Signal Processing 117 (2015) 188–191

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Fast communication

# An algorithm based on non-squared sum of the errors

Cristiane da Silva [b], Allan Kardec Barros [b], Ewaldo Santana [b], Marcos A.F. de Araújo [b], Marcus V. de S. Lopes [b], João V. Fonseca [b], Jose C. Principe [a]

[a] Computational NeuroEngineering Laboratory, University of Florida, United States
[b] Lab. for Biological Information Processing, São Luís, MA, Brazil

ARTICLE INFO

*Article history:*
Received 30 June 2014
Received in revised form
5 March 2015
Accepted 7 March 2015
Available online 20 March 2015

*Keywords:*
RLS
Adaptive filtering
Non-quadratic function
Convergence speed

ABSTRACT

In adaptive filtering, several algorithms are developed in the quest for greater convergence speed, mostly relying on second order statistics. Here we modify the Recursive Least Square (RLS) equations by using as performance surface a weighted sum of even error power. As a result, the equations turn out to be simple, elegant, while yielding faster convergence and preserving the computational cost when compared with the existing RLS algorithm.



## 1. Introduction

The least mean square (LMS) and the recursive least square (RLS) algorithms have been the hallmark of adaptive filtering, but they have been developed under two very different approaches. The first is a stochastic approximation of gradient descent and the latter is an on-line mathematical recursion of Wiener's solution that works with growing data. Normally one stresses as a major difference between the two algorithms the use of second order approximation in the RLS, i.e. the RLS implements an on-line Newton-like gradient search using a different stepsize for each direction. But the fact of the matter is that the instantaneous error never appears explicitly in the RLS algorithm, so the difference is deeper than the inclusion of second order information in the formulation. And this is unfortunate, because the error brings instantaneous information about the parametric fitting of the model to each sample of the time series. This is the reason that the LMS is a fast tracker, and why RLS has difficulty in tracking as it is well known in the literature [1]. Moreover the selection of the forgetting factor in RLS is most of the times heuristic, although old work such as [2] has recognized its importance to fit AR models to speech data.

This paper brings the error explicitly to the RLS algorithm formulation by proposing a new cost function that preserves the mean squared-error (MSE) solution, but allows for the exploitation of higher order moments of the error to speedup the convergence of the RLS algorithm. The idea is based on Widrow's least mean square fourth (LMF) algorithm [3], where the dynamics of learning are dissociated from the MSE solution. The geometry of the problem tells it all: the quadratic error cost function and the fourth power error cost function have the same global minimum, but in most of the search space, the fourth power has a higher slope (except near the neighborhood of the minimum), therefore the speed of adaptation is faster. The instantaneous error was used in [4], which developed an RLS algorithm. Chambers and Colleagues [5] proposed the Least Mean Mixed-Norm (LMMN) Algorithm, which combines the LMS and LMF algorithms, and also used the instantaneous error.

Previous work has also attempted to include the instantaneous error in the RLS algorithm, but the methods have been heuristic. An automatic gain control (AGC) scheme realized by estimating the cross correlation between the error and the input signal was included in [6]. In [7] a technique to improve the convergence rate of RLS algorithm is presented, where the

http://dx.doi.org/10.1016/j.sigpro.2015.03.012
0165-1684/

gain vector takes into consideration the value of $|1/\xi(n)|$, where $\xi(n)$ is a priori estimation of the error. Our work is different because its main goal is to create from first principles (new cost functions) a mechanism to include instantaneous error information in the RLS algorithm, make it track better, and allow for the design of an adaptive forgetting factor. As we will see the key aspect of our approach is to include the error in the "Kalman gain" that effectively controls the speed of adaptation of the RLS algorithm. The paper is organized as follows: Section 2 presents the proposed algorithm, Section 3 shows the simulations results and Section 4 presents discussion and conclusion.

## 2. Proposed algorithm

The basic structure of an adaptive FIR filter is composed of a desired signal $d_i$, an input vector $\mathbf{u}_i=[u_i,u_{i-1},\ldots,u_{i-L+1}]^T$, and an error $e_i$, which is used to update the filter weight vector $\mathbf{w}_n=[w_{0,n},w_{1,n},\ldots,w_{L-1,n}]^T$. The goal is to recover an approximation of $d_i$ by estimating the output signal $y_i=\mathbf{w}_{n-1}^T\mathbf{u}_i$, after calculating the error $e_i=d_i-y_i$, where $n$ represents the current sample with $1\le i\le n$ and $L$ is the filter length.

The famed RLS algorithm is an on-line implementation of the Wiener solution [1], which solves the regression problem in functional spaces. All these algorithms optimize the filter coefficients to minimize the mean square error cost function. In this work, we propose to use a weighted sum of even power of the error, $J=\sum_{j=1}^{m}k^{m-j}E[\lambda e^{2j}]$, leading to the empirical risk,

$$J_n=\sum_{j=1}^{m}k^{m-j}\sum_{i=1}^{n}\left\{\lambda^{n-i}[e_i]^{2j}\right\}, \tag{1}$$

where $j$, $m$, $k$ and $n$ are positive integers, subject to design. Note that when $j=1$, $k=1$, and $m=1$ we obtain the MSE cost function. The exponential weighting factor, $\lambda$, is the conventional forgetting factor that normally is selected close to, but less than one. The weighting term $k^{m-j}$ is just a multiplicative term that plays a role of accelerating the convergence. The more interesting terms are $j$ and $m$ which are related to the cost function. Note that $j>1$ effectively chooses an even power of the error and $m$ adds the different powers of the error, effectively meaning that the empirical cost function becomes a sum of $m$ even powers of the error.

