PROPORTIONATE-TYPE NORMALIZED LEAST MEAN SQUARE ALGORITHM WITH GAIN ALLOCATION MOTIVATED BY MINIMIZATION OF MEAN-SQUARE-WEIGHT DEVIATION FOR COLORED INPUT

Kevin T. Wagner
Naval Research Laboratory
Radar Division
Washington, DC 20375, USA

Miloš I. Doroslovački
The George Washington University
Department of Electrical and Computer Engineering
Washington, DC 20052, USA

ABSTRACT

In previous work, a water-filling algorithm was proposed which sought to minimize the mean square error (MSE) at any given time by optimally choosing the gains (i.e. step-sizes) each time instance. This work relied on the assumption that the input signal was white. In this paper, an algorithm is derived which operates when the input signal is colored. The proposed algorithm minimizes the mean square weight deviation which is important in many applications such as system identification. Additionally, it is shown that by minimizing the mean square weight deviation, an upper bound on the MSE is also minimized. The proposed algorithm offers improved misalignment and learning curve convergence rates relative to other standard algorithms.

Index Terms— Adaptive filtering, convergence, proportionate-type normalized least mean square (PtNLMS) algorithm.

1. INTRODUCTION

The water-filling algorithm [1] was designed with the goal of minimizing the mean square error (MSE) by choosing optimal gains at each time step. This algorithm relied on the limiting assumption that the input signal is white. In this work, a colored-water-filling (CWF) algorithm is derived and implemented which no longer requires that the input signal is white.

The algorithm results from minimizing the mean square weight deviation with respect to the gain. The implementation of this algorithm requires the estimation of the mean weight deviation and knowledge of the input signal covariance matrix. An estimation scheme for the mean weight deviation is given. One application of this algorithm is to perform system identification for echo cancelation.

The remainder of this paper is organized as follows. In section 2 we review the PtNLMS algorithm. In section 3 we introduce the new optimal gain motivated algorithm. In section 4 the relationship between the MSE and the mean square weight deviation is discussed. Finally, in section 5 we present simulation results regarding the performance of these algorithms.

2. PtNLMS ALGORITHM INTRODUCTION

We begin by introducing PtNLMS algorithms. All signals are real throughout this work. Let us assume there is some input signal denoted as \( x(k) \) for time \( k \) that excites an unknown system with impulse response \( w \). Let the output of the system be \( y(k) = w^T x(k) \) where \( x(k) = [x(k), x(k-1), \ldots, x(k-L+1)]^T \), and \( L \) is the length of the filter. The measured output of the system, \( d(k) \), contains zero-mean stationary measurement noise \( v(k) \) and is equal to the sum of \( y(k) \) and \( v(k) \). The impulse response of the system is estimated with the adaptive filter coefficient vector, \( \hat{w}(k) \), which has length \( L \) also. The output of the adaptive filter is given by \( \hat{y}(k) = \hat{w}^T(k) x(k) \). The error signal \( e(k) \) between the output of the adaptive filter \( \hat{y}(k) \) and \( d(k) \) drives the adaptive algorithm. The weight deviation vector is given by \( z(k) = w - \hat{w}(k) \). Note the \( i \)th component of any vector \( a \) is denoted as \( a_i \) and the \( (i,j) \)th entry of any matrix \( A \) is denoted as \( A_{ij} \) throughout this work.

The PtNLMS algorithm is shown in Table I. Here, \( \beta \) is the fixed stepsize parameter. The term \( F[|\hat{w}_l(k)|], k \) with \( l \in \{1, 2, \ldots, L\} \), governs how each coefficient is updated. In the case when \( F[|\hat{w}_l(k)|], k \) is less than \( \gamma_{\text{min}} \), the quantity \( \gamma_{\text{min}} \) is used to set the minimum gain a coefficient can receive. The constant \( \delta_p \), where \( \delta_p \geq 0 \), is important in the beginning of learning when all of the coefficients are zero and together with \( \rho \), where \( \rho \geq 0 \), prevents the very small coefficients from stalling. \( G(k) = \text{DIAG} \{g_1(k), \ldots, g_L(k)\} \) is the time-varying stepsize control diagonal matrix. The constant \( \delta \) is typically a small positive number used to avoid overflow. Note for vector \( a \) with length \( L \) we define the function \( \text{DIAG}(a) \) as an \( L \times L \) matrix whose diagonal entries are the \( L \) elements of \( a \) and all other entries are zero. For matrix \( A \), we define the function \( \text{diag}(A) \) as a column vector containing the \( L \) diagonal entries from \( A \).

