

Diffusion Strategies Outperform Incremental Strategies Distributed Estimation over Adaptive Networks

Vipul C.Zilpelwar, Sushant S.Chiwarkar

Abstract— Adaptive networks consist of a collection of nodes with adaptation and learning abilities. The nodes interact with each other on a local level and diffuse information across the network to solve estimation and inference tasks in a distributed manner. In this work, we compare the mean-square performance of two main strategies for distributed estimation over networks: incremental strategies and diffusion strategies. The analysis in the paper confirms that under constant step-sizes, diffusion strategies allow information to diffuse more thoroughly through the network and this property has a favorable effect on the evolution of the network: diffusion networks are shown to converge faster and reach lower mean-square deviation than consensus networks, and their mean-square stability is insensitive to the choice of the combination weights. In contrast, it is shown that consensus networks can become unstable even if all the individual nodes are stable and able to solve the estimation task on their own. When this occurs, cooperation over the network leads to a catastrophic failure of the estimation task. This phenomenon does not occur for diffusion networks: we show that stability of the individual nodes always ensures stability of the diffusion network irrespective of the combination topology.

Index Terms— Adaptive networks, diffusion strategy, distributed estimation, dRLS, incremental strategy, LC-dRLS, mean square error performance.

1 INTRODUCTION

Distributed and sensor networks are emerging as a formidable technology for a variety of applications, ranging from precision agriculture, to environment surveillance, to target localization. However, the advantages advocated by the use of distributed and cooperative processing [1] demand adaptive processing capabilities at the distributed nodes in order to be able to cope with time variations in the environment and the network. In addition, the adaptive processors should enable the network to respond to such variations in real-time and to adjust its performance accordingly. Inspired by incremental strategies [2], we propose distributed processing strategies over what we refer to as adaptive networks (e.g., [3]). The proposed strategies require the adaptive nodes to share information only locally and to exploit both spatial and temporal information in a cooperative fashion. Different cooperation policies will lead to different distributed algorithms. Each node k across an N -node network is assumed to have access to time realizations $\{\mathbf{d}_k(i), \mathbf{u}_k(i)\}$, of zero-mean random data $\{\mathbf{d}_k, \mathbf{u}_k\}$, with $\mathbf{d}_k(i)$ a scalar measurement and $\mathbf{u}_k(i)$ a regression row vector; both at time i – see Figure 1. The nodes should use the data to estimate some unknown common vector \mathbf{w}^o . Rather than expect each node to function independently of the other nodes, the nodes will instead collaborate with each other in an adaptive manner in order to achieve three objectives: (1) improve global performance with reduced communication; (2) allow the nodes to converge to the desired estimate without the need to share global information; (3) endow the network with learning abilities.

Vipul C. Zilpelwar is currently pursuing bachelor's degree program in Computer Science and Engineering in SGBAU University, India.

Email: vipulzilpelwar007@yahoo.com

Sushant C. Chiwarkar is currently pursuing bachelors degree program in Computer Science and Engineering in SGBAU University, India. Email: sushant_chiw@rediffmail.com.

2 LITERATURE REVIEW

2.1 History

In distributed processing, agents generally collect data generated by the same underlying unknown distribution and then solve the desired estimation and inference tasks cooperatively. In this paper, we consider the situation in which the data observed by the agents may arise from different distributions or models. Agents do not know beforehand which model accounts for the data of their neighbors. The objective for the network becomes that of guiding all agents towards the same common goal. In these situations, where agents are subject to data from unknown different sources, conventional distributed estimation strategies would lead to biased solutions. We first show how to modify existing strategies to guarantee unbiasedness. We then develop a classification scheme for the agents to identify the models that generated the data, and propose a procedure by which the entire network can be made to converge towards the same model through a collaborative decision-making process.

