DISTRIBUTED INFERENCE OVER NETWORKS UNDER SUBSPACE CONSTRAINTS

Roula Nassif [(1)], *Stefan Vlaski* [(1,2)], *Ali H. Sayed* [(1)]

[(1)]Ecole Polytechnique Fédérale de Lausanne, Switzerland
[(2)]University of California, Los Angeles, USA

ABSTRACT

This paper considers optimization problems over networks where agents have individual objectives to meet, or individual parameter vectors to estimate, subject to subspace constraints that enforce the objectives across the network to lie in a low-dimensional subspace. This constrained formulation includes consensus optimization as a special case, and allows for more general task relatedness models such as smoothness. While such formulations can be solved via projected gradient descent, the resulting algorithm is not distributed. Motivated by the centralized solution, we propose an iterative and distributed implementation of the projection step, which runs in parallel with the gradient descent update. We establish that, for small step-sizes μ, the proposed distributed adaptive strategy leads to small estimation errors on the order of μ.

***Index Terms*—Distributed optimization, subspace projection, gradient noise.**

I. INTRODUCTION

Distributed inference 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 *consensus* problems, where agents with separate objective functions need to agree on a common parameter vector corresponding to the minimizer of the aggregate sum of the individual costs [1]–[8]. In recent years, there has been interest in learning algorithms that operate over *multitask* networks, where agents need to estimate and track multiple objectives simultaneously [9]–[17]. Although agents may generally have distinct, though related, tasks to perform, they may still be able to capitalize on inductive transfer of information between them to improve estimation accuracy. Therefore, existing strategies to address multitask problems generally exploit prior knowledge on how the tasks across the network relate to each other. For example, one way to model relationships among tasks is to formulate convex optimization problems with appropriate co-regularizers between neighboring agents [10]–[13]. In other applications, it may happen that the parameter vectors to be estimated at neighboring agents are related according to a set of linear equality constraints [9], [14]–[16].

In this work, we consider inference problems over networks where each agent seeks to minimize an individual cost expressed as the expectation of some loss function. The collection of parameter vectors to be estimated across the network is required to lie in a low-dimensional subspace. That is, we consider a connected network of N nodes and let $w_k \in \mathbb{R}^{M_k}$ denote some parameter vector at node k. We also let $\mathcal{W} = \text{col}\{w_1, \ldots, w_N\}$ denote the collection of parameter vectors from across the network. We associate with each agent k a differentiable strongly-convex cost $J_k(w_k) : \mathbb{R}^{M_k} \to \mathbb{R}$, which is expressed as the expectation of some loss function $Q_k(\cdot)$ and written as $J_k(w_k) = \mathbb{E}Q_k(w_k; \boldsymbol{x}_k)$, where $\boldsymbol{x}_k$ denotes the random data. The expectation is computed over the distribution of the data (note that, in our notation, we use boldface letters for random quantities and normal letters for deterministic quantities). Let $M = \sum_{k=1}^{N} M_k$. We consider convex constrained optimization problems of the form:

$$\mathcal{W}^\star = \arg\min_{\mathcal{W}} \quad \sum_{k=1}^{N} J_k(w_k), \qquad \text{subject to} \quad \mathcal{W} \in \mathcal{R}(\mathcal{U}), \tag{1}$$

where $\mathcal{R}(\cdot)$ denotes the range space operator, and $\mathcal{U}$ is an $M \times P$ full-column rank matrix with $P \ll M$. Each agent k is interested in estimating the k-th $M_k \times 1$ subvector $w_k^\star$ of $\mathcal{W}^\star = \text{col}\{w_1^\star, \ldots, w_N^\star\}$. In order to solve (1) iteratively, the gradient projection method can be applied [18]:

$$\mathcal{W}_i = \mathcal{P}_{\mathcal{U}} \left(\mathcal{W}_{i-1} - \mu \,\text{col}\left\{\nabla_{w_k} J_k(w_{k,i-1})\right\}_{k=1}^{N} \right), \quad i \geq 0, \tag{2}$$

where $\mathcal{W}_i = \text{col}\{w_{1,i}, \ldots, w_{N,i}\}$ is the estimate of $\mathcal{W}^\star$ at iteration i, $\mu > 0$ is a small step-size parameter, $\nabla_{w_k} J_k(\cdot)$ is the gradient of $J_k(\cdot)$, and $\mathcal{P}_{\mathcal{U}}$ is the projection matrix onto $\mathcal{R}(\mathcal{U})$.

We are particularly interested in solving the problem in the *stochastic* setting when the distribution of the data $\boldsymbol{x}_k$ is generally unknown. This means that the risks $J_k(\cdot)$ and their gradients $\nabla_{w_k} J_k(\cdot)$ are unknown. As such, approximate gradient vectors need to be employed. A common construction in stochastic approximation theory is to employ the following approximation at iteration i:

$$\widehat{\nabla_{w_k} J_k}(w_k) = \nabla_{w_k} Q_k(w_k; \boldsymbol{x}_{k,i}), \tag{3}$$

where $\boldsymbol{x}_{k,i}$ represents the data observed at iteration i. The difference between the true gradient and its approximation is called *gradient noise*. This noise will seep into the operation of the algorithm and one main challenge is to show that despite its presence, agent k is able to approach $w_k^\star$ asymptotically.

