PARTIAL UPDATE EDS ALGORITHMS FOR ADAPTIVE FILTERING

Bei Xie and Tamal Bose

Wireless @ Virginia Tech
Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061, USA

ABSTRACT

In practice, computational complexity is an important consideration of an adaptive signal processing system. A well-known approach to controlling computational complexity is applying partial update (PU) adaptive filters. In this paper, a partial update Euclidean Direction Search (EDS) algorithm is employed. The theoretical analyses of mean and mean-square performance are presented. The simulation results of different PU EDS are shown.

Index Terms— EDS, partial updates

1. INTRODUCTION

Adaptive filters play an important role in the fields related to digital signal processing, such as system identification, noise cancellation, and channel equalization. In the real world, the computational complexity of an adaptive filter is an important consideration for applications which need long filters. Usually algorithms with high computational complexity, such as Recursive Least Square (RLS) and EDS [1], give better convergence performance. Therefore, a tradeoff must be made between computational complexity and performance. One option is to use partial update techniques [2] to reduce the computational complexity. The partial update adaptive filter only updates part of the coefficient vector instead of updating the entire vector. The theoretical results on the full-update case may not apply to the partial update case. Therefore, performance analysis of the partial update adaptive filter is very meaningful.

The EDS algorithm, developed by Xu and Bose [1], is similar to the RLS. It has fast a convergence rate and high computational complexity $O(N^2)$. A fast EDS algorithm was developed to achieve $O(N)$ computational complexity [1]. It combines the benefits of fast convergence of the RLS and the computational efficiency of the Least Mean Square (LMS) algorithm. Since the EDS algorithm is relatively new, there has been little attempt to analyze the performance of the PU EDS. In [3] and [4], the deficient-length EDS, which is a special case of PU EDS, was studied.

In this paper, partial update techniques are applied to the EDS algorithm. The theoretical mean and mean-square expressions of the PU EDS are derived. The performance of the PU EDS is analyzed. In Section 2, different PU EDS algorithms are developed. Theoretical analyses of PU EDS are given in Section 3. In Section 4, the simulation results for correlated inputs are shown.

2. PARTIAL UPDATE EDS

The basic adaptive filter system model can be written as:

$$d(n) = x^T(n)w^* + v(n),$$

where $d(n)$ is the desired signal, $x(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$ is the input data vector of the unknown system, $w^*(n) = [w_1^*, w_2^*, ..., w_N^*]^T$ is the impulse response vector of the unknown system, and $v(n)$ is zero-mean white noise, which is independent of any other signals.

Let $w$ be the coefficient vector of an adaptive filter. The estimated signal $y(n)$ is defined as

$$y(n) = x^T(n)w(n),$$

and the output signal error is defined as

$$e(n) = d(n) - x^T(n)w(n).$$

The partial update method aims to reduce the computational cost of the adaptive filters. Instead of updating all the $N \times 1$ coefficients, it usually only updates $M \times 1$ coefficients, where $M < N$. Basic partial update methods include periodic PU, sequential PU, the deficient-length method, stochastic PU, the MMax update method, etc. These methods will be applied to the EDS algorithm.

The partial update EDS has the uniform update equation:

$$w(n+1) = w(n) - \frac{g^T \hat{Q}(n)w(n) - \hat{p}(n)}{\hat{g}^T \hat{Q}(n)\hat{g}},$$

where

$$\hat{Q}(n) = \sum_{i=1}^{n} \lambda^{n-i}I_M x(i)x^T(i)I_M^T,$$

$$\hat{p}(n) = \sum_{i=1}^{n} \lambda^{n-i}d(i)I_M x(i),$$

$$I_M(n) = \begin{bmatrix} i_1(n) & 0 & \ldots & 0 \\ 0 & i_2(n) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & i_N(n) \end{bmatrix},$$

