Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2006) 9–13
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Using non-linear even functions for error minimization in adaptive filters

Allan Kardec Barros[a,*], Jose Principe[b], Yoshinori Takeuchi[c], Noboru Ohnishi[c]

[a]*Universidade Federal do Maranhao, Brazil*
[b]*University of Florida at Gainsville, USA*
[c]*Nagoya University, Japan*

Available online 18 August 2006

Abstract

In this work, we analyze algorithms for adaptive filtering based on non-linear cost function of the error, which we named *non-linear even moment* (NEM) algorithms. We assume that this non-linear function can be generally described in a Taylor series as a linear combination of the even moments of the error. NEM is a generalization of the well-known *least mean square* (LMS). We study the NEM convergence behavior and derive equations for misadjustment and convergence. We found a good approximation for the theoretical results and we show that there are various combinations of the even moments which yields better results than the LMS as well as other algorithms proposed in the literature.

Keywords: Non-linear error; Least mean square

1. Introduction

In many signal processing applications using adaptive filtering, there is a need of algorithms that yield small error, fast convergence and low computational complexity. Usually, these algorithms are analyzed under a framework where a number of linearizations are carried out, so that one can easily access both the convergence time and the misadjustment error. Moreover, underlying these methods are some assumptions about the statistics of the signals under study. This yields important simplifications in the analysis of the algorithm. However, those linearizations and assumptions may oversimplify the problem or hide important properties of the algorithms.

Among the adaptive filters, the *least mean square* algorithm (LMS) of Widrow and Hoff [5] appears as one of the most widely used. The LMS belongs to a class of algorithms that can be designated as *second order statistics* (SOS), in opposition to *higher order statistics* (HOS). The use of SOS methods are sufficient when the signals involved in the application are Gaussian distributed, yielding a number of simplifications in the algorithm analysis, as well as leading to computationally less expensive methods.

Interestingly, probably due to the increase in the computational power in the last decades, HOS methods have drawn more attention of the research community. Indeed, instead of dealing only with the signal's power (i.e., SOS), HOS allows access to the information contained in all moments of the signal [6], yielding therefore a better approximation of the actual distribution of the signal under study. As a result, one can expect that algorithms designed under the HOS framework behave more efficiently.

An interesting idea would be to explore the HOS of the error, such as carried out in the works of Walach and Widrow [7], Chambers et al. [1] or Erdogmus et al. [3]. There is an interesting property which is: the mean of the error raised to even powers is a convex function of the weight vector. This can be interpreted as the error cannot have local minima [4]. Here we generalize the work of Chambers et al. [1], that proposed a weighted sum of the moments of order two and four. The idea behind the sum

*Corresponding author.
E-mail addresses: akbarros@ieee.org, allan@dee.ufma.br (A.K. Barros).

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.07.001

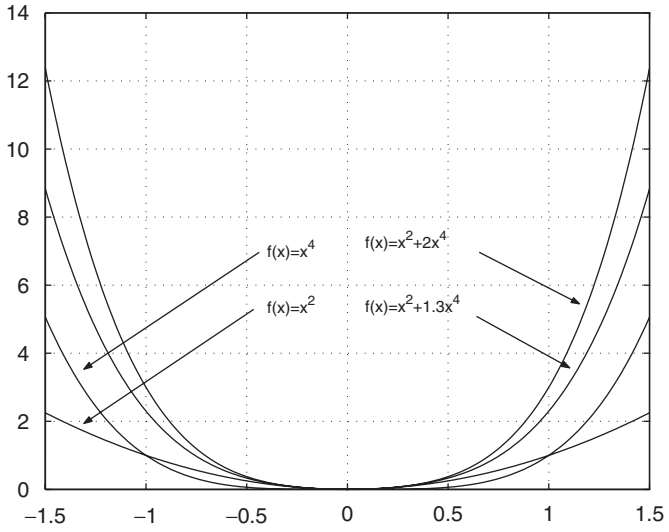


Fig. 1. Here we show the behavior of four functions to illustrate the idea of adding power functions. One is the usual square, another is the variable to the fourth, and a third is a weighted sum of the previous two. One can see that around zero the function has less variance in the case of x^2, while x^4 has a large drop as it gets far from zero. The idea in this work is to use the advantages of both, as shown in the third and fourth functions.

of errors is that one can have the good behavior of the second order moment in steady state allied to the fast convergence of higher order even moments, as shown in Fig. 1.