Some common examples of the term \( F[|\hat{w}_l(k)|], k \) are \( F[|\hat{w}_l(k)|], k = 1 \) and \( F[|\hat{w}_l(k)|], k = |\hat{w}_l(k)| \), which results in the normalized least mean square algorithm (NLMS) [2] and the proportionate NLMS algorithm (PNLMS) [3], respectively.

<table>
<thead>
<tr>
<th>Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td>PtNLMS Algorithm with Time-varying Stepsize Matrix</td>
</tr>
<tr>
<td>( x(k) )</td>
</tr>
<tr>
<td>( \hat{y}(k) )</td>
</tr>
<tr>
<td>( e(k) )</td>
</tr>
<tr>
<td>( F[</td>
</tr>
<tr>
<td>( \gamma_{\text{min}}(k) )</td>
</tr>
<tr>
<td>( \gamma_l(k) )</td>
</tr>
<tr>
<td>( g_l(k) )</td>
</tr>
<tr>
<td>( G(k) )</td>
</tr>
<tr>
<td>( \hat{w}(k+1) )</td>
</tr>
</tbody>
</table>
3. OPTIMAL GAIN ALGORITHM

In this section we seek the optimal gain at each time step \( k \). Our approach to this problem is to minimize the mean square weight deviation with respect to the gain under two constraints. The two constraints that are to be satisfied are that \( g_i(k) \geq 0 \, \forall \, i, k \) and \( \sum_{i=1}^{L} g_i(k) = L \, \forall \, k \). For notational simplicity we denote \( z(k+1) \) by \( z \), \( z(k-1) \) by \( z^- \), \( g(k) \) by \( g \), \( v(k) \) by \( v \), and \( x_i(k) \) by \( x_i \).

The recursion for the weight deviation vector is given by

\[
z^+ = z - \frac{\beta G x^T z}{\sum_j g_j + \delta} - \frac{\beta G z v}{\sum_j g_j + \delta}
\]  

(1)

which is obtained using the definitions in Table I.

Before proceeding further we make the following assumptions:

**Assumption I:** The input signal is a stationary Gaussian process with zero-mean and covariance \( R \), where \( [R]_{ii} = \sigma_i^2 \).

**Assumption II:** The input \( x \) and weight deviation vector \( z \) are independent. This is a reasonable assumption when \( \beta \) is sufficiently small, i.e. when \( z \) fluctuates much slower than \( x \) [2] [4].

**Assumption III:** The denominator terms, \( \sum_j g_j \), are assumed to be constant for all times. For \( L > \sqrt{2g_T[R]_{jj}\sigma_j^2} \), the standard deviation of the term \( \sum_j g_j \) becomes much smaller than the expected value, where \( \sigma_j \) represents the Hadamard product of the weight deviation vector and \( g = \text{diag}(G) \). A sufficient condition for this to hold is that

\[
S > > \frac{2\lambda_{\max}(R)}{\sigma_j^2}
\]

(2)

where \( S \) is the support size defined as

\[
S = \left[ \sum_j g_j \right]^2 = \frac{L^2}{\sum_j g_j^2}
\]

(3)

and \( \lambda_{\max}(R) \) is the largest eigenvalue of the covariance matrix \( R \). The condition in (2) is satisfied for large values of \( L \), and \( g \) that is not extremely sparse. Hence we can have that the denominator term is approximately constant.

**Assumption IV:** The measurement noise \( v \) is stationary white with zero-mean, variance \( \sigma_v^2 \), and \( g \) and \( v \) are independent of the input.

Next we employ Assumption III and define \( \mu = \beta/(L\sigma_j^2 + \delta) \). This allows us to rewrite the weight deviation recursion as

\[
z^+ = z - \mu G x^T z - \mu G z v.
\]

(4)

In this form the algorithm can be interpreted as a Least Mean Square (LMS) algorithm with time varying gains.

