Signal Processing 90 (2010) 2621–2627

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Fast communication

Performance analysis of quantized incremental LMS algorithm for distributed adaptive estimation

Amir Rastegarnia *, Mohammad Ali Tinati, Azam Khalili

Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz 51664, Iran

ARTICLE INFO

Article history:
Received 10 December 2009
Received in revised form
1 February 2010
Accepted 16 February 2010
Available online 24 February 2010

Keywords:
Adaptive networks
Distributed estimation
Energy conservation
DILMS algorithm

ABSTRACT

Recently distributed adaptive estimation algorithms have been proposed as a solution to the issue of linear estimation over distributed networks. In all previous works, the performance of such algorithms is considered only for infinite-precision arithmetic implementation. In this paper we study the performance of distributed incremental least mean square (DILMS) estimation algorithm when it is implemented in finite-precision arithmetic. To this aim, we first derive the quantized version of the DILMS algorithm. Then a spatial–temporal energy conservation argument is used to derive theoretical expressions that evaluate the steady-state performance of individual nodes in the network. Simulation results show that there is a good match between the theory and simulation.

1. Introduction

Distributed estimation deals with the extraction of information from data collected at nodes that are distributed over a geographic area. The main objective is to obtain an estimate that is as accurate as the one that would be obtained if each node had access to the information across the entire network [1]. In many applications, it is necessary to perform estimation in a constantly changing environment without having available a (statistical) model for the underlying processes of interest. This motivates the development of distributed adaptive estimation schemes. In these schemes, every node is equipped with local computing ability to derive local estimate and share it with its predefined neighbors. Particular merits of a distributed adaptive network are its abilities to collaborate and adapt. The adaptive property enables the network to track not only the variations of the environment but also the topology of the network. On the other hand due to cooperative structure the computational burden is shared over the individual nodes so that communications are reduced as compared to a centralized network, and power and bandwidth usage are also thereby reduced [2]. Distributed adaptive networks (networks with distributed adaptive estimation algorithms) can therefore find potential application in a wide number of fields, such as precision agriculture, environmental monitoring and military.

In [1–4] distributed incremental adaptive estimation algorithms which have a cyclic pattern of cooperation are developed and their transient and steady-state performance analyses are also provided. DILMS and distributed recursive least-square (DRLS) are the examples of such algorithms. This scheme inherently requires a Hamiltonian cycle through which the estimates are sequentially circulated from sensor to sensor. When more communication and energy resources are available a diffusion cooperative scheme can be applied. In these schemes each node updates its estimate using all available estimates from the neighbors, as well as data and its own past estimate. In [5–7], diffusion implementations of distributed adaptive estimation algorithms are developed. In these algorithms, each node can communicate with all

* Corresponding author.
E-mail addresses: rastegar@tabrizu.ac.ir (A. Rastegarnia), tinati@tabrizu.ac.ir (M.A. Tinati), a-khalili@tabrizu.ac.ir (A. Khalili).

0165-1684/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2010.02.019

its neighbors as dictated by the network topology. Both LMS-based and RLS-based diffusion algorithms have been considered in the literature.

In all of the previous works distributed adaptive estimation algorithms have been assumed to use infinite-precision arithmetic. In this paper we study the performance of DILMS estimation algorithm when it is implemented in finite-precision arithmetic. The importance of such a study arises from the fact that the performance of distributed adaptive estimation algorithms (like DILMS) is strongly based on the adaptive filter which is used in their structure. On the other hand, the performance of adaptive filters can vary significantly when they are implemented in finite-precision arithmetic.

Throughout the paper, we adopt boldface letters for random quantities and normal font for nonrandom (deterministic) quantities. We also use capital letters for matrices and small letters for vectors. The * symbol is used for both complex conjugation for scalars and Hermitian transpose for matrices. The weighted norm for a vector is defined as $\|x\|^2_\Sigma \triangleq x^*\Sigma x$.

2. Estimation problem and the adaptive distributed solution

2.1. Incremental LMS solution

Consider a network composed of N distributed nodes. The purpose is to estimate an unknown parameter $M\times 1$ vector w^o from multiple spatially independent but possibly time-correlated measurements collected at N nodes in a network. Each node k has access to time-realizations $\{d_k(i), u_{k,i}\}$ of zero-mean spatial data $\{\boldsymbol{d}_k,\boldsymbol{u}_k\}$ where each $\boldsymbol{d}_k$ is a scalar measurement and each $\boldsymbol{u}_k$ is a $1\times M$ row regression vector. Collecting regression and measurement data into global matrices results

$$\boldsymbol{U}\triangleq\begin{bmatrix}\boldsymbol{u}_1\\ \boldsymbol{u}_2\\ \vdots\\ \boldsymbol{u}_N\end{bmatrix}_{N\times M},\quad \boldsymbol{d}\triangleq\begin{bmatrix}\boldsymbol{d}_1\\ \boldsymbol{d}_2\\ \vdots\\ \boldsymbol{d}_N\end{bmatrix}_{N\times 1} \tag{1}$$

The objective is to estimate the $M\times 1$ vector w that solves

$$\min_w J(w) \quad \text{where } J(w)=E\{\|\boldsymbol{d}-\boldsymbol{U}w\|^2\} \tag{2}$$

The optimal solution w^o satisfies the normal equations [2]

$$R_{du}=R_u w^o \tag{3}$$

where

$$R_{du}=E\{\boldsymbol{U}^*\boldsymbol{d}\} \quad \text{and} \quad R_u=E\{\boldsymbol{U}^*\boldsymbol{U}\} \tag{4}$$