2.2 Broad Objective

To endow a network with adaptation capabilities:

- Individual nodes are adaptive.
- Individual nodes share local information.
- The network responds in real-time to excitations.
- Rather than expect each node to function independently, the nodes will instead collaborate with each other in an adaptive manner in order to achieve three objectives:
 - (1) improve global performance with reduced communication
 - (2) allow the nodes to converge to the desired estimate without the need to share global information
 - (3) endow the network with learning abilities.

2.3 Implementation

DISTRIBUTED ESTIMATION

We are interested in estimating an unknown vector w° from measurements collected at N nodes in a network. Each node k has access to realizations of zero-mean data $\{d_k u_k\}$, $k = 1, \dots, N$, where d_k is a scalar and u_k is $1 \times M$. We collect the regression and measurement data into global matrices:

$$U \triangleq \text{col}\{u_1, u_2, \dots, u_N\} \quad (N \times M) \quad (1)$$

$$d \triangleq \text{col}\{d_1, d_2, \dots, d_N\} \quad (N \times 1) \quad (2)$$

and pose the minimum mean-square error estimation problem:

$$\min_w J(w), \quad \text{where } J(w) = E\|d - Uw\|^2 \quad (3)$$

The optimal solution w° satisfies the normal equations [4]:

$$R_{du} = R_u w^\circ \quad (4)$$

where

$$R_u = E U^* U = \sum_{k=1}^N R_{u,k}, \quad R_{du} = E U^* d = \sum_{k=1}^N R_{du,k} \quad (5)$$

Computing w° from (4) requires every node to have access to the global statistical information $\{R_u, R_{du}\}$, thus draining communications and computational resources. In [3] we proposed a distributed solution (incremental LMS) that allows cooperation among nodes through limited local communications, while equipping the nodes with adaptive mechanisms to respond to time-variations in the underlying signal statistics.

3 VARIOUS ADAPTATIONS

3.1. INCREMENTAL LMS ADAPTATION

In this section, we review the distributed incremental LMS algorithm [3], which is a starting point for the later developments. The algorithm is obtained as follows. We start from the standard gradient-descent implementation

$$w_i = w_{i-1} - \mu [\nabla J(w_{i-1})]^* \quad (6)$$

for solving the normal equations (4), where $\mu > 0$ is a suitably chosen step-size parameter, w_i is an estimate for w° at iteration i , and $\nabla(\cdot)$ denotes the gradient vector of $J(w)$ evaluated at w_{i-1} . For μ sufficiently small we will have $w_i \rightarrow w^\circ$ as $i \rightarrow \infty$. This iterative solution could be applied at every node k or centrally at some central node. A distributed version can be motivated as follows. The cost function $J(w)$ can be decomposed as:

$$J(w) = \sum_{k=1}^N J_k(w), \quad \text{where } J_k(w) \triangleq E|d_k - u_k w|^2 \quad (7)$$

which allows us to rewrite (6) as

$$w_i = w_{i-1} - \mu \left[\sum_{k=1}^N \nabla J_k(w_{i-1}) \right]^* \quad (8)$$

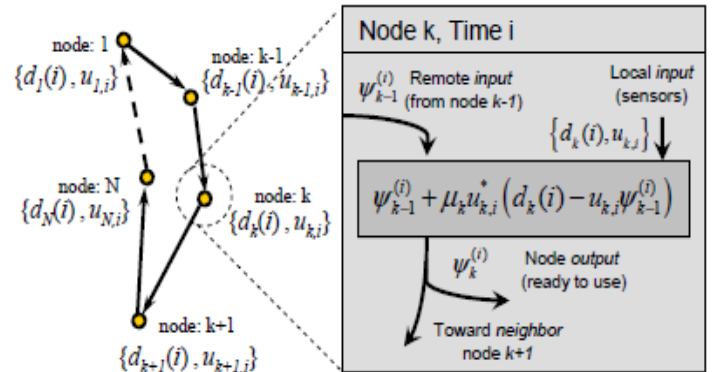


Fig 2. The Structure of Incremental LMS

3.2. EXACT DISTRIBUTED RLS ADAPTATION

We formulate in this section a least-squares solution for estimating the unknown parameter vector w° . At each time instant i , the network has access to the following space-time data:

$$y_i = \begin{bmatrix} d_1(i) \\ d_2(i) \\ \vdots \\ d_N(i) \end{bmatrix} \quad \text{and} \quad H_i = \begin{bmatrix} u_{1,i} \\ u_{2,i} \\ \vdots \\ u_{N,i} \end{bmatrix}.$$