3.1. Optimal Gain Resulting from Minimization of Mean Weight Deviation

The criterion we try to minimize is the mean weight deviation at time \( k + 1 \). The weight deviation at time \( k + 1 \) can be represented by

\[
z^+ z^+ = z^T z - \mu z^T G x^T z - \mu z^T G z v
\]

\[
- \mu z^T x G z + \mu z^T x G^2 z + \mu z^T x G^2 v
\]

\[
- \mu x G z v + \mu x G^2 v + \mu x G^2 v.
\]

(5)

Next taking the expectation of (5) given the prior weight deviation \( z \), and assuming that \( g \) depends only on \( z \) yields:

\[
E \left\{ z^+ z^+ \mid z \right\} = z^T z - 2\mu z^T \text{Diag}(R) z
\]

\[
+ 2\mu^2 z^T \text{Diag}^2(R) z + \mu^2 z^T R z \text{Diag} \text{Diag}(R) z
\]

(6)

In [1] we assumed that the gain was a deterministic function of time and took the expectation with respect to the input and prior weight deviation instead of assuming the prior weight deviations are given and \( g \) is only a function of \( z \). We could work in this way now, but the resulting feasible algorithm would be the same regardless of which set of the assumptions are employed.

Next, we make the substitution \( g = s \circ s \) to ensure the solution \( g_i \geq 0 \, \forall \, i \) (\( s \) is real) and construct the Lagrangian:

\[
T(s, \lambda) = z^T z - 2\mu z^T \text{Diag}(R) s + \mu^2 z^T R z \text{Diag} \text{Diag}(R) s
\]

\[
+ 4\mu^2 \sigma_s^2 z^T \text{Diag} \text{Diag}(R) s
\]

\[
+ \lambda \left( 1^T (s \circ s) - L \right)
\]

(7)

where \( 1 \) is the \( L \times 1 \) vector of ones.

We can calculate the gradient and Hessian of the Lagrangian which are given by

\[
\frac{\partial T(s, \lambda)}{\partial s} = \{ 2\lambda - 4\mu \text{Diag} [z \circ R z]
\]

\[
+ 8\mu^2 \text{Diag} [R z \circ R z] (s \circ s)
\]

\[
+ 4\mu^2 z^T R z \sigma_s^2 \text{Diag} (s \circ s)
\]

\[
+ 4\mu^2 \sigma_s^2 \sigma_s^2 \text{Diag} (s \circ s)
\]

(8)

and

\[
\frac{\partial^2 T(s, \lambda)}{\partial s^2} = 2I - 4\mu \text{Diag} [z \circ R z]
\]

\[
+ 8\mu^2 \text{Diag} [R z \circ R z] (s \circ s)
\]

\[
+ 4\mu^2 z^T R z \sigma_s^2 \text{Diag} (s \circ s)
\]

\[
+ 4\mu^2 \sigma_s^2 \sigma_s^2 \text{Diag} (s \circ s)
\]

(9)

In order to aid in finding minimums of the Lagrangian we rewrite equations (8) and (9) in component-wise form as follows

\[
\frac{\partial T(s, \lambda)}{\partial s_i} = \{ 2\lambda - 4\mu z_i [R z_i] + 8\mu^2 [R z_i]^2 s_i
\]

\[
+ 4\mu^2 z_i^T R z_i \sigma_s^2 s_i + 4\mu^2 \sigma_s^2 \sigma_s^2 s_i\}
\]

(10)

and

\[
\frac{\partial^2 T(s, \lambda)}{\partial s_i^2} = 2\lambda - 4\mu z_i [R z_i] + 8\mu^2 [R z_i]^2 s_i
\]

\[
+ 4\mu^2 z_i^T R z_i \sigma_s^2 s_i + 4\mu^2 \sigma_s^2 \sigma_s^2 s_i
\]

\[
+ 2 \left[ 8\mu^2 [R z_i]^2 + 4\mu^2 z_i^T R z_i \sigma_s^2 + 4\mu^2 \sigma_s^2 \sigma_s^2 \right] s_i.
\]

(11)
Examining the component-wise form of the gradient there are two solutions
\[ s_1 = 0 \quad \text{and} \quad s_2 = \frac{4\mu z_i [Rz]_i - 2\lambda}{8\mu^2 [Rz]_i^2 + 4\mu^2 z^T Rz\sigma_z^2 + 4\mu^2 \sigma_z^2 \zeta} \tag{12} \]

