Distributed learning over multitask networks with linearly related tasks

Roula Nassif[†], Cédric Richard[†], André Ferrari[†], Ali H. Sayed[‡]
[†]Université Côte d'Azur, OCA, CNRS, France
Email: {roula.nassif, cedric.richard, andre.ferrari}@unice.fr
[‡]University of California, Los Angeles, USA
Email: sayed@ee.ucla.edu

***Abstract*—In this work, we consider distributed adaptive learning over multitask mean-square-error (MSE) networks where each agent is interested in estimating its own parameter vector, also called task, and where the tasks at neighboring agents are related according to a set of linear equality constraints. We assume that each agent knows its own cost function of its vector and the set of constraints involving its vector. In order to solve the multitask problem and to optimize the individual costs subject to all constraints, a projection based diffusion LMS approach is derived and studied. Simulation results illustrate the efficiency of the strategy.**

I. Introduction

Distributed adaptive learning allows a collection of interconnected agents to perform parameter estimation tasks from streaming data by relying solely on local computations and interactions with immediate neighbors. Most prior literature focuses on *single-task* problems, where agents with separable objective functions need to agree on a common parameter vector corresponding to the minimizer of the aggregate sum of individual costs [1]–[8]. However, many network applications require more complex models and flexible algorithms than single-task implementations since their agents may need to estimate and track multiple objectives simultaneously [9]–[16]. Networks of this kind, where agents need to infer multiple parameter vectors, are referred to as *multitask* networks. Although agents may generally have distinct though related tasks to perform, they may still be able to capitalize on inductive transfer between them to improve their estimation accuracy [9], [11], [12]. For example, sensor networks deployed to estimate a spatially varying temperature profile need to exploit more directly spatiotemporal correlations that exist between measurements at neighboring nodes [13]. In another example, in distributed power system monitoring, the local state vectors to be estimated at neighboring control centers may overlap partially since the areas in a power system are interconnected [14]. Likewise, in distributed wireless acoustic sensor networks, neighboring agents need to estimate different but overlapping active noise control filters [15].

In this work, we consider multitask estimation problems where the parameter vectors to be estimated at neighboring agents are related according to a set of linear equality constraints. That is, we consider multitask optimization problems subject to linear equality constraints of the form:

$$\underset{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_N}{\text{minimize}}\; J^{\text{glob}}(\boldsymbol{w}_1,\ldots,\boldsymbol{w}_N) \triangleq \sum_{k=1}^{N} J_k(\boldsymbol{w}_k), \tag{1a}$$

$$\text{subject to} \sum_{\ell\in\mathcal{I}_p} \boldsymbol{D}_{p\ell}\,\boldsymbol{w}_\ell + \boldsymbol{b}_p = \boldsymbol{0}, \quad p = 1,\ldots,P, \tag{1b}$$

where N is the number of agents in the network. Each agent k seeks to estimate its own $M_k \times 1$ parameter vector $\boldsymbol{w}_k$, and has knowledge of its cost $J_k(\cdot)$ and the set of linear equality constraints that agent k is involved in. Each constraint is indexed by p, and defined by the $L_p \times M_\ell$ matrices $\boldsymbol{D}_{p\ell}$, the $L_p \times 1$ vector $\boldsymbol{b}_p$, and the set $\mathcal{I}_p$ of agent indices involved in this constraint. It is assumed that each agent k in $\mathcal{I}_p$ can collect estimates from all agents in $\mathcal{I}_p$ to satisfy the p-th constraint, i.e., $\mathcal{I}_p \subseteq \mathcal{N}_k$ where $\mathcal{N}_k$ denotes the neighborhood of agent k. In the sequel, we propose a primal technique (based on propagating and estimating primal variables) for solving problem (1) in a distributed manner. The technique relies on combining a diffusion adaptation principle with a stochastic gradient projection step, and on the use of constant step-sizes to enable continuous learning from streaming data. We illustrate the main results of the analysis of the proposed algorithm in the mean and mean-square-error sense. Simulation results are conducted to show the effectiveness of the proposed strategy.

Notation: We use normal font letters to denote scalars, boldface lowercase letters to denote column vectors, and boldface upper case letters to denote matrices. The symbols $(\cdot)^\top$, $(\cdot)^{-1}$, and $(\cdot)^\dagger$ denote matrix transpose, matrix inverse, and matrix pseudo-inverse, respectively.