Note that in order to use (3) to compute w^o each node must have access to the global statistical information $\{R_u, R_{du}\}$ which in turn, needs a lot of communication and computational resources. Moreover, such an approach, do not enable the network to response to changes in statistical properties of data. The DILMS algorithm is a solution to the above-mentioned problems [2,3]. The update equation in DILMS is given by

$$\boldsymbol{\psi}_{k,i}=\boldsymbol{\psi}_{k-1,i}-\mu\boldsymbol{u}^*_{k,i}[\boldsymbol{d}_k(i)-\boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k-1,i}] \tag{5}$$

where $\boldsymbol{\psi}_{k,i}$ denotes the local estimate of w^o at node k at time i and μ is the step size. The calculated estimates are sequentially circulated from node to node. Note that to implement the DILMS, the time realizations $\{d_k(i), u_{k,i}\}$ are used.

2.2. Quantized version of DILMS

The DILMS algorithm can be implemented in finite-precision at every node k as shown in Fig. 1, where the $\mathcal{Q}(x)$ block denotes the fixed-point quantization of x, $x^q=\mathcal{Q}(x)$ is the quantized value of x and $\hat{x}=x-x^q$ is the quantization error. Let n_r and L_r denote the number of bits and the saturation level of quantization for x respectively, where x could be a scalar or entries of a vector. Then, for real-valued data, the variance of quantization error is [8]

$$\sigma_r^2=\frac{1}{12}\frac{L_r^2}{2^{n_r}} \tag{6}$$

As it is shown in Fig. 1, the quantized DILMS algorithm operates on $\{\boldsymbol{d}^q_k(i),\boldsymbol{u}^q_{k,i},\boldsymbol{z}^q_k(i),\boldsymbol{\psi}^q_{k,i}\}$, which are related to their unquantized quantities via

$$\boldsymbol{d}_k(i)=\boldsymbol{d}^q_k(i)+\hat{\boldsymbol{d}}_k(i),\quad \boldsymbol{z}_k(i)\triangleq\boldsymbol{u}_{k,i}\boldsymbol{\psi}_{k-1,i}=\boldsymbol{z}^q_k(i)+\hat{\boldsymbol{z}}_k(i)$$

$$\boldsymbol{u}_{k,i}=\boldsymbol{u}^q_{k,i}+\hat{\boldsymbol{u}}_{k,i},\quad \boldsymbol{\psi}_{k,i}=\boldsymbol{\psi}^q_{k,i}+\hat{\boldsymbol{\psi}}_{k,i} \tag{7}$$

where $\{\hat{\boldsymbol{d}}_k(i),\hat{\boldsymbol{u}}_{k,i},\hat{\boldsymbol{z}}_k(i),\hat{\boldsymbol{\psi}}_{k,i}\}$ are the corresponding quantization errors. Using (6) we can conclude that the $\{\hat{\boldsymbol{d}}_k(i), \hat{\boldsymbol{z}}_k(i)\}$ have variance σ_r^2 and $\hat{\boldsymbol{u}}_{k,i}$ has covariance matrix $\sigma_r^2 I_M$. Note that we can use different number of bits to quantize different quantities like $\boldsymbol{\psi}_{k,i}$. Now, according to Fig. 1 we have

$$\begin{aligned}\boldsymbol{\psi}^q_{k,i}&=\mathcal{Q}(\boldsymbol{\psi}^q_{k-1,i}+\mathcal{Q}(\mu\boldsymbol{u}^{q*}_{k,i}\boldsymbol{e}^q_k(i)))\overset{(a)}{=}\boldsymbol{\psi}^q_{k-1,i}+\mu\mathcal{Q}(\boldsymbol{u}^{q*}_{k,i}\boldsymbol{e}^q_k(i))\\ &\quad-\boldsymbol{\alpha}_{k,i}=\boldsymbol{\psi}^q_{k-1,i}+\mu\boldsymbol{u}^{q*}_{k,i}\boldsymbol{e}^q_k(i)-\boldsymbol{\alpha}_{k,i}-\boldsymbol{\beta}_{k,i}\end{aligned} \tag{8}$$

where (a) is valid when μ is chosen as a power of 2^{-1}. In (8) $\boldsymbol{\alpha}_{k,i}$ and $\boldsymbol{\beta}_{k,i}$ are the quantization errors in the evaluation of corresponding $\mathcal{Q}(x)$ and both have covariance matrix $\sigma_r^2 I_M$. Substituting $\boldsymbol{u}^q_{k,i}$ from (7) into (8) and expanding, we find that the quantized version of DILMS can be obtained as

$$\boldsymbol{\psi}^q_{k,i}=\boldsymbol{\psi}^q_{k-1,i}+\mu\boldsymbol{u}^*_{k,i}\boldsymbol{e}^q_k(i)-\boldsymbol{p}_{k,i} \tag{9}$$

where

$$\boldsymbol{p}_{k,i}=\boldsymbol{\alpha}_{k,i}+\boldsymbol{\beta}_{k,i}+\mu\hat{\boldsymbol{u}}^*_{k,i}\boldsymbol{e}^q_k(i) \tag{10}$$

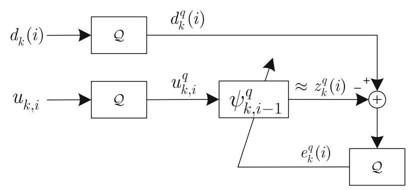

Fig. 1. A block diagram representation of quantized implementation of DILMS algorithm in node k.