AN IMPROVED PROPORTIONATE NLMS ALGORITHM BASED ON THE $l_0$ NORM

Constantin Paleologu$^1$, Jacob Benesty$^2$, and Silviu Ciochină$^1$

1: University Politehnica of Bucharest, Telecommunications Department
Iuliu Maniu Blvd., 1-3, Bucharest, Romania
e-mail: {pale, silviu}@comm.pub.ro
2: Universite du Quebec, INRS-EMT, Montreal, QC H5A 1K6, Canada
e-mail: benesty@emt.inrs.ca

ABSTRACT

The proportionate normalized least-mean-square (PNLMS) algorithm was developed in the context of network echo cancellation. It has been proven to be efficient when the echo path is sparse, which is not always the case in real-world echo cancellation. The improved PNLMS (IPNLMS) algorithm is less sensitive to the sparseness character of the echo path. This algorithm uses the $l_1$ norm to exploit sparseness of the impulse response that needs to be identified. In this paper, we propose an IPNLMS algorithm based on the $l_0$ norm, which represents a better measure of sparseness than the $l_1$ norm. Simulation results prove that the proposed algorithm outperforms the original IPNLMS algorithm.

Index Terms— Echo cancellation, adaptive filters, proportionate normalized least-mean-square (PNLMS) algorithm, sparseness.

1. INTRODUCTION

Echo cancellation is one of the most popular applications of adaptive filtering [1]. The basic approach is to build a model of the echo path impulse response using an adaptive filter. Theoretically, various kinds of adaptive algorithms are applicable for echo cancellation [2]. However, for many good reasons, the normalized least-mean-square (NLMS) algorithm is frequently the algorithm of choice in this important application.

In order to enhance the performance of the NLMS algorithm for network echo cancellation, a proportionate NLMS (PNLMS) algorithm has been proposed in [3]. It takes advantage of the fact that the network echo path is sparse in nature, i.e., only a small portion of the coefficients is significantly different from zero (active coefficients). The idea behind the PNLMS algorithm is to update each coefficient of the filter independently of the others by adjusting the adaptation step-size in proportion to the magnitude of the estimated filter coefficient. It redistributes the adaptation gains among all coefficients and emphasizes the large ones in order to speed up their convergence, achieving a fast initial convergence rate. Unfortunately, after this initial phase, the convergence rate of the PNLMS algorithm slows down significantly, even becoming slower than NLMS. In order to deal with this problem, several versions of the PNLMS algorithm were proposed (e.g., see [4] and the references therein). Most of them still assume the sparse character of the echo path impulse response. However, in real-world echo cancellation scenarios, it is difficult to a priori know the sparseness “degree” of the echo path. Motivated by this consideration, an improved PNLMS (IPNLMS) algorithm was proposed in [5]. This algorithm better takes advantage of the “proportionate” idea, by using the $l_1$ norm to exploit sparseness of the impulse response that needs to be identified. Consequently, it behaves well even in the case when the echo path is non-sparse or quasi-sparse (e.g., like in acoustic echo cancellation).

The motivation behind this work is that the $l_0$ norm can represent an even better measure of sparseness than the $l_1$ norm [6]. Thus, we propose an IPNLMS algorithm based on the $l_0$ norm. The remainder of the paper is organized as follows. Section 2 presents the backgrounds of the PNLMS-based algorithms. The proposed IPNLMS algorithm based on the $l_0$ norm is derived in Section 3. Experimental results are shown in Section 4. Finally, Section 5 concludes this work.

2. PNLMS ALGORITHMS BACKGROUND

Let us consider an echo canceller configuration, where we try to model an echo path with impulse response $h = [h_0, h_1, \ldots, h_{L-1}]^T$ using an adaptive filter, $\hat{h}(n) = [\hat{h}_0(n), \hat{h}_1(n), \ldots, \hat{h}_{L-1}(n)]^T$. Superscript $T$ denotes transposition, $L$ is the length of the systems, and $n$ is the time index. The desired signal for the adaptive filter is $y(n) = \mathbf{h}^T \mathbf{x}(n) + v(n)$, where $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T$ is a real-valued vector containing the $L$ most recent samples of the input signal (i.e., far-end signal) and $v(n)$ is the near-end signal.