II. Problem formulation and distributed solution

Consider a network consisting of N agents, labeled $k = 1,\ldots,N$. At each time instant i, each agent in the network has access to a zero-mean observation $d_k(i)$, and a zero-mean real-valued $M_k \times 1$ regression vector $\boldsymbol{x}_k(i)$, with positive covariance matrix $\boldsymbol{R}_{x,k} = \mathbb{E}\{\boldsymbol{x}_k(i)\boldsymbol{x}_k^\top(i)\} > \boldsymbol{0}$. We assume the data to be related via the linear data model:

$$d_k(i) = \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k^o + z_k(i), \tag{2}$$

The work of C. Richard and A. Ferrari was supported in part by ANR and DGA grant ANR-13-ASTR-0030 (ODISSEE project).
The work of A. H. Sayed was supported in part by NSF grants CCF-1524250 and ECCS-1407712.

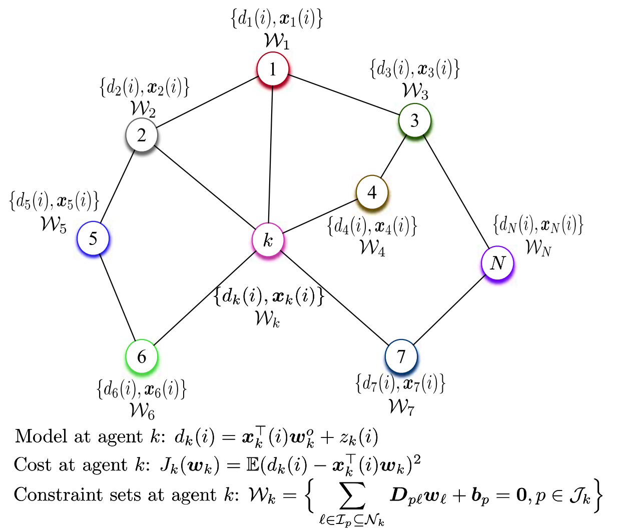

Fig. 1. Multitask MSE network with local linear equality constraints.

where $\boldsymbol{w}_k^o$ is an $M_k \times 1$ unknown parameter vector, and $z_k(i)$ is a zero-mean measurement noise of variance $\sigma_{z,k}^2$, independent of $\boldsymbol{x}_\ell(j)$ for all ℓ and j, and independent of $z_\ell(j)$ for $\ell \neq k$ or $i \neq j$. We let $\boldsymbol{r}_{dx,k} \triangleq \mathbb{E}\{d_k(i)\boldsymbol{x}_k(i)\}$ and $\sigma_{d,k}^2 \triangleq \mathbb{E}|d_k(i)|^2$.

Let $\boldsymbol{w}_k$ denote some generic $M_k \times 1$ vector that is associated with agent k. The objective at agent k is to find an estimate for $\boldsymbol{w}_k^o$, and we associate with this agent the mean-square-error criterion:

$$J_k(\boldsymbol{w}_k) = \mathbb{E}\left(d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k\right)^2. \quad (3)$$

In addition, P linear equality constraints ($1 \leq \sum_{p=1}^P L_p < \sum_{k=1}^N M_k$) of the form (1b) are imposed on the vectors $\{\boldsymbol{w}_k\}$ of neighboring agents. Let $\mathcal{I}_p$ be the set of agent indices involved in the p-th constraint. Consider agent $k \in \mathcal{I}_p$. We assume that k is aware of the p-th constraint, and that it can collect estimates from all agents in $\mathcal{I}_p$ in order to satisfy the p-th constraint, that is, $\mathcal{I}_p \subseteq \mathcal{N}_k$. This assumption is reasonable in many distributed monitoring applications over networks [17]. An illustrative example for the network considered in this work is provided in Fig. 1 where $\mathcal{J}_k$ denotes the set of constraint indices containing agent k, i.e., $\mathcal{J}_k = \{p \mid k \in \mathcal{I}_p\}$.

Let us collect the parameter vectors $\{\boldsymbol{w}_k\}$ and $\{\boldsymbol{w}_k^o\}$ from across the network into $N \times 1$ block column vectors:

$$\boldsymbol{w} = \text{col}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N\}, \quad \boldsymbol{w}^o = \text{col}\{\boldsymbol{w}_1^o, \ldots, \boldsymbol{w}_N^o\}, \quad (4)$$

and let us write the P linear equality constraints in (1b) more compactly as:

$$\boldsymbol{\mathcal{D}}\boldsymbol{w} + \boldsymbol{b} = \boldsymbol{0}, \quad (5)$$