Although the gradient update in (2) can be performed locally at agent k, the projection operation requires a fusion center. To see this, let us introduce an intermediate variable $\psi_{k,i}$ at node k:

$$\psi_{k,i} = w_{k,i-1} - \mu \nabla_{w_k} J_k(w_{k,i-1}). \tag{4}$$

After evaluating $\psi_{k,i}$ locally, each agent at each iteration needs to send its estimate $\psi_{k,i}$ to a fusion center, which performs the projection operation in (2) by computing $\mathcal{W}_i = \mathcal{P}_{\mathcal{U}}\text{col}\{\psi_{1,i}, \ldots, \psi_{N,i}\}$, and then sends the resulting estimates $w_{k,i}$ back to the agents. While centralized solutions can be powerful, decentralized solutions are more attractive since they are more robust and respect the privacy policy at each agent [2]. Thus, a second challenge we face in this paper is how to carry out the projection through a *distributed* network, with no fusion center, using a network where each node performs local computations and exchanges information only with its neighbors.

We propose an adaptive and distributed iterative algorithm allowing each agent k to converge, in the mean-square-error sense, within $O(\mu)$ from the solution $w_k^\star$ of (1), for sufficiently small μ. Conditions on the network topology and signal subspace ensuring the feasibility of a distributed implementation will be provided. We also show how some well-known network optimization problems, such as consensus optimization [1]–[3] and multitask smooth optimization [10], [11], can be recast in the form (1) and addressed with the strategy proposed in this paper.

This work was supported in part by NSF grant CCF-1524250. Emails: {roula.nassif, stefan.vlaski, ali.sayed}@epfl.ch.

II. DISTRIBUTED INFERENCE

We move on to propose and study a distributed solution for solving (1) with a continuous adaptation mechanism. The solution must rely on local computations and communications with immediate neighborhood, and operate in real-time on streaming data. To proceed with the analysis, one of the challenges we face is that the projection in (2) requires non-local exchange of information. Our strategy is to replace the $M \times M$ projection matrix $\mathcal{P}_{\mathcal{U}}$ in (2) by an $M \times M$ matrix $\mathcal{A}$ that satisfies the following conditions:

$$\begin{cases} \lim\limits_{i\to\infty} \mathcal{A}^i = \mathcal{P}_{\mathcal{U}}, & (5) \\ A_{k\ell} = [\mathcal{A}]_{k\ell} = 0, \quad \text{if } \ell \notin \mathcal{N}_k \text{ and } k \neq \ell, & (6) \end{cases}$$

where $[\mathcal{A}]_{k\ell}$ denotes the (k,ℓ)-th block of $\mathcal{A}$ of dimension $M_k \times M_\ell$ and $\mathcal{N}_k$ denotes the neighborhood of agent k, i.e., the set of nodes connected to agent k by an edge. The sparsity condition (6) characterizes the network topology and ensures local exchange of information at each time instant i. By replacing the projector $\mathcal{P}_{\mathcal{U}}$ in (2) by $\mathcal{A}$ and the true gradients $\nabla_{w_k} J_k(\cdot)$ by their stochastic approximations, we obtain the following distributed adaptive solution at each agent k:

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \widehat{\nabla_{w_k} J_k}(\boldsymbol{w}_{k,i-1}), \\ \boldsymbol{w}_{k,i} = \sum\limits_{\ell\in\mathcal{N}_k} A_{k\ell}\boldsymbol{\psi}_{\ell,i}, \end{cases} \tag{7}$$

where we used condition (6), and where $\boldsymbol{\psi}_{k,i}$ is an intermediate estimate and $\boldsymbol{w}_{k,i}$ is the estimate of $w_k^\star$ at agent k and iteration i. As we shall see in Section IV, condition (5) helps ensure convergence toward the optimum. Necessary and sufficient conditions for the matrix equation (5) to hold are given in the following lemma.

Lemma 1. *The matrix equation* (5) *holds, if and only if, the following conditions on the projector $\mathcal{P}_{\mathcal{U}}$ and the matrix $\mathcal{A}$ are satisfied:*

$$\mathcal{A}\mathcal{P}_{\mathcal{U}} = \mathcal{P}_{\mathcal{U}}, \tag{8}$$

$$\mathcal{P}_{\mathcal{U}}\mathcal{A} = \mathcal{P}_{\mathcal{U}}, \tag{9}$$

$$\rho(\mathcal{A} - \mathcal{P}_{\mathcal{U}}) < 1, \tag{10}$$

where $\rho(\cdot)$ denotes the spectral radius of its matrix argument. It follows that any $\mathcal{A}$ satisfying condition (5) *has one as an eigenvalue with multiplicity P, and all other eigenvalues are strictly less than one in magnitude.*

Proof. Proof omitted due to space limitations. The arguments are along the lines developed in [7] for distributed averaging with proper adjustments to handle general subspace constraints. □

Since $\mathcal{U}$ is an $M \times P$ full-column rank matrix, the projector $\mathcal{P}_{\mathcal{U}}$ onto the P-dimensional subspace of $\mathbb{R}^M$ spanned by the columns of $\mathcal{U}$ is given by:

$$\mathcal{P}_{\mathcal{U}} = \mathcal{U}(\mathcal{U}^\top\mathcal{U})^{-1}\mathcal{U}^\top. \tag{11}$$

If we replace $\mathcal{P}_{\mathcal{U}}$ by (11), conditions (8) and (9) reduce to:

$$\mathcal{A}\mathcal{U} = \mathcal{U}, \tag{12}$$

$$\mathcal{U}^\top\mathcal{A} = \mathcal{U}^\top. \tag{13}$$

Condition (12) states that the columns of $\mathcal{U}$ are right eigenvectors of $\mathcal{A}$ associated with the eigenvalue 1. Condition (13) states that the rows of $\mathcal{U}^\top$ are left eigenvectors of $\mathcal{A}$ associated with the eigenvalue 1.