Moreover, it is worth saying that in the study of convergence time or misadjustment of adaptive algorithms, one analyzes their behavior near the optimum solution, which yields interesting linearizations [2,7]. This policy makes sense in the case of misadjustment, which should be studied when the learning reaches steady state. However, it may lead to large errors in the case of convergence time, as it is an indication of how fast the algorithm has *started* the learning. Thus, we also propose a new way of evaluating the convergence time here, by analyzing the algorithm behavior in the beginning of the learning task.

2. The method

Let us consider that we observe a given signal d_j and a number of others, which can be included into a vector $\mathbf{X}_j = [x_{j,1}\ \ x_{j,2}\ \ \cdots\ \ x_{j,M}]$, called reference input. Moreover, let us define $d_j = s_j + n_j$, where s_j is the signal we want to extract and n_j is the noise. Let us also assume that n_j is statistically independent of s_j and $\mathbf{X}_j$, whereas all these variables have probability distributions which are not necessarily Gaussian. Our aim is to estimate s_j, after optimally calculating the weight $\mathbf{W}_j = [w_{j,1}\ \ w_{j,2}\ \ \cdots\ \ w_{j,M}]$ and the current error $\varepsilon_j = d_j - y_j$, where the output signal is given by $y_j = \mathbf{W}_j^{\mathrm{T}}\mathbf{X}_j$. We assume that the weight vector coefficients are statistically independent of the input vector.

In this optimization, the Widrow–Hoff algorithm uses an instantaneous estimation of the gradient of $E[\varepsilon_j^2]$. However, our interest is to minimize a general cost function $\zeta_K = f\{E[\varepsilon]\}$. We will assume that $f\{\cdot\}$ is a even function and therefore it can be rewritten in a Taylor series as a sum of even moments of the error. Thus, we can write

$$\zeta_K = \sum_{K=1}^{N} a_K (2K)^{-1} E[\varepsilon_j^{2K}], \tag{1}$$

where a_K is a scaling factor. The term $2K^{-1}$ was introduced only for ease of manipulation.

Thus, the instantaneous gradient of (1), $\nabla(\zeta_K) = -2(\sum_{K=1}^{N} a_K \varepsilon_j^{2K-1})\mathbf{X}_j$, will lead to the following simple update weight rule:

$$\mathbf{W}_{j+1} = \mathbf{W}_j + 2\mu\left(\sum_{K=1}^{N} a_K \varepsilon_j^{2K-1}\right)\mathbf{X}_j, \tag{2}$$

where μ is a learning constant, controlling the stability and rate of convergence.

3. Adaptation analysis

The first task for analyzing the algorithm behavior should be to check the conditions under which it converges to the desired solution, and how it behaves until it reaches steady state. This can be carried out by analyzing the misadjustment error and the convergence time.

Let us first make a change of variable, by defining the vector $\mathbf{V}_j = \mathbf{W}_j - \mathbf{W}_*$, where $\mathbf{W}_*$ is the optimum solution, i.e., $s_j = \mathbf{W}_*^{\mathrm{T}}\mathbf{X}_j$. Thus, (2) becomes,

$$\mathbf{V}_{j+1} = \mathbf{V}_j + 2\mu\left(\sum_{K=1}^{N} a_K \varepsilon_j^{2K-1}\right)\mathbf{X}_j. \tag{3}$$

More specifically, (3) can be rewritten in the form of a binomial expansion as follows:

$$\mathbf{V}_{j+1} = \mathbf{V}_j + 2\mu\left[\sum_{K=1}^{N}\sum_{i=0}^{2K-1} a_K \binom{2K-1}{i} \times n_j^i(-\mathbf{X}_j^{\mathrm{T}}\mathbf{V}_j)^{2K-1-i}\right]\mathbf{X}_j. \tag{4}$$