where $\boldsymbol{\mathcal{D}}$ is a $P \times N$ block matrix whose (p, ℓ)-th block is given by $\boldsymbol{D}_{p\ell}$ if $\ell \in \mathcal{I}_p$ and $\boldsymbol{0}$ otherwise, and $\boldsymbol{b}$ is a $P \times 1$ block column vector whose p-th block is given by $\boldsymbol{b}_p$. We assume that $\boldsymbol{\mathcal{D}}$ is full row rank to ensure that (5) has at least one solution. Combining (1), (3), and (5), we find that the closed form solution of problem (1) is unique and given by:

$$\boldsymbol{w}^\star = \boldsymbol{w}^o - \boldsymbol{\mathcal{R}}_x^{-1}\boldsymbol{\mathcal{D}}^\top(\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{R}}_x^{-1}\boldsymbol{\mathcal{D}}^\top)^{-1}(\boldsymbol{\mathcal{D}}\boldsymbol{w}^o + \boldsymbol{b}), \quad (6)$$

where $\boldsymbol{w}^\star$ is a block vector with N subvectors of size $M_k \times 1$ each. Moreover, the matrix $\boldsymbol{\mathcal{R}}_x$ is given by:

$$\boldsymbol{\mathcal{R}}_x \triangleq \text{diag}\{\boldsymbol{R}_{x,1}, \ldots, \boldsymbol{R}_{x,N}\}. \quad (7)$$

In the sequel, we show how each agent k in the network illustrated in Fig. 1 can estimate the k-th sub-vector $\boldsymbol{w}_k^\star$ of $\boldsymbol{w}^\star$ in (6) in a distributed and adaptive manner. To do so, we first transform (1) into an equivalent optimization problem exhibiting structure amenable for distributed optimization with separable constraints. Let j_k denote the number of constraints that agent k is involved in, i.e., $j_k = |\mathcal{J}_k|$. We expand each node k into a cluster $\mathcal{C}_k$ of j_k *virtual* sub-nodes, namely, $\mathcal{C}_k \triangleq \{k_m\}_{m=1}^{j_k}$. Each one of these sub-nodes is involved in a single constraint. Let $\boldsymbol{w}_{k_m}$ denote the $M_k \times 1$ auxiliary vector associated with sub-node k_m. In order to ensure that agent k satisfies simultaneously all the constraints at convergence, we will allow all sub-nodes at agent k to run diffusion learning to reach agreement on their estimates $\{\boldsymbol{w}_{k_m}\}$ asymptotically. An illustrative example is provided in Fig. 2. On the left of this panel is the original network topology with $N = 6$ agents and $P = 3$ constraints. On the right is the network topology *model* with clusters of sub-nodes shown in grey color. Observe that $\mathcal{I}_2 = \{1, 3, k\}$ and $\mathcal{I}_3 = \{4, k, \ell\}$, which means that agent k is involved in constraints 2 and 3. Thus, agent k is expanded into a cluster $\mathcal{C}_k = \{k_1, k_2\}$ of 2 sub-nodes. Sub-nodes k_1 and k_2 are assigned to constraints 2 and 3, respectively. Each other agent, say ℓ, involved in a single-constraint is renamed ℓ_1 and assigned to a single-node cluster $\mathcal{C}_\ell = \{\ell_1\}$ for consistency of notation. This leads to the sets $\mathcal{I}_{e,2} = \{1_2, 3_1, k_1\}$ and $\mathcal{I}_{e,3} = \{4_1, \ell_1, k_2\}$ where all sub-nodes are involved in a single constraint.

Accordingly, we can reformulate problem (1) into the following equivalent form by introducing the auxiliary variables $\{\boldsymbol{w}_{k_m}\}$:

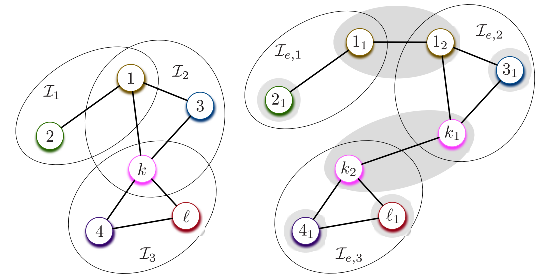

Fig. 2. (*Left*) Network topology with constraints identified by the sets $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$. (*Right*) Network topology model with virtual clusters shown in grey and constraints now identified by the sets of sub-nodes $\mathcal{I}_{e,1}$, $\mathcal{I}_{e,1}$, and $\mathcal{I}_{e,3}$ where all sub-nodes are involved in one constraint. Diffusion learning is run in clusters with more than one sub-node in order to reach agreement on local estimates while satisfying their respective constraints.

