TRADING OFF COMMUNICATIONS BANDWIDTH WITH ACCURACY IN ADAPTIVE DIFFUSION NETWORKS

Symeon Chouvardas\textsuperscript{1}  \hspace*{1cm} Konstantinos Slavakis\textsuperscript{2}  \hspace*{1cm} Sergios Theodoridis\textsuperscript{3}

\textsuperscript{1}University of Athens, Dept. of Informatics and Telecommunications, Athens 15784, Greece. Emails: schou@di.uoa.gr, stheodor@di.uoa.gr
\textsuperscript{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, etc. A typical WSN consists of a number of decentralized solutions have been proposed. The incremental, in practice, non-deterministic, type 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 $\omega^* \in \mathbb{R}^m$, 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 $(d_k(n), u_{k,n}) \in \mathbb{R} \times \mathbb{R}^m$, which are related according to the linear system

$$d_k(n) = u_{k,n}^T \omega^* + v_k(n),$$

where $v_k(n)$ is the noise process with standard deviation equal to $\sigma_k$. The general concept of a diffusion network can be summarized as follows. Each sensor collects information from its environment, i.e., the measurement pair $(d_k(n), u_{k,n})$, in order to proceed to the adaptation step, and it also exploits the estimates transmitted by its neighbouring nodes. From now on, $N_k$ 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, $u_l(n), l \in N_k$. For node $k$, at time instance $n$, the most common example of a combination strategy is:
\[ \phi_k(n) = \sum_{l \in N_k} c_{k,l} w_l(n), \] 
(4)

\[ w_k(n+1) = \phi_k(n) + \mu_k(n) \times \left( \sum_{j=n-q+1}^{q} \omega_{k,j} P_{S_{k,j}}(\phi_k(n)) - \phi_k(n) \right), \] 
(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 \( A \in \mathbb{R}^{m \times m} \) and a vector \( e \in \mathbb{R}^m \), the definition of the \( D \)-dimensional Krylov subspace is

\[ K_D(A, e) = \text{span} \{ e, Ae, A^2 e, \ldots, A^{D-1} e \}. \]

Let us define \( \mathcal{R} = \mathcal{E}(u_n, u_n^T) \) and \( \mathcal{P} = \mathcal{E}(d(n), u_n) \), where \( d(n), u_n \) are related according to (1); the celebrated Wiener-Hopf equation [10] states that \( u^* = \mathcal{R}^{-1} \mathcal{P} \). It has been proved, e.g., [5], that the reduced rank Wiener filter, of dimension \( D \), belongs to \( K_D(A, e) \). 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 \( \mathcal{L} : \mathbb{R}^m \rightarrow [0, \infty) \), for the whole network

\[ \mathcal{L}(w) = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E} \left\{ (d_k(n) - u_{k,n}^T w)^2 \right\} \]
\[ = \frac{1}{N} \sum_{k=1}^{N} (w^T R_k w - 2 w^T p_k + \sigma_{d_k}^2) \]
\[ = w^T R' w - 2 w^T p' + \frac{1}{N} \sum_{k=1}^{N} \sigma_{d_k}^2, \]
(6)

where \( \sigma_{d_k} = \mathbb{E}(d_k^2(n)) \), \( R' = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}(u_{k,n} u_{k,n}^T) = \frac{1}{N} \sum_{k=1}^{N} R_k \) and \( p' = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}(d_k(n) u_{k,n}) = \frac{1}{N} \sum_{k=1}^{N} p_k \).

