LMS-Based Leader Selection for Distributed Estimation

Sander Ulp, Yannick Le Moulléc, and Muhammad Mahtab Alam
Thomas Johann Seebeck Department of Electronics
Tallinn University of Technology
Tallinn, Estonia

Email: sander.ulp@ttu.ee, yannick.lemoullec@ttu.ee, muhammad.alam@ttu.ee

Abstract—In this paper we investigate the distributed leader selection algorithm in a LMS-based (least mean squares) adaptive network. We introduce the modified algorithm and express the MSD (mean squared deviation) performance of the algorithm. The outline of the communication and setting of the problem is given and the LMS-based distributed leader selection algorithm is described. We compare the analytical performance of the algorithm to the diffusion algorithms and illustrate that the leader selection has robust performance compared to the diffusion algorithms. Simulation results show that the diffusion algorithms outperform the leader selection algorithm in most cases. However and interestingly, the results also show that leader selection attains the same or even better network performance than the diffusion algorithms in a scenario where one of the nodes is in better conditions (e.g. lower noise power) than the other nodes in the network.

I. INTRODUCTION

Estimation over distributed networks is a method to enhance the performance of nodes that cooperate to solve a common task or problem. Different methods and algorithms have been proposed for estimation over distributed networks. These approaches include strategies such as centralized, diffusion, consensus and incremental [1]. Many of these methods, however, require some kind of architecture or a priori knowledge of the network.

It has been shown that diffusion algorithms achieve good steady state MSD performance in distributed networks and they do not require any predefined architecture [2], therefore we focus our interest on diffusion algorithms. It has also been shown that the ATC (adapt and then combine) diffusion network outperforms the CTA (combine and then adapt) diffusion network and thus, we consider the ATC diffusion network in this work [2]. The performance of these diffusion algorithms rely on the calculation of the communication weights which are based on a secondary parameter, for example noise powers at the nodes [1].

There are several methods for calculating the weights for the communication and in the depth analysis of the performance of different weights have been derived [3], [4]. By assuming that the noise powers are not uniform across the network the relative variance and relative degree variance weight calculation methods have been shown to achieve good performance [4] and will be considered in this paper. The secondary parameters are in most of the cases not known beforehand and have to be estimated [1], which directly impacts the weight calculation and therefore influences the performance of the diffusion network. Distributed leader selection offers an interesting alternative to the diffusion algorithms as it is more robust to the secondary parameter estimation [5].

There are many works on leader selection, however, they rely on different constraints or a priori knowledge [6]–[11]. In [8]–[11] works the leaders are known beforehand and have extra information or better sensors compared to the followers in the network. Some works also require the nodes to know the topology beforehand or estimate the paths to the leader [6], [12]. Therefore, for the comparison with the diffusion algorithms to be fair we are interested in a fully distributed leader selection solution that is able to adapt and learn in real-time.

The purpose of this work is to investigate the MSD performance of the modified version of our previously proposed distributed leader selection algorithm [5]. The algorithm is modified and implemented using LMS to have fair comparison with the diffusion algorithms proposed in [4]. We expected that the diffusion algorithms are able to outperform the distributed leader selection theoretically. However, the simulation results show that there are situations where the distributed leader selection is able to achieve similar or even beter network performance. We analyse the scenarios in which the distributed leader selection performance is able to perform equally or better compete with diffusion strategies.

In this paper, italic letters are used for scalars (c, E), lower case bold letters are denoting vectors (x). All vectors are column vectors except for regression vectors, which are denoted as uk,i. The operator E[·] stands for mathematical expectation of the subject and (y*) denotes the complex conjugate of (y).

The remainder of the paper is organized as follows. In Section 2 we introduce the problem setting. Section 3 gives an overview of the LMS-based leader selection algorithm.
Section 4 describes the performance of the algorithm, the comparison to the diffusion algorithms and presents the simulation results. The final section concludes the paper and summarizes the results of the paper.

II. PROBLEM FORMULATION

Assume that there are $K$ nodes. Each of the nodes have access to $d_k(i)$ and $u_{k,i}$ at each time instant $i$.

$$d_k(i) = u_{k,i}w^0 + v_k(i)$$  \hspace{1cm} (1)

where $d_k(i)$ is the measurement at node $k$, $u_{k,i}$ is the row regression vector, $w^0$ is the unknown column vector and $v_k(i)$ is zero-mean white random noise with power $\sigma_{v,k}^2$. The nodes estimate the $w^0$ to minimize the cost function:

$$J_k(w) = E \left[ d_k(i) - u_{k,i}w \right]^2$$  \hspace{1cm} (2)

Each of the nodes employs the LMS algorithm for adaptation:

$$w_{k,i} = w_{k,i-1} + \mu_k \cdot u_{k,i}^* [d_k(i) - u_{k,i}w_{k,i-1}]$$  \hspace{1cm} (3)

where $\mu_k$ is a positive step-size $0 < \mu_k < 1$.

This outlines the non-cooperating network where every node works independently. To improve the performance of the nodes at the network level, the nodes can cooperate by exchanging information between themselves.