We can then seek an estimate for w° by solving a regularized leastsquares problem of the form [4]:

$$\min_w \left[w^* \Pi w + \|Y_i - H_i w\|^2 \right]$$

where $\Pi > 0$ is a regularization matrix and Y_i and H_i collect all the data that are available up to time i :

$$Y_i = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_i \end{bmatrix} \quad \text{and} \quad H_i = \begin{bmatrix} H_0 \\ H_1 \\ \vdots \\ H_i \end{bmatrix}.$$

One could also incorporate weighting into (13) to account for spatial relevance, temporal relevance, and node relevance. Here we continue without weighting in order to convey the main idea. We are thus interested in deriving a distributed implementation of the least-squares solution. Some related work has been recently proposed where a global least-squares solution is achieved only approximately at each node, and the algorithm demands large communication and energy resources [5]. We proceed instead as follows. Assume that w_{i-1} is the solution to the following leastsquares (LS) problem using the data that are available up to time $i-1$:

$$\min_w \left[w^* \Pi w + \|\mathcal{Y}_{i-1} - \mathcal{H}_{i-1} w\|^2 \right].$$

We are interested in updating w_{i-1} to w_i by accounting for the incoming data blocks \mathcal{Y}_i and \mathcal{H}_i at time i . An algorithm that updates w_{i-1} incrementally is given by:

$$\begin{aligned} \psi_0^{(i)} &\leftarrow w_{i-1}; \quad P_{0,i} \leftarrow P_{i-1} \\ \text{for } k &= 1 : N \\ e_k(i) &= d_k(i) - u_{k,i} \psi_{k-1}^{(i)} \\ \psi_k^{(i)} &= \psi_{k-1}^{(i)} + \frac{P_{k-1,i} u_{k,i}^*}{1 + u_{k,i} P_{k-1,i} u_{k,i}^*} u_{k,i}^* e_k(i) \\ P_{k,i} &= P_{k-1,i} - \frac{P_{k-1,i} u_{k,i}^* u_{k,i} P_{k-1,i}}{1 + u_{k,i} P_{k-1,i} u_{k,i}^*} \\ \text{end} \\ w_i &\leftarrow \psi_N^{(i)}; \quad P_i \leftarrow P_{N,i}. \end{aligned}$$

3.3 LC- DISTRIBUTED RLS ADAPTATION

The algorithm proposed in the previous section implements exact RLS distributively, whereby the nodes share information about the local weight estimates $\{\psi_k^{(i)}\}$ and the matrices $\{P_{k,i}\}$. A less complex solution that only shares information about the weight estimates can be obtained by requiring the matrices $\{P_{k,i}\}$ to evolve locally. This strategy leads to:

$$\begin{aligned} \psi_0^{(i)} &\leftarrow w_{i-1} \\ \text{for } k &= 1 : N \\ e_k(i) &= d_k(i) - u_{k,i} \psi_{k-1}^{(i)} \\ \psi_k^{(i)} &= \psi_{k-1}^{(i)} + \frac{P_{k-1,i} u_{k,i}^*}{1 + u_{k,i} P_{k-1,i} u_{k,i}^*} u_{k,i}^* e_k(i) \\ P_{k,i} &= \left[P_{k-1,i} - \frac{P_{k-1,i} u_{k,i}^* u_{k,i} P_{k-1,i}}{1 + u_{k,i} P_{k-1,i} u_{k,i}^*} \right] \\ \text{end} \\ w_i &\leftarrow \psi_N^{(i)}. \end{aligned}$$

