Signal Processing 101 (2014) 1–10

Contents lists available at ScienceDirect

Signal Processing

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

A fast exact filtering approach to a family of affine projection-type algorithms

Feiran Yang, Ming Wu, Jun Yang*, Zheng Kuang

Key Laboratory of Noise and Vibration Research, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 16 October 2013
Received in revised form
6 January 2014
Accepted 30 January 2014
Available online 7 February 2014

Keywords:
Adaptive filter
Affine projection
Fast exact filtering
Low complexity

ABSTRACT

The affine projection (AP)-type algorithms produce a good tradeoff between convergence speed and complexity. As the projection order increases, the convergence rate of the AP algorithm is improved at a relatively high complexity. Many efforts have been made to reduce the complexity. However, most of the efficient versions of the AP-type algorithms are based on the fast approximate filtering (FAF) scheme originally proposed in the fast AP (FAP) algorithm. The approximation leads to degraded convergence performance. Recently, a fast exact filtering (FEF) AP (FEAP) algorithm was proposed by Y. Zakharov. In this paper, we propose a new FEF approach to further reduce the complexity of the FEAP algorithm given that the calculation of the weight vector is not the primary objective for the application at hand. The proposed FEF scheme is then extended to the dichotomous coordinate descent (DCD)-AP, affine projection sign (APS), and modified filtered-x affine projection (MFxAP) algorithms. The complexity of AP-type algorithms based on the proposed FEF approach is comparable to that based on the FAF scheme. Moreover, analysis results show that the complexity reduction of the new algorithms is achieved without any performance degradation.

1. Introduction

In adaptive filtering, the least-mean-square (LMS)-type algorithms are widely used but suffer from slow convergence for colored signals. The affine projection (AP) algorithm [1] was proposed to speed up the convergence, which produces a good tradeoff between the convergence speed and the complexity. Due to the good properties, several variants of the AP algorithm have been developed in the context of blind multiuser detection [2], acoustic echo cancellation (AEC) [3,4], active noise control (ANC) [5], and acoustic feedback cancellation (AFC) [6].

When the projection order P increases, the convergence rate of the AP algorithm is improved at the price of a considerable rise of the computational complexity. Many efforts have been made to reduce the complexity of the AP algorithm [3–20].

The complexity of the direct calculation of the error vector is proportional to the projection order. The fast AP (FAP) [3,4] algorithm and its variants [5–16] present a fast approximate filtering (FAF) scheme to reduce the complexity. Since the FAP algorithm is based on an implicit "small regularization parameter" assumption, the FAP algorithm is not exactly equal to the standard AP algorithm [17]. Most of the existing fast AP-type algorithms [3–16] are based on the FAF approach. The FAF approach reduces the complexity efficiently but also leads to degraded performance.

To overcome this limitation, a fast exact filtering (FEF) approach to the AP algorithm (FEAP) [18] was presented by Y. Zakharov. However, in the FEAP algorithm, calculation of the error vector requires the update of the weight vector explicitly that provides the largest contribution towards the

* Corresponding author.
E-mail address: jyang@mail.ioa.ac.cn (J. Yang).

http://dx.doi.org/10.1016/j.sigpro.2014.01.030
0165-1684

Table 1
FAP algorithm [3,4].

Equation	×	+
$\mathbf{R}(n)=\mathbf{X}^T(n)\mathbf{X}(n)$	$2P$	$2P$
$\boldsymbol{\alpha}(n)=\mathbf{X}^T(n-1)\mathbf{x}(n)$	0	0
$e(n)=d(n)-\mathbf{x}^T(n)\hat{\mathbf{w}}(n-2)-\boldsymbol{\alpha}^T(n)\boldsymbol{\varphi}(n-1)$	$L+P$	$L+P$
$\mathbf{e}(n)=\begin{bmatrix} e(n) \\ (1-\mu)\overline{\mathbf{e}}(n-1)\end{bmatrix}$	$P-1$	0
$\boldsymbol{\varepsilon}(n)=\mu[\mathbf{R}(n)+\delta\mathbf{I}]^{-1}\mathbf{e}(n)$	P_m	P_a
$\boldsymbol{\varphi}(n)=\boldsymbol{\varepsilon}(n)+\begin{bmatrix} 0 \\ \overline{\boldsymbol{\varphi}}(n-1)\end{bmatrix}$	0	$P-1$
$\hat{\mathbf{w}}(n)=\hat{\mathbf{w}}(n-1)+\mathbf{x}(n-P+1)\varphi_{P-1}(n)$	L	L
Total $2L+4P+P_m$ multiplications $2L+4P+P_a$ additions		

algorithm complexity. Many AP-type algorithms [19–21] adopt the FEF approach to reduce the complexity and have a similar problem. In many applications such as AEC and ANC, calculation of the weight vector is not the main concern [3–15]. In this paper, we will extend the work in [18] and propose an enhanced FEF approach to the AP (EFEAP) algorithm. The complexity of the proposed EFEAP algorithm is comparable to that of the FAP algorithm. We then extend the proposed FEF approach to a family of AP-type algorithms such as the dichotomous coordinate descent (DCD)-AP [18], affine projection sign (APS) [22], and modified filtered-x affine projection (MFxAP) algorithms [14,15]. Computer simulations demonstrate the effectiveness of the proposed approach.

