Abstract

A two-dimensional fast recursive least squares algorithm is presented using a geometrical formulation based on the mathematical concepts of vector space, orthogonal projection, and subspace decomposition.

By appropriately ordering the 2-D data, the algorithm provides an exact least-squares solution to the deterministic Normal equations. The method is further extended to the general FIR Wiener filter and to ARMA modeling. The size and shape of the support region for both the MA and AR coefficients of the filter can be chosen arbitrarily.

1 Introduction

Adaptive algorithms have been used successfully in a wide range of signal processing applications involving unknown or non-stationary data. Real time implementation of these algorithms has recently become possible with the latest VLSI technology partly as a result of efficient computational algorithms such as the ‘fast’ recursive least squares (fast RLS). The fast RLS methods are based upon an elegant geometric approach involving the concepts of linear vector spaces, orthogonality, projection operators, and their relation to linear least squares prediction [1,2,3]. The development of adaptive algorithms for two-dimensional (2-D) problems has been much slower than their development for one-dimensional (1-D) problems. In this paper we describe a fast RLS method for 2-D signals. We show that besides the 2-D filter, the method involves the use and simultaneous update of two additional multichannel filters, and related gain transversal filters. In this way the algorithm is similar but more extensive than the 1-D counterpart.

2 2-D RLS Algorithm

2.1 2-D Adaptive Filtering

In the following we consider a 2-D linear predictive filter of the form

\[ \hat{y}(n_1, n_2) = \sum_{i=0}^{N} \sum_{j=0}^{M} a_{ij} y(n_1 - i, n_2 - j) \]

\[ (i, j) \neq (0, 0) \]  \hspace{1cm} (1)

and a 2-D data sequence with \( K \times L \) points as shown in Fig. 1(a). We assume that the entire data set is processed by scanning along rows and that in processing the data, the data is considered to 'wrap around' at the end points in a helicoidal fashion, so the prediction filter has the support depicted in Fig. 1(b) at the beginning and ends of rows. This is an important assumption in developing the filter update procedure. If the prediction error is defined as

\[ e(n_1, n_2) = y(n_1, n_2) - \hat{y}(n_1, n_2) \]  \hspace{1cm} (2)

Figure 1: First Quadrant \((N + 1) \times (M + 1)\) Filter

A New 2-D Fast RLS Algorithm

A. M. Sequeira and C. W. Therrien*
Naval Postgraduate School
Monterey, California 93943

*Sr. Member of the IEEE.
then the optimal least-squares filter is defined to be the filter that minimizes the accumulated squared error

$$c(n_1, n_2) = \sum_{i=0}^{n_1} [e(i, n_2)]^2 + \sum_{i=0}^{K-1} \sum_{j=0}^{n_2-1} [e(i, j)]^2$$

(3)

The purpose of this paper is to describe a fast RLS algorithm that solves this problem.

Because of limited space and the length and complexity of the derivation we will not be able to derive the algorithm here. (A complete derivation is given in [4] and requires several pages.) The derivation relies heavily on the use of projection operations and geometrical concepts; these will be more difficult to appreciate in this limited discussion. In what follows we will merely define the components of the algorithm and give a brief outline of its steps.

### 2.2 Algorithm Components

The key to developing a fast 2-D RLS algorithm lies in the consideration of a forward and a backward multi-channel filtering problem, and their interrelation with the 2-D problem. These components of the algorithm will be discussed with the aid of Fig. 2.

![Figure 2: M + 1 Channel Analogy](image)

Let us consider a “channel” to be the $i^{th}$ row covered by the mask and denote the data in the $i^{th}$ channel by $y_i(n)$ where $n$ is the linear scanning index $n = n_2K + n_1$. With this new notation, the 2-D linear prediction problem has the form

$$\hat{y}_i(n) = Y_{1,N}^T(n) a(n)$$

(4)

where $Y_{1,N}(n)$ is the data covered by the filter mask (shaded in Fig. 2) and $a(n)$ is the correspondingly ordered vector of 2-D filter coefficients.

$$y_{1,N}(n) = [y_1(n-1), \ldots, y_1(n-N), \ldots, y_M(n+1), \ldots, y_M(n-N)]^T$$

(5)

$$a(n) = [a_{10}(n), a_{11}(n), \ldots, a_{1M}(n), a_{20}(n), \ldots, a_{2M}(n)]^T$$

(6)

and the prediction error (2) is given by

$$e_1(n) = y_1(n) - \hat{y}_1(n)$$

(7)

Let $x_F(n)$ be a (M+1)-channel signal formed by the data acquired by the 2-D mask when it is moved from time $n$ to $n+1$ (unshaded boxes in Fig. 2). The matrix $E(n)$ is the set of optimal filter coefficients for predicting this data from the data under the filter mask. This prediction is given by

$$\hat{x}_F(n) = E^T(n)Y_{1,N}(n)$$

(8)

and we define the corresponding prediction error covariance matrix as $\Sigma_F(n)$.