One can study the misadjustment, which is a measure of how far the output differs from the ideal solution. The misadjustment calculation can be performed in the neighborhood of the optimal solution, i.e., $\mathbf{V}_j \to 0$. Hence, we can neglect the higher powers of $\mathbf{V}_j$ in (4). By remembering that $\varepsilon_j = s_j + n_j - \mathbf{W}_j^{\mathrm{T}}\mathbf{X}_j = n_j - \mathbf{V}_j^{\mathrm{T}}\mathbf{X}_j$, we have,

$$\mathbf{V}_{j+1} \simeq \mathbf{V}_j + 2\mu\left[\sum_{K=1}^{N} a_K \mathbf{X}_j (n_j^{2K-1} - (2K-1)n_j^{2K-2}\mathbf{X}_j^{\mathrm{T}}\mathbf{V}_j)\right], \tag{5}$$

where we made an approximation up to the second order.

Defining $\mathbf{R} = E[\mathbf{X}_j\mathbf{X}_j^{\mathrm{T}}]$, and recalling that $\mathbf{X}_j$ and n_j were assumed to be mutually independent, we can study the behavior of $\mathbf{V}_j$, by taking the expectations at

both sides of (5), which yields

$$E[\mathbf{V}_{j+1}] = \left[\mathbf{I} - 2\mu\left(\sum_{K=1}^{N} a_K(2K-1)E[n_j^{2K-2}]\mathbf{R}\right)\right]E[\mathbf{V}_j]. \tag{6}$$

The equation above is recursive, therefore the convergence condition is given by

$$0<\mu<\frac{1}{(\sum_{K=1}^{N} a_K(2K-1)m_{2K-2}\lambda_{\max})}, \tag{7}$$

where m_q is the qth moment of n_j and $\lambda_{\max}$ is the maximum eigenvalue of $\mathbf{R}$. As we do not have, in principle, access to the noise information, a simpler condition would be to use d which is observed and is function of the noise. Thus, we have $0<\mu<1/(\sum_{K=1}^{N} a_K(2K-1)E[d^{2K-2}]\,\mathrm{tr}[\mathbf{R}])$.

The misadjustment is a measure of the distance between $E[\varepsilon_j^2]$ and m_2, or, in other words, $\chi = \xi_{\mathrm{ex}}/m_2$, where $\xi_{\mathrm{ex}} = \mathrm{tr}[\mathbf{R}\mathbf{Z}_j]$ is the excess mean squared error, and $\mathbf{Z}_j = E[\mathbf{V}_j\mathbf{V}_j^{\mathrm{T}}]$ (see [5]). To analyze it, we shall first find the steady-state value for $\mathbf{Z}_j$. By using (5), defining $\alpha_1 = \sum_{K=1}^{N} a_K^2 m_{4K-2}$, $\alpha_2 = \sum_{K=1}^{N} a_K(2K-1)m_{2K-2}$ and $\alpha_3 = \sum_{K=1}^{N} a_K^2(2K-1)^2 m_{4K-4}$, we find,

$$\mathbf{Z}_{j+1} = \mathbf{Z}_j + 4\mu^2\alpha_1\mathbf{R} - 2\mu\alpha_2(\mathbf{Z}_j\mathbf{R} + \mathbf{R}\mathbf{Z}_j) + 4\mu^2\alpha_3\mathbf{R}\mathbf{Z}_j\mathbf{R}. \tag{8}$$

By using the properties of the trace, one can easily find the relation $\xi_{\mathrm{ex}} = \mathrm{tr}[\mathbf{R}\mathbf{Z}_j] = \mathrm{tr}[\Lambda\Psi_j]$, where Λ and Ψ_j are diagonal matrices where their non-zero elements are the eigenvalues of $\mathbf{R}$ and $\mathbf{Z}_j$, respectively [5].

Finally, we shall find the value of Ψ_j in steady state. This can be easily estimated from (8), by calculating the recursive values of Ψ_j when j is large enough.

Thus, for a small enough μ, the misadjustment will become

$$\chi \approx \sum_{k=1}^{M} \frac{\mu\alpha_1\,\mathrm{tr}[\mathbf{R}]}{\alpha_2 m_2}. \tag{9}$$

3.1. Convergence time

Along with the misadjustment, it is important to study the convergence time. However, we can no longer study the algorithm near the solution, as the convergence time—or equivalently its time constant—should analyze the algorithm behavior in the *beginning* of the learning. Usually, the time constant is defined for first order linear circuits, and it measures the time which an algorithm takes to fall down to around 38% (or $1/e$) of its initial error.

