Low-Complexity Set-Membership Normalized LMS Algorithm for Sparse System Modeling

Javad Sharafi and Mohsen Mehrali-Varjani

Abstract

In this work, we propose two low-complexity set-membership normalized least-mean-square (LCSM-NLMS1 and LCSM-NLMS2) algorithms to exploit the sparsity of an unknown system. For this purpose, in the LCSM-NLMS1 algorithm, we employ a function called the discard function to the adaptive coefficients in order to neglect the coefficients close to zero in the update process. Moreover, in the LCSM-NLMS2 algorithm, to decrease the overall number of computations needed even further, we substitute small coefficients with zero. Numerical results present similar performance of these algorithms when comparing them with some state-of-the-art sparsity-aware algorithms, whereas the proposed algorithms need lower computational cost.

Index Terms

Adaptive learning, set-membership filtering, NLMS, sparsity, computational burden.

I. INTRODUCTION

Adaptive filtering has applications in many areas such as communications, control, radar, acoustics, and speech processing. Nowadays, sparsity is an ubiquitous characteristic in signal or system parameters. Unfortunately, traditional adaptive filtering algorithms, such as the least-mean square (LMS) and the normalized LMS (NLMS) algorithms, do not exploit the sparsity in the signal or system models to improve the learning performance.

We know that by exploiting signal sparsity, we can remarkably improve the convergence rate and/or the steady-state performance of the learning process. Therefore, many improvement in the classical algorithms were introduced to exploit sparsity. A well-known family of algorithms to exploit sparsity is the family of proportionate algorithms [1]–[5]. Another famous strategy to exploit sparsity is obtained by including a penalty function, such as the $l_1$-norm and the $l_0$-norm, to the cost function of traditional algorithms [6]–[10]. Note that in these methods something is added to the conventional algorithms, thus they increase computational cost of the algorithms.

In this work, to exploit systems sparsity, we employ a different approach from adding some features to the algorithms. Indeed, we ignore coefficients close to zero; thus, we can decrease the computational resources. It is good to mention that a sparse impulse response of a system contains a few coefficients with high energy, and most of the coefficients are close to zero. Therefore, by using some knowledge about the uncertainty of small coefficients, we may substitute the coefficients smaller than assumed uncertainty with zero to reduce computational costs. In other words, we discard small coefficients.

Besides the mentioned approach to exploit sparsity, we also use the set-membership filtering (SMF) technique [11]–[17] to propose the Low-Complexity Set-Membership Normalized Least-Mean-Square (LCSM-NLMS1) algorithm. This algorithm combines the set-membership normalized least-mean-square (SM-NLMS) algorithm [18] with the discarding technique to exploit sparsity. Indeed, the SMF strategy avoids new update when the error is greater than the pre-determined upper bound; thus, it can decrease the computational load even further.

This paper is organized as follows. In Section [II] we review the SMF technique. The LCSM-NLMS1 and the LCSM-NLMS2 algorithms are proposed in Section [III]. Numerical results and conclusions are described in Sections [IV] and [V] respectively.

(J. Sharafi), Imam Ali University, Tehran, Iran. E-mail address: javadsharafi@grad.kashanu.ac.ir.
(M. Mehrali-Varjani), Imam Ali University, Tehran, Iran. E-mail address: Mohsenmehrali@yahoo.com.
II. SET-MEMBERSHIP FILTERING (SMF)

In the last decades, the SMF is proposed to acquire the adaptive coefficients \( \mathbf{w} \) so that the magnitude of their error is upper bounded by a constant positive factor \( \gamma \in \mathbb{R}_+ \). For appropriately adopted \( \gamma \), we can have various acceptable estimates for \( \mathbf{w} \). Assume that \( S \) is the set of all input-desired data \((\mathbf{x}, d)\), and define the feasibility set \( \Theta \) as

\[
\Theta = \bigcap_{(\mathbf{x}, d)\in S} \{ \mathbf{w} \in \mathbb{R}^{N+1} : |d - \mathbf{w}^T \mathbf{x}| \leq \gamma \}.
\]

Moreover, define the constraint set \( \mathcal{H}(k) \) containing all vectors \( \mathbf{w} \) so that, at a given iteration \( k \), the magnitude of their errors are upper bounded by \( \gamma \), that is

\[
\mathcal{H}(k) = \{ \mathbf{w} \in \mathbb{R}^{N+1} : |d(k) - \mathbf{w}^T \mathbf{x}(k)| \leq \gamma \},
\]

where \( \mathbf{w}, \mathbf{x}(k), \) and \( d(k) \) are the weight vector, the input vector, and the desired signal, respectively. The membership set \( \psi(k) \) can be described by

\[
\psi(k) = \bigcap_{i=0}^{k} \mathcal{H}(i).
\]

Note that, as \( k \to \infty \), the membership set converges to the feasibility set. However, we cannot compute \( \psi(k) \) because of practical issues; thus, we calculate a point estimate by the information obtained by the constraint set \( \mathcal{H}(k) \) [18].