$$\underset{\boldsymbol{w}_e}{\text{minimize}} \quad \sum_{k=1}^{N}\sum_{m=1}^{j_k} c_{k_m} J_k(\boldsymbol{w}_{k_m}), \tag{8a}$$

$$\text{subject to} \quad \sum_{\ell_n \in \mathcal{I}_{e,p}} \boldsymbol{D}_{p\ell_n}\, \boldsymbol{w}_{\ell_n} + \boldsymbol{b}_p = \boldsymbol{0}, \quad p = 1,\ldots,P, \tag{8b}$$

$$\boldsymbol{w}_{k_1} = \ldots = \boldsymbol{w}_{k_{j_k}}, \quad k = 1,\ldots,N, \tag{8c}$$

where $\boldsymbol{w}_e$ is an $N_e \times 1$ network block column vector:

$$\boldsymbol{w}_e \triangleq \text{col}\left\{\text{col}\left\{\boldsymbol{w}_{k_m}\right\}_{m=1}^{j_k}\right\}_{k=1}^{N}, \tag{9}$$

with $N_e \triangleq \sum_{k=1}^{N} j_k$, and $\{c_{k_m}\}$ are free positive coefficients chosen by the user such that:

$$c_{k_m} > 0, \text{ for } m = 1,\ldots,j_k, \quad \sum_{m=1}^{j_k} c_{k_m} = 1. \tag{10}$$

Let $\boldsymbol{w}_k^\star$ denote the $(M_k \times 1)$ k-th block of $\boldsymbol{w}^\star$ in (6). The closed form solution $\boldsymbol{w}_e^\star$ of problem (8) can be written as:

$$\boldsymbol{w}_e^\star = \text{col}\{\mathbb{1}_{j_k \times 1} \otimes \boldsymbol{w}_k^\star\}_{k=1}^{N}. \tag{11}$$

In the following, we shall address the equality constraints (8c) with a single-task diffusion algorithm [3], [6]–[8] within each cluster of sub-nodes with the objective of reaching an agreement within each cluster (all sub-nodes at agent k converge to the same estimate).

Based on the gradient projection and diffusion strategies principles, we propose in the following an iterative algorithm to solve (8) in a distributed manner [17]. Let $\boldsymbol{w}_{e,p}$ denote the $i_p \times 1$ block column vector given by $\boldsymbol{w}_{e,p} \triangleq \text{col}\{\boldsymbol{w}_{\ell_n}\}_{\ell_n \in \mathcal{I}_{e,p}}$ where i_p is the number of nodes (or equivalently sub-nodes) involved in the p-th constraint and let Ω_p denote the linear manifold corresponding to the p-th constraint in (8b), namely, $\Omega_p \triangleq \{\boldsymbol{\mathcal{D}}_p \boldsymbol{w}_{e,p} + \boldsymbol{b}_p = \boldsymbol{0}\}$ where $\boldsymbol{\mathcal{D}}_p$ is a $1 \times i_p$ block matrix. We assume that $k_m \in \mathcal{I}_{e,p}$ and we let $\boldsymbol{w}_{k_m}(i)$ denote the estimate of $\boldsymbol{w}_k^\star$ at sub-node k_m and iteration i. Starting from $\boldsymbol{w}_{k_m}(0) = \boldsymbol{w}_k(0)$ for $m = 1,\ldots,j_k$, our multitask diffusion algorithm consists of three steps [17], namely, an adaptation step (12a), a projection step (12b) involving exchange of estimates between agents, and a combination step (12c):

$$\boldsymbol{\psi}_{k_m}(i+1) = \boldsymbol{w}_{k_m}(i) + \mu c_{k_m} \boldsymbol{x}_k(i)\left(d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_{k_m}(i)\right), \tag{12a}$$

$$\boldsymbol{\phi}_{k_m}(i+1) = [\boldsymbol{\mathcal{P}}_p]_{k_m,\bullet} \cdot \text{col}\{\boldsymbol{\psi}_{\ell_n}(i+1)\}_{\ell_n \in \mathcal{I}_{e,p}} - [\boldsymbol{f}_p]_{k_m}, \tag{12b}$$

$$\boldsymbol{w}_{k_m}(i+1) = \sum_{k_n \in \mathcal{N}_{k_m} \cap \mathcal{C}_k} a_{k_n,k_m} \boldsymbol{\phi}_{k_n}(i+1), \tag{12c}$$