Remark 1–Distributed consensus optimization: Let $M_k = L$ for all agents. If we set in (1) $P = L$ and $\mathcal{U} = \frac{1}{\sqrt{N}}(\mathbb{1}_N \otimes I_L)$, where $\mathbb{1}_N$ is the $N \times 1$ vector of all ones and $\otimes$ denotes the Kronecker product operation, then solving problem (1) will be equivalent to finding at each node k the $L \times 1$ vector $w^\star$ that solves:

$$w^\star = \arg\min_{w} \sum_{k=1}^{N} J_k(w), \tag{14}$$

which corresponds to the minimizer of the aggregate sum of individual costs. Problems of the form (14) have been well studied in the literature of consensus optimization [1]–[6]. Different algorithms for solving (14) have been proposed. Diffusion strategies are particularly attractive due to their enhanced adaptation performance and stability [1]–[3]. When the network is strongly connected, by picking any $N \times N$ doubly-stochastic matrix $A = [a_{k\ell}]$ satisfying:

$$a_{k\ell} \geq 0,\ A\mathbb{1}_N = \mathbb{1}_N,\ \mathbb{1}_N^\top A = \mathbb{1}_N^\top,\ a_{k\ell} = 0 \text{ if } \ell \notin \mathcal{N}_k \text{ and } k \neq \ell \tag{15}$$

the diffusion strategy for solving (14) takes the form [1]–[3]:

$$\begin{cases} \boldsymbol{\psi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \widehat{\nabla_{w_k} J_k}(\boldsymbol{w}_{k,i-1}), \\ \boldsymbol{w}_{k,i} = \sum\limits_{\ell\in\mathcal{N}_k} a_{k\ell}\boldsymbol{\psi}_{\ell,i}. \end{cases} \tag{16}$$

Observe that strategy (16) can be written in the form of (7) with $A_{k\ell} = a_{k\ell} I_L$ and $\mathcal{A} = A \otimes I_L$. It can be verified that, when A satisfies conditions (15) over a strongly connected network, the matrix $\mathcal{A} = A \otimes I_L$ will satisfy conditions (6), (12), (13), and (10). Similarly, with a proper selection of $\mathcal{U}$, multitask inference problems with overlapping parameter vectors [9] can also be recast in the form (1).

Remark 2–Distributed inference under smoothness: Let $M_k = L$ for all agents. In such problems, each agent k in the network has an individual cost $J_k(w_k)$ to minimize subject to a smoothness condition over the graph. The smoothness requirement softens the transition in the tasks $\{w_k\}$ among neighboring nodes and can be measured in terms of a quadratic form of the graph Laplacian [11]:

$$S(\mathcal{W}) = \mathcal{W}^\top \mathcal{L} \mathcal{W} = \frac{1}{2}\sum_{k=1}^{N}\sum_{\ell\in\mathcal{N}_k} c_{k\ell}\|w_k - w_\ell\|^2, \tag{17}$$

where $\mathcal{L} = L_c \otimes I_L$ with $L_c = \text{diag}\{C\mathbb{1}_N\} - C$ denoting the graph Laplacian. The matrix $C = [c_{k\ell}]$ is an $N \times N$ symmetric weighted adjacency matrix with $c_{k\ell} \geq 0$ if $\ell \in \mathcal{N}_k$ and $c_{k\ell} = 0$ otherwise. The smaller $S(\mathcal{W})$ is, the smoother the signal $\mathcal{W}$ on the graph is. Since L_c is symmetric positive semi-definite, it can be decomposed as $L_c = V\Lambda V^\top$ where $\Lambda = \text{diag}\{\lambda_1, \ldots, \lambda_N\}$ with λ_m the non-negative eigenvalues ordered as $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$ and $V = [v_1, \ldots, v_N]$ is the matrix of orthonormal eigenvectors. When the graph is connected, there is only one zero eigenvalue with corresponding eigenvector $v_1 = \frac{1}{\sqrt{N}}\mathbb{1}_N$ [19]. Using the eigenvalue decomposition $\mathcal{L} = (V\Lambda V^\top) \otimes I_L$, $S(\mathcal{W})$ can be written as:

$$S(\mathcal{W}) = \overline{\mathcal{W}}^\top(\Lambda \otimes I_L)\overline{\mathcal{W}} = \sum_{m=1}^{N} \lambda_m \|\overline{w}_m\|^2, \tag{18}$$

where $\overline{\mathcal{W}} = (V^\top \otimes I_L)\mathcal{W}$ and $\overline{w}_m = (v_m^\top \otimes I_L)\mathcal{W}$. Given that $\lambda_m \geq 0$, the above expression shows that $\mathcal{W}$ is considered to be smooth if $\|\overline{w}_m\|^2$ corresponding to large λ_m is negligible. Thus, for a smooth $\mathcal{W}$, $S(\mathcal{W})$ will be equal to $\sum_{m=1}^{p} \lambda_m\|\overline{w}_m\|^2$ with $p \ll N$. By choosing $\mathcal{U} = U \otimes I_L$ where $U = [v_1, \ldots, v_p]$, the smooth signal $\mathcal{W}$ will be in the range space of $\mathcal{U}$ since it can be written as $\mathcal{W} = \mathcal{U}s$ with $s = \text{col}\{\overline{w}_1, \ldots, \overline{w}_p\}$. Therefore, distributed inference problems under smoothness can be recast in the form (1).