To illustrate the operation of both algorithms dRLS and its low communication counterpart (LC-dRLS), we consider a network with $N = 15$ nodes where each local filter has $M = 10$ taps. The system evolves for 30000 iterations and the results

are averaged over 100 independent experiments. The steady-state meansquare error values are obtained by averaging the last 500 time samples. Each node accesses time-correlated spatially independent Gaussian repressors' $u_{k,i}$ with correlation functions

$$\sigma_{u,k}^2 \cdot (\alpha_k)^{|i|}, \quad i = 0, \dots, M-1, \quad \text{with } \{\alpha_k\} \text{ and } \{\sigma_{u,k}^2\} \text{ randomly hosen in } [0; 1].$$

The background noise $w_k(i)$ has variance $\sigma_{w,k}^2 = 10^{-3}$ across the network.

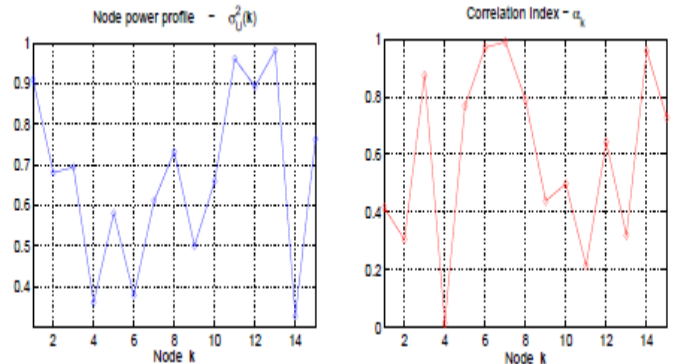


Fig 3. Network Statistical Profile

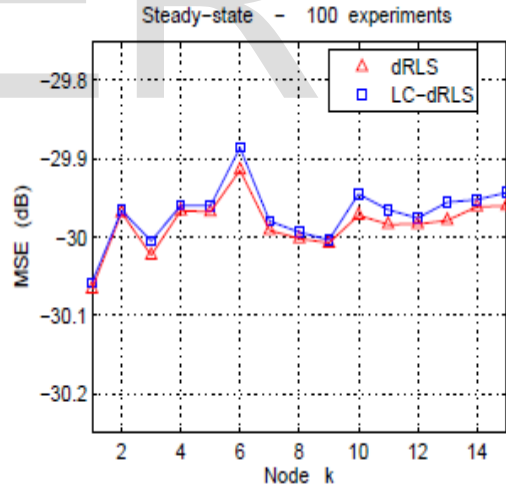


fig 4. MSE performance for both algorithms dRLS and LC-dRLS across the network

Both distributed RLS algorithms have similar performance in the mean-square error sense, suggesting that the low-communication implementation can be a quite competitive suboptimal implementation.

3.4 DIFFUSION LMS ADAPTATION

When more communication and energy resources are available, we may take advantage of the network connectivity and

devise more sophisticated peer-to-peer cooperation rules. We explore a simple diffusion protocol. Each individual node k consults its peer nodes from the neighborhood $\mathcal{N}_k(i-1)$ and combines their past estimates $\{\psi_l^{(i-1)}; l \in \mathcal{N}_k(i-1)\}$ with its own past estimate $\psi_k^{(i-1)}$. The node generates an aggregated estimate $\phi_k^{(i-1)}$ and feeds it in its local adaptive filter. This strategy can be expressed as follows:

$$\phi_k^{(i-1)} = f_k \left(\psi_\ell^{(i-1)}; \ell \in \mathcal{N}_k(i-1) \right)$$

$$\psi_k^{(i)} = \phi_k^{(i-1)} + \mu u_{k,i}^* \left(d_k(i) - u_{k,i} \phi_k^{(i-1)} \right)$$

for some combiner $f_k(\cdot)$

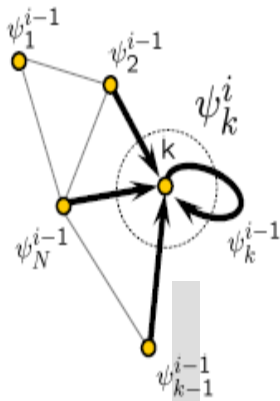


Fig 5. A Network with Diffusion cooperation Strategy

In this work we explore a simple combining rule in which the aggregated estimate is generated by averaging local and neighbours' previous estimates, i.e.