It is important to remember that the time constant for the LMS algorithm, given by $\tau_{i,\mathrm{LMS}} = 1/2\mu\lambda_i$ was deduced by truncating a non-linear function to become a first order linear one [8]. However, if we draw a straight line, in the $w_{j,i} \times t$ plane, starting from w_0^i and passing by $w_{1,i}$, then the time at which the line crosses the t-axis will be exactly the one found by the linearization: $\tau_{i,\mathrm{LMS}} = 1/2\mu\lambda_i$.

We can generalize this concept to any algorithm. Thus, we can easily find that the "time constant" will be given by $\tau = w_*^i/w_{1,i}$. As we have the value for the optimum weight, w_*^i, all we have to do is find $w_{1,i}$. By remembering that $d_0 = s_0 + n_0$, assuming $\mathbf{W}_0 = \mathbf{0}$, and using (2), we find, after some easy manipulations, the time constant to be $\tau_{\mathrm{NEM}} = 1/2\mu\alpha_4$, where,

$$\alpha_4 = 2\mu\sum_{K=1}^{N} a_K \sum_{i=0}^{2K-1}\binom{2K-1}{i} E[x_{j,i}s_j^{2K-i-1}]E[n_j^i]. \tag{10}$$

Here we assumed that all signals are stationary and therefore the statistics of, for example, s_a and s_b are the same, $\forall a, b$.

3.2. Comparison to LMS

An important figure of merit would be to compare the two different algorithms by checking the balance between convergence time and misadjustment. We can carry out this as Walach and Widrow did [7], by using an index $\beta(K)$, which measured the convergence time for the two algorithms, for the same misadjustment. Using the LMS as standard, we can define $\beta(K) = \tau_{\mathrm{LMS}}/\tau_{\mathrm{NEM}}$. It is important to notice that it would be advantageous to use the NEM rather than the LMS when $\beta(K)>1$.

We can accomplish this comparison by equating the misadjustment for the LMS given by $\chi_{\mathrm{LMS}} = \lambda_i\mu$ [5] and the one given by (9), and finding a rate between the two different learning constants, which is $\mu_{\mathrm{LMS}} = \alpha_1\mu_{\mathrm{NEM}}/\alpha_2$. From this, we find,

$$\beta_i(K) = \alpha_2\alpha_4/(\lambda_i\alpha_1). \tag{11}$$

4. Results

In order to check the validity of the theoretical results found here, we carried out simulations. There are a number of ways to carry that out, either because the number of parameters in the NEM algorithm is large or because there are different types of probability distribution for the noise model in the plant.

We carried out the simulation by estimating a plant model of an FIR filter with 31 coefficients. One can easily manipulate the NEM coefficients in order to get better performance than the LMS, by having a higher convergence speed along with less misadjustment. Thus, here we

Table 1
Theoretical and actual misadjustment found for two types of noise: Gaussian and uniform

μ	Gaussian		Uniform	
	Theor.	Actual	Theor.	Actual
1e−2	0.0189	0.0121	0.0190	0.0143
1e−3	0.0019	0.0054	0.0019	0.0049

The parameters of the error were set to be [1 5e−1 1e−4].

