TRADING OFF COMMUNICATIONS BANDWIDTH WITH ACCURACY IN ADAPTIVE DIFFUSION NETWORKS

Symeon Chouvardas[1] *Konstantinos Slavakis*[2] *Sergios Theodoridis*[1]

[1]University of Athens,
Dept. of Informatics and Telecommunications,
Athens 15784, Greece.
Emails: schouv@di.uoa.gr, stheodor@di.uoa.gr

[2]University of Peloponnese,
Dept. of Telecommunications Science and Technology,
Tripolis 22100, Greece.
Email: slavakis@uop.gr

ABSTRACT

In this paper, a novel algorithm for bandwidth reduction in adaptive distributed learning is introduced. We deal with diffusion networks, in which the nodes cooperate with each other, by exchanging information, in order to estimate an unknown parameter vector of interest. We seek for solutions in the framework of set theoretic estimation. Moreover, in order to reduce the required bandwidth by the transmitted information, which is dictated by the dimension of the unknown vector, we choose to project and work in a lower dimension Krylov subspace. This provides the benefit of trading off dimensionality with accuracy. Full convergence properties are presented, and experiments, within the system identification task, demonstrate the robustness of the algorithmic technique.

Index Terms— Adaptive distributed learning, Krylov subspaces, projections.

1. INTRODUCTION

Wireless sensor networks (WSNs) have attracted considerable interest over the recent years, due to the plethora of applications in which they contribute. Typical examples of these are: acoustic source localization, life sciences, e.t.c. A typical WSN consists of a number of nodes, which sense an amount of data from the environment, and perform the essential computations, in order to estimate an unknown vector of interest. This paper deals with the case where all the nodes take part in the computations, which is known as the decentralized mode of operation. In such a scenario, nodes do not act as individual learners, but cooperate with each other. Such a cooperation is known that results in an enhanced performance, [1]. Two types of decentralized solutions have been proposed. The incremental, in which each node communicates with only one node, called neighbour, and henceforth the network has a cyclic topology, e.g., [2], and the diffusion, where a node, say k, is able to communicate with a number of nodes, that constitute the neighbourhood of k, e.g., [1, 3].

In this paper, we consider a diffusion network in which the nodes are scheduled to estimate, adaptively, an unknown, yet common to all the nodes, parameter vector, which is assumed to live in the m-dimensional Euclidean space $\mathbb{R}^m$. The problem is attacked within the set theoretic framework; instead of seeking for a single solution, we seek for a set of possible solutions. This set is formed by the intersection of a sequence of closed convex sets. Each one of these convex sets defines a region in $\mathbb{R}^m$, which consists of all the points that are in agreement with a measurement point in the training data set. The term in agreement means that it results in an error that obeys a bounding condition. Such an approach is in line with robust statistics loss functions, which were recently popularized in the context of Support Vector Regression. For the specific error bounding condition adopted in this paper, the aforementioned closed convex sets take the form of hyperslabs.

In addition, since cooperation implies the exchange between nodes, at every time instant, of the m coefficients of the obtained estimates, the required communications bandwidth is directly related to the dimensionality of $\mathbb{R}^m$. In order to reduce the bandwidth budget, we choose to project and work in a subspace $\mathbb{R}^D,\ D \leq m$, of lower dimension. In order to "control" the optimality of the projection, the $\mathbb{R}^D$ subspace is selected to be the respective Krylov one, due to its strong connection with the optimal Wiener solution [4, 5]. It turns out that the basic recursion of the developed algorithm consists of projections of points, lying in the Krylov subspace, onto the intersection of this subspace with hyperslabs defined in $\mathbb{R}^m$. An analytic formula will be presented, as well as the theoretic analysis of the algorithm, which enjoys a number of nice convergence properties. Finally, experiments verify the robustness of the algorithm even in cases when the subspace is of significantly lower dimension than the original unknown vector.

2. NETWORK AND PROBLEM FORMULATION

The set of real numbers and the set of non-negative integers will be denoted by $\mathbb{R}$ and $\mathbb{N}$ respectively. Moreover, vectors will be denoted by boldface letters, matrices by uppercase letters, and $(\cdot)^T$ will stand for the transpose of the respective vector or matrix. Finally, $\|\cdot\|$ will stand for the Euclidean norm and $E\{\cdot\}$ for the expectation operator.