This work was supported in part by NASA grant #NNG06GE95G, NSF Grant #0809036, and the Institute for Critical Technology and Applied Science.
\[
\sum_{k=1}^{N} i_k(n) = M, \quad i_k(n) \in \{0, 1\}, \tag{8}
\]

where \( \hat{x} = I_M x \) and the \( N \times 1 \) vector \( g \) is the search direction at iteration \( n \), which is taken to be "Euclidean directions." It is defined as \( g_k = [0, \ldots, 0, 1, 0, \ldots, 0]^T \), where the 1 appears in the \( i \)-th position. For PU EDS, the entire weight vector \( w(n) \) is partially updated by cycling through the Euclidean directions \( g_k \) at each iteration \( n \) and \( k \) is determined by \( i_k \). This means if \( i_k = 1 \), then the \( k \)-th element of \( g \) will be set to be 1. If \( i_k = 0 \), then the \( k \)-th element of \( w(n) \) will not be updated. The total number of multiplications needed for this uniform PU EDS becomes \( M(2N + M + 3) \). The original EDS needs \( N(3N + 3) \) \([1]\) multiplications. For different partial update algorithms, only \( I_M \) will be different.

2.1. Periodic Partial Update EDS

The periodic partial update method \([2]\) only updates the coefficients at every \( S \)th iteration and copies the coefficients at the other iterations. The periodic PU EDS doesn’t partially update the coefficient vector, but updates the entire vector at every \( S \)th iteration. The update equation for EDS can be written as:

\[
w((n+1)S) = w(nS) - \frac{g^T(Q(nS)w(nS) - \bar{x}(nS))g}{g^T Q(nS)g}, \tag{9}
\]

where \( S = \lceil \frac{N}{M} \rceil \), which is the ceiling of \( \frac{N}{M} \). The total number of multiplications needed for every \( S \) number of samples is \( N(3N + 3) \). Since the periodic PU EDS still updates the whole vector, the steady-state performance will be the same as the original EDS. However, the convergence rate of periodic PU EDS will be \( S \) times slower than the original EDS.

2.2. Deficient-length EDS

The deficient-length method only updates the first \( 1 \ldots M \) elements of the coefficient vector \( w \) at each iteration. Then \( i_k(n) \) becomes

\[
i_k(n) = \begin{cases} 1 & \text{if } k = 1, \ldots, M \\ 0 & \text{otherwise} \end{cases}, \tag{10}
\]

2.3. Sequential Partial Update EDS

The sequential partial update method \([2]\) designs \( i_k(n) \) as:

\[
i_k(n) = \begin{cases} 1 & \text{if } k \in K_n \mod S + 1 \\ 0 & \text{otherwise} \end{cases}, \tag{11}
\]

where \( K_1 = \{1, 2, \ldots, M\} \), \( K_2 = \{M + 1, M + 2, \ldots; 2M\}, \ldots \), \( K_S = \{(S-1)M + 1, (S-1)M + 2, \ldots, N\} \).

2.4. Stochastic Partial Update EDS

The stochastic partial update algorithm chooses coefficients subsets randomly. The \( i_k(n) \) becomes

\[
i_k(n) = \begin{cases} 1 & \text{if } k \in K_{m(n)} \\ 0 & \text{otherwise} \end{cases}, \tag{12}
\]

where \( m(n) \) is a random process with probability mass function \([2]\):

\[
Pr\{m(n) = i\} = p_i, \quad i = 1, \ldots, S, \quad \sum_{i=1}^{S} p_i = 1. \tag{13}
\]

Usually a uniformly distributed random process will be applied. Therefore, for each iteration, \( M \) of \( N \) weight elements will be chosen and updated with probability \( p_i = 1/S \).

2.5. MMax EDS

The update equations (4) and (7) can still be applied to the MMax EDS algorithm. The condition of \( i_k(n) \) \([2]\) becomes

\[
i_k(n) = \begin{cases} 1 & \text{if } |x_k(n)| \in \max_{1 \leq i \leq N} \{ |x_i(n)|, M \} \\ 0 & \text{otherwise} \end{cases}, \tag{14}
\]

which means the elements of the weight vector \( w \) are updated according to the first \( M \) max elements of the input signal \( x \).

3. PERFORMANCE ANALYSIS OF PARTIAL UPDATE EDS

To determine the steady-state behaviors of partial update EDS, two assumptions are needed.

Assumption I: The coefficient error \( z(n) \) is independent of the input signal \( x(n) \) in steady state.