Notations: throughout this paper, we use uppercase and lowercase bold fonts to denote matrices and vectors, respectively, e.g., $\mathbf{R}$ and $\mathbf{r}$. Superscript T denotes the transpose operator, and $\mathbf{I}$ is the $P\times P$ identity matrix.

2. Proposed FEF approach to the AP algorithm

Consider the desired response $d(n)$ arising from the linear model

$$d(n)=\mathbf{w}_o^T\mathbf{x}(n)+v(n) \tag{1}$$

where $\mathbf{w}_o=[w_0,w_1,\ldots,w_{L-1}]^T$ is the L-length weight vector of the unknown system, $\mathbf{x}(n)=[x(n),x(n-1),\ldots,x(n-L+1)]^T$ denotes the input signal vector, and $v(n)$ represents the system noise.

The adaptive weight vector is $\mathbf{w}(n)=[w_0(n),w_1(n),\ldots,w_{L-1}(n)]^T$. To describe the AP algorithm, we define the input signal, the desired signal, the filtered-out, and the error vectors as follows:

$$\mathbf{X}(n)=[\mathbf{x}(n),\mathbf{x}(n-1),\ldots,\mathbf{x}(n-P+1)] \tag{2}$$

$$\mathbf{d}(n)=[d(n),d(n-1),\ldots,d(n-P+1)]^T \tag{3}$$

$$\mathbf{y}(n)=[y_0(n),y_1(n),\ldots,y_{P-1}(n)]^T =\mathbf{X}^T(n)\mathbf{w}(n-1) \tag{4}$$

$$\mathbf{e}(n)=[e_0(n),e_1(n),\ldots,e_{P-1}(n)]^T =\mathbf{d}(n)-\mathbf{y}(n). \tag{5}$$

The update equation of the AP algorithm is

$$\boldsymbol{\varepsilon}(n)=[\varepsilon_0(n),\varepsilon_1(n),\ldots,\varepsilon_{P-1}(n)]^T =\mu[\mathbf{X}^T(n)\mathbf{X}(n)+\delta\mathbf{I}]^{-1}\mathbf{e}(n) \tag{6}$$

$$\mathbf{w}(n)=\mathbf{w}(n-1)+\mathbf{X}(n)\boldsymbol{\varepsilon}(n) \tag{7}$$

where μ is the step size, and δ is a regularization parameter.

The complexity of the AP algorithm is mainly due to the following three operations: (i) calculation of the filtered-out vector $\mathbf{y}(n)$ in (4), (ii) update of the weight vector $\mathbf{w}(n)$ in (7), and (iii) the matrix inversion operation in (6). For a direct implementation, the first two steps need $2PL$ operations per sample, which is very expensive especially for a long impulse response. We now briefly review the state-of-the-art fast filtering approaches.

2.1. FAF approach

The FAP algorithm [3,4] updates the error vector $\mathbf{e}(n)$ via the following approximation

$$\mathbf{e}(n)\approx\begin{bmatrix} d(n)-\mathbf{x}^T(n)\mathbf{w}(n-1) \\ (1-\mu)\overline{\mathbf{e}}(n-1)\end{bmatrix} \tag{8}$$

where $\overline{\mathbf{e}}(n-1)$ consists of the $P-1$ upper elements of $\mathbf{e}(n-1)$. For Clarity, we present the FAP algorithm in Table 1, where the definitions of $\hat{\mathbf{w}}(n)$ and $\boldsymbol{\varphi}(n)$ can be found in (13) and (14). The only difference among many variants of FAP algorithm is the calculation of the linear system of equations. We assume that solving $[\mathbf{R}(n)+\delta\mathbf{I}]\boldsymbol{\varepsilon}(n)=\mu\mathbf{e}(n)$ requires P_m multiplications and P_a additions.

Using (8), the complexity of the filtering step reduces from $O(PL)$ operations in (4) to $O(L)$ operations. But (8) is only an approximate implementation of (5) under the condition that δ is significantly smaller than the eigenvalue of the matrix $\mathbf{R}(n)=\mathbf{X}^T(n)\mathbf{X}(n)$ [4,17]. When the regularization parameter δ is large, the approximation in (8) can cause discrepancy between the FAP and AP algorithms. Indeed, the regularization parameter can vary from very small to very large, depending on the level of the additive noise [23]. Thus, the assumption used in the derivation of (8) has its shortcomings [17].

2.2. FEF approach

In the FAP algorithm, only the first component of the error vector is calculated, and the others are approximated. More recently, a low-complexity FEAP algorithm [18] was proposed where all the error vector components can be exactly calculated. The basic idea is summarized as follows.

Substituting (7) into (4), one has

$$\mathbf{y}(n)=\mathbf{X}^T(n)\mathbf{w}(n-1) =\mathbf{z}(n)+\mathbf{G}(n)\boldsymbol{\varepsilon}(n-1) \tag{9}$$

where $\mathbf{G}(n)=\mathbf{X}^T(n)\mathbf{X}(n-1)$ and $\mathbf{z}(n)=\mathbf{X}^T(n)\mathbf{w}(n-2)$. Taking (4) into account, $\mathbf{z}(n)$ can be expressed as

$$\mathbf{z}(n)=\mathbf{X}^T(n)\mathbf{w}(n-2) =[z_0(n),y_0(n-1),\ldots,y_{P-2}(n-1)]^T \tag{10}$$