where

$$\boldsymbol{\mathcal{P}}_p \triangleq \boldsymbol{I} - \boldsymbol{\mathcal{D}}_p^\dagger \boldsymbol{\mathcal{D}}_p, \quad \boldsymbol{f}_p \triangleq \boldsymbol{\mathcal{D}}_p^\dagger \boldsymbol{b}_p, \tag{13}$$

where the notation $\boldsymbol{\mathcal{D}}_p^\dagger$ represents the pseudo-inverse matrix of $\boldsymbol{\mathcal{D}}_p$. Moreover, $\mathcal{N}_{k_m} \cap \mathcal{C}_k$ is a *virtual* set of neighboring sub-nodes of k_m in $\mathcal{C}_k$ chosen by the designer ($\mathcal{C}_k$ is allowed to be strongly connected in order to reach an agreement at each sub-node k_m and satisfy all the constraints at agent k [17]). The non-negative combination coefficients $\{a_{k_n,k_m}\}$ are chosen to satisfy:

$$\sum_{k_m \in \mathcal{N}_{k_n} \cap \mathcal{C}_k} a_{k_n,k_m} = 1, \quad \sum_{k_n \in \mathcal{N}_{k_m} \cap \mathcal{C}_k} a_{k_n,k_m} = 1, \quad \text{and } a_{k_n,k_m} = 0 \text{ if } k_n \notin \mathcal{N}_{k_m} \cap \mathcal{C}_k. \tag{14}$$

By setting c_{k_m} to $\frac{1}{j_k}$ for all $m = 1,\ldots,j_k$, and combining the intermediate estimates $\boldsymbol{\phi}_{k_m}(i+1)$ at each sub-node k_m with the estimates of all other sub-nodes at agent k using uniform combination coefficients, i.e., $\mathcal{N}_{k_m} \cap \mathcal{C}_k = \mathcal{C}_k$ and $a_{k_n,k_m} = \frac{1}{j_k}$ for $n = 1,\ldots,j_k$, (12a) and (12c) reduce to:

$$\boldsymbol{\psi}_{k_m}(i+1) = \boldsymbol{\psi}_k(i+1), \quad \text{and} \quad \boldsymbol{w}_{k_m}(i+1) = \boldsymbol{w}_k(i+1), \tag{15}$$

for $m = 1,\ldots,j_k$, where $\boldsymbol{\psi}_k(i+1)$ and $\boldsymbol{w}_k(i+1)$ are computed by agent k according to:

$$\boldsymbol{\psi}_k(i+1) = \boldsymbol{w}_k(i) + \frac{\mu}{j_k}\boldsymbol{x}_k(i)\big(d_k(i) - \boldsymbol{x}_k^\top(i)\boldsymbol{w}_k(i)\big), \tag{16}$$

$$\boldsymbol{w}_k(i+1) = \frac{1}{j_k}\sum_{n=1}^{j_k} \boldsymbol{\phi}_{k_n}(i+1). \tag{17}$$

III. PERFORMANCE ANALYSIS RESULTS

Due to space limitations, we only list the main results of the analysis without showing the proofs[1]. Note that, throughout the analysis, the regression vectors $\boldsymbol{x}_k(i)$ are assumed to be zero-mean, temporally white, and spatially independent.

A. Mean behavior

Let us introduce the $N_e \times 1$ network block error vector:

$$\widetilde{\boldsymbol{w}}_e(i) = \boldsymbol{w}_e^\star - \text{col}\{\text{col}\{\boldsymbol{w}_{k_m}(i)\}_{m=1}^{j_k}\}_{k=1}^{N}. \tag{18}$$

The mean error vector of algorithm (12) evolves according to [17]:

$$\mathbb{E}\,\widetilde{\boldsymbol{w}}_e(i+1) = \boldsymbol{\mathcal{B}}\,\mathbb{E}\,\widetilde{\boldsymbol{w}}_e(i) - \mu \boldsymbol{r}, \tag{19}$$

where

$$\boldsymbol{\mathcal{B}} \triangleq \boldsymbol{\mathcal{A}}^\top \boldsymbol{\mathcal{P}}_e(\boldsymbol{I} - \mu \boldsymbol{\mathcal{R}}_{x,e}), \tag{20}$$

$$\boldsymbol{r} \triangleq \boldsymbol{\mathcal{A}}^\top \boldsymbol{\mathcal{P}}_e \boldsymbol{\mathcal{R}}_{x,e}(\boldsymbol{w}_e^o - \boldsymbol{w}_e^\star), \tag{21}$$

