Int. J. Electron. Commun. (AEÜ) 68 (2014) 97–110

Contents lists available at ScienceDirect

International Journal of Electronics and Communications (AEÜ)

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

Low computational complexity family of affine projection algorithms over adaptive distributed incremental networks

Mohammad Shams Esfand Abadi*, Ali-Reza Danaee[1]

Faculty of Electrical and Computer Engineering, Shahid Rajaee Teacher Training University, P.O. Box 16785-163, Tehran, Iran

ARTICLE INFO

Article history:
Received 26 January 2013
Accepted 5 July 2013

Keywords:
Affine projection algorithm
Adaptive distributed estimation
Incremental network
Selective partial update
Dynamic selection
Mean-square performance

ABSTRACT

This paper presents the problem of distributed estimation in an incremental network based on the family of affine projection (AP) adaptive algorithms. The distributed selective partial update normalized least mean squares (dSPU-NLMS), the distributed SPU-AP algorithm (dSPU-APA), the distributed selective regressor APA (dSR-APA), the distributed dynamic selection of APA (dDS-APA), dSPU-SR-APA and dSPU-DS-APA are introduced in a unified way. These algorithms have low computational complexity feature and close convergence speed to ordinary distributed adaptive algorithms. In dSPU-NLMS and dSPU-APA, the weight coefficients are partially updated at each node during the adaptation. In dSR-APA, the optimum number of input regressors is selected during the weight coefficients update. The dynamic selection of input regressors is used in dDS-APA. dSPU-SR-APA and dSPU-DS-APA combine SPU with SR and DS approaches. In these algorithms, the weight coefficients are partially updated and the input regressors are optimally/dynamically selected at every iteration for each node. In addition, a unified approach for mean-square performance analysis of each individual node is presented. This approach can be used to establish a performance analysis of classical distributed adaptive algorithms as well. The theoretical expressions for stability bounds, transient, and steady-state performance analysis of various distributed APAs are introduced. The validity of the theoretical results and the good performance of dAPAs are demonstrated by several computer simulations.

1. Introduction

One of the important features of sensor networks is the cooperative attempt of sensor nodes. Sensor nodes use their processing abilities to locally perform simple computations and transfer only the required and partially processed data. Some applications of sensor networks include environment monitoring, target localization, military surveillance, transportation, as well as medical research [1,2].

In contrast to classical centralized methods, distributed processing uses local computations at each node and communications among neighboring nodes to solve problems over the entire network which is a practical use among a wide range of applications. Because of data statistical variations and network topology changes, the processing should be adaptive and requires two time scales in distributed networks. During the initial period of time, each node makes a special estimation and then through consensus iterations, the nodes incorporate further estimations to attain the desired estimate. In incremental learning algorithms over a distributed network, each node cooperates only with one adjacent node to utilize the spatial dimension, whilst doing local computations in the time dimension [3]. Within the last decade, different incremental adaptive strategies such as distributed least mean squares (dLMS), distributed normalized LMS (dNLMS), and distributed affine projection algorithm (dAPA) were introduced over distributed networks [3,4]. Unfortunately, in comparison with dLMS and dNLMS, the computational complexity of dAPA is high.

In some of applications such as adaptive distributed networks, a high computational load is needed to achieve an acceptable performance. Therefore the computational complexity is an important problem in these applications. Several single adaptive filter algorithms such as adaptive filter algorithms with selective partial updates have been proposed to reduce the computational complexity. These algorithms update only a subset of filter coefficients at each time iteration. The Max-NLMS [5], the N-Max NLMS [6,7] (N is the number of filter coefficients to update), and the variants of the selective partial update normalized least mean square (SPU-NLMS) [8,9] are important examples of this family of adaptive filter algorithms. The SPU approach was also successfully extended to APA in [8].

In the case of APA, other algorithms such as selective regressor APA (SR-APA) and dynamic selection of APA (DS-APA) were

* Corresponding author. Tel.: +98 21 22970003; fax: +98 21 22970003.
E-mail addresses: mshams@srttu.edu (M.S.E. Abadi), ar.danaee@srttu.edu (A.-R. Danaee).
[1] Tel.: +98 21 22970003; fax: +98 21 22970003.

