USE OF THE AVAILABLE INFORMATION FOR AN ADEQUATE STEP-SIZE ACTUALIZATION

Gregori Vázquez and Antoni Gasull.
E.T.S. Ingenieros de Telecomunicación. U.P.C.
C/ Jordi Girona Salgado, s/n. 08034 Barcelona, Spain.

This paper deals with the use of the available information for the update of the step-size in gradient-based adaptive methods. Several proposals are made and due to the relatively high computational cost, more efficient frequency domain versions are studied. Unconstrained and constrained minimum mean square error algorithms are both considered.

I. INTRODUCTION

A number of algorithms and methods are formulated in terms of minimum mean square error or as constrained least mean-square problems. These two alternatives lead to two respective optimal solutions. As it is well known, under actual situations, very usually, the exact or block methods are substituted by adaptive approximated ones, with a sensitivity to computation error propagations and non-stationary conditions better covered. Besides, very often, the adaptive algorithms assume a great computation cost reduction. Thus, the election or the design of an algorithm will be influenced by several pragmatic aspects, that they are, mainly, the available information and the computation capability among others. Very common options are the gradient based algorithms and the recursive methods, which main difference is the kind of objective they try to minimize. Gradient methods used to follow from stochastic error cost functions instead of the recursive ones, which usually minimize deterministic objectives that are exactly reached in each iteration. Any case, from a unified point of view, it is possible to see all them as gradient methods with the particularity that the recursive algorithms include an adaptive step-size and very often they try 'to focus' the gradient to the direction of the minimum for quadratic function.

The main objective of the paper will be the discussion of the step size updating in such kind of adaptive gradient-based algorithms. Besides, frequency-domain studies are also included.

For simplicity, we analyze the m.m.s.e. linear filtering problem, leaving the constrained LMS algorithm for the moment. From figure 1, we dispose of an estimate \( \hat{y}(n) \) of a reference \( y(n) \) at the output of a \( Q \) order FIR filter determined by a weight vector \( w \) and a data vector \( x_n \), given by:

\[
\hat{y}(n) = w^T_n x_n
\]

This work has been supported by CAICYT by grant number 21096/84.

\[
\begin{align*}
\mathbf{x}_n^T &= (x(n), x(n-1), \ldots, x(n-Q+1)) \\
\mathbf{w}^T &= (w(0), w(1), \ldots, w(Q-1))
\end{align*}
\]

The mean square error (m.m.s.e.) between the reference \( y(n) \) and the filter output \( \hat{y}(n) \) is expressed by:

\[
\varepsilon_F^2 \triangleq E[(y(n) - \hat{y}(n))^2] = \mathbf{e}^T \mathbf{e}
\]

For obtaining the optimum solution, we determined the gradient of this cost function with respect to the weights:

\[
\nabla \varepsilon_F^2 = R_{xx} w - P
\]

with:

\[
R_{xx} = E x_n x_n^T \\
P = E y(n)x_n
\]

and setting expression (1.3) to vector zero, we reach the well known optimal Wiener solution given by:

\[
\mathbf{w}_{opt} = R_{xx}^{-1} P
\]

Often, as has been said, (1.4) is substituted by adaptive gradient-based solutions. The most familiar approach is the Steepest Descent Method based in the gradient (1.3) of the filtering error function (1.2):

\[
\mathbf{w}_{n+1} = \mathbf{w}_n - \eta \varepsilon_F^2
\]

Another proposal, that we want to remark, differs in the minimization objective. Now, the error is the m.s.e. associated to the weights, defined by:

\[
\varepsilon_{mss}^2(n) = E[(w - \mathbf{w}_{opt})^2]
\]

Obviously, the solution to (1.6) is the same that for (1.2), that is, equation (1.4). Nevertheless, important differences can be found between this two cost functions. For (1.6) the gradient with respect the weight vector can be expressed as:
\[ V^{-2}(n) = R^{-1} \begin{bmatrix} x_x & x_y \\ x_y & y_n \end{bmatrix} \] (1.7)

that is:
\[ V^{-2}(n) = R^{-1} V^{-2}(n) \] (1.8)

The main goal of this variable parameter would be the determination how much noisy the successive gradient estimates are, and any case, then decide how much the solution in this direction will be moved.