III. THE LOW-COMPLEXITY SET-MEMBERSHIP NLMS ALGORITHM

To exploit sparsity in unknown systems with a computational complexity lower than that of the existing sparsity-aware algorithms, we propose the Low-Complexity Set-Membership NLMS (LCSM-NLMS) algorithm. To this end, we avoid updating the small adaptive coefficients of the sparse system. Thus, first, in Subsection III-A, we propose the LCSM-NLMS1. Then, we discuss some important issues of the LCSM-NLMS1 algorithm. Moreover, in Subsection III-C, we introduce the LCSM-NLMS2 algorithm (an improved version of the LCSM-NLMS1) intending to reduce computational cost even further.

A. LCSM-NLMS1 algorithm

To exploit systems sparsity by the LCSM-NLMS1 algorithm, the thresholding approach by the discard function is applied. The discard function, \( f_\epsilon : \mathbb{R} \to \mathbb{R} \), is defined by [19], [20]

\[
f_\epsilon(w) = \begin{cases} 
 w & \text{if } |w| \geq \epsilon \\
 0 & \text{if } |w| < \epsilon 
\end{cases},
\]

where \( \epsilon \) is a positive constant. In fact, this function disregards those entries of \( w \) that are close to zero. Note that \( \epsilon \) determines which values are close to zero, and some prior knowledge about the sparse system can be used to select this parameter. It is worth mentioning that \( f_\epsilon(w) \) is not differentiable at \( \pm \epsilon \), and we need its derivative in the optimization problem. To resolve this problem, we assume that the derivative of \( f_\epsilon(w) \) at \( +\epsilon \) and \( -\epsilon \) is zero. We can know introduce the discard vector function \( f_\epsilon : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1} \) by

\[
f_\epsilon(\mathbf{w}) = [f_\epsilon(w_0), \ldots, f_\epsilon(w_N)]^T.
\]

When the output estimation error is greater than the predetermined positive value \( \gamma \), the LCSM-NLMS1 algorithm updates the adaptive coefficients whose absolute values are greater than \( \epsilon \). Whenever \( \mathbf{w}(k) \notin \mathcal{H}(k) \), i.e., \( |e(k)| = |d(k) - \mathbf{w}^T(k)\mathbf{x}(k)| > \gamma \), the optimization criterion of the LCSM-NLMS1 is given by

\[
\min \frac{1}{2} \|f_\epsilon(\mathbf{w}(k+1)) - \mathbf{w}(k)\|^2
\]

subject to

\[
d(k) - \mathbf{w}^T(k+1)\mathbf{x}(k) = \gamma.
\]
To compute the solution of this optimization problem, the Lagrangian \( L \) should be formed as
\[
L = \frac{1}{2} \| f_\epsilon(w(k + 1)) - w(k) \|^2 + \lambda(k)[d(k) - w^T(k + 1)x(k) - \gamma],
\]
where \( \lambda(k) \in \mathbb{R} \) is the Lagrange multiplier. By taking the differential of the above equation with respect to \( w(k + 1) \) and letting it equal to zero, we get
\[
f_\epsilon(w(k + 1)) = w(k) + \lambda(k)F_\epsilon^{-1}(w(k + 1))x(k),
\]
where \( F_\epsilon(w(k + 1)) \) is the Jacobian matrix of \( f_\epsilon(w(k + 1)) \). To form the recursion, we use the projection approximation subspace tracking with deflation approach, as in [21], in (8), then we can substitute \( f_\epsilon(w(k + 1)) \) and \( F_\epsilon^{-1}(w(k + 1)) \) with \( w(k + 1) \) and \( F_\epsilon^{-1}(w(k)) \), respectively. Thus, we attain
\[
w(k + 1) = w(k) + \lambda(k)F_\epsilon^{-1}(w(k))x(k).
\]
By replacing this equation into the constraint relation (6), we obtain \( \lambda(k) \) as
\[
\lambda(k) = \frac{e(k) - \gamma}{x^T(k)F_\epsilon^{-1}(w(k))x(k)}.
\]
Then, by substituting (10) into (9), we get the following recursion rule
\[
w(k + 1) = w(k) + \frac{(e(k) - \gamma)F_\epsilon^{-1}(w(k))x(k)}{x^T(k)F_\epsilon^{-1}(w(k))x(k) + \delta},
\]
where \( \delta \) is a small positive constant to avoid division by zero. Note that \( F_\epsilon(w(k)) \) is a singular matrix; thus, the Moore-Penrose pseudoinverse (generalization of the inverse matrix) can be employed in the place of the standard matrix inversion. But \( F_\epsilon(w(k)) \) is a diagonal matrix whose diagonal components are zero or one. In fact, when a coefficient of \( w(k) \) has the absolute value greater than \( \epsilon \), then its corresponding entry on the diagonal of \( F_\epsilon(w(k)) \) is one, but the remaining entries are zero. Note that the pseudoinverse of \( F_\epsilon(w(k)) \) is \( F_\epsilon(w(k)) \). Hence, the update rule is given by
\[
w(k + 1) = w(k) + \frac{(e(k) - \gamma)F_\epsilon^{-1}(w(k))x(k)}{x^T(k)F_\epsilon^{-1}(w(k))x(k) + \delta}.
\]
By changing \( \gamma \) to \( \frac{\gamma e(k)}{|e(k)|} \), we obtain the update equation of the LCSM-NLMS1 algorithm as follows
\[
w(k + 1) = w(k) + \mu(k) \frac{e(k)F_\epsilon^{-1}(w(k))x(k)}{x^T(k)F_\epsilon^{-1}(w(k))x(k) + \delta},
\]
where
\[
\mu(k) = \begin{cases} 
1 - \frac{\gamma}{|e(k)|} & \text{if } |e(k)| > \gamma, \\
0 & \text{otherwise.}
\end{cases}
\]