The network performance can be further improved by assigning weights based on the situation of nodes which is related to some secondary parameter. The noise power $\sigma_{v,k}^2$ can be used as a secondary parameter to calculate the weights. We assume that noise powers are known a priori to the nodes for simplicity and for the sake of a fair comparison to the diffusion algorithms in the performance evaluation section. In practice, the noise powers are not usually known and have to be estimated. There are works on estimating noise powers at different nodes [4], [13]. We have also covered a possible solution in our last work which is based on the MDL (minimum description length) subspace algorithm [5].

III. LMS-BASED DISTRIBUTED LEADER SELECTION

The goal of the distributed leader selection algorithm is to determine the leader node in a network of nodes in a distributed manner. Whether the noise powers are known or estimated, the nodes are able to use the secondary parameter information to determine which of the nodes is the best performing node and follow its lead. We have previously outlined the algorithm for the distributed leader selection [5]. In this paper, we look at the distributed leader selection algorithm using a LMS-based adaptive network, see Algorithm 1.

The nodes exchange $w_{k,i}$ estimates between themselves, together with the corresponding weight $\alpha_k(i)$ that is assigned to the estimate at iteration $i$. The weights are calculated based on the noise powers; the algorithm is given as follows:

Each of the nodes $k$ performs the LMS adaptation of $w_{k,i}$ at each iteration $i$ as can be seen on Line 3. The nodes exchange their results of the adaptation and the corresponding weight with nodes that they are connected in the neighbourhood. The nodes proceed to determine whether their weight is the largest of the neighbouring nodes (Line 4). If this statement holds true, the node uses its own LMS adaptation result as the best available information (Line 5). The node also calculates adaptively a new weight value for the next iteration based on the noise power (Line 6).

![Algorithm 1: LMS-Based Distributed Leader Selection](image)

If the statement on Line 4 is false and the weight at node $k$ is smaller than any of the weights from connected nodes then the node determines the best neighbouring source of information (Line 8). This corresponds to the neighbouring node $c_k$ with the largest weight $\alpha_k(i-1)$. The node uses the LMS adaptation result from the best neighbour as can be seen on Line 9. The node will calculate the next iteration weight based on the own noise power and the weight that corresponds to the best neighbour, as can be seen on Line 10. This ensures that if the noise powers of the nodes change the node with the best information has its information distributed during the next iterations.

IV. PERFORMANCE EVALUATION

The LMS-based distributed leader selection algorithm performance can be obtained from the single node performance from the non-cooperative network. All of the nodes in the cooperating network have different MSD performance due to their respective noise powers $\sigma_{v,k}^2$. The MSD performance of the non-cooperating node (denoted as $ncoop$) employing LMS is given as [1]:
\[ \text{MSD}_{\text{coop},k} = \frac{\mu M}{2} \sigma_{v,k}^2 \]  
(4)

where \( M \) is the length of the adaptive filter, \( \mu \) is the step size of the LMS algorithm, \( \sigma_{v,k}^2 \) is the noise power at node \( k \).

The performance of the non-cooperating LMS network is given as [1]:

\[ \text{MSD}_{\text{coop}} = \frac{\mu M}{2} \left( \frac{1}{N} \sum_{k=1}^{N} \sigma_{v,k}^2 \right) \]  
(5)

If we employ the leader selection algorithm, the network achieves the best performing nodes MSD performance. In this setting this is the node with the lowest noise power. We can express the network MSD performance of the leader selection algorithm (denoted as \( l_s \)) as:

\[ \text{MSD}_{l_s} = \min_k \text{MSD}_{\text{coop},k} \]  
(6)

It is important to note that the network MSD performance of the leader selection algorithm is equal to the individual node performance. Thus, we can express the network and node performance of the leader selection algorithm as:

\[ \text{MSD}_{l_s} = \text{MSD}_{l_s,k} = \frac{\mu M}{2} \min_k \sigma_{v,k}^2 \]  
(7)

A. Comparison of algorithms

The analytical MSD performance of the diffusion network (denoted as coop) as well as the node performance with optimal weights is given as [1]:

\[ \text{MSD}_{\text{coop}} = \text{MSD}_{\text{coop},k} = \frac{\mu M}{2} \left( \sum_{k=1}^{N} \frac{1}{\sigma_{v,k}^2} \right)^{-1} \]  
(8)

If we compare the MSD network performance of the diffusion network with optimal weights (7) and the performance of the leader selection algorithm (8) we can see that the leader selection does not outperform the diffusion network. To demonstrate this we write:

\[ \frac{\mu M}{2} \min_k \sigma_{v,k}^2 \geq \frac{\mu M}{2} \left( \sum_{k=1}^{N} \frac{1}{\sigma_{v,k}^2} \right)^{-1} \]  
(9)

which we can write as:

\[ \min_k \sigma_{v,k}^2 \geq \left( \sum_{k=1}^{N} \frac{1}{\sigma_{v,k}^2} \right)^{-1} \]  
(10)

To further elaborate, if we rewrite (10) and define the node with the lowest noise power as \( p \):

\[ p = \arg \min_k \sigma_{v,k}^2 \]  
(11)

\[ \left( \frac{1}{\sigma_{v,p}^2} \right)^{-1} \geq \left( \frac{1}{\sigma_{v,p}^2} + \sum_{k=1, k \neq p}^{N} \frac{1}{\sigma_{v,k}^2} \right)^{-1} \]  
(12)