It can be seen, that the solution minimizing (6) is given by \( w^* = R'^{-1} p' \). This argument indicates that it may be reasonable to select \( R' \) and \( p' \) (i.e., the average values) in order to construct the respective Krylov subspace. The question, now, is how to construct \( R', p' \), since we assume that there is no a-priori knowledge of \( R_k, p_k \). A possible strategy, followed also in [4], is to resort to approximations of the unknown quantities, in which the measurements, \( d_k(n), u_{k,n} \), are exploited. To be more specific, \( R_{k,n} = \gamma R_{k,n-1} + u_{k,n} u_{k,n}^T \) and \( p_{k,n} = \gamma p_{k,n-1} + d_k(n) u_{k,n} \), where \( \gamma \) is an 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 \( R_{k,n}, p_{k,n} \) at every time instance; we assume, instead, that

---

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

2In [3], an extra step which was a projection of \( \phi_k(n) \) onto a hyperslab took place, here, for simplicity purposes this step is omitted.
coefficients of the estimate and the rest needed for node 2 of the communications follow similar philosophy. The large bandwidth is estimated, now, are given by:

\[ \phi_k(n) = \sum_{i \in N_k} c_{k,i} \omega_i(n), \]

\[ \omega_k(n+1) = \phi_k(n) + \mu_k(n) \]

where \( c_{k,i} \) denote the projection onto the subspace, in the estimations section it will be verified that even a small number of outliers not to share intersection, can provide asymptotic consensus.

\[ \omega_k(n+1) = K_{n+1}K_n^T \left( \phi_k(n) + \mu_k(n) \right) \]

Remark 1: From (7), it can be seen that the estimate transmitted from the nodes, at time instance \( n \), 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 \( P_{K_D(R',p')}^T(\omega^*) \), where with \( P_{K_D(R',p')}^T \) we denote the projection onto the subspace, in the \( \Omega^* \) norm sense, instead of \( \omega^* \).

\[ ||\omega(n+1) - \omega|| \leq ||\omega(n) - \omega||, \quad \forall n \geq n_0, \]

where \( \omega_k = [\omega_1, \ldots \omega_N]^T \in \mathbb{R}_+^N \) where \( \Omega = \mathbb{R} = \{\omega \in \mathbb{R}^N \mid \omega_k \geq 0 \} \). 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 \( n \) such that \( \hat{\omega}_k = \bar{\omega}_k, \forall n \geq n_1 \) and \( \bar{\omega}_k = \bar{\omega}_k, \forall n \geq n_1 \) then asymptotic consensus holds, i.e.

\[ \lim_{n \to \infty} ||\omega_k(n) - \omega_l(n)|| = 0, \quad \forall k, l = 1, \ldots, N. \]

Strong Convergence: Let us define \( \mathcal{O} := \{z \in \mathbb{R}_+^N : z = [v^T, \ldots v^T]^T, v \in \mathbb{R}^m \} \). If the previously mentioned assumptions

\[ 1 \leq p < \infty, \quad \rho < 1, \quad \lambda_{\max} < 1. \]

in a parallel processing environment, this complexity drops to \( O(D) \).

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 \( R', p' \) will be good and it will not be essential for the subspace to change.
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 $q = 4$, $\epsilon_k = \sqrt{2} \times \sigma_k$, $\mu_k (n) = \frac{\epsilon_k^2}{\sigma_k^2}$ 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., $\frac{1}{N} \sum_{k=1}^{N} (d_k(n) - u_{k,n}^T w_k(n))^2$. In the first experiment (Fig. 2) we let $\gamma = 1$ and we alter $\gamma$. From Fig. 3 it can be seen that until the channel changes, the best performance is achieved for $\gamma = 1$ whereas for smaller $\gamma$ 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 $\gamma$ 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.

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 $q = 4$, $\epsilon_k = \sqrt{2} \times \sigma_k$, $\mu_k (n) = \frac{\epsilon_k^2}{\sigma_k^2}$ 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., $\frac{1}{N} \sum_{k=1}^{N} (d_k(n) - u_{k,n}^T w_k(n))^2$. In the first experiment (Fig. 2) we let $\gamma = 1$ and we alter $\gamma$. From Fig. 3 it can be seen that until the channel changes, the best performance is achieved for $\gamma = 1$ whereas for smaller $\gamma$ 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 $\gamma$ 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