IMPROVED SET-MEMBERSHIP PARTIAL-UPDATE AFFINE PROJECTION ALGORITHM

Paulo S. R. Diniz, Fellow, IEEE, and Hamed Yazdanpanah

Universidade Federal do Rio de Janeiro
DEL/Poli & PEE/COPPE/UFRJ
P.O. Box 68504, Rio de Janeiro, RJ, 21941-972, Brazil

ABSTRACT
In this paper, we present an improved set-membership partial-update affine projection (I-SM-PUAP) algorithm, aiming at accelerating the convergence, and decreasing the update rates and the computational complexity of the set-membership partial-update affine projection (SM-PUAP) algorithm. To meet these targets, we constrain the weight vector perturbation to be bounded by a hypersphere instead of the threshold hyperplanes as in the standard algorithm. We use the distance between the present weight vector and the expected update in the standard set-membership affine projection (SM-AP) algorithm to construct the hypersphere. With this strategy, the new algorithm shows better behavior in the early iterations. Simulation results verify the excellent performance of the proposed algorithm related to the convergence rate and the required number of updates.

Index Terms— adaptive filtering, set-membership filtering, partial-update, identification problem

1. INTRODUCTION
Adaptive filters have applications in a wide range of areas such as noise cancellation, signal prediction, echo cancellation, communications, radar, and speech processing. In several applications, the large number of coefficients to be updated leads to high computational complexity, turning the adaptation of the filter coefficients prohibitive in terms of hardware requirements. In these applications, the convergence would entail a large number of iterations, calling for more sophisticated updating rule which is inherently more computationally intensive. For a given adaptive filter, the computational complexity can be reduced by updating only part of the filter coefficients at each iteration, forming a family of algorithms called partial-update (PU) algorithms. In the literature, several variants of adaptive filtering algorithms with partial-update have been proposed [1–14].

Another powerful approach to decrease the computational complexity of an adaptive filter is to employ set-membership filtering (SMF) approach [2, 15]. The SMF allows the reduction in computational complexity, since the filter coefficients are updated only when the estimation error is higher than a pre-determined threshold. Algorithms developed from the SMF framework employ a deterministic objective function related to a bounded error constraint on the filter output, such that the updates belong to a set of feasible solutions. Implementation of SMF algorithms involves two main steps: 1) information evaluation, 2) parameter update. As compared with the standard normalized least mean square (NLMS) and affine projection (AP) algorithms, the set-membership normalized least mean square and affine projection (SM-NLMS and SM-AP) algorithms lead to reduced computational complexity chiefly due to data-selective updates [15–22].

The use of PU strategy decreases the computational complexity while reducing convergence speed. We employ SMF technique to reduce further the computational load due to lower number of updates. However applying the SMF and PU strategies together might result in slow convergence speed. One approach to accelerate the convergence speed is choosing a smaller error estimation bound but it might increase the number of updates. Also, if we adopt a higher error estimation threshold to reduce the number of updates, the convergence rate decreases. Therefore, convergence speed and computational complexity are conflicting requirements.

In this paper, we introduce an interesting algorithm which can accelerate the convergence speed and simultaneously reduce the number of updates (and as a result decrease the computational complexity) in the SM-PUAP algorithm. In the SM-PUAP algorithm, some updates move too far from their SM-AP update; especially when the angle between the updating direction and the threshold hyperplane is small. In this case we might have a large disturbance in the coefficient update while attempting to reach the feasibility set. Therefore, to limit the distance between two consecutive updates, first we will construct a hypersphere centered at the present weight vector whose radius equals the distance between the present weight vector and the weight vector that would be obtained with the SM-AP algorithm. This radius is an upper bound on the Euclidean norm of the coefficient disturbance that is allowed in the proposed I-SM-PUAP algorithm.

The organization of the paper is as follows. In Section 2,
the concept of SMF is briefly described. Section 3 reviews
the SM-PUAP algorithm. In Section 4, we derive the M-SM-
PUAP algorithm. Section 5 presents simulations of the algo-
rithms. Section 6 contains the conclusions.