Our general goal is to estimate a parameter vector of interest $\boldsymbol{w}^* \in \mathbb{R}^m$, through measurements collected at the N nodes of the diffusion network. We assume that each node, k, at time instance n, has access to the measurements $(d_k(n), \boldsymbol{u}_{k,n}) \in \mathbb{R} \times \mathbb{R}^m$, which are related according to the linear system

$$d_k(n) = \boldsymbol{u}_{k,n}^T \boldsymbol{w}^* + v_k(n), \tag{1}$$

where $v_k(n)$ is the noise process with standard deviation equal to σ_k. The general concept of a diffusion network can be summarized as follows. Each sensor collects information from its environment, i.e., the measurement pair $(d_k(n), \boldsymbol{u}_{k,n})$, in order to proceed to the adaptation step, and it also exploits the estimates transmitted by its neighbouring nodes. From now on, $\mathcal{N}_k$ will stand for the neighbourhood of node k, i.e., the nodes with which communication is possible. Such a scenario can be seen as a fusion of the estimates collected by the nodes of the neighbourhood, $\boldsymbol{w}_l(n),\ l \in \mathcal{N}_k$. For node k, at time instance n, the most common example of a combination strategy is:

This research has been co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

$\boldsymbol{\phi}_k(n) = \sum_{l\in\mathcal{N}_k} c_{k,l}\boldsymbol{w}_l(n)$, where $c_{k,l} = 0$ if $l \notin \mathcal{N}_k$, $c_{k,l} > 0$ if $l \in \mathcal{N}_k$ and $\sum_{l\in\mathcal{N}_k} c_{k,l} = 1$. It has been verified ([3]), that for a properly chosen adaptation algorithm, the combination strategy can lead to asymptotic *consensus*, which implies that the nodes will reach the same estimate, and that the performance of the respective adaptive filters is better, compared to the case where the nodes work individually, e.g., [6]. Depending on the way with which this fused information takes part in the computations, we can define the following combination strategies: combine-adapt, in which the information collected by the neighbourhood is fused under a certain protocol and then is put into the adaptation step, e.g., [1, 3]. Adapt-combine, where the adaptation step precedes the combination part, e.g., [6], and consensus based, where the computations are made in parallel and there is no clear distinction between the combine and the adapt steps, e.g., [7].

3. THE ALGORITHMIC FRAMEWORK

A set, $\mathcal{C} \subset \mathbb{R}^m$, will be called convex if $\forall \boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathcal{C}$ and $\forall \alpha \in [0,1]$, $\alpha\boldsymbol{b}_1 + (1-\alpha)\boldsymbol{b}_2 \in \mathcal{C}$. This implies that every line segment with endpoints $\boldsymbol{b}_1, \boldsymbol{b}_2$ will lie in $\mathcal{C}$. Moreover, the projection mapping, $P_{\mathcal{C}}$ onto $\mathcal{C}$, is the mapping which takes a $\boldsymbol{w}$ to the uniquely existing point, $P_{\mathcal{C}}(\boldsymbol{w}) \in \mathcal{C}$, such that $\|\boldsymbol{w} - P_{\mathcal{C}}(\boldsymbol{w})\| = \inf\{\boldsymbol{v} \in \mathcal{C} : \|\boldsymbol{w} - \boldsymbol{v}\|\}$.

The algorithm, to be described, belongs to the family of the Adaptive Projected Subgradient Method (APSM) [8]. The general notion is to find points that are in *agreement* with the measurements. To be more specific, every point $\boldsymbol{w}$ that satisfies the bounded condition [1]

$$S_n := \{\boldsymbol{w} \in \mathbb{R}^m : |d(n) - \boldsymbol{u}_n^T\boldsymbol{w}| \le \epsilon\}, \tag{2}$$

will be in agreement with the current measurements set. All the points that are defined by (2) lie in a hyperslab in $\mathbb{R}^m$. The user-defined parameter ϵ determines the hyperslab's width, and it is chosen so as to account for the noise, e.g., [3]. Our initial task, now, becomes to seek for points lying in the intersection of these hyperslabs, which "arrive" sequentially. This can be achieved by a sequence of projections onto them, and the occurring algorithmic scheme is