III. FINDING A COMBINATION MATRIX $\mathcal{A}$

In some cases, one may find a family of matrices $\mathcal{A}$ satisfying conditions (10), (12), and (13) under the sparsity constraints (6). For example, in consensus optimization where $\mathcal{U} = \frac{1}{\sqrt{N}}(\mathbb{1}_N \otimes I_L)$, by ensuring that the underlying graph is strongly connected and by choosing any doubly stochastic A satisfying the sparsity constraints, the resulting matrix $\mathcal{A} = A \otimes I_L$ will satisfy the required conditions. In general, finding an $\mathcal{A}$ satisfying conditions (5) and (6) is a challenging problem. Not all network topologies satisfying (6) guarantee the existence of an $\mathcal{A}$ satisfying condition (5). For the purpose of this work, we shall assume that the sparsity constraints (6) and the signal subspace lead to a feasible problem.

Assumption 1. *The problem of finding an $\mathcal{A}$ satisfying constraints* (6)*,* (10)*,* (12)*, and* (13) *is assumed to be feasible.*

As a remedy for the violation of Assumption 1, one may increase the connectivity of the network (i.e., add more links) [17].

In the simulations section, we shall find an $\mathcal{A}$ by solving the following constrained spectral norm minimization problem:

$$\begin{aligned} \underset{\mathcal{A}}{\text{minimize}} \quad & \|\mathcal{A} - \mathcal{P}_{\mathcal{U}}\| \\ \text{subject to} \quad & \mathcal{A}\mathcal{U} = \mathcal{U}, \\ & \mathcal{U}^\top \mathcal{A} = \mathcal{U}^\top, \\ & [\mathcal{A}]_{k\ell} = 0, \text{ if } \ell \notin \mathcal{N}_k \text{ and } k \neq \ell. \end{aligned} \tag{19}$$

The convex spectral norm is used instead of the non-convex spectral radius function. Since the spectral radius of a matrix is upper bounded by any of its norms, minimizing $\|\mathcal{A} - \mathcal{P}_{\mathcal{U}}\|$ is equivalent to minimizing an upper bound on $\rho(\mathcal{A} - \mathcal{P}_{\mathcal{U}})$. For symmetric $\mathcal{A}$, we have $\|\mathcal{A} - \mathcal{P}_{\mathcal{U}}\| = \rho(\|\mathcal{A} - \mathcal{P}_{\mathcal{U}}\|)$. Since the objective in (19) is convex and the constraints are linear equalities, the problem is convex. It can be expressed as a semidefinite program (SDP) and solved efficiently [7], [20], [21].

IV. STOCHASTIC PERFORMANCE ANALYSIS

Since the iterates $\boldsymbol{w}_{k,i}$ generated by algorithm (7) are random, we shall measure performance by examining the average squared distance between $\boldsymbol{w}_{k,i}$ and $w_k^\star$, $\limsup_{i\to\infty} \mathbb{E}\|w_k^\star - \boldsymbol{w}_{k,i}\|^2$. We analyze (7) under conditions (6), (10), (12), and (13) on $\mathcal{A}$, and the following assumptions on the risks $\{J_k(\cdot)\}$ and on the gradient noise processes $\{\boldsymbol{s}_{k,i}(\cdot)\}$ defined as:

$$\boldsymbol{s}_{k,i}(w) \triangleq \nabla_{w_k} J_k(w) - \widehat{\nabla_{w_k} J_k}(w). \tag{20}$$

As explained in [1]–[3], these conditions are satisfied by many objective functions of interest in learning and adaptation such as quadratic and logistic risks. Besides, regularization is a common technique to ensure strong convexity.

Assumption 2. *The individual costs $J_k(w_k)$ are assumed to be twice differentiable and strongly convex such that:*

$$0 < \lambda_{k,\min} I_{M_k} \leq \nabla^2_{w_k} J_k(w_k) \leq \lambda_{k,\max} I_{M_k}, \tag{21}$$

where $\lambda_{k,\min} > 0$ for $k = 1, \ldots, N$.

Assumption 3. *The gradient noise process defined in* (20) *satisfies for any $\boldsymbol{w} \in \boldsymbol{\mathcal{F}}_{i-1}$ and for all $k, \ell = 1, \ldots, N$:*

$$\mathbb{E}[\boldsymbol{s}_{k,i}(\boldsymbol{w})|\boldsymbol{\mathcal{F}}_{i-1}] = 0, \tag{22}$$

$$\mathbb{E}[\|\boldsymbol{s}_{k,i}(\boldsymbol{w})\|^2|\boldsymbol{\mathcal{F}}_{i-1}] \leq \beta_k^2\|\boldsymbol{w}\|^2 + \sigma^2_{s,k}, \tag{23}$$

for some $\beta_k^2 \geq 0$, $\sigma^2_{s,k} \geq 0$, and where $\boldsymbol{\mathcal{F}}_{i-1}$ denotes the filtration generated by the random processes $\{\boldsymbol{w}_{\ell,j}\}$ for all $\ell = 1, \ldots, N$ and $j \leq i-1$.

Without loss of generality, we shall introduce the following assumption on the matrix $\mathcal{U}$.