2. SET-MEMBERSHIP FILTERING (SMF)

The goal of the SMF [2] is to find the filter coefficient vec-
tor \( w \) such that the magnitude of the estimation error is upper
bounded by a prescribed parameter \( \gamma \). Several valid estimates
of \( w \) satisfy the chosen bound \( \gamma \) for the estimation error at
instant \( k \). Let the constraint set \( \mathcal{H}(k) \) denote the set consist-
ing of all vectors \( w \) such that their estimation errors at time
instant \( k \) are upper bounded in magnitude by \( \gamma \), i.e.,
\[
\mathcal{H}(k) = \{ w \in \mathbb{R}^N : |d(k) - w^T x(k)| \leq \gamma \}, \tag{1}
\]
where \( \mathbb{R} \) and the superscript \( T \) denote the real numbers and
transpose operator, respectively. The quantities \( x(k), w, \) and
d \( (k) \) are input vector, weight vector, and desired signal, re-
drespectively. The membership set \( \psi(k) \) is defined as
\[
\psi(k) = \bigcap_{i=0}^k \mathcal{H}(i). \tag{2}
\]

The idea of set-membership recursion techniques is to
adapt the coefficient vector such that it always remain
within the feasibility set. Due to difficulties to compute
\( \psi(k) \), we calculate a point estimate using, for example, the
information provided by the constraint set \( \mathcal{H}(k) \) like in the
set-membership NLMS algorithm [15], or several previous
constraint sets as is done in the set-membership affine projection
algorithm [17].

3. SET-MEMBERSHIP PARTIAL-UPDATE AFFINE
PROJECTION ALGORITHM

In this section we introduce the SM-PUAP algorithm [2].
The main objective in the partial-update adaptation is to per-
form updates in \( M \) out of \( N \) adaptive filter coefficients, where
\( N \) is the order of adaptive filter. The \( M \) coefficients to be
updated at time instant \( k \) are specified by an index set
\( \mathcal{I}_M(k) = \{ i_1(k), \ldots, i_M(k) \} \) with \( \{ i_j(k) \}_{j=1}^M \) chosen from
the set \( \{ 1, \ldots, N \} \). Note that \( \mathcal{I}_M(k) \) varies with the time
instant \( k \). As a result, the \( M \) coefficients to be updated can
change according to the time instant. The choice of which \( M \)
coefficients should be updated is related to the optimization
criterion chosen for algorithm derivation. The SM-PUAP
algorithm [2] takes the update vector \( w(k+1) \) as the vector
minimizing the Euclidean distance \( ||w(k+1) - w(k)||^2 \)
subject to the constraint \( w(k+1) \in \mathcal{H}(k) \) in such a way that
only \( M \) coefficients are updated.

The optimization criterion in the SM-PUAP algorithm is
following described. Let \( \psi^k(k) \) indicate the intersection of
the last \( L \) constraint sets. A coefficient update is implemented
whenever \( w(k) \notin \psi^L(k) \) as follows
\[
\begin{align*}
\text{min} & \ ||w(k+1) - w(k)||^2 \\
\text{subject to} & : \ d_{ap}(k) - X_{ap}^T(k)w(k+1) = \gamma(k) \\
& \ C_{I_M(k)}[w(k+1) - w(k)] = 0
\end{align*}
\tag{3}
\]
where \( d_{ap}(k) \in \mathbb{R}^{L \times 1} \) contains the desired output
from the \( L \) last time instants;
\( \gamma(k) \in \mathbb{R}^{L \times 1} \) specifies the point in \( \psi^L(k) \);
\( X_{ap}(k) \in \mathbb{R}^{N \times L} \) contains the corresponding
input vectors, i.e.,
\[
\begin{align*}
& d_{ap}(k) = [d(k) d(k-1) \cdots d(k-L+1)]^T, \\
& \gamma(k) = [\gamma_0(k) \gamma_1(k) \cdots \gamma_{L-1}(k)]^T, \\
& X_{ap}(k) = [x(k) x(k-1) \cdots x(k-L+1)],
\end{align*}
\tag{4}
\]
with \( x(k) \) being the input-signal vector
\[
x(k) = [x(k) x(k-1) \cdots x(k-N+1)]^T. \tag{5}
\]
Moreover, the matrix \( C_{I_M(k)} = I - C_{I_M(k)} \) is a com-
plementary matrix that gives \( \hat{C}_{I_M(k)} w(k+1) = \hat{C}_{I_M(k)} w(k) \),
which means that only \( M \) coefficients are updated. The
threshold vector elements are such that \( |\gamma_i(k)| \leq \gamma \), for
(4), \( \delta \) and \( I \) are a small positive constant and
\( n \times L \) identity matrix, respectively. The diagonal ma-
trix \( \delta I \) is added to the matrix to be inverted in order to avoid
numerical problems in the inversion operation in the cases
\( X_{ap}(k) C_{I_M(k)} X_{ap}(k) + \delta I \) is ill conditioned.

