\ell k}$ are the weights that node k assigns to information coming from each node ℓ of its cluster. In the second step, that is, in the combination stage, node k combines through the coefficients $a_{\ell k}$ the intermediate estimates $\boldsymbol{\psi}_\ell(i+1)$ from its neighbors that belong to its cluster.

The non-negative coefficients $a_{\ell k}$ and $c_{\ell k}$ in (8) are required to satisfy the following constraints:

$$\sum_{k\in\mathcal{N}_\ell\cap\mathcal{C}(\ell)} c_{\ell k} = 1, \text{ and } c_{\ell k} = 0 \text{ if } k \notin \mathcal{N}_\ell \cap \mathcal{C}(\ell), \quad (9)$$

$$\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} a_{\ell k} = 1, \text{ and } a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \cap \mathcal{C}(k). \quad (10)$$

Coefficients $a_{\ell k}$ and $c_{\ell k}$ are grouped into a left-stochastic matrix $\boldsymbol{A}$ and a right-stochastic matrix $\boldsymbol{C}$.

3. PERFORMANCE ANALYSIS

3.1. Error vector recursion

Let $\boldsymbol{w}(i)$, $\boldsymbol{w}^\star$ and $\widetilde{\boldsymbol{w}}(i)$ denote the block weight estimate vector, the block optimum vector, and the block weight error vector, namely,

$$\begin{aligned} \boldsymbol{w}(i) &\triangleq \text{col}\{\boldsymbol{w}_1(i),\ldots,\boldsymbol{w}_N(i)\} && (11)\\ \boldsymbol{w}^\star &\triangleq \text{col}\{\boldsymbol{w}_1^\star,\ldots,\boldsymbol{w}_N^\star\} && (12)\\ \widetilde{\boldsymbol{w}}(i) &\triangleq \boldsymbol{w}^\star - \boldsymbol{w}(i). && (13)\end{aligned}$$

Using the linear data model (1), the error recursion for the diffusion strategy (8) can be written in the following form:

$$\widetilde{\boldsymbol{w}}(i+1) = \boldsymbol{\mathcal{B}}(i)\widetilde{\boldsymbol{w}}(i) - \boldsymbol{g}(i) + \boldsymbol{b}(i), \quad (14)$$

where

$$\begin{aligned} \boldsymbol{\mathcal{B}}(i) &\triangleq \boldsymbol{\mathcal{A}}^\top(\boldsymbol{I}_{LN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}_{\boldsymbol{x}}(i)), && (15)\\ \boldsymbol{g}(i) &\triangleq \boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{C}}^\top\text{col}\{\boldsymbol{x}_k(i)z_k(i)\}_{k=1}^N, && (16)\\ \boldsymbol{b}(i) &\triangleq \eta\,\boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}\,\boldsymbol{r}(i), && (17)\end{aligned}$$

with $\boldsymbol{\mathcal{A}} \triangleq \boldsymbol{A}\otimes\boldsymbol{I}_L$, $\boldsymbol{\mathcal{C}} \triangleq \boldsymbol{C}\otimes\boldsymbol{I}_L$. Matrices $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{R}}_{\boldsymbol{x}}(i)$ are $N\times N$ block diagonal with k-th block given by $\mu_k\boldsymbol{I}_L$ and $\sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\boldsymbol{x}_\ell(i)\boldsymbol{x}_\ell^\top(i)$, respectively. Let us denote by $p_{k\ell}$ the quantity $(\rho_{k\ell}+\rho_{\ell k})/2$, and introduce the $LN\times 1$ vector:

$$\boldsymbol{r}(i) = \text{col}\Big\{\sum_{\ell\in\mathcal{N}_k\setminus\mathcal{C}(k)} p_{k\ell}\,\partial_{\boldsymbol{w}_k} f(\boldsymbol{w}_k(i)-\boldsymbol{w}_\ell(i))\Big\}_{k=1}^N. \quad (18)$$

Recursion (14) can be used to examine the performance of the algorithm in the mean and mean-square-error sense. Due to space limitations, we only list the main results without showing the proofs. The arguments are along the lines developed in [3, 16] for single-task diffusion with proper adjustments to handle the multitask scenario.