We see that the both equation sides include the term that includes the lowest noise power. From this we can see that the performance difference between the diffusion network and the leader selection algorithm is related to the sum term. We can conclude that the diffusion network outperforms the leader selection algorithm and we can write:

\[ \text{MSD}_{\text{coop}} \leq \text{MSD}_{l_s} \]  
(13)

Let us consider the case where one of the nodes is in a more favourable situation and has significantly lower noise power compared to the other nodes in the network:

\[ \sum_{k=1, k \neq p}^{N} \frac{1}{\sigma_{v,k}^2} \ll \frac{1}{\sigma_{v,p}^2} \]  
(14)

In this scenario we see that the performance of the cooperating network largely depends on the node with lowest noise power:

\[ \text{MSD}_{\text{coop}} \approx \frac{\mu M}{2} \left( \frac{1}{\sigma_{v,p}^2} \right)^{-1} \]  
(15)

The resulting equation is the equation for the performance of the distributed leader selection algorithm (7) and we can write that under these conditions:

\[ \text{MSD}_{\text{coop}} \approx \text{MSD}_{l_s} \]  
(16)

To summarize, the diffusion network outperforms the leader selection algorithm in most cases and in the presence of a node in more favorable conditions the performance difference between the two algorithms is marginal. It is important to note that the above results (13,16) hold for optimal combinational weights for the diffusion network and are valid if all the nodes obtain the same performance. However, simulations results show that it is not the case and there is variation among the performance of the nodes in the diffusion network as can be seen in [13], [14]. Furthermore, it is also important to note that the combination weight optimality heavily relies on the estimation accuracy of the secondary parameter. Inaccuracies of the estimation leads to the degradation of the diffusion network performance in comparison to the leader selection as the inaccuracy of the secondary parameter does not affect the performance to such large extent. Taking this into account the leader selection becomes an interesting alternative and in the next subsection we explore these scenarios in simulations.
Fig. 1: Topology of the network.

Fig. 2: Noise powers at nodes.

B. Simulation results

Let us assume a strongly connected network of nodes with randomly generated topology and the amount of nodes in the generated network is $K = 21$ (Fig. 1).

The nodes share their estimates based on the connections and it is assumed that the nodes have a self-loop which allows them to communicate with themselves. The communication between the nodes is full duplex and is modelled as lossless and noiseless. A constant step size $\mu = 0.01$ is selected for all the nodes in the simulations. The noise power levels are randomly generated between $\sigma^2_{n,k} = [-5 -30]$ dB and are modelled as AWGN (additive white Gaussian noise) for each node. The regression vector power $P_n = 1$ is constant at each node. The results are averaged over 100 independent trials.

We examine two scenarios in our simulations. In the first scenario, the distribution of the noise powers can be seen on Fig. 2a. In the second scenario there is one node that has noticeably lower noise power than the other nodes in the network, as can be seen in Fig. 2b.

For the diffusion network we use the ATC algorithm and for the weight calculation the relative-variance rule and the relative-degree-variance rule [4]. We compare the network level MSD performance of the non-cooperating, the diffusion algorithms and the leader selection algorithm (Fig. 3). In the first scenario, the diffusion network algorithms' performances are close to each other and have better performance than the other algorithms, including the leader selection algorithm (Fig. 3a). However, for the other scenario the leader selection algorithm is able to achieve better network performance than the diffusion algorithms as can be seen in Fig. 3b.

The results can be explained by observing the MSD steady state performance of the individual nodes (Fig. 4). In the first scenario the diffusion network nodes' performances are closer to each other, but for the second scenario there are larger variations among different nodes' performances, as can be seen in Fig. 4a and Fig. 4b. In the first scenario the steady state performance of the nodes in the leader selection algorithm is poorer than the performances of the nodes of the diffusion algorithms. In the second scenario we see that some of the nodes of the diffusion algorithms are unable to achieve better performance than the leader nodes performance in the leader selection algorithm. This results in a better MSD network performance for the leader selection algorithm.
V. CONCLUSION

In this paper, we investigated the distributed leader selection algorithm in a LMS-based adaptive network. The algorithm analytical MSD performance was given and the performance was compared to the diffusion algorithm performance with optimal weights. The analytical performance shows that the diffusion networks are able to achieve better performance than the leader selection algorithm. It is shown that the leader selection achieves similar performance under the condition where one of the nodes is in significantly better conditions. For the simulations, the relative variance and relative degree variance diffusion algorithms were compared to the leader selection algorithm. The diffusion algorithms were able to achieve better performance than the leader selection in most cases. However, in the presence of one node with significantly better conditions, the leader selection algorithm is able to obtain similar or even better network performance than the algorithms in comparison.

ACKNOWLEDGMENT

This research was supported by the European Regional Development Fund and the Information Technology Foundation for Education. This project has also received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 668995 and the Institutional Research Project IUT19-11 financed by the Estonian Research Council.

REFERENCES