We can observe that for a fixed value of \( ||e_{ap}(k) - \gamma (k)||^2 \),
the value of \( ||w(k+1) - w(k)||^2 \) is minimized if \( ||P'(k)|| \) is
minimized. As a consequence, a natural choice for the \( M \)
coefficients to be updated are those that will be multiplied
by the elements of \( X_{ap}(k) \) with the largest norm. Figure 1
illustrates the update in SM-PUAP algorithm in \( \mathbb{R}^2 \) for \( L = 1 \).
4. IMPROVED SET-MEMBERSHIP PARTIAL-UPDATE AFFINE PROJECTION ALGORITHM

In this section we propose the I-SM-PUAP algorithm aiming at accelerating the convergence speed of SM-PUAP algorithm and decreasing the number of updates.

Since the partial update strategy deviates the updating direction from the one determined by the input signal vector \( x(k) \) utilized by the SM-PUAP algorithm, it is natural that the size of the step for a partial update algorithm should be smaller than the corresponding algorithm that updates all coefficients. A solution to this problem is to constrain the Euclidean norm of the coefficient disturbance of the partial update algorithm to the disturbance implemented by the originating non partial updating algorithm, in our case the SM-AP algorithm. For that we build hypersphere, \( S(k) \), whose radius is the distance between the \( w(k) \) and SM-AP update. The SM-AP update takes a step towards the hyperplanes \( d(k) - w^T x(k) = \pm \gamma \) with the minimum disturbance, i.e., when the step in the direction \( x(k) \) touches perpendicularly the hyperplane. Therefore, the radius of the hypersphere \( S(k) \) is given by

\[
\mu(k) = \min \left( \frac{w^T(k)x(k) - d(k) \pm \gamma}{\|x(k)\|_2} \right),
\]

where \( \|\cdot\|_2 \) is the Euclidean norm in \( \mathbb{R}^N \). The equation describing the hypersphere \( S(k) \) with the radius \( \mu(k) \) centered at \( w(k) \) is as follows

\[
(w_1 - w_1(k))^2 + \cdots + (w_N - w_N(k))^2 = \mu^2(k).
\]

As can be observed in Figure 1, \( w(k + 1) \) is the point where, starting from \( w(k) \), the vector representing the \( w(k + 1) \) direction touches the hyperplane \( d(k) - w^T x(k) = \gamma \). Unlike the SM-PUAP algorithm, in the I-SM-PUAP algorithm \( w(k + 1) \) is the point where, starting from \( w(k) \), the vector representing the partial direction touches the defined \( N \) dimensional hypersphere \( S(k) \) and points at a sparse version of \( x(k) \). The process of deriving the I-SM-PUAP algorithm is described in Figure 2.

Define \( \hat{w}(k) \) as the update result of equation (6) with \( \hat{\gamma}(k) = [0 \cdots 0]^T \). In order to find the update of \( w(k) \) to the boundary of hypersphere \( S(k) \) such that \( C_{\mathcal{I}_M}(k)w(k + 1) = C_{\mathcal{I}_M}(k)w(k) \) we have to find the intersection of hypersphere \( S(k) \) with the line \( l(k) \) passing through \( w(k) \) and \( \hat{w}(k) \). This line, shown in Figure 2, is parallel to the vector \( u(k) = a(k) / \|a(k)\|_2 \), where \( a(k) = [\hat{w}_1(k) - w_1(k) \cdots \hat{w}_N(k) - w_N(k)]^T \). Hence, the equation of the line \( l(k) \) is given as follows

\[
\begin{cases}
  w_1 - w_1(k) = \cdots = w_N - w_N(k) / u_N(k), & \text{for } i \in \mathcal{I}_M(k) \\
  w_i = w_i(k), & \text{for } i \notin \mathcal{I}_M(k)
\end{cases}
\]

In order to find the intersection of the line \( l(k) \) with the hypersphere \( S(k) \), we should replace equation (12) in equation (11). Thus, we will attain \( w_i = w_i(k) \) for \( i \notin \mathcal{I}_M(k) \), and for \( i \in \mathcal{I}_M(k) \) we have

\[
\frac{u_1^2(k)}{u_1^2(k)}(w_i - w_i(k))^2 + \cdots + \frac{u_N^2(k)}{u_N^2(k)}(w_i - w_i(k))^2 = \mu^2(k).
\]

Then,

\[
(w_i - w_i(k))^2 = u_i^2(k)\mu^2(k),
\]

where we obtained the last equality owing to \( \|u(k)\|_2 = 1 \). Therefore, the intersections of the line \( l(k) \) and the hypersphere \( S(k) \) are given by

\[
w_i = w_i(k) \pm u_i(k)\mu(k).
\]

We will choose the positive sign in equation (15) since the direction of the vector \( a(k) \) is from \( w(k) \) to \( \hat{w}(k) \). As a result, the vector \( w(k + 1) \) becomes as below

\[
w(k + 1) = w(k) + \mu(k)u(k).
\]