Assumption 4. *The full-column rank matrix $\mathcal{U}$ in* (1) *is assumed to be semi-orthogonal, i.e., its column vectors are orthonormal and $\mathcal{U}^\top\mathcal{U} = I_P$.*

Theorem 1. (Network mean-square-error stability) *Consider a network of N agents satisfying Assumption 1 and running the distributed strategy* (7) *with a matrix $\mathcal{A}$ satisfying conditions* (6)*,* (10)*,* (12)*, and* (13)*. Under Assumptions 2, 3, and 4, the network is mean-square-error stable for sufficiently small step-sizes, namely, it holds that:*

$$\limsup_{i\to\infty} \mathbb{E}\|w_k^\star - \boldsymbol{w}_{k,i}\|^2 = O(\mu), \quad k = 1, \ldots, N, \tag{24}$$

for small enough μ.

Proof. Let $\widetilde{\boldsymbol{w}}_{k,i} = w_k^\star - \boldsymbol{w}_{k,i}$. Using (20) and the mean-value theorem [22, pp. 24], [2, Appendix D], we can express $\widehat{\nabla_{w_k} J_k}(\boldsymbol{w}_{k,i-1})$ as follows:

$$\widehat{\nabla_{w_k} J_k}(\boldsymbol{w}_{k,i-1}) = b_k - \boldsymbol{H}_{k,i-1}\widetilde{\boldsymbol{w}}_{k,i-1} - \boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}), \tag{25}$$

where

$$b_k \triangleq \nabla_{w_k} J_k(w_k^\star), \tag{26}$$

$$\boldsymbol{H}_{k,i-1} \triangleq \int_0^1 \nabla^2_{w_k} J_k(w_k^\star - t\widetilde{\boldsymbol{w}}_{k,i-1})dt. \tag{27}$$

By introducing the following extended vectors and matrices, which collect quantities from across the network:

$$\widetilde{\boldsymbol{\mathcal{W}}}_i \triangleq \mathcal{W}^\star - \text{col}\{\boldsymbol{w}_{1,i}, \ldots, \boldsymbol{w}_{N,i}\}, \tag{28}$$

$$\boldsymbol{\mathcal{H}}_{i-1} \triangleq \text{diag}\{\boldsymbol{H}_{1,i-1}, \ldots, \boldsymbol{H}_{N,i-1}\}, \tag{29}$$

$$\boldsymbol{\mathcal{B}}_{i-1} \triangleq \mathcal{A}(I_M - \mu\boldsymbol{\mathcal{H}}_{i-1}), \tag{30}$$

$$\boldsymbol{s}_i \triangleq \text{col}\{\boldsymbol{s}_{1,i}(\boldsymbol{w}_{1,i-1}), \ldots, \boldsymbol{s}_{N,i}(\boldsymbol{w}_{N,i-1})\}, \tag{31}$$

$$b \triangleq \text{col}\{b_1, \ldots, b_N\}, \tag{32}$$

we can show that the network weight error vector $\widetilde{\boldsymbol{\mathcal{W}}}_i$ in (28) evolves according to the following dynamics:

$$\widetilde{\boldsymbol{\mathcal{W}}}_i = \boldsymbol{\mathcal{B}}_{i-1}\widetilde{\boldsymbol{\mathcal{W}}}_{i-1} - \mu\mathcal{A}\boldsymbol{s}_i + \mu\mathcal{A}b, \tag{33}$$

where we used (25) and the fact that $\mathcal{W}^\star$ is the solution of the constrained problem (1), and thus:

$$\mathcal{A}\mathcal{W}^\star = \mathcal{A}\mathcal{P}_{\mathcal{U}}\mathcal{W}^\star \overset{(8)}{=} \mathcal{P}_{\mathcal{U}}\mathcal{W}^\star = \mathcal{W}^\star. \tag{34}$$

Under conditions (10), (12), (13), and Assumption 4, the matrix $\mathcal{A}$ admits a Jordan decomposition of the form $\mathcal{A} = \mathcal{V}_\epsilon\Lambda_\epsilon\mathcal{V}_\epsilon^{-1}$ with:

$$\mathcal{V}_\epsilon = \left[\begin{array}{c|c} \mathcal{U} & \mathcal{V}_{R,\epsilon} \end{array}\right], \; \Lambda_\epsilon = \left[\begin{array}{c|c} I_P & 0 \\ \hline 0 & \mathcal{J}_\epsilon \end{array}\right], \; \mathcal{V}_\epsilon^{-1} = \left[\begin{array}{c} \mathcal{U}^\top \\ \hline \mathcal{V}_{L,\epsilon}^\top \end{array}\right], \tag{35}$$

where $\mathcal{J}_\epsilon$ is a Jordan matrix with the eigenvalues λ (which may be complex but have magnitude less than one) on the diagonal and $\epsilon > 0$ on the first lower sub-diagonal. Multiplying both sides of (33) from the left by $\mathcal{V}_\epsilon^{-1}$ and introducing the transformed iterate $\overline{\boldsymbol{\mathcal{W}}}_i = \mathcal{V}_\epsilon^{-1}\widetilde{\boldsymbol{\mathcal{W}}}_i$, we obtain:

$$\overline{\boldsymbol{\mathcal{W}}}_i = \mathcal{V}_\epsilon^{-1}\boldsymbol{\mathcal{B}}_{i-1}\mathcal{V}_\epsilon\overline{\boldsymbol{\mathcal{W}}}_{i-1} - \mu\mathcal{V}_\epsilon^{-1}\mathcal{A}\boldsymbol{s}_i + \mu\mathcal{V}_\epsilon^{-1}\mathcal{A}b. \tag{36}$$