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

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

3.2. Mean behavior analysis

For any initial conditions, the multitask diffusion strategy (8) asymptotically converges in the mean if the step-sizes satisfy:

$$0 < \mu_k < \frac{2}{\lambda_{\max}(\boldsymbol{R}_k)}, \quad k = 1,\ldots,N, \quad (19)$$

where $\boldsymbol{R}_k \triangleq \sum_{\ell\in\mathcal{N}_k\cap\mathcal{C}(k)} c_{\ell k}\boldsymbol{R}_{\boldsymbol{x},\ell}$ and $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue of its matrix argument. The asymptotic mean bias is given by

$$\lim_{i\to\infty}\mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\} = \eta(\boldsymbol{I}_{LN}-\boldsymbol{\mathcal{B}})^{-1}\boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}\lim_{i\to\infty}\mathbb{E}\{\boldsymbol{r}(i)\}, \quad (20)$$

where

$$\boldsymbol{\mathcal{B}} = \mathbb{E}\{\boldsymbol{\mathcal{B}}(i)\} = \boldsymbol{\mathcal{A}}^\top(\boldsymbol{I}_{LN} - \boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{R}}), \quad (21)$$

with $\boldsymbol{\mathcal{R}}$ denoting the $N\times N$ block diagonal matrix whose k-th block is $\boldsymbol{R}_k$. Recall that the block maximum norm of an $N\times 1$ block vector $\boldsymbol{x}$ is defined as [3]:

$$\|\boldsymbol{x}\|_{b,\infty} = \max_{1\le k\le N}\|\boldsymbol{x}_k\|_2 \quad (22)$$

where $\boldsymbol{x}_k$ is the k-th block entry. The block maximum norm of the mean bias (20) can be bounded as follows:

$$\lim_{i\to\infty}\|\mathbb{E}\{\widetilde{\boldsymbol{w}}(i)\}\|_{b,\infty} \le \frac{\eta\mu_{\max}r_{\max}}{1-\|\boldsymbol{\mathcal{B}}\|_{b,\infty}}, \quad (23)$$

where $\mu_{\max}$ is the largest step-size and $r_{\max} \triangleq \max_i\|\boldsymbol{r}(i)\|_{b,\infty}$. Note that $r_{\max}$ is finite since $\|\boldsymbol{r}(i)\|_{b,\infty}$ is upper bounded by $\max_{1\le k\le N}\sum_{\ell=1}^N p_{k\ell}\|\partial_{\boldsymbol{w}_k} f(\boldsymbol{w}_k(i)-\boldsymbol{w}_\ell(i))\|_2$ and the Eucledian norm of (3) and (6) is bounded by $\sqrt{L}$ and $\frac{\sqrt{L}}{\epsilon}$, respectively. Under condition (19), the induced block maximum norm of $\boldsymbol{\mathcal{B}}$ is strictly less than 1.

3.3. Mean-square-error stability

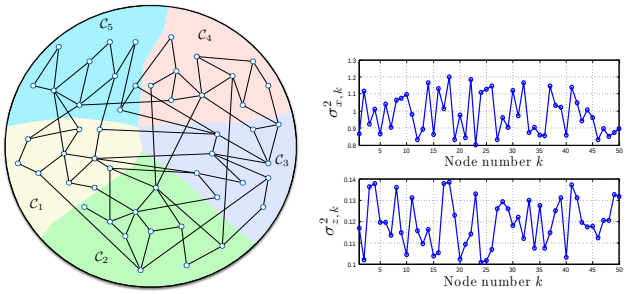
To examine mean-square-error stability, we study the weighted mean-square deviation $\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|_{\boldsymbol{\Sigma}}^2\}$, where $\boldsymbol{\Sigma}$ is a positive semi-definite matrix that we are free to choose. Let us denote by $\boldsymbol{\sigma}$ the vectorized version of $\boldsymbol{\Sigma}$. We obtain from (14) the following recursion:

$$\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|_{\boldsymbol{\sigma}}^2\} = \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|_{\boldsymbol{\mathcal{F}}\boldsymbol{\sigma}}^2\} + [\text{vec}(\boldsymbol{G})]^\top\boldsymbol{\sigma} + h(i), \quad (24)$$

where we use the notation $\|\boldsymbol{x}\|_{\boldsymbol{\Sigma}}^2$ and $\|\boldsymbol{x}\|_{\boldsymbol{\sigma}}^2$ interchangeably to denote the same quantity $\boldsymbol{x}^\top\boldsymbol{\Sigma}\,\boldsymbol{x}$. The other terms in (24) are given by:

$$\begin{aligned} \boldsymbol{G} &\triangleq \boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}\,\boldsymbol{\mathcal{C}}^\top\boldsymbol{\mathcal{S}}\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{A}} && (25)\\ \boldsymbol{\mathcal{F}} &\triangleq \mathbb{E}\{\boldsymbol{\mathcal{B}}^\top(i)\otimes\boldsymbol{\mathcal{B}}^\top(i)\} \approx \boldsymbol{\mathcal{B}}^\top\otimes\boldsymbol{\mathcal{B}}^\top && (26)\\ h(i) &\triangleq \eta^2\,\mathbb{E}\{\|\boldsymbol{r}(i)\|_{\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{A}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{M}}}^2\} + \\ &\quad 2\eta\,\mathbb{E}\{\boldsymbol{r}^\top(i)\,\boldsymbol{\mathcal{M}}\,\boldsymbol{\mathcal{A}}\,\boldsymbol{\Sigma}\,\boldsymbol{\mathcal{B}}\,\widetilde{\boldsymbol{w}}(i)\}, && (27)\end{aligned}$$

where $\boldsymbol{\mathcal{S}}$ is an $N\times N$ block diagonal matrix whose k-th block is $\sigma_{z,k}^2\boldsymbol{R}_{\boldsymbol{x},k}$. The approximation in (26) follows from Assumption 2 and requires sufficiently small step-sizes. For any



(a) Network topology. (b) Regression and noise variances.

Fig. 1. Experimental setup.

initial conditions, the multitask diffusion algorithm with sparsity based regularization (8) is mean-square stable if the error recursion (14) is mean stable and the matrix $\boldsymbol{\mathcal{F}}$ is stable. Once convergence is achieved, then the variance of the weight error vector $\widetilde{\boldsymbol{w}}(i)$ satisfies the following relation in steady-state:

$$\lim_{i\to\infty} \mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i+1)\|^2_{(\boldsymbol{I}-\boldsymbol{\mathcal{F}})\boldsymbol{\sigma}}\} = [\mathrm{vec}(\boldsymbol{G})]^\top \boldsymbol{\sigma} + h_\infty, \quad (28)$$

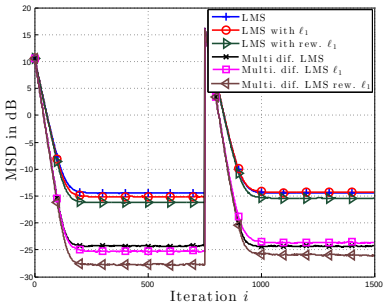
where it can be argued that $h_\infty \triangleq \lim_{i\to\infty} h(i)$ exists. Through a proper selection of the weighting matrix $\boldsymbol{\Sigma}$ or vector $\boldsymbol{\sigma}$, relation (28) allows us to derive several performance metrics such as the mean-square deviation (MSD) for network or nodes. For example, the network MSD given by $\lim_{i\to\infty} \frac{1}{N}\mathbb{E}\{\|\widetilde{\boldsymbol{w}}(i)\|^2\}$ is obtained for

$$\boldsymbol{\sigma} = \frac{1}{N}(\boldsymbol{I}-\boldsymbol{\mathcal{F}})^{-1}\mathrm{vec}(\boldsymbol{I}_{NL}). \quad (29)$$