Assumption II: Noise \( v(n) \) is independent of the input signal \( x(n) \) and has zero mean.

The desired signal can be rewritten as

\[
d(n) = \hat{x}^T(n)w^* + \bar{x}(n)w^* + v(n), \tag{15}
\]

where \( \bar{x} = x - \hat{x} = (I - I_M)x = I_{N-M}x \).

Introducing the coefficient error vector

\[
z(n) = w(n) - w^*, \tag{16}
\]

and subtracting \( w^* \) from both sides of (4), we have

\[
z(n+1) = z(n) - \frac{gg^T(\hat{Q}(n)w(n) - \bar{r}(n))}{g^T Q(n)g}. \tag{17}
\]

Now consider the term \( \hat{Q}(n)w(n) - \bar{r}(n) \). Since \( \hat{Q}(n) = \lambda Q(n-1) + \hat{x}(n)\hat{x}^T(n) \), \( \bar{r}(n) = \lambda \bar{r}(n-1) + d(n)\hat{x}(n) \), and from (15), (16), we get

\[
\hat{Q}(n)w(n) - \bar{r}(n) = \sum_{i=1}^{n} \lambda^{n-i} \hat{x}(i) \hat{x}^T(i) z(n) - \sum_{i=1}^{n} \lambda^{n-i} v(i) \hat{x}(i)
\]

\[
= \hat{Q}(n)z(n) - B(n)w^* - C(n), \tag{18}
\]

where

\[
B(n) = \sum_{i=1}^{n} \lambda^{n-i} \hat{x}(i) \hat{x}^T(i), \quad \tag{19}
\]

\[
C(n) = \sum_{i=1}^{n} \lambda^{n-i} v(i) \hat{x}(i). \quad \tag{20}
\]

Now we can rewrite the PU EDS coefficient error recursion in (17) as

\[
z(n+1) = z(n) - \frac{gg^T(\hat{Q}(n)z(n) - B(n)w^* - C(n))}{g^T \hat{Q}(n)g}. \tag{21}
\]

Using the direct-averaging method \([6]\), (21) can be replaced by

\[
z(n+1) = z(n) - E\{\frac{gg^T \hat{Q}(n)g}{g^T Q(n)g}\}z(n)
\]

\[
+ E\{\frac{gg^T B(n)g}{g^T Q(n)g}\}w^* + \frac{gg^T C(n)}{g^T \hat{Q}(n)g}. \tag{22}
\]

Note \( w^* \) is independent of the input signal.
3.1. Mean Performance

The mean behavior of the EDS coefficient error vector can be determined by taking the expectation of both sides of (21)

$$E\{z(n+1)\} = (I - J(n)) E\{z(n)\} + S(n),$$

(23)

where $J(n) = E\{g g^T Q(n)g\}$ and $S(n) = E\{g g^T v(n)g\} w^*$. We have used $E\{g g^T C(n)g\} = 0$.

The necessary and sufficient condition for $E\{z\}$ to converge is

$$\rho(I - J(n)) < 1,$$

(24)

where $\rho$ is the spectral radius and this condition is equivalent to

$$0 < \lambda_{max}(J(n)) < 2.$$

(25)

The full-update EDS algorithm was proven to converge to the optimal filter $w^*$ in [5]. Since the maximal eigenvalue of $J(n)$ must be smaller than or equal to the maximal eigenvalue of $E\{g g^T Q(n)g\}$ for stationary signals, the PU EDS can also converge, but to the optimal filter plus an error weight. The convergence rate of the PU EDS depends on the term $J(n)$. As $g$ is only the direction and $g^T Q(n)g$ is only a scalar to normalize the $Q(n)$, the convergence can be further related to the partial input correlation matrix $\hat{R} = E\{\hat{x}\hat{x}^T\}$.