We now partition $\overline{\boldsymbol{\mathcal{W}}}_i$ into $\overline{\boldsymbol{\mathcal{W}}}_i = \text{col}\{\overline{\boldsymbol{\mathcal{W}}}_{c,i}, \overline{\boldsymbol{\mathcal{W}}}_{r,i}\}$ where $\overline{\boldsymbol{\mathcal{W}}}_{c,i} = \mathcal{U}^\top\widetilde{\boldsymbol{\mathcal{W}}}_i$ is a $P \times 1$ vector and $\overline{\boldsymbol{\mathcal{W}}}_{r,i} = \mathcal{V}_{L,\epsilon}^\top\widetilde{\boldsymbol{\mathcal{W}}}_i$ is an $(M-P) \times 1$ vector. Then, recursion (36) can be decomposed as:

$$\overline{\boldsymbol{\mathcal{W}}}_{c,i} = (I_P - \boldsymbol{\mathcal{D}}_{11,i-1})\overline{\boldsymbol{\mathcal{W}}}_{c,i-1} - \boldsymbol{\mathcal{D}}_{12,i-1}\overline{\boldsymbol{\mathcal{W}}}_{r,i-1} - \boldsymbol{s}_{c,i}, \tag{37}$$

$$\overline{\boldsymbol{\mathcal{W}}}_{r,i} = (\mathcal{J}_\epsilon - \boldsymbol{\mathcal{D}}_{22,i-1})\overline{\boldsymbol{\mathcal{W}}}_{r,i-1} - \boldsymbol{\mathcal{D}}_{21,i-1}\overline{\boldsymbol{\mathcal{W}}}_{c,i-1} - \boldsymbol{s}_{r,i} + b_r, \tag{38}$$

where $\boldsymbol{s}_{c,i} \triangleq \mu\mathcal{U}^\top\mathcal{A}\boldsymbol{s}_i$, $\boldsymbol{s}_{r,i} \triangleq \mu\mathcal{V}_{L,\epsilon}^\top\mathcal{A}\boldsymbol{s}_i$, $b_r \triangleq \mu\mathcal{V}_{L,\epsilon}^\top\mathcal{A}b$, and:

$$\begin{aligned} \boldsymbol{\mathcal{D}}_{11,i-1} &\triangleq \mu\mathcal{U}^\top\boldsymbol{\mathcal{H}}_{i-1}\mathcal{U}, & \boldsymbol{\mathcal{D}}_{12,i-1} &\triangleq \mu\mathcal{U}^\top\boldsymbol{\mathcal{H}}_{i-1}\mathcal{V}_{R,\epsilon} \\ \boldsymbol{\mathcal{D}}_{21,i-1} &\triangleq \mu\mathcal{J}_\epsilon\mathcal{V}_{L,\epsilon}^\top\boldsymbol{\mathcal{H}}_{i-1}\mathcal{U}, & \boldsymbol{\mathcal{D}}_{22,i-1} &\triangleq \mu\mathcal{J}_\epsilon\mathcal{V}_{L,\epsilon}^\top\boldsymbol{\mathcal{H}}_{i-1}\mathcal{V}_{R,\epsilon}, \end{aligned}$$

and where we used the fact that $\mathcal{U}^\top\mathcal{A}b \overset{(13)}{=} \mathcal{U}^\top b = 0$ since the constrained problem (1) can be written alternatively as:

$$\begin{aligned} \underset{\mathcal{W}}{\text{minimize}} \quad & \sum_{k=1}^N J_k(w_k) \\ \text{subject to} \quad & (I_M - \mathcal{P}_{\mathcal{U}})\mathcal{W} = 0, \end{aligned} \tag{39}$$

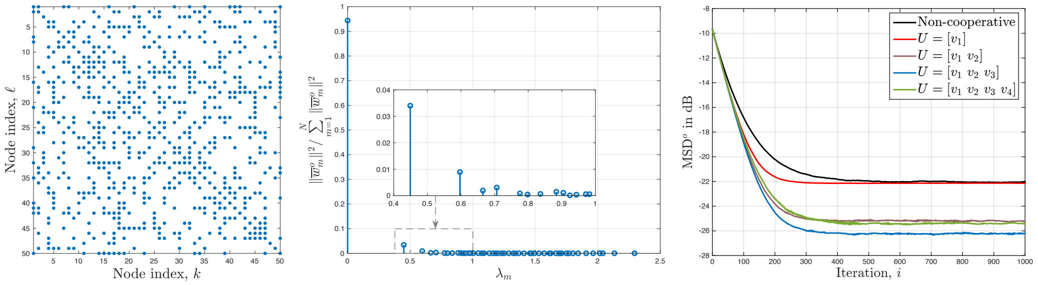

Fig. 1. Inference under smoothness. *(Left)* Link matrix. *(Middle)* Graph spectral content of $\mathcal{W}^o$ with $\overline{w}_m^o = (v_m^\top \otimes I_L)\mathcal{W}^o$. *(Right)* Performance of algorithm (7) w.r.t. $\mathcal{W}^o$ for 4 different choices of the matrix $\mathcal{U}$ in (1) with $\mathcal{U} = U \otimes I_L$, and non-cooperative strategy.

with the Lagrangian given by:

$$\mathcal{L}(\mathcal{W};\gamma) = \sum_{k=1}^N J_k(w_k) + \gamma^\top (I_M - \mathcal{P}_{\mathcal{U}})\mathcal{W}, \tag{40}$$

where γ is the $M \times 1$ vector of Lagrange multipliers. From the optimality conditions, we obtain the following condition on $\mathcal{W}^\star$:

$$b + (I_M - \mathcal{P}_{\mathcal{U}})\lambda = 0. \tag{41}$$

where b is given by (32). By multiplying both sides of the previous equation by $\mathcal{U}^\top$ and using (11), we obtain $\mathcal{U}^\top b = 0$.