By substituting the first solution into the component-wise form of the Hessian it turns out the first solution results in minimum when \( \lambda - 2\mu z_i [Rz]_i > 0 \) and a maximum when \( \lambda - 2\mu z_i [Rz]_i < 0 \). In contrast, when the second candidate solution is substituted into the component-wise form of the Hessian a minimum occurs when \( \lambda - 2\mu z_i [Rz]_i < 0 \) and no solution exists when \( \lambda - 2\mu z_i [Rz]_i > 0 \). As a result the gain which minimizes the mean square weight deviation at time \( k + 1 \) can be written as
\[ g_k = \left[ \frac{2\mu z_i [Rz]_i - \lambda}{4\mu^2 [Rz]_i^2 + 4\mu^2 z^T Rz\sigma_z^2 + 2\mu^2 \sigma_z^2 \zeta} \right]_+ \tag{13} \]

where \( [\cdot]_+ = 0 \) for \( a < 0 \) and \( [\cdot]_+ = a \) for \( a > 0 \).

Now the solution can be obtained by a water filling algorithm similar to the one proposed for the white input case \[1\]. First, make the following definitions:
\[ c_i = 2\mu z_i [Rz]_i \quad \text{and} \quad q_i = 4\mu^2 [Rz]_i^2 + 2\mu^2 z^T Rz\sigma_z^2 + 2\mu^2 \sigma_z^2 \zeta. \tag{14} \]

Next find the constant \( \lambda \) according to the following procedure. First we sort the entries of \( e = [e_1, e_2, \ldots, e_L]^T \) in ascending order to form a new vector such that \( c_{(1)} < c_{(2)} < \ldots < c_{(L)} \). We subsequently rearrange the elements of \( q = [q_1, q_2, \ldots, q_L]^T \) to match the position of the original indices in the sorted \( e \) and to form a new vector whose elements are \( q_{(1)}, q_{(2)}, \ldots, q_{(L)} \). The optimal value of \( \lambda \) solves the following equation:
\[ \sum_{j=1}^{L} \left[ \frac{c_{(j)} - \lambda}{q_{(j)}} \right]_+ = L. \tag{15} \]

From (15) the candidate solutions are
\[ \lambda = \frac{\sum_{j=1}^{L} c_{(j)} - L}{\sum_{j=1}^{L} q_{(j)}}. \tag{16} \]

We choose \( \lambda = \lambda_i \) if \( c_{(i-1)} < \lambda_i < c_{(i)} \), where \( c_{(0)} = -\infty \).

### 3.2. Implementation

The algorithm presented to this point is not feasible because it requires the knowledge of \( z_i [Rz]_i, [Rz]_i^2, \) and \( z^T Rz \). We propose replacing these quantities with an estimate of their corresponding mean values. For notational convenience let \( E\{a\} = \pi \).

#### 3.2.1. Estimate of Terms Involving \( z \)

We begin with
\[ E\{p\} = E\{xc\} = E\{x(x^T z + v)\} = Rz \tag{17} \]
then \( z = R^{-1}p \). We update our estimate of \( p \) in the following fashion: \( \hat{p} = \alpha \tilde{p} + (1 - \alpha)p \), where \( 0 < \alpha < 1 \).

Now we make three approximations by replacing \( z_i [Rz]_i, \approx \sqrt{\pi [\mu z_i]^2}, \) and \( z^T Rz \approx \sum \tilde{z}_j^2 \). Note that these estimates are only approximate and good when the variance of \( z_i \) is small compared to the mean of \( z_i \) which is typically true in the transient regime.

If \( \alpha \) is chosen too large (i.e. close to 1) the transient performance can be unsatisfactory. Conversely, if \( \alpha \) is set to small the steady-state error will be large.