$$\boldsymbol{\mathcal{P}}_e \triangleq \boldsymbol{I} - \boldsymbol{\mathcal{D}}_e^\dagger \boldsymbol{\mathcal{D}}_e, \tag{22}$$

$$\boldsymbol{\mathcal{R}}_{x,e} \triangleq \text{diag}\{\boldsymbol{C}_k \otimes \boldsymbol{R}_{x,k}\}_{k=1}^{N}, \tag{23}$$

where $\boldsymbol{\mathcal{A}} \triangleq \text{diag}\{\boldsymbol{A}_k \otimes \boldsymbol{I}\}_{k=1}^{N}$, $\boldsymbol{A}_k$ is the $j_k \times j_k$ doubly-stochastic matrix whose (n,m)-th element is a_{k_n,k_m}, $\boldsymbol{C}_k \triangleq \text{diag}\{c_{k_m}\}_{m=1}^{j_k}$, $\boldsymbol{\mathcal{D}}_e$ is a $P \times N_e$ matrix constructed according to (8b), and $\boldsymbol{w}_e^o \triangleq \text{col}\{\mathbb{1}_{j_k \times 1} \otimes \boldsymbol{w}_k^o\}$. Recursion (19) converges as $i \to \infty$ if the matrix $\boldsymbol{\mathcal{B}}$ is stable. The stability of the matrix $\boldsymbol{\mathcal{B}}$ is ensured by choosing μ such that:

$$0 < \mu < \frac{2}{c_{k,\max} \cdot \lambda_{\max}(\boldsymbol{R}_{x,k})}, \quad \forall k = 1,\ldots,N, \tag{24}$$

[1]The arguments are available in the technical report [17].

where $c_{k,\max} \triangleq \max_{1\leq m\leq j_k} c_{k_m}$. The asymptotic mean bias is given by:

$$\lim_{i\to\infty} \mathbb{E}\,\widetilde{\boldsymbol{w}}_e(i) = -\mu(\boldsymbol{I}-\boldsymbol{\mathcal{B}})^{-1}\boldsymbol{r}. \tag{25}$$

The bias is zero in two cases: 1) in the perfect model scenario where $\{\boldsymbol{w}_k^o\}$ satisfy the constraints ($\boldsymbol{w}_e^o = \boldsymbol{w}_e^\star$); 2) if each agent is involved in at most one constraint.

B. Mean-square-error behavior

Algorithm (12) is stable in the mean-square-error sense, i.e., the quantity $\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_e(i)\|^2_{\boldsymbol{\Sigma}}$ converges as $i\to\infty$ for any positive semi-definite matrix $\boldsymbol{\Sigma}$ that we are free to choose, if the algorithm is mean stable and if the matrix $\boldsymbol{\mathcal{F}}$ given by:

$$\boldsymbol{\mathcal{F}} \triangleq \mathbb{E}\{\boldsymbol{\mathcal{B}}(i)\otimes_b \boldsymbol{\mathcal{B}}(i)\} \tag{26}$$

is stable, where $\otimes_b$ denotes the block Kronecker product operator. In this case, the steady-state performance with metric $\boldsymbol{\Sigma}_{ss}$ defined as $\zeta^\star \triangleq \lim_{i\to\infty}\mathbb{E}\,\|\widetilde{\boldsymbol{w}}_e(i)\|^2_{\boldsymbol{\Sigma}_{ss}}$ can be obtained as follows [17]:

$$\begin{aligned}\zeta^\star = &[\text{bvec}(\boldsymbol{\mathcal{Y}}(\infty))]^\top(\boldsymbol{I}-\boldsymbol{\mathcal{F}})^{-1}\text{bvec}(\boldsymbol{\Sigma}_{ss}) - \\ &2\boldsymbol{h}^\top\boldsymbol{\Sigma}_{ss}(\boldsymbol{w}_e^o-\boldsymbol{w}_e^\star)+\|\boldsymbol{w}_e^o-\boldsymbol{w}_e^\star\|^2_{\boldsymbol{\Sigma}_{ss}}\end{aligned} \tag{27}$$

where bvec$(\cdot)$ is the block vectorization operator and

$$\boldsymbol{\mathcal{Y}}(\infty) \triangleq \mu^2\boldsymbol{\mathcal{G}}^\top + \boldsymbol{q}\boldsymbol{q}' + 2\boldsymbol{q}\boldsymbol{h}^\top\boldsymbol{\mathcal{B}}^\top, \tag{28}$$

$$\boldsymbol{h} \triangleq (\boldsymbol{I}-\boldsymbol{\mathcal{B}})^{-1}\boldsymbol{q}, \tag{29}$$