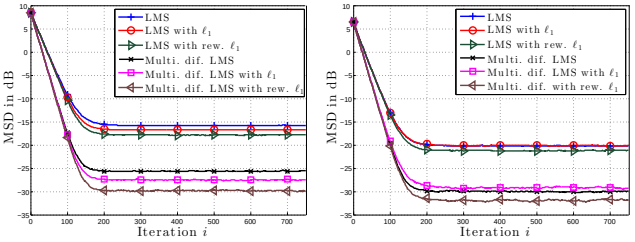
4. SIMULATION RESULTS

We considered a network consisting of 50 agents with the topology shown in Fig. 1(a). The regression vectors were zero-mean Gaussian distributed with covariance $\boldsymbol{R}_{\boldsymbol{x},k} = \sigma^2_{\boldsymbol{x},k}\boldsymbol{I}_L$. The noises $z_k(i)$ were zero-mean i.i.d. Gaussian random variables, independent of any other signal, with variance $\sigma^2_{z,k}$. The variances $\sigma^2_{\boldsymbol{x},k}$ and $\sigma^2_{z,k}$ are shown in Fig. 2(b). We ran the proposed algorithm by setting $c_{\ell k} = |\mathcal{N}_\ell \cap \mathcal{C}(\ell)|^{-1}$ for $k \in \mathcal{N}_\ell \cap \mathcal{C}(\ell)$ and $a_{\ell k} = |\mathcal{N}_k \cap \mathcal{C}(k)|^{-1}$ for $\ell \in \mathcal{N}_k \cap \mathcal{C}(k)$. The regularization weights were set to $\rho_{k\ell} = |\mathcal{N}_k \setminus \mathcal{C}(k)|^{-1}$ for $\ell \in \mathcal{N}_k \setminus \mathcal{C}(k)$. We used a constant step-size $\mu_k = 0.03$ for all k, a sparsity strength $\eta = 0.03$ for the ℓ_1-regularization, $\eta = 0.015$ for the reweighted ℓ_1-regularization with $\epsilon = 0.1$. The results were averaged over 100 Monte-Carlo runs.

The optimum vectors were set to $\boldsymbol{w}^\star_{\mathcal{C}_j} = \boldsymbol{w}_0 + \boldsymbol{\delta}_{\mathcal{C}_j}$ at each cluster with $\boldsymbol{w}_0 = [-2\ -1\ -1\ \mathbf{0}_3\ -2\ 0\ 1\ \mathbf{0}_3\ 1\ 2\ 1]^\top$. First, we set $\boldsymbol{\delta}_{\mathcal{C}_1}$ to $\mathbf{0}_{15}$, $\boldsymbol{\delta}_{\mathcal{C}_2}$ to $[2\ \mathbf{0}_{11}\ -1\ \mathbf{0}_2]^\top$, $\boldsymbol{\delta}_{\mathcal{C}_3}$ to $[2\ \mathbf{0}_5\ 2\ \mathbf{0}_8]^\top$, $\boldsymbol{\delta}_{\mathcal{C}_4}$ to $[2\ 1\ \mathbf{0}_4\ 2\ \mathbf{0}_5\ -1\ \mathbf{0}_2]^\top$ and $\boldsymbol{\delta}_{\mathcal{C}_5}$ to $[0\ 1\ \mathbf{0}_4\ 2\ \mathbf{0}_8]^\top$. Observe that at most 4 entries over 15 differed between clusters. After 750 iterations, we set $\boldsymbol{\delta}_{\mathcal{C}_2}$ to $[2\ 1\ 1\ 2\ 1\ 1\ 2\ \mathbf{0}_8]$, $\boldsymbol{\delta}_{\mathcal{C}_3}$ to $[3\ 2\ 2\ 3\ 2\ 2\ 3\ \mathbf{0}_8]^\top$, $\boldsymbol{\delta}_{\mathcal{C}_4}$ to $[4\ 3\ 3\ 4\ 3\ 3\ 4\ \mathbf{0}_8]^\top$ and $\boldsymbol{\delta}_{\mathcal{C}_5}$ to $[5\ 4\ 4\ 5\ 4\ 4\ 5\ \mathbf{0}_8]^\top$. In this way, 7 entries over 15 differed between clusters. We compared 6 algorithms: the non-cooperative LMS algorithm, the so-called spatially regularized LMS algorithm [17] ($\boldsymbol{A} = \boldsymbol{C} = \boldsymbol{I}$) with ℓ_1-norm and reweighted ℓ_1-norm, the multitask diffusion LMS algorithm obtained from (8) by setting $\eta = 0$, and the multitask diffusion LMS algorithm with ℓ_1 and reweighted ℓ_1-norm regularization.