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu(n)\left(\sum_{j=n-q+1}^{n}\omega_j P_{S_j}(\boldsymbol{w}(n)) - \boldsymbol{w}(n)\right), \tag{3}$$

where q determines the number of hyperslabs considered at time n, and controls the convergence speed [9], $\mu(n)$ is the step-size that guarantees convergence, $\sum_{j=n-q+1}^{n}\omega_j = 1$ and $P_{S_n}(\cdot)$ stands for the projection operator onto S_n, given by: $P_{S_n}(\boldsymbol{w}) = \boldsymbol{w} + \beta_n\boldsymbol{u}_n$, $\forall \boldsymbol{w} \in \mathbb{R}^m$ with

$$\beta_n = \begin{cases} \dfrac{d(n) - \boldsymbol{u}_n^T\boldsymbol{w} + \epsilon}{\|\boldsymbol{u}_n\|^2}, & d(n) - \boldsymbol{u}_n^T\boldsymbol{w} < -\epsilon, \\ 0, & |d(n) - \boldsymbol{u}_n^T\boldsymbol{w}| \le \epsilon, \\ \dfrac{d(n) - \boldsymbol{u}_n^T\boldsymbol{w} - \epsilon}{\|\boldsymbol{u}_n\|^2}, & d(n) - \boldsymbol{u}_n^T\boldsymbol{w} > \epsilon. \end{cases}$$

4. REDUCED RANK DIFFUSION ALGORITHM

A modified version of (3) with application to diffusion networks was presented in [3]. The steps of the algorithm, in each node, are the following [2]

$$\boldsymbol{\phi}_k(n) = \sum_{l\in\mathcal{N}_k} c_{k,l}\boldsymbol{w}_l(n), \tag{4}$$

$$\boldsymbol{w}_k(n+1) = \boldsymbol{\phi}_k(n) + \mu_k(n)\times \left(\sum_{j=n-q+1}^{n}\omega_{k,j}P_{S_{k,j}}(\boldsymbol{\phi}_k(n)) - \boldsymbol{\phi}_k(n)\right), \tag{5}$$

where $S_{k,j}$ and $\omega_{k,j}$ are defined in a similar way as in (2). It can be readily seen that (4) is the combination step, whereas (5) is the adaptation one. Hence, the algorithm belongs to the family of the combine adapt algorithms.

From (4) it is not difficult to see that every node, at every time instance, transmits its estimate to the neighbouring nodes, which amounts to m coefficients to be transmitted. In order to reduce this number, a possible strategy is to restrict the initial solution space ($\mathbb{R}^m$) to a subspace of lower dimension, say D, where $D < m$. In this paper, we will consider Krylov subspaces for dimensionality reduction (see also [4, 5]). For a given matrix $\boldsymbol{A}$ $(m \times m)$ and a vector $\boldsymbol{c}$ $(m \times 1)$, the definition of the D-dimensional Krylov subspace is $K_D(\boldsymbol{A}, \boldsymbol{c}) = \text{span}\{\boldsymbol{c}, \boldsymbol{A}\boldsymbol{c}, \ldots, \boldsymbol{A}^{D-1}\boldsymbol{c}\}$.

Let us define $\boldsymbol{R} = E\{\boldsymbol{u}_n\boldsymbol{u}_n^T\}$ and $\boldsymbol{p} = E\{d(n)\boldsymbol{u}_n\}$, where $d(n), \boldsymbol{u}_n$ are related according to (1); the celebrated Wiener-Hopf equation [10] states that $\boldsymbol{w}^* = \boldsymbol{R}^{-1}\boldsymbol{p}$. It has been proved, e.g., [5], that the reduced rank Wiener filter, of dimension D, belongs to $K_D(\boldsymbol{R}, \boldsymbol{p})$. In other words, it is a reasonable strategy to seek for a possible solution in this subspace. However, in distributed networks, despite the fact that every node seeks for the same unknown vector, the statistics in each node may be different. This implies that a different viewpoint has to be followed. Let us define the mean square error loss function $\mathcal{L} : \mathbb{R}^m \to [0, +\infty)$, for the whole network