A corresponding backward multichannel problem can be defined as follows. Let $x_B(n)$ represent the data left out by the 2-D mask when it is moved from time $n$ to $n+1$ (data in left column in Fig. 2). Then proceeding as for the forward problem, define a matrix of prediction coefficients $B(n)$ and the corresponding prediction error covariance matrix $\Sigma_B(n)$. This prediction is of the form

$$\hat{x}_B(n) = B(n)^T Y_{0,N-1}(n)$$

(9)

where $Y_{0,N-1}(n)$ is the data covered by the mask in Fig. 2 when it is moved from time $n$ to $n+1$. (This includes the unshaded boxes.)

As a final step we proceed along lines similar to what has been done for 1-D single channel problems in developing the fast RLS. We introduce the fixed $n+1$-dimensional vector

$$\pi(n) = [0, 0, 0, \ldots, 0, 0, 1]^T$$

(10)

and consider the filter $g(n)$ necessary for predicting this vector using all of the data up to time $n$. The estimate of $\pi(n)$ is of the form

$$\hat{\pi}(n) = Y_{0,N-1}(n)g(n)$$

(11)

where $Y_{0,N-1}(n)$ is the data matrix whose rows are the transposed vectors $Y_{k,N-1}^T(k), k = 0, \ldots, n$. This filter $g$ is known as the \textit{gain transversal filter} and
its consideration leads to a method for quantifying
the angular change between the subspaces associated
with data matrices at times \( n \) and \( n - 1 \). From geometric
arguments it can be shown that

\[
\gamma(n) = 1 - Y_{0,N-1}^T(n) g(n) = \cos^2 \theta
\]

(12)

where \( \theta \) is the angle between \( \pi(n) \) and the vector
normal to the data subspace. Updating this filter
and the associated angular change parameter \( \gamma(n) \) is key to updating the 2-D filter.

One further set of terms is necessary for the 2-D
fast RLS problem. This is the concept of the extended
gain transversal filter \( g'(n) \). Its interpretation
is that of estimating \( \pi(n) \) from the extended data sets \( \{Y_{0,N}(n)\} \) where \( Y_{0,N}(n) \) is all of the data
shown in Fig. 2 at time \( n \) (both shaded and unshaded boxes). Since the data for \( Y_{0,N}(n) \) is defined by cate-
nerating rows of the extended area, consideration of the
extended problem necessarily requires the use of forward
and backward permutation matrices \( \Psi_F \) and
\( \Psi_B \) which are defined implicitly by

\[
Y_{0,N}(n) = [x_F(n), y_{1,N}(n)] \Psi_F
\]

(13)

where \( y_{0,N-1}(n), x_B(n) \) are set to \( 1.0 \) since the initial data is zero and since all of the
subspaces associated with previous data are the null space. Since a positive forward prediction error
variance is necessary for the algorithm to start \( \Sigma_F^{-1} \)
set to zero for

\[
\Sigma_F^{-1}(n) = \Sigma_F^{-1}(n-1) - \frac{\Sigma_F^{-1}(n-1) e_F(n) e_F^T(n) \Sigma_F^{-1}(n-1)}{\gamma(n-1) + e_F^T(n) \Sigma_F^{-1}(n-1) e_F(n)}
\]

(18)

Inverse error covariance matrix for the multichannel
forward filter (1.5\( K_1^2 + 2.5K_1 \) operations):

\[
\Sigma_F^{-1}(n) = \Sigma_F^{-1}(n-1) - \frac{\Sigma_F^{-1}(n-1) e_F(n) e_F^T(n) \Sigma_F^{-1}(n-1)}{\gamma(n-1) + e_F^T(n) \Sigma_F^{-1}(n-1) e_F(n)}
\]

(19)

Multichannel forward filter update (\( K_1^2 + K_1K_2 \) operations):

\[
F(n) = F(n-1) + g(n) e_F(n) e_F^T(n) F(n-1)
\]

(20)

Extended gain transversal filter (\( K_1^2 + K_1K_2 \) operations):

\[
\Psi_B \Psi_F^T \left[ \begin{bmatrix} 0_{M+1} & \Psi_F^T \right] + \left[ \begin{bmatrix} I_{M+1} & -F(n) \right] \right] \Psi_F^{-1} e_F(n)
\]

(21)

A priori multichannel backward prediction error (\( K_1K_2 \) operations):

\[
e_F(n) = e_F(n-1) - \gamma_F(n) \Sigma_F^{-1}(n-1) e_F(n)
\]