Now, using similar arguments as in [2, Theorem 9.1], we can show that, under Assumptions 2, 3, and 4, the variances of $\overline{\boldsymbol{\mathcal{W}}}_{c,i}$ in (37) and $\overline{\boldsymbol{\mathcal{W}}}_{r,i}$ in (38) are coupled and recursively bounded as:

$$\left[\begin{array}{c} \mathbb{E}\|\overline{\boldsymbol{\mathcal{W}}}_{c,i}\|^2 \\ \mathbb{E}\|\overline{\boldsymbol{\mathcal{W}}}_{r,i}\|^2 \end{array}\right] \preceq \Gamma \left[\begin{array}{c} \mathbb{E}\|\overline{\boldsymbol{\mathcal{W}}}_{c,i-1}\|^2 \\ \mathbb{E}\|\overline{\boldsymbol{\mathcal{W}}}_{r,i-1}\|^2 \end{array}\right] + O(\mu^2), \tag{42}$$

where

$$\Gamma = \left[\begin{array}{cc} 1 - O(\mu) & O(\mu) \\ O(\mu^2) & \|\mathcal{J}_\epsilon\| + O(\mu^2) \end{array}\right], \tag{43}$$

with $\|\mathcal{J}_\epsilon\| < 1$. It then follows that, for sufficiently small μ, we have:

$$\limsup_{i\to\infty} \left[\begin{array}{c} \mathbb{E}\|\overline{\boldsymbol{\mathcal{W}}}_{c,i}\|^2 \\ \mathbb{E}\|\overline{\boldsymbol{\mathcal{W}}}_{r,i}\|^2 \end{array}\right] \preceq \left[\begin{array}{c} O(\mu) \\ O(\mu^2) \end{array}\right], \tag{44}$$

from which we can conclude that:

$$\limsup_{i\to\infty} \mathbb{E}\|\widetilde{\boldsymbol{\mathcal{W}}}_i\|^2 = O(\mu). \tag{45}$$

The proof of (42)–(45) is omitted due to space limitations. □

V. SIMULATION RESULTS

We apply strategy (7) to solve distributed inference under smoothness (described in Remark 2 of Section II). We consider a connected mean-square-error (MSE) network of $N = 50$ nodes and $M_k = L = 5$, generated randomly with the link matrix shown in Fig. 1 (left). Each agent is subjected to streaming data $\{\boldsymbol{d}_k(i), \boldsymbol{u}_{k,i}\}$ assumed to satisfy a linear regression model [2]:

$$\boldsymbol{d}_k(i) = \boldsymbol{u}_{k,i}^\top w_k^o + \boldsymbol{v}_k(i), \quad k = 1,\dots,N, \tag{46}$$

for some unknown $L \times 1$ vector w_k^o to be estimated with $\boldsymbol{v}_k(i)$ denoting a zero-mean measurement noise. For these networks, the risk functions take the form of mean-square-error costs:

$$J_k(w_k) = \frac{1}{2}\mathbb{E}|\boldsymbol{d}_k(i) - \boldsymbol{u}_{k,i}^\top w_k|^2, \quad k = 1,\dots,N. \tag{47}$$

The processes $\{\boldsymbol{u}_{k,i}, \boldsymbol{v}_k(i)\}$ are zero-mean jointly wide-sense stationary with: i) $\mathbb{E}\boldsymbol{u}_{k,i}\boldsymbol{u}_{\ell,i}^\top = R_{u,k} = \sigma_{u,k}^2 I_L$ if $k = \ell$ and zero otherwise; ii) $\mathbb{E}\boldsymbol{v}_k(i)\boldsymbol{v}_\ell(i) = \sigma_{v,k}^2$ if $k = \ell$ and zero otherwise; and iii) $\boldsymbol{u}_{k,i}$ and $\boldsymbol{v}_k(i)$ are independent of each other. The variances $\sigma_{u,k}^2$ and $\sigma_{v,k}^2$ are generated from the uniform distributions unif$(0.5, 2)$ and unif$(0.1, 0.4)$, respectively. Let $\mathcal{W}^o = \text{col}\{w_1^o, \dots, w_N^o\}$. The signal $\mathcal{W}^o$ is generated by smoothing a signal $\mathcal{W}_o$ by a diffusion kernel. Particularly, we generate $\mathcal{W}^o$ according to $\mathcal{W}^o = [(Ve^{-\tau\Lambda}V^\top) \otimes I_L]\mathcal{W}_o$ with $\tau = 4$, $\mathcal{W}_o$ a randomly generated vector from the Gaussian distribution $\mathcal{N}(0.1 \times \mathbb{1}_{NL}, I_{NL})$, and $\{V = [v_1, \dots, v_N], \Lambda = \text{diag}\{\lambda_1, \dots, \lambda_N\}\}$ are the matrices of eigenvectors and eigenvalues of $L_c = \text{diag}\{C\mathbb{1}_N\} - C$. The adjacency matrix C is chosen such that the (k,ℓ)-th entry $[C]_{k\ell} = c_{k\ell} = 0.1$ if $\ell \in \mathcal{N}_k$ and 0 otherwise. Figure 1 (middle) illustrates the normalized squared ℓ_2-norm of the spectral component $\overline{w}_m = (v_m^\top \otimes I_L)\mathcal{W}^o$. It can be observed that the signal is mainly localized in $[0, 0.6]$.

Table I. Performance of strategy (7) w.r.t. $\mathcal{W}^\star$ in (1) for 4 different choices of μ.