### 3.2.2. Adaptive Convex Gain Combination

A potential solution to the above problems is presented here as an adaptive convex gain combination of the NLMS gain with the NLMS gain. Let \( g^* \) represent the gain generated from the CWF algorithm. The gain used in the implementation of the proposed algorithms is a mixture of \( g^* \) and \( 1 \) given as \( g = (1 - \zeta)g^* + \zeta \). Here \( \zeta \) is a mixing parameter defined as
\[ \zeta = \min \left[ 1, \frac{\omega^2 \sigma_x^2}{\sum_{j=1}^{L} (\pi_j^2 + \zeta^2)} \right] \tag{18} \]
where \( \omega \geq 0 \). The denominator term in (18) is related to an estimate of the mean square deviation error (MSDE). Hence as the estimated MSDE approaches zero the algorithm acts more like the NLMS by equally distributing gain to all coefficients. When the estimated MSDE is large relative to \( \sigma_x^2/\sigma_x^2 \) then the algorithm inherits the characteristics of CWF algorithm. Now \( \alpha \) can be chosen to ensure satisfactory transient performance. Some modification of (18) can be used as well. For example, the denominator term in (18) could be replaced by an estimate of the MSE.

This gain selection methodology is a variant of the gain proposed in the SC-PNLMS \[5\]. But instead of estimating sparsity, an estimate of the distance to steady-state noise is used to set the gain.

### 4. RELATIONSHIP BETWEEN MINIMIZATION OF MSE AND MINIMIZATION OF MEAN SQUARE WEIGHT DEVIATION

Let the MSE at time \( k + 1 \) be represented by \( J(k + 1) = J^+ \). The MSE can be written as
\[ J^+ = \sigma_0^2 + E\{(x^+)^T z^+ (z^+)^T x^+\}. \tag{19} \]

Applying assumption II we can take the expectation with respect to the input signal first which yields
\[ J^+ = \sigma_0^2 + E\{(z^+)^T Rz^+\}. \tag{20} \]

Next applying the Cauchy-Schwarz inequality gives
\[ J^+ \leq \sigma_0^2 + \sqrt{E\{(z^+)^T z^+\}} \sqrt{E\{(z^+)^T R^2 z^+\}}. \tag{21} \]

Let \( \lambda_{\text{max}}(R) \) represent the largest eigenvalue of the covariance matrix \( R \). Then we can write
\[ J^+ \leq \sigma_0^2 + \sqrt{E\{(z^+)^T z^+\}} \sqrt{E\{(z^+)^T \lambda R z^+\}} = \sigma_0^2 + E\{(z^+)^T z^+\} \lambda_{\text{max}}(R). \tag{22} \]

Hence by minimizing the mean square weight deviation we minimize an upper bound for \( J^+ \).
In Figures 1 and 2 we plot the misalignment and MSE for the NLMS, PNLMS, MPNLMS [6], color-water-filling, and ideal color-water-filling algorithms with $\beta = 0.02$, $\rho = 0.01$, $\sigma^2 = 10^{-4}$, $\delta = 10^{-4}$, $\delta_\rho = 0.01$, and $L = 50$, respectively. The misalignment at time $k$ is defined by $M(k) = z^T z - \hat{w}^T \hat{w}$. The MPNLMS used the value 22941 in the $\mu$-law. The ideal color-water-filling algorithm uses the instantaneous value of the weight deviation, $z(k) = w - \hat{w}(k)$, to drive the water-filling algorithm. The color-water-filling algorithm uses the implementation described in 3.2 with $(\omega, \alpha) = (5, 0.999)$. The impulse response used in this simulation is given in Figure 3. The input signal consists of colored noise generated by a single pole autoregressive model with variance $\gamma = 0.9$, $x(0) = n(0)$, and $n(k)$ is a white Gaussian random variable with variance $\sigma^2 = 1$. Therefore $\sigma^2 = \sigma^2 / (1 - \gamma^2) = 5.263$.

The NLMS algorithm has the slowest convergence followed by the PNLMS algorithm and then the MPNLMS algorithm. The feasible and ideal CWF algorithms offer significant improvement in the convergence rate relative to the other algorithms.

6. CONCLUSION

In this paper we have introduced the colored-water-filling algorithm. The algorithm provides gains at each time step in a manner that tries to minimize the mean square weight deviation. Through simulation in colored input scenarios, it was shown that the feasible and ideal colored-water-filling algorithms have impressively superior learning curve and misalignment convergence rates relative to other standard algorithms such as the MPNLMS, PNLMS, and NLMS algorithms, when the steady-state errors are of the same level. This work generalized the approach in [1] to a much wider class of signals. Implementations providing lower computational complexity is an area for future work.

7. REFERENCES