$$\boldsymbol{q} \triangleq \boldsymbol{\mathcal{A}}^\top(\boldsymbol{I}-\boldsymbol{\mathcal{P}}_e)\boldsymbol{w}_e^o + \boldsymbol{\mathcal{A}}^\top\boldsymbol{f}_e, \tag{30}$$

$$\boldsymbol{\mathcal{G}} \triangleq \boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{P}}_e\text{diag}\{\boldsymbol{c}_k\boldsymbol{c}_k^\top \otimes \sigma^2_{z,k}\boldsymbol{R}_{x,k}\}_{k=1}^N\boldsymbol{\mathcal{P}}_e\boldsymbol{\mathcal{A}}, \tag{31}$$

with $\boldsymbol{f}_e \triangleq \boldsymbol{\mathcal{D}}_e^\dagger\boldsymbol{b}$ and $\boldsymbol{c}_k \triangleq \text{col}\{c_{k_m}\}_{m=1}^{j_k}$. The steady-state network MSD is defined as:

$$\text{MSD}_{\text{net}} \triangleq \frac{1}{N}\sum_{k=1}^N\left(\frac{1}{j_k}\sum_{m=1}^{j_k}\lim_{i\to\infty}\mathbb{E}\|\widetilde{\boldsymbol{w}}_{k_m}(i)\|^2\right) \tag{32}$$

which is obtained by setting $\boldsymbol{\Sigma}_{ss} = \frac{1}{N}\text{diag}\left\{\frac{1}{j_k}\boldsymbol{I}_{j_k\cdot M_k}\right\}_{k=1}^N$.

IV. Simulation Results

In the following, we provide an example to illustrate the behavior of algorithm (12). We considered a network of 14 agents with the topology shown in Fig. 3 (left). The regressors had size $M_k = 2\ \forall k$, were zero-mean Gaussian, independent in time and space and had covariance matrices $\boldsymbol{R}_{x,k} > 0$. The noises $z_k(i)$ were zero-mean i.i.d. Gaussian random variables with variances $\sigma^2_{z,k}$. Figure 3 (right) shows how the signal and noise powers vary across the agents. We randomly sampled $P = 9$ linear equality constraints of the form (1b) where the matrices $\boldsymbol{D}_{p\ell}$ are of dimension 1×2 if $p = \{1,3,5,7,9\}$ and 2×2 otherwise. The entries of the matrices $\boldsymbol{D}_{p\ell}$ and vectors $\boldsymbol{b}_p$ were sampled from the Gaussian distribution $\mathcal{N}(0,1)$. The factors c_{k_m} were set equal to $\frac{1}{j_k}$ and the sets $\mathcal{N}_{k_m}\cap\mathcal{C}_k = \mathcal{C}_k$ for $m = 1,\ldots,j_k$. We ran algorithm (12) with $a_{k_n,k_m} = \frac{1}{j_k}$ for $n = 1,\ldots,j_k$.

We used a constant step-size $\mu = 0.02$ for all agents. The results were averaged over 200 Monte-Carlo runs. Let

$$\boldsymbol{w}^o(\sigma) = \text{col}\{\boldsymbol{w}_k^o(\sigma)\}_{k=1}^N = \boldsymbol{w}_o + \boldsymbol{u}(\sigma), \tag{33}$$

where $\boldsymbol{w}_o$ is a parameter vector satisfying the constraint $\boldsymbol{\mathcal{D}}\boldsymbol{w}+\boldsymbol{b} = \boldsymbol{0}$ and $\boldsymbol{u}(\sigma)$ is a vector whose entries are sampled from the Gaussian distribution $\mathcal{N}(0,\sigma^2)$. To test the tracking ability of the algorithm, we set the parameter vectors $\boldsymbol{w}_k^o$ in (2) equal to $\boldsymbol{w}_k^o(\sigma)$ in (33) and we modified σ every 500 iterations as shown in Fig. 4. We observe that the theoretical model (27) matches well the actual performance of the network.

In order to characterize the influence of the step-size μ on the performance, we report in Fig. 5 (left) the theoretical steady-state MSD (27) for different values of μ when $\sigma = 0.5$. We observe that the steady-state MSD is on the order of μ. Furthermore, we report in Fig. 5 (right) the squared norm of the bias (25) for different values of μ. As expected, this quantity is on the order of μ^2.

V. Conclusion

In this work, we proposed a multitask diffusion LMS algorithm for solving problems that require the simultaneous estimation of multiple parameter vectors that are related locally via linear equality constraints. Our primal technique was based on the stochastic gradient projection algorithm with constant step-size. We showed through simulations that for sufficiently small step-sizes the agents are able to reach the optimal solution with arbitrarily good precision.