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}) &= \frac{1}{N}\sum_{k=1}^{N} E\left\{(d_k(n) - \boldsymbol{u}_{k,n}^T\boldsymbol{w})^2\right\} \\ &= \frac{1}{N}\sum_{k=1}^{N}(\boldsymbol{w}^T\boldsymbol{R}_k\boldsymbol{w} - 2\boldsymbol{w}\boldsymbol{p}_k + \sigma_{d_k}^2) \\ &= \boldsymbol{w}^T\boldsymbol{R}'\boldsymbol{w} - 2\boldsymbol{w}\boldsymbol{p}' + \frac{1}{N}\sum_{k=1}^{N}\sigma_{d_k}^2, \end{aligned} \tag{6}$$

where $\sigma_{d_k} = E\{d_k^2(n)\}$, $\boldsymbol{R}' = \frac{1}{N}\sum_{k=1}^{N}E\{\boldsymbol{u}_{k,n}\boldsymbol{u}_{k,n}^T\} = \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{R}_k$ and $\boldsymbol{p}' = \frac{1}{N}\sum_{k=1}^{N}E\{d_k(n)\boldsymbol{u}_{k,n}\} = \frac{1}{N}\sum_{k=1}^{N}\boldsymbol{p}_k$. It can be seen, that the solution minimizing (6) is given by $\boldsymbol{w}^* = \boldsymbol{R}'^{-1}\boldsymbol{p}'$. This argument indicates that it may be reasonable to select $\boldsymbol{R}'$ and $\boldsymbol{p}'$ (i.e., the average values) in order to construct the respective Krylov subspace. The question, now, is how to construct $\boldsymbol{R}', \boldsymbol{p}'$, since we assume that there is no a-priori knowledge of $\boldsymbol{R}_k$, $\boldsymbol{p}_k$. A possible strategy, followed also in [4], is to resort to approximations of the unknown quantities, in which the measurements, $d_k(n), \boldsymbol{u}_{k,n}$, are exploited. To be more specific, $\hat{\boldsymbol{R}}_{k,n} = \gamma\hat{\boldsymbol{R}}_{k,n-1} + \boldsymbol{u}_{k,n}\boldsymbol{u}_{k,n}^T$ and $\hat{\boldsymbol{p}}_{k,n} = \gamma\hat{\boldsymbol{p}}_{k,n-1} + d_k(n)\boldsymbol{u}_{k,n}$, where $\gamma \in (0,1]$ is the forgetting factor, also met in the RLS algorithm [10]. The previous relations, imply that in order to construct the respective subspace, every node must have access to measurements coming out from the other nodes of the network, something that is, in general, infeasible in distributed networks. However, it is not essential to update $\hat{\boldsymbol{R}}_{k,n}$, $\hat{\boldsymbol{p}}_{k,n}$ at every time instance; we assume, instead, that

[1] Here, the subscript which denotes the node is suppressed.

[2] In [3], an extra step which was a projection of $\boldsymbol{\phi}_k(n)$ onto a hyperslab took place. Here, for simplicity purposes this step is omitted.

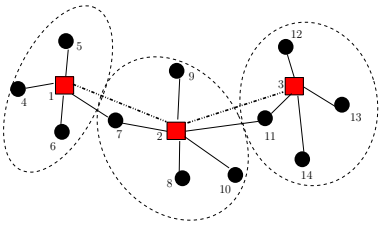

Fig. 1. Illustration of a hierarchical network with $L = 5$. The solid lines denote the simple communication links, whereas the dashed-dotted ones the hierarchical communication links. At every time instant, nodes have to transmit to their neighbourhood D coefficients. In addition to that, at time instance n node 1 transmits to node 2, $u_{1,n'}, d_1(n')$, at $n+1$, $u_{4,n'}, d_4(n')$, at $n+2$, $u_{5,n'}, d_5(n')$ and at $n+3$, $u_{6,n'}, d_6(n')$. Node 2, at time instance n, transmits to 3, $u_{2,n'}, d_2(n'), u_{7,n'}, d_7(n')$. At $n+1$, $u_{1,n'}, d_1(n'), u_{8,n'}, d_8(n')$, at $n+2$, $u_{4,n'}, d_4(n'), u_{10,n'}, d_{10}(n')$, at $n+3$, $u_{5,n'}, d_5(n'), u_{9,n'}, d_9(n')$ and at $n+4$, $u_{6,n'}, d_6(n')$. The rest of the communications follow similar philosophy. The largest bandwidth is needed for node 2 and amounts to $D+4$, where D originates from the D coefficients of the estimate and the rest 4 from the information needed to construct the subspace.

