Signal Processing 142 (2018) 423–430

Contents lists available at ScienceDirect

Signal Processing

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

Diffusion least logarithmic absolute difference algorithm for distributed estimation☆

Feng Chen[a,b,c], Tao Shi[a,c], Shukai Duan[a,*], Lidan Wang[a], Jiagui Wu[a,c]

[a] College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China
[b] School of Mathematics and Statistics, Southwest University, Chongqing 400715, China
[c] Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Chongqing 400715, China

ARTICLE INFO

Article history:
Received 23 November 2016
Revised 26 April 2017
Accepted 17 July 2017
Available online 18 July 2017

Keywords:
Impulsive interference
Distributed estimation
Least logarithmic absolute difference

ABSTRACT

The popular distributed estimation algorithms based on the mean-square error criterion is not robust against impulsive noise in the adaptive networks. To address the problem, the diffusion least logarithmic absolute difference (LLAD) algorithm is proposed in this article, which adopts both the logarithm operation and sign operation to the error. The algorithm can elegantly and gradually adjust the conventional cost functions in its optimization based on the error variation. Compared with centralized LLAD algorithm, the diffusion LLAD algorithm performs a good balance between communications and performance. The theoretical stability of mean and mean-square performance of the algorithm is analyzed. Simulation results indicate that the algorithm achieves a better performance, compared with diffusion LMS and diffusion sign-error LMS algorithms, even in the impulsive noise environment.

1. Introduction

Noisy signal is assumed to be Gaussian in many signal processing applications, but impulsive interferences widely exist in most practical contexts. Such as underwater acoustics [1,2], traffic prediction [4], multiuser detection [5,6] and broad-band powerline communications [7]. Adaptive filtering algorithm minimizing the mean-square criterion may suffer from severely degraded convergence performance, when the noise involves impulsive noise [3,8,9,24,25].

In recent studies, it is found that the diffusion sign-error LMS (DSE-LMS) [9] and the diffusion sign subband adaptive filtering (DSSAF) [20] are robust against impulse noise. The merit of DSSAF algorithm has a lower computational complexity than diffusion LMS algorithm (DLMS) [17].

In our work, the diffusion LLAD algorithm is presented for distributed estimation over adaptive network in which each node only communicates with its neighbors at each iteration. The proposed algorithm based on least logarithmic absolute difference cost function is resistant to impulsive interference [3]. By using such a relative cost measure, we can effectively adjust the conventional cost functions in its optimization based on the error variation. The diffusion LLAD algorithm achieves a better performance than DSE-LMS and DLMS in impulsive noise environment. Compared with centralized LLAD algorithm, the diffusion LLAD algorithm performs a good balance between communications and performance. In the theoretical analysis, we study the stability and convergence of our algorithm. Our simulation results demonstrate that the diffusion LLAD algorithm is robust against impulse interference than diffusion LMS and diffusion sign-error LMS algorithms.

The paper is organized as follows. In Section 2, the diffusion LLAD algorithm is formulated. The theoretical stability of the diffusion LLAD algorithm is analyzed in Section 3. Simulation results are performed to evaluate the proposed algorithm in Section 4. Finally, we draw conclusions in Section 5.

Notation: In this paper, $(.)^T$, sign(.), and $\otimes$ denote the transpose, sign and Kronecker product operators, respectively. $(.)_0$ denotes a real value of parameter. col{.} denotes a column vector. Here, {.} stands for a set. |.| and abs(.) stand for the absolute value of a scalar. Tr{.} calculates the trace of a matrix.

☆ This work was supported in party by the Fundamental Research Funds for the Central Universities (Grant Nos. XDJK2017B053, XDJK2017D176, XDJK2017D180, XDJK2017D181), Chongqing Research Program of Basic Research and Frontier Technology (No.cstc2017jcyjAX0265) and program for New Century Excellent Talents in University (Grant Nos. [2013] 47).