The proportionate-type NLMS algorithms are summarized by the following equations:
\[
\begin{align*}
    e(n) &= y(n) - \hat{h}^T(n-1)x(n), \\
    \hat{h}(n) &= \hat{h}(n-1) + \frac{\mu G(n-1)x(n)e(n)}{\delta + x^T(n)G(n-1)x(n)},
\end{align*}
\]

where \(\mu\) is the normalized step-size parameter and \(\delta\) is the regularization. The main idea behind these algorithms is to assign an individual step-size to each filter coefficient, using the diagonal matrix \(G(n)\), in such a way that a larger coefficient receives a larger increment, thus increasing the convergence rate of that coefficient. Consequently, the active coefficients are adjusted faster than non-active coefficients (i.e., small or zero coefficients), so that this type of algorithms converge faster than NLMS especially for sparse impulse responses (i.e., when only a small percentage of coefficients is significant). Regarding (2), it can be noticed that the NLMS algorithm is obtained when \(G(n) = I\), where \(I\) is the identity matrix.

In the case of the original PNLMS algorithm [3], the diagonal elements of \(G(n)\), denoted here by \(g_d(n)\), with \(0 \leq i \leq L - 1\), are calculated using the following procedure:

\[
\gamma_i(n) = \max \left\{ \rho \max \left[ \frac{\gamma}{l} h_i(n), \ldots, \frac{\gamma}{l} h_{L-1}(n), \frac{\gamma}{l} h_L(n) \right] \right\},
\]

\[
g_i(n) = \frac{\gamma_i(n)}{\sum_{i=0}^{L-1} \gamma_i(n)}.
\]

Parameters \(\rho\) and \(\gamma\) are positive numbers with typical values \(\rho = 5/L\) and \(\gamma = 0.01\). The constant \(\rho\) prevents the very small coefficients from stalling and the parameter \(\gamma\) regularizes the updating when all coefficients are zero at initialization. The main limitation of the PNLMS algorithm is that it assigns too much adaptation gains to large coefficients to speed up their convergence at the cost of small coefficients. Thus, after an initial fast convergence phase, this algorithm slows down and becomes even slower than NLMS.

The IPNLMS algorithm proposed in [5] uses a smoother choice as compared to (3), i.e.,

\[
\gamma_i(n) = (1 - \alpha) \frac{\|\hat{h}(n)\|}{L} + (1 + \alpha) \left| \hat{h}_i(n) \right|,
\]

where \(-1 \leq \alpha < 1\) and

\[
\|\hat{h}(n)\| = \sum_{i=0}^{L-1} \left| \hat{h}_i(n) \right|
\]

is the \(l_1\) norm of the vector \(\hat{h}(n)\). Therefore, substituting (5) into (4) and taking (6) into account, we find that

\[
g_i(n) = \frac{1 - \alpha}{2L} + (1 + \alpha) \frac{\left| \hat{h}_i(n) \right|}{\left| \sum_{i=0}^{L-1} \hat{h}_i(n) \right| + \epsilon}.
\]

For practical reasons, a very small positive constant \(\epsilon\) was added to the denominator of the second term of (7), in order to avoid division by zero (especially at the beginning of the adaptation when all the filter taps are initialized to zero). It can be checked that for \(\alpha = -1\) the IPNLS and NLMS algorithms are equivalent, while for \(\alpha\) close to 1 the IPNLMS algorithm behaves like the PNLMS algorithm. In practice, good choices for the parameter \(\alpha\) are 0 or –0.5. Also, it is very important to correctly choose the regularization parameters of the PNLMS-type algorithms. It was explained in [5] that the relation between the regularization parameters of the discussed algorithms is \(\delta_{\text{IPNLMS}} = \delta_{\text{PNLMS}}(1-\alpha)/2 = \delta_{\text{NLMS}}(1-\alpha)/2L\). The IPNLMS algorithm performs better than PNLMS for both sparse and non-sparse impulse responses.

### 3. IPNLMS Algorithm Based on \(l_0\) Norm

It can be noticed that the IPNLMS algorithm uses the \(l_1\) norm to exploit the sparseness of the impulse response we need to identify. A better measure can be the \(l_0\) norm, since it is the natural mathematical measure of sparsity [7]. The \(l_0\) norm of a vector is equal to the number of its non-zero components.