$\hat{\boldsymbol{R}}_{k,n}$, $\hat{\boldsymbol{p}}_{k,n}$ will be updated every L time instances and the approximations, now, are given by: $\hat{\boldsymbol{R}}_{k,n'} = \gamma\hat{\boldsymbol{R}}_{k,n'-1} + \boldsymbol{u}_{k,n'}\boldsymbol{u}_{k,n'}^T$ and $\hat{\boldsymbol{p}}_{k,n'} = \gamma\hat{\boldsymbol{p}}_{k,n'-1} + d_k(n')\boldsymbol{u}_{k,n'}$, with $n' = \lfloor \frac{n}{L} \rfloor + 1$, where $\lfloor\cdot\rfloor$ denotes the floor function. If one recalls that $\boldsymbol{u}_{k,n'} = [u_{k,n'}\ u_{k,n'-1} \dots u_{k,n'-m+1}]^T$, it can be readily seen that inside a time window, of size L, the newly arriving information from each node consists of two numbers: $u_{k,n'}$ and $d_k(n')$, and this information must be delivered to the other nodes of the network.

In order to improve the network's flow, we adopt a hierarchical model [6], in which the nodes are clustered, under a predefined protocol, and we can classify them into two subclasses: the hierarchical and the non-hierarchical ones. The former are able to communicate over three hops, whereas the latter are not, and every non-hierarchical node is connected to a hierarchical one. An example which illustrates how the information is distributed over the network can be seen in Fig. 1. Obviously, for a given network and a specific value of L, different scenarios can be adopted. The critical point is that the information related to the updates of $\hat{\boldsymbol{R}}'$ and $\hat{\boldsymbol{p}}'$, can be spread over L, thus reducing the bandwidth demand. Now, assume that $\boldsymbol{K}_n$ is a $m \times D$ matrix[3], whose columns form an orthonormal basis of $K_D(\hat{\boldsymbol{R}}'_{n'}, \hat{\boldsymbol{p}}'_{n'})$, with $\hat{\boldsymbol{R}}'_{n'} = \frac{1}{N}\sum_{k=1}^N \hat{\boldsymbol{R}}_{k,n'}$ and $\hat{\boldsymbol{p}}'_{n'} = \frac{1}{N}\sum_{k=1}^N \hat{\boldsymbol{p}}_{k,n'}$. It holds that $\forall \boldsymbol{w} \in K_D(\hat{\boldsymbol{R}}'_{n'}, \hat{\boldsymbol{p}}'_{n'})$ there exists $\tilde{\boldsymbol{w}} \in \mathbb{R}^D$ s.t. $\tilde{\boldsymbol{w}} = \boldsymbol{K}_n^T\boldsymbol{w}$. The resulting algorithm in the lower dimension space is

$$\tilde{\boldsymbol{\phi}}_k(n) = \sum_{l\in\mathcal{N}_k} c_{k,l}\tilde{\boldsymbol{w}}_l(n) = \sum_{l\in\mathcal{N}_k} c_{k,l}\boldsymbol{K}_n^T\boldsymbol{w}_l(n), \tag{7}$$

$$\tilde{\boldsymbol{w}}_k(n+1) = \tilde{\boldsymbol{\phi}}_k(n) + \tilde{\mu}_k(n)\times \left(\sum_{j=n-q+1}^{n} \omega_{k,j} P_{\tilde{S}_{k,j}}(\tilde{\boldsymbol{\phi}}_k(n)) - \tilde{\boldsymbol{\phi}}_k(n)\right), \tag{8}$$

where $\tilde{S}_{k,n} := \{\tilde{\boldsymbol{w}} \in \mathbb{R}^D : |d_k(n) - \tilde{\boldsymbol{u}}_{k,n}^T\tilde{\boldsymbol{w}}| \le \epsilon_k\}$, with $\tilde{\boldsymbol{u}}_{k,n} = \boldsymbol{K}_n^T\boldsymbol{u}_{k,n}$. Furthermore, $\tilde{\mu}_k(n) \in [0, 2\tilde{\mathcal{M}}_{k,n}]$ where