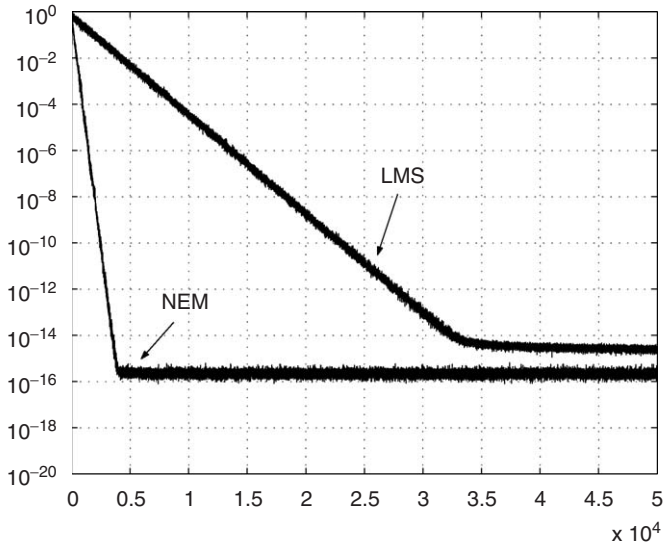


Fig. 2. Mean squared error when estimating the coefficients of a low-pass filtered Gaussian signal of order 31. The parameters were estimated by the LMS and the NEM algorithm with coefficients [1 1.3]. Notice that there is a quite small difference in steady state. That may be due to the larger variance of the squared error, around zero.

Table 2
Theoretical and actual time constant for different level of additive Gaussian noise

	Noise level			
	Zero	0.001	0.01	0.1
Theor.	151	156	152	151
Actual	111	109	111	126

show two examples where we varied the parameters of the NEM algorithm while we used two types of probability density functions: Gaussian and uniform. Firstly, we examined the misadjustment given by (9). The results are shown in Table 1. Moreover, we show in Fig. 2 the simulation results for 200 Monte-Carlo runs. The result for the time-constant estimation is shown in Table 2 along with an example in Fig. 3.

5. Discussions and conclusions

One can see from (11) that we can obtain better performance for the NEM than the LMS by accessing the statistics of the input signal d_j, or by simply controlling λ, independently of the distribution of the noise. Some words are in order here. By examining (11), we can see that the only parameters which we can actually change are α_4 and λ_i. Indeed, from (10), we can find that α_4 can be changed if we manipulate $x_{i,j}$. Thus, depending upon the coefficients, the value of $\beta(K)$, can be changed in (11).

We also saw that there is a reasonable agreement between the deduced equation to the misadjustment and the practical results, either to Gaussian, sinusoidal or uniform type of noise. Moreover, we provided a new way of calculating the time of convergence. Regarding to misadjustment, one can see in Fig. 2 that the non-linear even moment algorithm reached a slightly lower misadjustment than that of the LMS. This can be explained through the form of the curve close to zero, as in Fig. 1, where one can see that there is a sharper drop of the linearly combined curve when compared to the one of the squared error. Moreover, one can see that the approximations for the

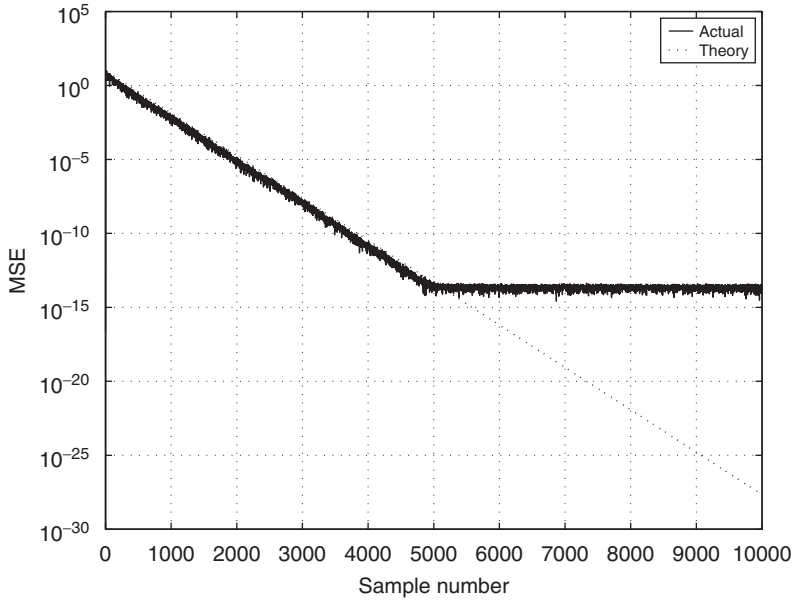


Fig. 3. Mean squared error when estimating the coefficients of a low-pass filtered Gaussian signal of order 5. We also plotted the actual envelope for the proposed time constant.

time-constant estimation as in (10) showed to fit well to the actual results.