Let us consider the vector \(a = [a_0, a_1, \ldots, a_{L-1}]^T\). Defining the function

\[
f(a) = \begin{cases} 
1, & a_i \neq 0 \\
0, & a_i = 0,
\end{cases}
\]

the \(l_0\) norm of the vector \(a\) is

\[
\|a\|_0 = \sum_{i=0}^{L-1} f(a_i).
\]

The function \(f(a)\) is not continuous and the fact that many elements of the vector \(a\) can be very small but not exactly equal to zero, it is better to approximate it by a smooth and continuous function. A good approximation is [6]

\[
f(a) \approx 1 - e^{-\beta |a|},
\]

where \(\beta\) is a large positive value. Therefore,

\[
\|a\|_0 = \lim_{\beta \to +\infty} \sum_{i=0}^{L-1} \left[ 1 - e^{-\beta |a_i|} \right] \approx \sum_{i=0}^{L-1} \left[ 1 - e^{-\beta |a_i|} \right].
\]

Now we can use this norm to estimate the elements of \(G(n)\). Following the idea of the IPNLMS algorithm, we have

\[
\gamma_i(n) = (1 - \alpha) \frac{\|\hat{h}(n)\|}{L} + (1 + \alpha) \left[ 1 - e^{-\beta |\hat{h}_i(n)|} \right].
\]

Next, substituting (12) into (4) and taking (11) into account, we find that
\[ g_l(n) = \frac{1-\alpha}{2L} + (1+\alpha) \frac{1-e^{-\beta ||\hat{v}(n)||}}{2\sum_{\ell=1}^{L-1} 1-e^{-\beta ||\hat{v}(n)||}} + \epsilon. \]  

(13)

The very small positive constant \( \epsilon \) plays the same role as in (7), i.e., to avoid division by zero. Using (13) instead of (7) we obtain the IPNLMS algorithm based on the \( l_0 \) norm.

Finally, some practical issues should be outlined. First, the choice of the parameter \( \beta \) is important. In principle, it should depend on the energy distribution of the impulse response that needs to be identified; a dispersive system could require a larger value for \( \beta \). Several considerations regarding the choice of this parameter can be found in [6]. Also, we can obtain some “a priori” information about the impulse response by using a regular IPNLMS algorithm in the first \( N \) iterations (with \( N > L \)); then, having a rough model of the echo path, we can evaluate its sparseness degree using a certain measure, e.g.,

\[ \zeta [\hat{h}(N)] = 1 - \frac{1}{L} \left\| \hat{h}(N) \right\|_0 \]  

(14)

and choose the value of \( \beta \) accordingly. It is obvious that the value of the parameter from (14) is between 0 and 1. Note that the closer this measure is to 1, the sparser is the impulse response. On the contrary, the closer the measure is to 0, the denser or less sparse is the impulse response.

Second, the evaluation of the exponential term from (13) could be problematic in practical implementations. A possible solution is based on the first order Taylor series expansions of exponential functions [6]. Another practical option is to use a look-up table.

4. SIMULATION RESULTS

For the experiments, we have chosen two impulse responses (Fig. 1) of length \( L = 512 \) (using an 8 kHz sampling rate). The first one (Fig. 1a) is a network echo path (sparse), according to ITU-T G.168 [8]; the second one (Fig. 1b) is a measured acoustic echo path (quasi-sparse). The input signal, \( x(n) \), is either a white Gaussian noise or a speech signal. An independent white Gaussian noise, \( v(n) \), is added to the output of the echo path, resulting the desired signal, \( y(n) \); the signal-to-noise ratio (SNR) is 30 dB.

It would be very interesting to compare the proposed algorithm with most of the PNLMS-type algorithms but it is beyond the scope of this paper. For this reason we choose to compare the IPNLMS algorithm based on the \( l_0 \) norm (referred in the following as IPNLMS-\( l_0 \)) with the classical PNLMS algorithm [3] and with the original IPNLMS algorithm [5]. These two algorithms are the benchmarks to which most PNLMS-type algorithms are compared. Besides, it is important to notice the improvements of the proposed IPNLMS-\( l_0 \) as compared to the IPNLMS algorithm, taking into account the improvements of the IPNLMS as compared to the PNLMS.