This idea is very clearly exhibited by the Recursive Least Square method. This algorithm adopts as gradient an instantaneous version of (1.7), that is:
\[ \hat{V}^{-2}(n) = R^{-1}(n) \hat{e}(n+1) \hat{e}^T(n+1) \] (1.13)
with:
\[ e(n+1) = y(n+1) - X^T(n+1) \hat{\theta} \] (1.14)
\[ R_{xx}(n) = \sum_{i=0}^{n} x_i x_i^T \] (1.15)

We see as the method rejects (1.13) (1.15) the last sample \( \hat{e}(n) \) for the evaluation of the autocorrelation matrix inverse \( R^{-1} \) in the actualization of the weight \( \hat{\theta} \). It will be in the next step when this sample will be included. From the matrix inversion Lemma:
\[ R^{1}_{xx}(n+1) = R^{1}_{xx}(n) - R^{1}_{xx}(n) x_{n+1} x_{n+1}^T R^{1}_{xx}(n) \] (1.16)

and it is here where we see how the method decides from \( R^{1}_{xx}(n) \) how good the sample \( x_{n+1} \) is. This is made by the denominator term of (1.16) where the algorithm includes the variable:
\[ e(n+1) = R^{1}_{xx}(n) \] (1.17)

If \( x_{n+1} \) is assumed a zero mean jointly Gaussian random vector with covariance matrix \( R_{xx}(n) \), we can give to \( \hat{e}(n) \) the meaning of a certain measure of how the new sample \( \hat{e}(n) \) fits the Gaussian process determined by \( R_{xx}(n) \).

Leaving the discussion of how the focusing matrix is evaluated, the next aspect we want to remark is that the method exhibits very good performances in spite of the fact the gradient it uses is purely noisy (1.13). Thus, the parameter that is really reducing a possible erratic behaviour is the step size. For obtaining this value, we will use the deterministic accumulated filtering error:
\[ F^{2}(k) = \sum_{i=0}^{k} (y(i) - \hat{y}(i))^2 \] (1.18)

and assuming that \( \hat{W} \) is the weight vector that minimizes (1.18) for kon the parameter \( \mu \) will be chosen such that \( \hat{W} \) minimizes (1.18) for \( \mu \). We get that for \( \mu \) in expression (1.9) with (1.13):
\[ \mu = \frac{1}{R^{1}_{xx}(n)} \] (1.19)
and if the process is ergodic, objective (1.18) ensures the convergence to (1.4) in the limit. We see again that the method introduces the term (1.17) to decide how much
I. Step-size evaluation

The central idea of the paper is to use similar considerations with a more simple scheme as (1.5), where the gradient moves the solution in a direction that locally is only of maximum filtering error slopes. In our opinion, these are the possible objectives that could substitute (1.18) in this case, they are:

a. The step-size will be evaluated such that the filtering error is minimized in some statistical sense.

b. The step size will try to approach the coefficient vector to the optimum (1.4) in each update, under some quadratic weight error objective.

c. For a cost function as the proposed in (1.2) it is always possible to find as much directions as the filter error such that the gradient (1.3) really is well oriented to the minimum of the quadratic performance curve. This directions correspond to the main axis of expression (1.2). Thus, the step size would try to move the instantaneous solution to this main axis, establishing theoretically an exact inversion method in a finite number of steps.

It is easy to see how these three objectives are completely equivalent when the scheme is the reflected in (1.3) but really different when the updating is made thought (1.5).

The next paragraph will present expressions for each of the three proposals and besides, several efficient alternatives in the frequency domain thought circular, linear or both simultaneous convolutions will be discussed.

2. TIME DOMAIN OPTIMUM STEP-SIZE EXPRESSIONS /4/ /9/

For convenience let's consider the generalized cost function given by:

\[ \bar{z}^2(n) = \left( \bar{x} - \bar{x}_n \right) \text{cov}^{-1} \left( \bar{x} - \bar{x}_n \right)^T + \epsilon^2 \]  (2.1)

This objective includes the previous (1.2) for K=0 and (1.6) for \( K=1 \). Besides, it will be shown to be very useful for the evaluation of the three optimum expressions for the step sizes.

For any \( K \) value, the step size that minimizes (2.1) for the updating expression under analysis (1.5) is shown to be:

\[ K = \frac{T - n \text{cov}^{-1} \bar{x} \text{cov}^{-1} \bar{x}}{T - n \text{cov}^{-1} \bar{x}} \]  (2.2)

where \( \bar{x} \) denotes the filtering error gradient \( (1.3) \). Next, we will see that three particular \( K \) values for (2.2) give us the three solutions we are looking for.