References

[1] J.A. Chambers, O. Tanrikulu, A.G. Constantinides, Least mean mixed-norm adaptive filtering, Electron. Lett. 30 (19) (1994) 1574–1575.

[2] S.C. Douglas, T. Meng, Stochastic gradient adaptation under general error criteria, Signal Process. 42 (1994) 1335–1351.

[3] D. Erdogmus, J. Principe, K. Hild II, Beyond second-order statistics for learning: a pairwise interaction model for entropy estimation, Neural Comput. (1) (2002) 85–108.

[4] A. Gersho, Some aspects of linear estimation with non-mean square error criteria, Proceedings of Asilomar Ckts. and Systems Conference, 1969.

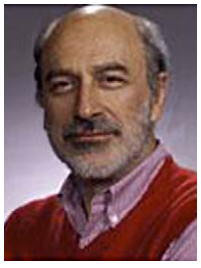
[5] S. Haykin, Adaptive Filter Theory, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[6] A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, New York, 1991.

[7] E. Walach, B. Widrow, The least mean fourth (LMF) adaptive algorithm and its family, IEEE Trans. Inform. Theory IT-30 (2) (1984).

[8] B. Widrow, S.D. Stearns, Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1985.

Allan Kardec Barros received the B.S. degree in electrical engineering from Universidade Federal do Maranhao, Brazil, in 1991, the M.S. degree in information engineering from Toyohashi University of Technology, Toyohashi, Japan, in 1995, and the D.Eng. degree from Nagoya University, Nagoya, Japan, in 1998. He worked as a Frontier Researcher from 1998 to 2000 at The Institute of Physical and Chemical Research (RIKEN), Japan. Currently, he is an Associate Professor at Universidade Federal do Maranhao, Brazil. His research interests include biomedical engineering, information coding, and speech signal processing.



Jose C. Principe (M'83-SM'90-F'00) is Distinguished Professor of Electrical and Biomedical Engineering at the University of Florida, Gainesville, where he teaches advanced signal processing and artificial neural networks (ANNs) modeling. He is BellSouth Professor and Founder and Director of the University of Florida Computational NeuroEngineering Laboratory (CNEL). He has been involved in biomedical signal processing, in particular the electroencephalogram (EEG), and the modeling and applications of adaptive systems. Dr. Principe is Editor in Chief of IEEE Transactions on Biomedical Engineering, President Elect of the International Neural Network Society, and formal Secretary of the Technical Committee on Neural Networks of the IEEE Signal Processing Society. He is also a member of the Scientific Board of the Food and Drug Administration, and a member of the Advisory Board of the University of Florida Brain Institute. He has more than 90 publications in refereed journals, 10 book chapters, and over 190 conference papers. He has directed 39 Ph.D. degree dissertations and 57 master degree theses.

Yoshinori Takeuchi received the degrees of B. Eng., M. Eng. and Dr. Eng. from Nagoya University in 1994, 1996 and 1999, respectively. In 1999, he was a Research Fellow of the Japan Society for the Promotion of Science. In 2000, he was a member of the Graduate School of Engineering, Nagoya University. Currently, he is an Associate Professor at the Information Security Promotion Agency, Nagoya University. His research interests include computer vision and computer audition. He is a member of IEEE, IEICE and RSJ.

Noboru Ohnishi received the B. Eng., M. Eng. and D. Eng. degrees from Nagoya University, Nagoya, Japan, in 1973, 1975 and 1984, respectively. From 1975 to 1986 he was with the Rehabilitation Engineering Center under the Ministry of Labor. From 1986 to 1989 he was an Assistant Professor in the Department of Electrical Engineering, Nagoya University. From 1989 to 1994, he was an Associate Professor. Since 1994, he is a professor in Nagoya University. From 1993 to 2001, he concurrently holded a Head of Laboratory for Bio-mimetic Sensory System at the Bio-mimetic Control Research Center of RIKEN. He is now in the Graduate School of Information Science. His research interests include computer vision and computer audition, robotics, bio-cybernetics, and rehabilitation engineering. Dr. Ohnishi is a member of IEEE, IEEJ, IEICE, IPSJ, SICE, JNNS, IIITE and RSJ.