References

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

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

[3] 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.

[4] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2718–2723, May 2013.

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

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

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

[8] Z. J. Towfic and A. H. Sayed, "Adaptive penalty-based distributed stochastic convex optimization," *IEEE Trans. Signal Process.,*, vol. 62, no. 15, pp. 3924–3938, Aug. 2014.

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

[10] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, June 2015.

[11] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Proximal multitask learning over networks with sparsity-inducing coregularization," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6329–6344, Dec. 2016.

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

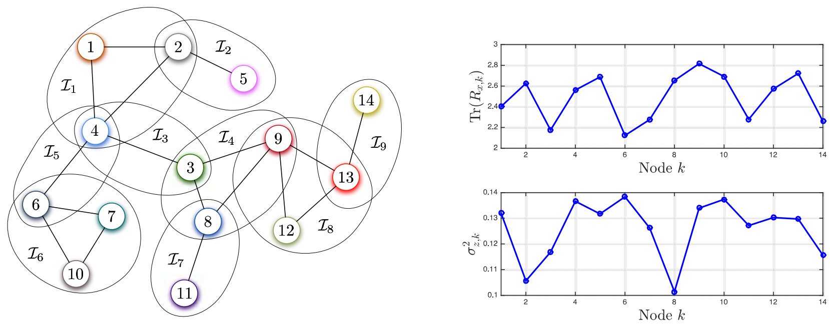

Fig. 3. Experimental setup. (Left) Network topology with constraints identified by the subsets $\mathcal{I}_p$. (Right) Network statistical profile.

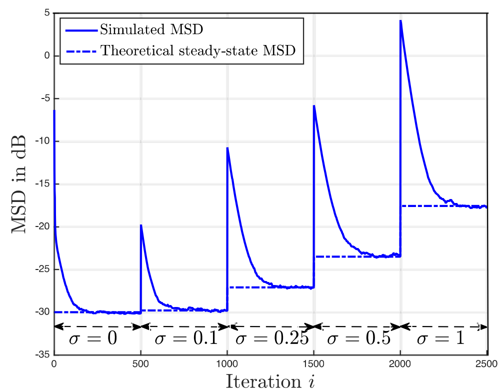

Fig. 4. Network MSD performance of algorithm (12) and tracking ability.

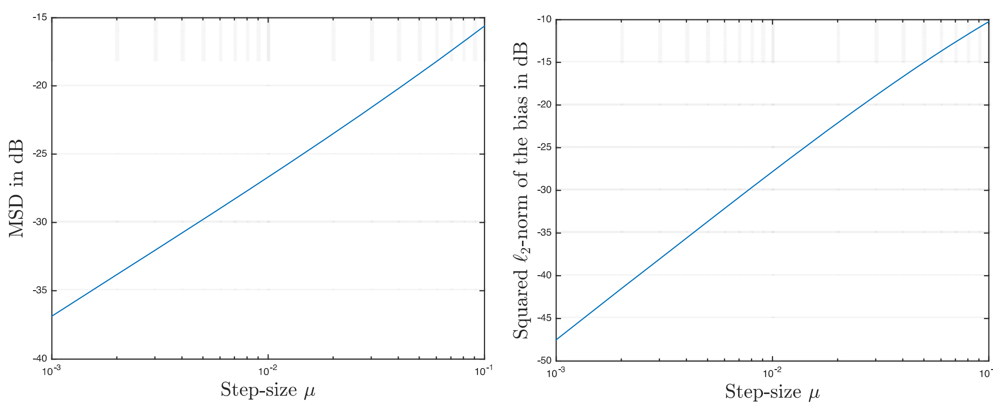

Fig. 5. Influence of the step-size μ. (Left) Network steady-state MSD performance. (Right) Squared ℓ_2-norm of the bias.

[13] 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, 2014.

[14] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1617–1626, May 2013.

[15] J. Plata-Chaves, A. Bertrand, and M. Moonen, "Incremental multiple error filtered-X LMS for node-specific active noise control over wireless acoustic sensor networks," in *IEEE Sensor Array and Multichannel Signal Processing Workshop*, Rio de Janeiro, Brazil, July 2016, pp. 1–5.

[16] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 1001–1016, Feb. 2015.

[17] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Diffusion LMS for multitask problems with local linear equality constraints," *Submitted for publication.* Also available as arXiv:1610.02943, Oct. 2016.