In filtering problem, case (a.-) seems the most adequate. If we minimize the filtering error step by step, then (1.6) for \( K=0 \), and then, the step size we will use is:

\[ \epsilon = \frac{T}{\bar{x} \text{cov}^{-1} \bar{x}} \]  (2.3)

This expression is already known since it is used in accelerated adaptive methods as Powell's one, or conjugated gradient algorithm, among others. In this case, the step size moves the coefficient vector as far as an iso-error curve is reached tangently (see fig. 2).

In the other hand, in identification problem or when we dispose of previous information about the input data autocorrelation matrix we could propose as objective the minimization of the weight error, that corresponds to (2.1) for \( K=1 \). Then, in this case, the best step size would be the following:

\[ \epsilon = \frac{T}{\text{cov}^{-1}} \]  (2.4)

This results corresponds to a normalized projection of the focused gradient (1.7) over the filtering error gradient (1.3). The evolution in a bidimensional case is shown in figure 3.

Finally, if we want to move the weight vector to the main axis associated to the largest eigenvalue we could take the limit \( K \rightarrow \infty \) over expression (2.1). The best step size is then:

\[ \epsilon = \frac{T}{\text{cov}^{-1} \bar{x} \text{cov}^{-1} \bar{x}} \]  (2.5)

Assuming that \( \text{cov} \) has \( N \) different eigenvalues that conveniently ordered:

\[ \lambda_1 > \lambda_2 > \ldots > \lambda_N \]  (2.6)

modal decomposition of expression (2.5) shows that successive application of (1.3), (2.5) and (1.5) supplies the following sequence for the step size (2.5):

\[ \epsilon = \frac{1}{\lambda_1^2} \frac{1}{\lambda_2^2} \ldots \frac{1}{\lambda_N^2} \]  (2.7)

that is, the eigenvalue decomposition of matrix \( \text{cov} \). Besides, each iteration matches the node associated to the largest eigenvalue not already adopted. Then, the method theoretically, converge in a finite number of \( N \) iterations.

Once the optimum expressions for the step-size parameter have been founded, we see that the computation would require a high number of operations, and this is specially clear in (2.4), where a matrix inversion is required. Besides, for the whole of cases, it will be necessary a good estimate of matrix \( \text{cov} \) and of the filtering error gradient (1.3). Both considerations make the frequency domain implementation very adequate as it will be seen in the next paragraph.
3. FREQUENCY DOMAIN IMPLEMENTATIONS

When frequency domain adaptation is adopted, there are the alternatives we dispose. They are referred basically to the kind of convolution the filter will compute. They are circular, linear or unconstrained, that is, either circular or linear, whichever best minimizing the mean-square error. Here, circular case is commented.

Let's define $F$ as the N×N DFT operator matrix, such that $F^*F = NI$. As algorithms will adapt in the transformed domain, the DFT of the weight vector is computed:

$$\tilde{W}(n) = F^{-1} W(n)$$

and $X(n)$ will be the N×N diagonal matrix generated from the DFT of the input data vector:

$$X(n) = \text{diag.}(F x(n))$$

The transformed filter output vector $\tilde{Y}(n)$ is defined by:

$$\tilde{Y}(n) = X(n) \tilde{W}(n)$$

and the error $E(n)$:

$$E(n) = Y(n) - \tilde{Y}(n)$$

where $Y(n)$ is the DFT of the desired vector.

The transformed weight vector is actualized by:

$$\tilde{W}(n+1) = \tilde{W}(n) + \gamma(n)X(n)E(n)n$$

It easy to see that in the time-domain (3.5) is given by:

$$W(n+1) = W(n) + \gamma(n)y(n)X(n)w(n)$$

with

$$X(n) = F^{-1} X(k) F$$

the circulant data matrix. Expression (3.6) shows that method uses a certain averaging of the instantaneous gradients, evaluated with successive circulant data vectors. Considering the quadratic function:

$$\tilde{J}_F = \frac{1}{2} (y(k) - \tilde{y}(k))^T (y(k) - \tilde{y}(k))$$

and taking into account that $F^*F = NI$, we can express (3.8) as:

$$\tilde{J}_F = \frac{1}{2} (\tilde{Y}(n))^T W(n) \tilde{Y}(n)$$

where:

$$R_f = E(X(n)X(n))$$

(3.10)

The main advantage of this transformed domain is basically that matrix $R_f$ is diagonal. Thus the evident option would be to use a diagonal matrix structure for the step-size, updating each bin as a function of the mean power of it.