* Corresponding author.
E-mail address: duansk@swu.edu.cn (S. Duan).

http://dx.doi.org/10.1016/j.sigpro.2017.07.014
0165-1684/

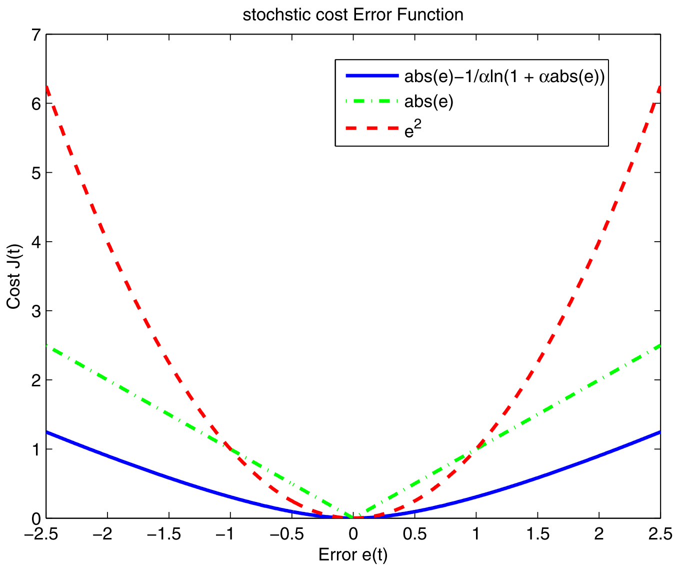

Fig. 1. Stochastic cost functions.

2. The estimate problem

2.1. Least logarithm absolute difference algorithm

At each time i, we observe an unknown vector $\mathbf{w}_0$ of length M through a linear system

$$d_i = \mathbf{u}_i^T \mathbf{w}_0 + n(i) \tag{1}$$

where d_i is the output random scalar and $\mathbf{u}_i$ is the length $1 \times M$. We define the error signal as $e_i = d_i - \mathbf{u}_i^T \mathbf{w}_i$ where $\mathbf{w}_i$ is the weight vector, and $n(i)$ denotes the background noise.

Given the data model, the objective of the least logarithmic absolute difference method [3] is minimizing the perturbations in the output data, which can be formulated as

$$J(e_i) = F(e_i) - \frac{1}{\alpha} ln(1 + \alpha F(e_i)) \tag{2}$$

where α is a small systemic parameter and $F(e_i) = E[|e_i|]$.

We compare different stochastic cost functions including the absolute cost function $|e_i|$, the mean-square cost function $[e_i]^2$ and the introduced LLAD cost function (2) in Fig. 1.

We observe that the logarithm based cost function is less steep for small perturbations on the error while both the logarithmic cost and absolute difference cost functions exhibit comparable steepness for relatively larger error values.

2.2. Diffusion least logarithmic absolute difference algorithm

We consider a network composed of $K \in \mathbb{N}$ nodes where each node measures its data $\{\mathbf{u}_{k,\,i}, d_{k,\,i}\}$ to estimate a deterministic $\mathbf{w}_0$ of length M vector through a linear model:

$$d_{k,i} = \mathbf{u}_{k,i}^T \mathbf{w}_0 + n_k(i). \tag{3}$$

Here, $k \in \{1, 2, \ldots, K\}$ is node index and i is the time index. The random process $d_{k,\,i}$ is a scalar, and the input vector $\mathbf{u}_{k,\,i}$ is spatially independent with other regression vectors. The noise $n_k(i)$ is a mixture signal of zero-mean Gaussians.