1434-8411/$ – see front matter © 2013 Elsevier GmbH. All rights reserved.
http://dx.doi.org/10.1016/j.aeue.2013.07.004

proposed to reduce the computational complexity [10,11]. In SR-APA, the input regressors are optimally selected at each iteration. The dynamic selection of input regressors during the filter coefficients update were introduced in [11]. Also by combination of SPU and SR approaches, SPU-SR-APA was suggested in [12]. In comparison with APA, the introduced algorithms are good convergence speed, low steady-state error, and low computational complexity.

In the present study, we therefore focus on a new class of adaptive algorithms-based distributed learning mechanism in incremental networks. The distributed SPU-APA (dSPU-APA), the distributed SR-APA (dSR-APA) and the distributed DS-APA (dDS-APA) are introduced in a unified way. Also, dSPU-SR-APA, and dSPU-DS-APA are presented by combining SPU and DS approaches. These algorithms have good convergence speed, low steady-state error, and low computationally complexity features. In the following, a unified approach to performance analysis of the spatial and temporal family of affine projection algorithms is presented and closed-form expressions for the transient and steady-state performances of each node are established. Moreover, the theoretical expressions for mean and mean-square stability of dAPAs are introduced. The analysis is based on spatial-temporal energy conservation relation which incorporates the space–time structure of the data. For the classical dAPA, this anslysis was presented in [4]. But the present analysis is the unified formalism for transient performance of various adaptive distributed networks. The general expressions for steady-state MSE and MSD are established,

What we propose in this paper can be summarized as follows:

- The establishment of the family of dAPAs. In comparison with classical dAPA in [4], the introduced dAPAs have close performance to dAPA with low computational complexity feature. In dSPU-APA, and dSPU-NLMS, the coefficients are partially updated at each node for every iteration. In dSR-APA, and dDS-APA, the number of recent regressors are optimally selected. By combination of SPU and DS approaches, the dSPU-SR-APA and dSPU-DS-APA are established. In these algorithms the coefficients are partially updated and the number of recent regressors are optimally selected at each node for every iteration.
- Mean-square performance analysis of the family of dAPAs in a unified way. The general theoretical expressions for transient and steady-state performance of the network are obtained. The presented expressions can also be used to predict the performance of dLMS, dNLMS, and dAPA.
- Analysis of the mean and mean-square stability bounds for the family of dAPAs.
- Demonstrating the presented algorithms in an incremental network. The performance of dNLMS, dAPA, dSPU-NLMS, dSPU-APA, dSR-APA, dDS-APA, dSPU-SR-APA, and dSPU-DS-APA are compared and analyzed. The validity of the theoretical relations is also justified in both transient and steady-state.

This paper is organized as follows. In Section 2, the generic adaptive weight coefficients update equation in distributed network is introduced. Section 3 describes the establishment of classical adaptive distributed networks based on the generic update equation. The family of dAPAs is presented in Section 4. In the next section, the mean-square performance analysis of each node is explained. Section 6 presents the stability bounds of step-size for various dAPAs. The computational complexity of dAPAs is discussed in Section 7. Finally, the paper ends with a comprehensive set of simulations supporting the validity of the results.

Throughout the paper, the following notations are used:

$\lvert . \rvert$	Norm of a scalar.
$\Vert . \Vert^2$	Squared Euclidean norm of a vector.
$\Vert \mathbf{t} \Vert^2_{\mathbf{\Sigma}}$	$\mathbf{\Sigma}$-Weighted Euclidean norm of a column vector $\mathbf{t}$ defined as $\mathbf{t}^T\mathbf{\Sigma}\mathbf{t}$.
vec($\mathbf{T}$)	Creates an $M^2 \times 1$ column vector $\mathbf{t}$ through stacking the columns of the $M \times M$ matrix $\mathbf{T}$.
vec($\mathbf{t}$)	Creates an $M \times M$ matrix $\mathbf{T}$ from the $M^2 \times 1$ column vector $\mathbf{t}$.
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$.
Tr(.)	Trace of a matrix.
$(.)^{-1}$	Inverse of a square matrix.
$(.)^T$	Transpose of a vector or a matrix.
diag{.}	Has the same meaning as the MATLAB operator with the same name: If its argument is a vector, a diagonal matrix with the diagonal elements given by the vector argument results. If the argument is a matrix, its diagonal is extracted into a resulting vector.
$E\{\cdot\}$	Expectation operator.