Our proposal is to obtain the expression for $\gamma$ in (3.5) such that minimizes (3.9), that is:

$$\begin{align*}
\gamma & = \frac{E(\tilde{Y}(n)X(n))^T (\tilde{Y}(n) - \tilde{Y}(n)))}{E(\tilde{Y}(n)X(n))^T X(n)E(n))} \\
& \text{or taking instantaneous values:} \\
\gamma & = \frac{E(\tilde{Y}(n)X(n))^T (\tilde{Y}(n) - \tilde{Y}(n)))}{E(\tilde{Y}(n)X(n))^T X(n)E(n))} \\
& \text{Simpler than (3.13) is step-size found when it minimizes} \\
& \tilde{J}_F = \frac{1}{2} (\tilde{Y}(n) - \tilde{Y}(n))^T (\tilde{Y}(n) - \tilde{Y}(n))) \\
& \text{where:} \\
\gamma_{\text{opt}} & = \frac{1}{2} \frac{1}{\text{opt}} \\
& \text{The expression obtained is:} \\
\gamma & = \frac{E(\tilde{Y}(n)X(n))^T (\tilde{Y}(n) - \tilde{Y}(n)))}{E(\tilde{Y}(n)X(n))^T X(n)E(n))} \\
& \text{and taking instantaneous estimates:}
JOC—

Tre where: J. retrnd. Tre prior information

Any case, frequency domain is always attractive because it will improve convergence rate due eigenvalue spread is reduced:

\[ \lambda_{\text{max}} = \max \left| \lambda_w(x) \right|^2 \]

\[ \lambda_{\text{min}} = \min \left| \lambda_w(x) \right|^2 \]

(3.18)

cost that is specially important when filtering error gradient-based methods are adopted.

4. SIMILARITIES WITH THE CONSTRAINED LEAST MEAN-SQUARES.

The versatile constrained LMS algorithm is formulated as a minimization of the filter output mean power subject to a set of linear constraints, that often reflect prior information available. Then we have to:

\[ \text{minimize} \quad W^T R W \]

subject to \( C^T W = 0 \)  

(4.1.a)

(4.1.b)

The problem is solved through Lagrange multipliers method. The cost function is:

\[ H(W) = W^T R W + \frac{1}{\lambda} (C^T W)^T (C^T W) \]

(4.2)

where: \( \lambda \) is the vector formed by the Lagrange multipliers. By setting the gradient of (4.2) to zero and imposing the final solution to satisfy the constraints (4.1.b), we get the optimum:

\[ W_{\text{opt}} = \lambda^{-1} C^T (R^{-1} C) \]

(4.3)

Again, adaptive schemes used to be adopted. The simplest one is, as usual, the Steepest Descent:

\[ W_n = W_{n-1} - \mu \frac{\partial H(W)}{\partial W} \]

(4.4)

expression quite equivalent to previous (1.5). The gradient expression is function of the Lagrange multiplier vector \( \lambda(x) \). This vector is chosen such \( \lambda(x) \) satisfy the constraints (4.1.b). Finally, the algorithm is founded to be:

\[ W_n = W_{n-1} - \mu (y(n)x_n) \]

(4.5)

with

\[ y(n) = y(n)x_n \]  \text{(FILTER OUTPUT)}

\[ A = 1 - C(C^T C)^{-1} C^T \]

\[ F = (C^T C)^{-1} F \]

(4.6)

Again, it is easy to show how matrix \( R^{-1} \) could focus gradient \( H(W) \) of function (4.2) to the optimum. The improved algorithm is as in (1.9):

\[ W_n = W_{n-1} - \mu \frac{\partial H(W)}{\partial W} \]

This both problems are shown to be quite equivalent and this suggests to use the same ideas in this algorithm. Expression for the actualization of the step size have been found and will be presented in further papers.

5. CONCLUSIONS

The paper has presented several time and frequency domain expressions for the step size actualization in gradient-based algorithms. Besides, frequency domain seems to be much more adequate to find attractive expressions, more simple and ensuring that in this domain, conditions for the convergence are better.

6. REFERENCES