![Fig. 1. (a) Sparse network echo path according to ITU-T G.168; (b) measured acoustic echo path (quasi-sparse).](image)

The measure of performance is the normalized misalignment (in dB), defined as \( 20 \log_{10} \left( \frac{||\hat{h} - \hat{h}(n)||_2}{||\hat{h}||_2} \right) \), where \( ||\cdot||_2 \) denotes the \( l_2 \) norm; the results are averaged over 20 independent trials. The normalized step-size for all the algorithms is \( \mu = 0.2 \). The regularization parameter for the PNLMS algorithm is \( \delta_{PNLMS} = 20 \sigma_i^2/L \), while the IPNLMS algorithms use \( \delta_{IPNLMS} = 20 \sigma_i^2/2L \) (where \( \sigma_i^2 \) is the input signal variance). For the IPNLMS algorithms we fix \( \alpha = 0 \).

First, we evaluate the impact of the parameter \( \beta \) on the performance of the proposed IPNLMS-\( l_0 \) algorithm. The experiments are performed for both impulse responses from Fig. 1; the results are presented in Fig. 2, using a white Gaussian noise as input signal. In both cases it can be noticed that the performance is improved when the value of \( \beta \) increases, but up to a value which seems to be closer to the number of active coefficients (i.e., \( \beta = 50 \) in Fig. 2a and \( \beta = 500 \) in Fig. 2b). This could be a motivation for using the approach related to (14) in order to choose the value of \( \beta \).

Fig. 3 compares the performance of the PNLMS, IPNLMS, and IPNLMS-\( l_0 \) algorithms for both sparse and quasi-sparse impulse responses; the input signal is a white Gaussian noise. The IPNLMS-\( l_0 \) algorithm uses \( \beta = 50 \) in Fig. 3a and \( \beta = 500 \) in Fig. 3b. It can be noticed that the proposed algorithm outperforms IPNLMS in both cases, but the differences are more significant for the sparse impulse response (Fig. 3a).

In Fig. 4, the tracking capabilities of the algorithms are tested. The impulse responses from Fig. 1 are shifted to the right by 12 samples. The other conditions are the same as in the previous experiment (Fig. 3). According to this simulation, the IPNLMS-\( l_0 \) algorithm has a similar tracking reaction as compared to the IPNLMS algorithm.

Finally, the speech input is used in the last experiment (Fig. 5). The other conditions are the same as in Fig. 4. It is clear that the IPNLMS-\( l_0 \) algorithm outperforms the other algorithms. In the case of the sparse impulse response (Fig. 5a), it is interesting to notice that the improvement of the IPNLMS-\( l_0 \) over the IPNLMS is similar with the improvement of the IPNLMS over the PNLMS.
Fig. 2. Misalignment of the IPNLMS-$\ell_0$ algorithm for different values of the parameter $\beta$. (a) Impulse response from Fig. 1a; (b) Impulse response from Fig. 1b. The input signal is a white Gaussian noise.

Fig. 3. Misalignment of the PNLMS, IPNLMS, and IPNLMS-$\ell_0$ algorithms. (a) Impulse response from Fig. 1a, $\beta = 50$; (b) Impulse response from Fig. 1b, $\beta = 500$. The input signal is a white Gaussian noise.

Fig. 4. Misalignment of the PNLMS, IPNLMS, and IPNLMS-$\ell_0$ algorithms. Echo path changes at time 1.5. Other conditions as in Fig. 3.

Fig. 5. Misalignment of the PNLMS, IPNLMS, and IPNLMS-$\ell_0$ algorithms. The input signal is speech. Echo path changes at time 3. Other conditions as in Fig. 3.

5. CONCLUSIONS AND PERSPECTIVES

In this paper we proposed a new proportionate-type NLMS algorithm. It is based on the IPNLMS algorithm but it uses the $\ell_0$ norm to exploit the sparseness of the system that needs to be identified. Simulation results indicate that the IPNLMS-$\ell_0$ algorithm outperforms IPNLMS especially for sparse impulse responses. The main challenge associated with the IPNLMS-$\ell_0$ algorithm is to find a practical way to choose the value of the parameter $\beta$. Several strategies for this purpose were also discussed in the paper. This idea can obviously be extended to the affine projection algorithm.

6. REFERENCES