2. The generic adaptive weight update equation

Due to special features of family of affine projection algorithms such as high convergence speed, low computational complexity and low steady state mean square error, we apply different APAs for a distributed estimation in an incremental network. The classical adaptive distributed algorithms, dSPU-NLMS, dSPU-APA, dSR-APA, dDS-APA, dSPU-SR-APA, and dSPU-DS-APA can be established based on a unified formalism. By selecting the proper matrices and parameters in this approach, various dAPAs are obtained. This new proposition will be useful to show the differences and similarities of dAPAs and to analyze the performance of dAPAs in a unified way.

The weight vector update equation for the family of dAPAs in an incremental network over J nodes is introduced as

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \sum_{k=1}^{J} \mu_k \mathbf{C}_k(n)\mathbf{X}_k(n)\mathbf{Y}_k(n)\mathbf{e}_k(n), \quad (1)$$

where

$$\mathbf{e}_k(n) = \mathbf{d}_k(n) - \mathbf{X}_k^T(n)\mathbf{h}_{k-1}(n), \quad (2)$$

is the error vector, and μ_k is the step-size of each node. In (2), $\mathbf{X}_k(n) = [\mathbf{x}_k(n), \mathbf{x}_k(n-1), \ldots, \mathbf{x}_k(n-(P-1))]$ is the $M \times P$ block data signal matrix and $\mathbf{d}_k(n) = [d_k(n), d_k(n-1), \ldots, d_k(n-(P-1))]^T$ is the $P \times 1$ data vector where $\mathbf{x}_k(n) = [x_k(n), x_k(n-1), \ldots, x_k(n-M+1)]^T$. We assume the linear data model between the unknown system vector $\mathbf{h}_t$, $\mathbf{d}_k(n)$, and $\mathbf{X}_k(n)$ as $\mathbf{d}_k(n) = \mathbf{X}_k^T(n)\mathbf{h}_t + \mathbf{v}_k(n)$, where $\mathbf{v}_k(n)$ is a temporally and spatially independent noise sequence with variance $\sigma^2_{v,k}$ at each node. By selecting matrices $\mathbf{C}_k(n)$ and $\mathbf{Y}_k(n)$ from Table 1, the family of distributed affine projection adaptive algorithms such as dAPA, dSPU-APA, dSR-APA, dDS-APA, dSPU-SR-APA, and dSPU-DS-APA as well as classical adaptive distributed algorithms can be established. Fig. 1 shows the learning algorithm in an incremental network for the family of dAPAs [4]. Based on the local data ($\mathbf{d}_k(n)$, $\mathbf{X}_k(n)$), and $\mathbf{h}_{k-1}(n)$ from its previous node $(k-1)$ in the cycle, the update for local estimation is performed at each time instant n. Finally, the local estimation $\mathbf{h}_J(n)$ is used as the global estimation for $\mathbf{w}(n)$. This value is used as the initial local estimation $\mathbf{h}_1(n+1)$ for the next time instant $n+1$. The same as [4], the pseudo-code implementation of dAPAs has been described in Table 2.

3. Derivation of classical distributed adaptive algorithms

We can now make specific choices for the matrices $\mathbf{C}_k(n)$ and $\mathbf{Y}_k(n)$ as well as for the parameter P. Different adaptive distributed algorithms can be viewed as specific instantiations of the generic adaptive weight update equation. By setting the parameter P equal to 1, and substituting the matrices $\mathbf{C}_k(n)$ and $\mathbf{Y}_k(n)$ from Table 1 in the generic adaptive weight update equation, dLMS, dNLMS, dAPA,