If we remove the sum in (1), the weighting term $k^{m-j}$ and for $j=2$, we will have the cost function that becomes Widrow's Least mean fourth (LMF) algorithm. Basically the LMF displays a large slope when compared with the MSE cost function in most of the areas of the parameter space, except in a neighborhood of the minimum. The larger the $j$ the steeper is the cost function and the higher is the gradient, except near the minimum where the slope becomes essentially flat. Therefore, a gradient descent algorithm working with a single value of $j$ will face an intrinsic compromise of very quickly decreasing the error, but then crawling to the minimum value. This intuition can be translated mathematically since when increasing the power of the error the eigenvalue spread of the input autocorrelation matrix increases proportionally.

In [4], we have shown that the cost function with $k=1$, $m=1$ and $j>1$ in fact supports an RLS like algorithm that has faster convergence than the RLS algorithm. But of course the sole use of one single power of the error is not optimal as we discussed. We now show how Eq. (1) can be used to derive a new family of even faster RLS type algorithms.

In order to get the optimum weight vector $\hat{\mathbf{w}}_n$, we calculate the instantaneous gradient of $J_n$ as $\nabla J_n=\sum_{j=1}^{m}\{-2\cdot a_j\sum_{i=1}^{n}[\lambda^{n-i}\cdot e_i^{\alpha_j}\cdot e_i\cdot\mathbf{u}_i]\}$, where $a_j=j\cdot k^{m-j}$ and $\alpha_j=2j-2$. Now, equating $\nabla J_n$ to zero, we define the optimum value of the weight vector $\hat{\mathbf{w}}_n$ by the matrix equation: $\hat{\mathbf{w}}_n=\Phi_n^{-1}\mathbf{z}_n$, where the L-by-L correlation matrix $\Phi_n$ and the L-by-1 cross-correlation vector $\mathbf{z}_n$ are now defined by

$$\Phi_n=\sum_{j=1}^{m}\left\{a_j\sum_{i=1}^{n}\left[\lambda^{n-i}\cdot e_i^{\alpha_j}\cdot\mathbf{u}_i\mathbf{u}_i^T\right]\right\},$$

$$\mathbf{z}_n=\sum_{j=1}^{m}\left\{a_j\sum_{i=1}^{n}\left[\lambda^{n-i}\cdot e_i^{\alpha_j}\cdot d_i\cdot\mathbf{u}_i\right]\right\}. \tag{2}$$

Notice that both the autocorrelation function and the cross-correlation vectors are now an explicit function of the instantaneous error unlike the conventional quantities in adaptive filtering. This means that the instantaneous error is now affecting the shape of the performance surface, but as we shall prove below the optimal solution is still independent of the error and remains the Wiener solution. This means that we have gained further control on the dynamics of learning without affecting the final solution.

Isolating the term corresponding to $i=n$ from (2) we obtain

$$\Phi_n=\lambda\Phi_{n-1}+\left[\sum_{j=1}^{m}\left\{a_j\cdot e_n^{\alpha_j}\right\}\right]\mathbf{u}_n\cdot\mathbf{u}_n^T,$$

$$\mathbf{z}_n=\lambda\mathbf{z}_{n-1}+\left[\sum_{j=1}^{m}\left\{a_j\cdot e_n^{\alpha_j}\right\}\right]d_n\cdot\mathbf{u}_n, \tag{3}$$

where $\Phi_{n-1}$ is the "old" value of the correlation matrix and $\mathbf{z}_{n-1}$ is the "old" value of the cross-correlation vector.

In order to compute the optimal estimate $\hat{\mathbf{w}}_n$ using the RLS algorithm, we have to determine the inverse of the correlation matrix $\Phi_n$, and will also use a basic result in matrix algebra known as the *matrix inversion lemma*. When applying this lemma as shown in [1] to $\Phi_n$ in (3) we obtain

$$\Phi_n^{-1}=\lambda^{-1}\Phi_{n-1}^{-1}-\frac{\lambda^{-1}\Phi_{n-1}^{-1}\mathbf{u}_n\mathbf{u}_n^T\Phi_{n-1}^{-1}}{\lambda\cdot\left[\sum_{j=1}^{m}\left\{a_j\cdot e_n^{\alpha_j}\right\}\right]^{-1}+\mathbf{u}_n^T\Phi_{n-1}^{-1}\mathbf{u}_n}, \tag{4}$$

where $\mathbf{P}_n=\Phi_n^{-1}$ and the vector,

$$\mathbf{g}_n=\frac{\mathbf{P}_{n-1}\mathbf{u}_n}{\lambda\cdot\left[\sum_{j=1}^{m}\left\{a_j\cdot e_n^{\alpha_j}\right\}\right]^{-1}+\mathbf{u}_n^T\mathbf{P}_{n-1}\mathbf{u}_n}, \tag{5}$$

is referred to as the Kalman gain vector. When comparing this result with the conventional gain in the RLS algorithm, it is clear that the error term affects directly the forgetting factor. Therefore we can think that our approach effectively modulates the forgetting factor with the instantaneous error information, which makes the gain not only dependent upon the input signal dynamics, but also a function of the desired response through the error. Defining $\mathbf{P}_n=\lambda^{-1}(\mathbf{P}_{n-1}-\mathbf{g}_n\mathbf{u}_n^T\mathbf{P}_{n-1})+\beta\mathbf{I}$, where $\beta$ is a very small constant to avoid singularities. We can