$$\tilde{\mathcal{M}}_{k,n} = \begin{cases} \frac{\sum_{j=n-q+1}^n \omega_{k,j}\|P_{\tilde{S}_{k,j}}(\tilde{\boldsymbol{\phi}}_k(n)) - \tilde{\boldsymbol{\phi}}_k(n)\|^2}{\sum_{j=n-q+1}^n \|\omega_{k,j}P_{\tilde{S}_{k,j}}(\tilde{\boldsymbol{\phi}}_k(n)) - \tilde{\boldsymbol{\phi}}_k(n)\|^2}, & \text{if } \tilde{\boldsymbol{\phi}}_k(n) \notin \bigcap_{j=n-q+1}^n \tilde{S}_{k,j}, \\ 1, & \text{otherwise.} \end{cases} \tag{9}$$

The complexity of the algorithm is of order: $O(qD)$ coming[4] from (8), $O(\frac{Nm}{L})$ from the update of $\hat{\boldsymbol{R}}'_{n'}$, $\hat{\boldsymbol{p}}'_{n'}$, and $O(\frac{Dm^2}{L})$ due to the computation of $\boldsymbol{K}_n$, [4].

Claim 1 *Eq. (7) is equivalent to*

$$\boldsymbol{\phi}_k(n) = \sum_{l\in\mathcal{N}_k} c_{k,l}\boldsymbol{w}_l(n),$$

$$\boldsymbol{w}_k(n+1) = \boldsymbol{K}_{n+1}\boldsymbol{K}_n^T\Bigg(\boldsymbol{\phi}_k(n) + \mu_k(n)\times \left(\sum_{j=n-q+1}^{n} \omega_{k,j}P_{S_{k,j}\cap K_D(\hat{\boldsymbol{R}}'_{n'}, \hat{\boldsymbol{p}}'_{n'})}(\boldsymbol{\phi}_k(n)) - \boldsymbol{\phi}_k(n)\right)\Bigg),$$

and $\mu_k(n) \in [0, 2\tilde{\mathcal{M}}_{k,n}]$.

Proof: Proof is omitted due to lack of space. □

Remark 1: From (7), it can be seen that the estimate transmitted from the nodes, at every time instance, is of length D. In the simulations section it will be verified that even a small D can provide considerably good performance of the respective algorithm.

Remark 2: Following a similar philosophy as in [4], it can be proved that (7) tracks $P^{(\boldsymbol{R}')}_{K_D(\boldsymbol{R}',\boldsymbol{p}')}(\boldsymbol{w}^*)$, where with $P^{(\boldsymbol{R}')}_{K_D(\boldsymbol{R}',\boldsymbol{p}')}$ we denote the projection onto the subspace, in the $\boldsymbol{R}'$ norm sense, instead of $\boldsymbol{w}^*$.

Theorem 2 ***Monotone Approximation****: Assume that there exists a non-negative integer, say* n_0*, for which* $\Omega = \bigcap_{n\ge n_0}\Omega_n \neq \emptyset$ *where* $\Omega_n = K_D(\hat{\boldsymbol{R}}'_{n'}, \hat{\boldsymbol{p}}'_{n'}) \cap \Omega'_n$ *with* $\Omega'_n := \bigcap_{k=1}^N\bigcap_{j=n-q+1}^n S_{k,j}$. *Then it holds that*

$$\|\boldsymbol{w}(n+1) - \underline{\boldsymbol{w}}_*\| \le \|\boldsymbol{w}(n) - \underline{\boldsymbol{w}}_*\|,\ \forall n \ge n_0,$$

where $\underline{\boldsymbol{w}}_* = \left[\boldsymbol{w}_*^T \dots \boldsymbol{w}_*^T\right]^T \in \mathbb{R}^{Nm}$, $\forall \boldsymbol{w}_* \in \Omega$ *and* $\boldsymbol{w}(n) = \left[\boldsymbol{w}_1^T(n) \dots \boldsymbol{w}_N^T(n)\right]^T \in \mathbb{R}^{Nm}$. *The previous inequality states that every iteration leads us closer to the feasible region, i.e., the intersection of the respective hyperslabs with the Krylov subspace. Notice here, that we let a finite number of outliers not to share intersection, without affecting the convergence of the algorithm.*