Given the data model, all nodes are observing data arising from the common parameter, it is natural to expect cooperation among the nodes to be beneficial over adaptive network. It means that neighboring nodes can share information with each other as permitted by the network topology. Therefore, we have defined the global cost function that enable all nodes to carry out adaptation in a cooperative manner. The global cost function of LLAD method based on references [11,17] can be formulated as:

$$f^{glob}(\mathbf{w}) = \sum_{l=1}^{K} \left(E\left[\left|d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}\right|\right] - \frac{1}{\alpha} ln\left(1 + \alpha E\left[\left|d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}\right|\right]\right)\right) \tag{4}$$

the centralized LLAD algorithm is based on global cost function, which is not distributed. We focus on the distributed solution over network in this paper. To develop the distributed LLAD algorithm, we define the local objective function for each node k based on references [11,17,18,27]. This local cost function can be formulated as a liner combination of local weighted least logarithmic absolute difference [3].

$$f_k^{loc}(\mathbf{w}) = \sum_{l \in \mathbf{N}_k} c_{l,k} \left(E\left[\left|d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}\right|\right] - \frac{1}{\alpha} ln\left(1 + \alpha E\left[\left|d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}\right|\right]\right)\right) \tag{5}$$

where $\mathbf{N}_k$ denotes the neighbor set of node k (including k itself), and the coefficients $\{c_{l,\,k}\}$ are non-negative real weights satisfying

$$\sum_{l \in \mathbf{N}_k} c_{l,k} = 1, \text{ and } c_{l,k} = 0 \text{ for all } l \notin \mathbf{N}_k. \tag{6}$$

We consider minimizing the local objective function (5) using the steepest descent method [26].

$$\mathbf{w}_{k,i} = \mathbf{w}_{k,i-1} - \mu_k \Delta \mathbf{w}_{k,i} \tag{7}$$

where μ_k is the step size. $\mathbf{w}_{k,\,i}$ is the local estimate obtained by node k at time i, and $\Delta(.)$ denotes the increment of computation, namely

$$\Delta w_{k,i} = \frac{\partial f_k^{loc}(\mathbf{w})}{\partial w} = \sum_{l \in \mathbf{N}_k} c_{l,k} f_l^{cg}(\mathbf{w})|_{w_{k,i-1}} \tag{8}$$

where

$$f_l^{cg}(\mathbf{w}) = -\frac{\alpha E\left[\mathbf{u}_{l,i}\left(d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}\right)\right]}{1 + \alpha E\left[\left|d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}\right|\right]}. \tag{9}$$

Eq. (9) requires the knowledge of the second moments, we replace the second moments by local instantaneous approximations, namely, $E[d_{l,\,i}\mathbf{u}_{l,\,i}] \approx [d_{l,\,i}\mathbf{u}_{l,\,i}]$, $E\left[\mathbf{u}_{l,i}\mathbf{u}_{l,i}^T\right] \approx \left[\mathbf{u}_{l,i}\mathbf{u}_{l,i}^T\right]$, $E\left|d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}\right| \approx \left|d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}\right|$. It is noted that removing the expectation of (9) obtains the stochastic gradient approximations of $f_l^{cg}(\mathbf{w})$ [3,29].

$$f_l^{ii}(\mathbf{w}) = -\mathbf{u}_{l,i} \frac{\alpha\left(d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}\right)}{1 + \alpha\left|d_{l,i} - \mathbf{u}_{l,i}^T \mathbf{w}\right|}. \tag{10}$$

From Eqs. (9) and (10), it is found that the gradient error is induced by this replacement [15].

Given (7), (8) and (10), we obtain the updated equation of the parameter vector $\mathbf{w}_{k,\,i}$. However, transmitting the original data $\{\mathbf{u}_{k,\,i}, d_{k,\,i}\}$ of each node k is a considerable communication load. In order to solve this problem and get more information, the local estimate $\mathbf{w}_{k,\,i}$ is assumed to be a linear combination of the intermediate estimates $\boldsymbol{\psi}_{k,\,i}$ in distributed estimation literatures [10,11,17,28,29].

$$\mathbf{w}_{k,i} = \sum_{l \in \mathbf{N}_k} c_{l,k} \psi_{l,i}. \tag{11}$$

In this way, $\Delta \mathbf{w}_{k,\,i}$ can be written as

$$\Delta \mathbf{w}_{k,i} = \sum_{l \in \mathbf{N}_k} c_{l,k} \Delta \psi_{l,i} \tag{12}$$