The steady-state mean weight error of the PU EDS is

$$E\{z(\infty)\} = \hat{R}^{-1} h,$$

(26)

where $\hat{R} = E\{\hat{x}\hat{x}^T\}$ and $h = E\{\hat{x}w^*\}$. From (23) and (26), we can see that the partial input correlation matrix $\hat{R}$ has a large effect on the convergence rate and the steady-state mean weight error. For the sequential PU EDS and the stochastic PU EDS $\hat{R} = \frac{g^T}{g^T} R$. Obviously, the error will increase and the convergence rate will decrease when $M$ decreases. For the MMax EDS, $\hat{R}$ is close to $R$, therefore the convergence rate should be close to that of the EDS. For the deficit-length EDS, the convergence rate should be slightly faster than the EDS because it uses a shorter filter.

For white inputs, $E\{\hat{x}\hat{x}^T\} = 0$. Therefore, $h = 0$. The steady-state mean weight is equal to the optimal weight.

3.2. Mean-Square Performance

The MSE equation of the PU EDS algorithm is the same as the full-update EDS, which is

$$\epsilon(n) = \epsilon_{min} + tr(\hat{R} E\{z(n)z^T(n)\}),$$

(27)

where $\epsilon_{min} = E\{v^2(n)\}$ is the variance of the input noise, and $\hat{R} = E\{z(n)z^T(n)\}$ is the autocorrelation matrix of the input data. The only difference for PU EDS is the weight-error autocorrelation matrix $K(n) = E\{z(n)z^T(n)\}$. By using Assumptions I, II, and (22), $K(n)$ is determined as follows:

$$K(n+1) = K(n) - J(n)J^T(n) - J(n)K(n) + J(n)K(n)J^T(n) + E\{z(n)z^T(n)\} + S(n)E\{z^T(n)\}$$

$$- J(n)E\{z(n)z^T(n)\} - S(n)E\{z^T(n)\} - J(n)K(n)J^T(n) + S(n)S^T(n)$$

$$+ S(n)S^T(n) + E\{g g^T C(n)C^T(n)g g^T\} \frac{g^T Q(n)g}{g^T Q(n)Q(n)g}.$$  

(28)

The MSE convergence is affected by the mean convergence.

4.1. System Identification

The system identification model is shown in Fig. 1 and is taken from [3]. The same unknown system was used as in [7]. It was a 16-order FIR filter (N=16), with the impulse response

$$w^* = [0.01, 0.02, -0.04, -0.08, 0.15, -0.3, 0.45, 0.6, 0.6, 0.45, -0.3, 0.15, -0.08, -0.04, 0.02, 0.01]^T.$$

In our simulations, the length of the partial update filter was M=12. The variance of the input noise $v(n)$ was $\epsilon_{min} = 0.0001$. The initial weights were $w = 0$. The parameter $\lambda$ of EDS was equal to 0.9995. The results were obtained by averaging 400 independent runs.

The input of the system [3] has the following form

$$x(n) = 0.8x(n-1) + \eta(n),$$

(31)

where $\eta(n)$ was zero-mean white Gaussian noise with unit variance.
4.2. Channel Equalization

Equalization performance is also examined among PU EDS algorithms. We use a simple short FIR channel [8]

\[ C(z) = 0.5 + 1.2z^{-1} + 1.5z^{-2} - z^{-3}. \]  

(32)

Fig. 3. Comparison of theoretical and simulation curves of the mean behavior of the 8th coefficient error of the PU EDS.

Fig. 4. Comparison of theoretical and simulation curves of the mean-square behavior of the PU EDS for correlated input.

Fig. 5. Comparison of SER among PU EDS algorithms.

We assume the full length of the equalizer is 30 and let the PU length be 15. The input sequence is 4-QAM. The results are obtained by averaging 50 independent runs. Fig. 5 illustrates the symbol-error-rate (SER) in log-scale among PU EDS algorithms. The SER performance of different PU EDS algorithms is still related to the MSE performance shown in Fig. 2. When SNR equals 15 dB, the SERs of the deficient-length EDS and the MMax EDS become zeros, which are the same as the full-update EDS. However, the sequential and the stochastic EDS algorithms perform much worse than the full-update EDS for high SNR.

5. CONCLUSION

In this paper, different PU EDS algorithms are developed. The expressions for mean and mean-square errors of the PU EDS are derived for both correlated and white inputs. Simulation results agree with the derived theoretical expressions in steady state.

6. REFERENCES