(22)

Angle parameter (\( K_1 + 1 \) operations):

\[
\gamma(n) = \gamma(n-1) e_F^T(n-1) \Sigma_F^{-1}(n-1) e_F(n)
\]

(23)

Gain transversal filter (\( K_1K_2 + K_2 \) operations using previous results):

\[
\Psi_B(n) = \Psi_B(n-1) + g(n) e_B^T(n-1)
\]

(24)

Multichannel backward filter update (\( K_1K_2 \) operations):

\[
\Psi_B(n) = \Psi_B(n-1) + g(n) e_B^T(n-1)
\]

(25)

The total number of operations (multiplications or
divisions) required per iteration by the algorithm is

\[
2.5K_1^2 + 6K_1K_2 + 4.5K_1 + 3K_2 + 1
\]

(26)

\[
A(n) = a(n-1) + g(n-1)e_1(n-1)
\]

(15)

\[
e_1(n-1) = y_1(n) - Y_{1,N}^T(n) a(n-1)
\]

(14)

\[
B(n) = B(n-1) + g(n)e_B^T(n-1)
\]

(27)

\[
2-D \text{ filter update (} K_2 \text{ operations)}:
\]

\[
a(n) = a(n-1) + g(n-1)e_1(n-1)
\]

(15)
3 ARMA Model Extensions

The ARMA version of the 2-D fast RLS can be viewed as follows. Let us call the output or observed data $y_l(n)$ and the input data $w_l(n)$. For the present let us assume that this latter sequence is also known or observed. Let us separate the coefficients that operate on the two different sequences and call $g(n)$ the vector of AR coefficients of the filter, and $b(n)$ the vector of MA coefficients of the filter. As before, we develop this extension of the 2-D fast RLS to ARMA models assuming a first quadrant $(N + 1) \times (M + 1)$ quarter plane mask for both the AR and MA components of the filter, noting that more general forms are possible. Using the scanning index $n$ defined before, we proceed by defining an ARMA prediction filter of the form

$$\hat{y}_1(n) = Y_{1,N}(n) a(n) + W_{1,N}(n) b(n)$$

with $Y_{1,N}(n)$ and $W_{1,N}(n)$ defined using the same ordering as in (5). We want to find $a(n)$ and $b(n)$ to minimize the sum of squared errors

$$e_1(n) = y_1(n) - \hat{y}_1(n)$$

This can be written in vector notation as

$$e_1(n) = g_1^T(n) g_1(n)$$

with

$$g_1(n) = Y_1(n) - \hat{Y}_1(n)$$

We can combine the AR and MA coefficients in one single vector $g(n)$ as

$$g(n) = [a^T(n), b^T(n)]^T$$

The data under the mask can then be expressed as

$$z_{1,N}(n) = [Y_{1,N}(n), W_{1,N}(n)]^T$$

Now we have for the estimate of $Y_1(n)$

$$\hat{Y}_1(n) = Z_{1,N}(n) g(n)$$

where $Z_{1,N}(n)$ is the data matrix

$$Z_{1,N}(n) = [Y_{1,N}(n), W_{1,N}(n)]$$

formed by $Y_{1,N}(n)$ and $W_{1,N}(n)$, the data matrices whose rows are the transposed vectors vectors $Y_{1,N}(k)$ and $W_{1,N}(k)$, $k = 0, \ldots, n$. The least squares solution for $g(n)$ is given by the pseudo-inverse of $Z_{1,N}(n)$

$$g(n) = (Z_{1,N}(n) Z_{1,N}(n))^+ Z_{1,N}(n) Y_1(n)$$

After defining new projection matrices and transversal filter operators associated with the new data matrices, the algorithm to recursively update $g(n)$ closely follows the procedures developed above.

4 Conclusions

A two-dimensional fast recursive least squares algorithm was described in this paper. The derivation is based on the relation between least squares prediction and the concepts of orthogonality associated with vector spaces. The ordering necessary to develop the recursive algorithm was imposed on the data by using a linear scanning index.

The algorithm has been compared in performance with a more standard form RLS algorithm for 2-D and a 2-D LMS algorithm [4] A substantial reduction in computational cost is obtained when compared with the basic 2-D RLS algorithm. The 2-D fast RLS algorithm requires on the order of $6K_1K_2$ arithmetic operations per iteration compared with 1.5$K_2^2$ for the basic RLS, where $K_1$ is the number of channels defined for the 2-D fast RLS algorithm and $K_2$ is the total number of coefficients in the 2-D filter. The 2-D LMS algorithm due to its simplicity, is more economical than our algorithm in terms of computational cost, but lacks the excellent convergence performance experienced for the 2-D fast RLS.

References