Asymptotic Consensus*: As mentioned in section 2, a desirable property of distributed learning is consensus. Under the previously mentioned assumptions and if there exists* n_1 *such that* $\hat{\boldsymbol{R}}'_n = \hat{\boldsymbol{R}}'_{n_1}$, $\forall n \ge n_1$ *and* $\hat{\boldsymbol{p}}'_n = \hat{\boldsymbol{p}}'_{n_1}$, $\forall n \ge n_1$ [5] *then asymptotic consensus holds, i.e.,*

$$\lim_{n\to\infty} \|\boldsymbol{w}_k(n) - \boldsymbol{w}_l(n)\| = 0,\ \forall k, l \in 1, \dots, N.$$

Strong Convergence*: Let us define* $\mathcal{O} := \{\boldsymbol{z} \in \mathbb{R}^{Nm} : \boldsymbol{z} = [\boldsymbol{v}^T \dots \boldsymbol{v}^T]^T,\ \boldsymbol{v} \in \mathbb{R}^m\}$. *If the previously mentioned assumptions*

[3] This is constructed locally, with the Gram-Schmidt method.

[4] In a parallel processing environment, this complexity drops to $O(D)$.

[5] This assumption does not pose a problem to us, if the statistics of the nodes remain unchanged, due to the fact that for a large n_1 the approximations of $\boldsymbol{R}'$, $\boldsymbol{p}'$ will be good and it will not be essential for the subspace to change.

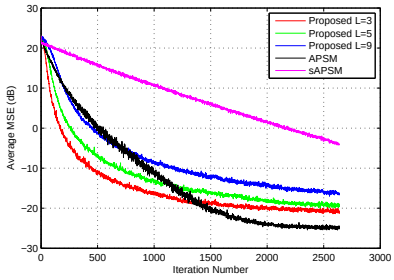

Fig. 2. Average MSE in the first experiment.

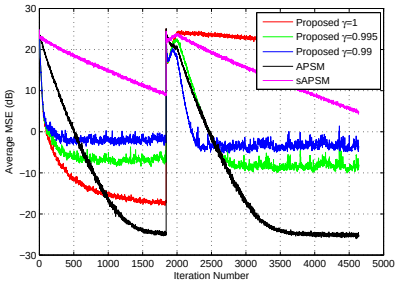

Fig. 3. Average MSE in the second experiment.

hold, and under some other mild assumptions, which are omitted to save space, there exists $\hat{\boldsymbol{w}}_* \in \mathcal{O}$ *such that*

$$\lim_{n\to\infty} \boldsymbol{w}(n) = \hat{\boldsymbol{w}}_*.$$

Proof: The proof is omitted due to lack of space. This theorem states that the algorithm, for the whole network, converges asymptotically to a point, in $\mathbb{R}^{Nm}$, which respects the consensus property. Furthermore, the distance of the estimate occurring, at each node, from the set of the desirable solutions, i.e., the intersection of the subspace with the hyperslabs, tends to zero as $n \to \infty$. □