$$\phi_k^{(i-1)} = \sum_{\ell \in \mathcal{N}_k} a(k, \ell) \psi_\ell^{(i-1)}$$

$$\psi_k^{(i)} = \phi_k^{(i-1)} + \mu u_{k,i}^* \left(d_k(i) - u_{k,i} \phi_k^{(i-1)} \right)$$

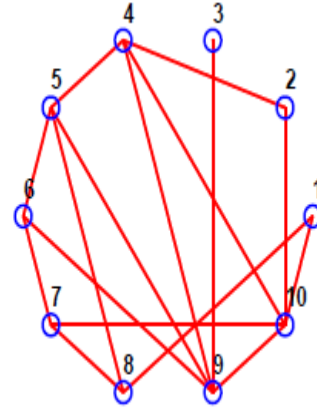


Fig 6. Network topology for the diffusion LMS example

4. Advantages and Disadvantages

Advantages:

1. Incremental growth: Computing power can be added in small increments.
2. Speed: A distributed system may have more total computing power than a mainframe.
3. Reliability: If one machine crashes, the system as a whole can still survive.
4. Performance: very often a collection of processors can provide higher performance than a centralized computer.

Disadvantages:

1. Troubleshooting: Troubleshooting and diagnosing problems in a distributed system can also become more difficult, because the analysis may require connecting to remote nodes or inspecting communication between nodes.
2. Networking: If the network gets saturated then problems with transmission will surface.
3. Software: There is currently very little less software support for Distributed system.

6. Conclusion

Thus, we concluded from the paper that by comparing several cooperative estimation strategies, the results show that diffusion networks are more stable than incremental networks. Moreover, the stability of diffusion networks is independent of the combination weights, whereas incremental networks can become unstable even if all individual nodes are stable.

7. Acknowledgments

It gives us a great pleasure to submit paper titled 'Diffusion Strategies Outperform Incremental Strategies Distributed Estimation over Adaptive Networks'. We wish to take this opportunity to express my heartiest gratitude with pleasure to J.D.I.E.T., Yavatmal, which gave me an opportunity in fulfilling my most cherished desire of reaching our goals. we are very much indebted to our HOD & Dean Academic, Dr.D.N.Chaudhari and

Principal, Dr. A.W. Kolhatkar, for providing all the facilities needed for successful completion of this paper. We are very much thankful to all Professors, Lecturers and staff members of Computer Science and Engineering Department. We express my deep gratitude to our parents and all my friends who directly and indirectly helped me to prepare for this Paper.

6. References

- [1] C. Lopes and A. H. Sayed, "Distributed Processing over Adaptive Networks," University of California, Los Angeles, Feb 2007.
- [2] D. Estrin, G. Pottie and M. Srivastava. "Instrumenting the world with wireless sensor networks," *Proc. IEEE ICASSP*, pp. 2033-2036, Salt Lake City, UT, May 2001.
- [3] D. Bertsekas, "A new class of incremental gradient methods for leastsquares problems," *SIAM J. Optim.*, vol.7, no. 4, pp. 913-926, Nov. 1997.
- [4] C. Lopes and A. H. Sayed, "Distributed adaptive incremental strategies: formulation and performance analysis," *Proc. ICASSP*, Toulouse, France, May 2006.
- [5] A. H. Sayed. *Fundamentals of Adaptive Filtering*, Wiley, NJ, 2003.
- [6] L. Xiao, S. Boyd and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," *Fourth IPSN*, Los Angeles, UCLA, Apr. 2005, pp. 63-70.