μ	10^{-5}	10^{-4}	10^{-3}	10^{-2}
MSD*	-64.12dB	-54.05dB	-42.56dB	-28.62dB

We run algorithm (7) for 4 different choices of matrix $\mathcal{U}$ in (1) with $\mathcal{U} = U \otimes I_L$: i) matrix U chosen as the first eigenvector of the Laplacian $U = [v_1] = \frac{1}{\sqrt{N}}\mathbb{1}_N$; ii) matrix U chosen as the first two eigenvectors of the Laplacian $U = [v_1\ v_2]$; iii) matrix U chosen as $U = [v_1\ v_2\ v_3]$; iv) matrix U chosen as $U = [v_1\ v_2\ v_3\ v_4]$. In each case, the combination matrix $\mathcal{A}$ is set as the solution of the optimization problem (19). We set $\mu = 0.01$. We report the network MSD^o learning curves $\frac{1}{N}\mathbb{E}\|\mathcal{W}^o - \boldsymbol{\mathcal{W}}_i\|^2$ in Fig. 1 (right). The results are averaged over 200 Monte-Carlo runs. The learning curve of the non-cooperative solution, obtained from (7) by setting $\mathcal{A} = I_{LN}$, is also reported. The results show that the best performance is obtained when $\mathcal{U} = [v_1\ v_2\ v_3] \otimes I_L$. This is due to the fact that the columns of $\mathcal{U}$ constitute a basis spanning the useful signal subspace (see Fig. 1 (middle)). As a consequence, a strong noise reduction may be obtained by projecting onto this subspace compared with the non-cooperative strategy where each agent estimates w_k^o without any cooperation. By forcing consensus (i.e., by choosing $U = [v_1]$), the resulting estimate $\boldsymbol{w}_{k,i}$ will be biased with respect to w_k^o, which is not common across agents. The performance obtained when $U = [v_1\ v_2\ v_3\ v_4]$ is worse than the case where $U = [v_1\ v_2\ v_3]$ due to a smaller noise reduction.

Finally, we illustrate Theorem 1 in Table I by reporting the steady-state MSD$^\star = \limsup_{i\to\infty} \frac{1}{N}\mathbb{E}\|\mathcal{W}^\star - \boldsymbol{\mathcal{W}}_i\|^2$ when $\mathcal{U} = [v_1\ v_2\ v_3] \otimes I_L$ for 4 different values of the step-size $\mu = \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$. A closed form solution for $\mathcal{W}^\star$ in (1) exists and is given by:

$$\mathcal{W}^\star = \mathcal{U}(\mathcal{U}^\top \mathcal{H}\mathcal{U})^{-1}\mathcal{U}^\top \mathcal{H}\mathcal{W}^o, \tag{48}$$

where $\mathcal{H} = \text{diag}\{R_{u,k}\}_{k=1}^N$. We observe that, in the small adaptation regime, i.e., when $\mu \to 0$, the network MSD* increases approximately 10dB per decade (when μ goes from μ_1 to $10\mu_1$). This means that the steady-state MSD* is on the order of μ.

VI. REFERENCES

[1] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, “Diffusion strategies for adaptation and learning over networks,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[2] A. H. Sayed, “Adaptation, learning, and optimization over networks,” *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[3] A. H. Sayed, “Adaptive networks,” *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[4] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

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

[6] S. Chouvardas, K. Slavakis, and S. Theodoridis, “Adaptive robust distributed learning in diffusion sensor networks,” *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4692–4707, Oct. 2011.

[7] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[8] P. Braca, S. Marano, and V. Matta, “Enforcing consensus while monitoring the environment in wireless sensor networks,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3375–3380, Jul. 2008.

[9] J. Plata-Chaves, A. Bertrand, M. Moonen, S. Theodoridis, and A. M. Zoubir, “Heterogeneous and multitask wireless sensor networks – Algorithms, applications, and challenges,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 450–465, 2017.

[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, 2014.

[11] R. Nassif, S. Vlaski, and A. H. Sayed, “Distributed inference over multitask graphs under smoothness,” in *Proc. IEEE International Workshop on Signal Processing Advances in Wireless Communications*, Kalamata, Greece, Jun. 2018, pp. 1–5.

[12] A. Koppel, B. M. Sadler, and A. Ribeiro, “Proximity without consensus in online multi-agent optimization,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Shanghai, China, May 2016, pp. 3726–3730.

[13] D. Hallac, J. Leskovec, and S. Boyd, “Network LASSO: Clustering and optimization in large graphs,” in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, Australia, Aug. 2015, pp. 387–396.

[14] J. Chen, C. Richard, A. O. Hero, and A. H. Sayed, “Diffusion LMS for multitask problems with overlapping hypothesis subspaces,” in *Proc. IEEE International Workshop on Machine Learning for Signal Processing*, Reims, France, Sept. 2014, pp. 1–6.

[15] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, “Distributed optimization with local domains: Applications in MPC and network flows,” *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 2004–2009, Jul. 2015.

[16] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, “Diffusion LMS for multitask problems with local linear equality constraints,” *IEEE Transactions on Signal Processing*, vol. 65, no. 19, pp. 4979–4993, 2017.

[17] S. Barbarossa, G. Scutari, and T. Battisti, “Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, Apr. 2009, pp. 2893–2896.

[18] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.

[19] F. R. K Chung, *Spectral Graph Theory*, American Mathematical Society, 1997.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.

[21] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” http://cvxr.com/cvx, Mar. 2014.

[22] B. T. Polyak, *Introduction to Optimization*, Optimization Software, New York, 1987.