Also, as an alternative method, we can get \( w(k + 1) \) through an elegant geometrical view. Denote \( w(k + 1) \) in equation (6) as \( \tilde{w}(k) \) while taking \( \hat{\gamma}(k) = [0 \cdots 0]^T \). Define \( a(k) \) as

\[
a(k) = \tilde{w}(k) - w(k) = C_{\mathcal{I}_M}(k)x_{ap}(k)\Phi(k)e_{ap}(k).
\]

Since we want the update up to the boundary of hypersphere \( S(k) \) centered at \( w(k) \) with radius \( \mu(k) \) in the direction of \( a(k) \), we get the update equation as follows

\[
w(k + 1) = w(k) + \mu(k) \frac{a(k)}{\|a(k)\|_2} = w(k) + \mu(k)u(k).
\]

Table 1 summarizes the I-SM-PUAP algorithm.
The correlated input signal is chosen as \( x(k) = 0.95x(k-1) + 0.19x(k-2) + 0.09x(k-3) - 0.5x(k-1) + m(k-4) \), where \( m(k) \) is a zero-mean Gaussian noise with unit variance.

The average number of updates performed by the I-SM-PUAP algorithm are 8.3%, 6.5%, and 2% for \( L = 2, 5, \) and 70, respectively, and 20% in the case of correlated input signal. The average number of updates implemented by the SM-PUAP algorithm are 14% and 25% for \( L = 70 \) and 65, respectively. Note that in both algorithms we have to find the inverse of an \( L \times L \) matrix, thus large \( L \) implies high computational complexity. Therefore, the I-SM-PUAP algorithm has lower computational complexity since it presents fast convergence even for small value of \( L \). Also, it is worth mentioning that for \( L < 65 \) the SM-PUAP algorithm does not reach its steady-state in 10000 iterations. From the results, we can observe that the proposed algorithm, I-SM-PUAP, has faster convergence speed, lower number of updates, and lower computational complexity as compared to the SM-PUAP algorithm.

5. SIMULATIONS

In this section, the SM-PUAP algorithm [2] and the proposed I-SM-PUAP algorithm are applied to a system identification problem. The unknown system has order \( N = 80 \) and its coefficients are random scalars drawn from the standard normal distribution. The input signal is zero-mean Gaussian noise with \( \sigma_0^2 = 1 \). The signal-to-noise ratio (SNR) is set to 20 dB, i.e., \( \sigma_0^2 = 0.01 \). The bound on the output estimation error is chosen as \( \gamma = \sqrt{25 \sigma_0^2} \). Also, we adopt the threshold bound vector \( \gamma_k(k) = \gamma_0(k) = \frac{\sigma_0(k)}{|\nu_0(k)|} \) and \( \gamma_i(k) = d(k) - w^T(k)x(k-i) \) for \( i = 1, \ldots, L-1 \) [2, 23]. The regularization constant, \( \delta \), is set to \( 10^{-12} \) and \( w(0) = [1 \cdots 1]^T \) which is not close to the unknown system. All learning curves averaged over 200 trials. We are updating 50 percent of the components randomly chosen of the filter to illustrate the partial updating, i.e., half of the elements of \( \Theta_M(k) \) are nonzero at each time instant \( k \). Figure 3 shows the learning curves for the I-SM-PUAP algorithm with \( L = 2, 5, \) and 70, and the learning curves for the SM-PUAP algorithm with \( L = 65 \) and 70. Also, in Figure 3 a blue curve is depicted using correlated inputs and \( L = 2 \). In fact, for the blue curve all of the specifications of the system are the same as explained above and the only difference is the input signal. The correlated input signal is chosen as \( x(k) = 0.95x(k-1) + 0.19x(k-2) + 0.09x(k-3) - 0.5x(k-1) + m(k-4) \), where \( m(k) \) is a zero-mean Gaussian noise with unit variance.

6. CONCLUSIONS

We have introduced an improved set-membership partial-update affine projection (I-SM-PUAP) algorithm aiming at accelerating the convergence rate of the set-membership partial-update affine projection (SM-PUAP) algorithm, with lower computational complexity and reduced number of updates. In order to achieve this goal, we use the distance between the present weight vector and the one obtained with the SM-AP update, in order to provide a hypersphere that upperbounds the coefficient disturbance. Numerical simulations for system identification problem have confirmed that the I-SM-PUAP algorithm has not only faster convergence rate, but also lower computational complexity and lower number of updates as compared with the previously proposed SM-PUAP algorithm.
7. REFERENCES