5. EXPERIMENTS

In this section, we present experiments within the system identification task, in order to study the performance of the developed algorithm. We compare the proposed algorithm with a modification of the algorithm given in (4), (5), denoted as subsampled APSM (sAPSM), where each node, instead of transmitting the whole estimate vector, at every time instance, transmits a subset of D coefficients of it. Such a scenario falls within the spirit of partial updating. To be more specific, at time instance 1, the first D coefficients are transmitted, at time instance 2, the coefficients $\#D+1,\ldots,\#2D$ and so on. In the first experiment we consider a distributed network consisted of $N = 10$ nodes and the unknown vector to be estimated is of length $m = 160$. The standard deviation of the noise, which is assumed to be zero-mean and Gaussian, is given by $\sigma_k = \sqrt{\alpha_k \times 0.1}$ where $\alpha_k \in (0, 0.5)$ under the uniform distribution. Furthermore, $u_{k,n} = \tau_k u_{k,n-1} + \chi_{k,n}$, where $\tau_k \in (0, 0.5)$ and respects the uniform distribution, and $\chi_{k,n}$ is zero-mean Gaussian with standard deviation equal to 1. We also choose $D = 10$ for the Krylov based algorithms and for the sAPSM, and $q = 4$, $\epsilon_k = \sqrt{2} \times \sigma_k$, $\mu_k(n) = \frac{\tilde{\mathcal{M}}_{k,n}}{2}$ for all the algorithms. Finally, the combiners $c_{k,l}$ are chosen with respect to the Metropolis rule [1], the experiments are averaged over 100 experiments, for smoothing purposes, and the comparative metric presented is the average Mean Square Error (MSE), i.e., $\frac{1}{N}\sum_{k=1}^{N}(d_k(n) - \boldsymbol{u}_{k,n}^T \boldsymbol{w}_k(n))$. In the first experiment (Fig. 2) we let $\gamma = 1$ and we alter the parameter L. It can be seen that the smaller the update window, the faster the convergence. This is expected, because for a small window we update the estimate of the subspace more often, and we reach sooner to a good approximation of it, compared to the case of a larger window. Furthermore, it can be readily seen, that the Krylov-based algorithms outperform significantly sAPSM. When the Krylov based algorithms are compared with the standard APSM, i.e., full dimensionality is used, we observe that there is a slight loss of performance with respect to the error floor, although the Krylov based algorithms converge faster. In the second experiment (Fig. 3), the parameters remain the same as in the previous one, albeit at $n = 1800$ the channel suddenly changes. This experiment takes place in order to check the tracking ability of the proposed algorithm. Now, we fix $L = 1$ and we alter γ. From Fig. 3 it can be seen that until the channel changes, the best performance is achieved for $\gamma = 1$ whereas for smaller γ the steady state error floor is increased. However, as in the RLS case [10], if $\gamma = 1$, the algorithm has a long memory of the old subspace that has to change and its tracking ability is not good. On the contrary, the other choices of γ provide a good tracking ability. Of course for large L the tracking ability may be affected. However, different scenarios can be considered, which will be presented elsewhere due to lack of space.

6. CONCLUSIONS

A novel algorithm, for bandwidth reduction in adaptive learning in diffusion networks, is introduced in the framework of set theoretic estimation. To achieve this reduction, the estimates are forced to lie in a lower dimension Krylov subspace. The results show that substantial bandwidth reduction can be achieved at the expense of only slight performance degradation, with respect to the error floor.

7. REFERENCES

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

[2] C.G. Lopes and A.H. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Trans. on Signal Processing*, vol. 55, no. 8, pp. 4064–4077, 2007.

[3] Symeon Chouvardas, Konstantinos Slavakis, and Sergios Theodoridis, "A novel adaptive algorithm for diffusion networks using projections onto hyperslabs," in *2010 IAPR Workshop on Cognitive Information Processing*, Elba Island, Italy.

[4] Masahiro Yukawa, Rodrigo C. de Lamare, and Isao Yamada, "Robust reduced-rank adaptive algorithm based on parallel subgradient projection and krylov subspace," *IEEE Trans. on Signal Processing*, vol. 57, no. 12, pp. 4660–4674, 2009.

[5] G.K.E. Dietl, *Linear estimation and detection in Krylov subspaces*, Springer Verlag, 2007.

[6] F.S. Cattivelli and A.H. Sayed, "Hierarchical diffusion algorithms for distributed estimation," in *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*. IEEE, 2009, pp. 537–540.

[7] G. Mateos, I.D. Schizas, and G.B. Giannakis, "Performance Analysis of the Consensus-Based Distributed LMS Algorithm," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, pp. 19, 2010.

[8] K. Slavakis, I. Yamada, and N. Ogura, "The adaptive projected subgradient method over the fixed point set of strongly attracting nonexpansive mappings," *Numerical Functional Analysis and Optimization, 27*, vol. 7, no. 8, pp. 905–930, 2006.

[9] K. Slavakis, Y. Kopsinis, and S. Theodoridis, "Adaptive algorithm for sparse system identification using projections onto weighted l1 balls," *IEEE Intl. Conference on Acoustics Speech and Signal Processing, ICASSP, Dallas*, 2010.

[10] A.H. Sayed, *Fundamentals of adaptive filtering*, John Wiley & Sons, New Jersey, 2003.