(a) Network MSD comparison.



(b) MSD over identical entries. (c) MSD over distinct components.

Fig. 2. MSD learning curves.

As shown in Fig. 2(a), when the optimums share a sufficient number of common entries, the multitask strategies with ℓ_1-norm and reweighted ℓ_1-norm regularization enhance the network MSD performance. When the number of common entries decreases, sparsity-promoting regularizers become less efficient and only the reweighted ℓ_1-norm regularizer allows to improve the performance. In Fig. 2(b), we show the MSD learning curves for the common parameter vector entries among clusters. Due to cooperation among clusters, we observe that the multitask approach makes the estimation of these entries more accurate. In Fig. 2(c), we report the learning curves over instants $[0\ 750]$ for entries that differ among clusters. We note that the reweighted ℓ_1-norm algorithm outperforms the other algorithms.

5. CONCLUSION

In this work, we proposed a diffusion-type algorithm for solving problems that require a simultaneous estimation of multiple parameter vectors with a prior information on similarities between neighboring clusters. Two different sparsity-based regularization terms were used, the ℓ_1-norm and the reweighted ℓ_1-norm. We examined conditions for stability in the mean and mean-square sense. Simulations results were presented to illustrate the benefit of multitask learning with similarity measures.

6. REFERENCES

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

[2] F. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, March 2010.

[3] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Libraray in Signal Processing*, Elsevier, Ed., vol. 3, pp. 322–454. Elsevier, 2014.

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

[5] J. Chen and A. H. Sayed, "Distributed Pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, April 2013.

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

[7] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *To appear in IEEE Transactions on Signal Processing*, 2015. Extended report available as arXiv:1404.6813 [cs.MA], Apr. 2014.

[8] J. Chen, C. Richard, A. O. Hero, and A. H. Sayed, "Diffusion LMS for multitask problems with overlapping hypothesis subspaces," in *Proc. IEEE MLSP*, Reims, France, September 2014, pp. 1–6.

[9] N. Bogdanovic, J. Plata-Chaves, and K. Berberidis, "Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks," in *Proc. IEEE ICASSP*, Florence, Italy, May 2014, pp. 7223–7227.

[10] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, August 2014.

[11] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Sparse diffusion LMS for distributed adaptive estimation," in *Proc. ICASSP*, Kyoto, Japan, March 2012, pp. 3281–3284.

[12] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4480–4485, August 2012.

[13] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5412–5425, October 2012.

[14] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. IEEE ICASSP*, Taipei, Taiwan, April 2009, pp. 3125–3128.

[15] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 2008.

[16] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Transactions on Signal Processing*, vol. 61, no. 6, pp. 1419–1433, March 2013.

[17] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS for clustered multitask networks," in *Proc. IEEE ICASSP*, Florence, Italy, May 2014, pp. 5487–5491.