B. Some discussion of the LCSM-NLMS1 algorithm

The recursion rules of the LCSM-NLMS1 and SM-NLMS algorithms are similar, but the LCSM-NLMS1 algorithm only updates the subset of coefficients of \( w(k) \) that their absolute values are greater than \( \epsilon \). Therefore, the LCSM-NLMS1 algorithm has lower computational cost in comparison with the SM-NLMS algorithm.

In Table [III-B] for each update, we describe the computational complexity of the set-membership proportionate NLMS (SM-PNLMS) [4], the set-membership \( l_0 \)-NLMS (SM-\( l_0 \)-NLMS) [22], and the LCSM-NLMS1 algorithms. It is worth mentioning that, in Table [III-B], the number of multiplications and additions are described for the update of all coefficients. In means that we presented the worst situation for the LCSM-NLMS1 algorithm (i.e., \( \epsilon = 0 \)). In this case, the computational load of the LCSM-NLMS1 algorithm is identical to that of the SM-NLMS algorithm. However, in practice \( \epsilon \) is different from zero. Also, note
TABLE I
NUMBER OF REQUIRED REAL MULTIPLICATIONS AND DIVISIONS FOR THE SM-PNLMS, SM-\(l_0\)-NLMS, AND LCSM-NLMS1 ALGORITHMS

| Algorithm        | Addition & Subtraction | Multiplication | Division |
|------------------|-------------------------|----------------|----------|
| SM-PNLMS         | \(N^2 + 5N + 5\)        | \(7N + 8\)     | \(2N + 4\) |
| SM-\(l_0\)-NLMS | \(7N + 7\)              | \(9N + 11\)    | \(N + 3\) |
| LCSM-NLMS1       | \(3N + 4\)              | \(3N + 4\)     | 1        |

TABLE II
THE COEFFICIENTS OF UNKNOWN SYSTEMS SYSTEM 1, SYSTEM 2, AND SYSTEM 3

| System 1 | 0.02 0 0 0 0 | 0.6 0 0 | 0.25 0 0 0 0 |
|----------|-------------|--------|-------------|
| System 2 | 0 0 0 0 0   | 0.3 0.6 | -0.5 0.7   | 0 0 0 0 0 |
| System 3 | 0 0 0 0 0   | 0.3 0.5 | 0.7 0.5 0.3 | 0 0 0 0 |

that the number of divisions in the LCSM-NLMS1 is only one, while the SM-PNLMS and the SM-\(l_0\)-NLMS algorithms require \(2N + 4\) and \(N + 3\) divisions, respectively. Furthermore, note that the memory requirements of the LCSM-NLMS1 algorithm are exactly the same as the NLMS algorithm.

Moreover, we must remind that the weight vector of the LCSM-NLMS1 algorithm cannot be initialized with the zero vector. As the matter of fact, for this algorithm, the adaptive coefficients must be initialized by some values outside the interval \([-\epsilon, \epsilon]\); i.e., \(|w_i(0)| > \epsilon\) for \(i = 0, 1, \ldots, N\).

C. The LCSM-NLMS2 algorithm

In the recursion rule of the LCSM-NLMS1, we can observe that for those coefficients of the adaptive filter fall inside \([-\epsilon, +\epsilon]\), the algorithm does not update them in the subsequent iterations since they are disregarded by the discard function. Furthermore, the coefficients inside \([-\epsilon, +\epsilon]\) are close to zero and the they can be estimated by zero. Also, inserting zero for these coefficients leads to a reduction of computational cost in computing the output signal \(y(k) = x^T(k)w(k)\). To this end, we introduce the LCSM-NLMS2 algorithm by multiplying \(w(k)\) by \(F_\epsilon(w(k))\). Thus the update equation of the LCSM-NLMS2 is given by

\[
w(k + 1) = F_\epsilon(w(k))w(k) + \mu(k)\frac{e(k)F^{-1}_\epsilon(w(k))x(k)}{x^T(k)F^{-1}_\epsilon(w(k))x(k) + \delta},
\]

where \(\mu(k)\) is described by (14).

IV. SIMULATIONS

In this section, we use the SM-PNLMS, the SM-\(l_0\)-NLMS, and the LCSM-NLMS2 algorithms, in system identification scenarios, to identify three sparse systems of order 12. The impulse response of the sparse systems are presented in Table IV. We only presents the performance of the LCSM-NLMS2 algorithms since it requires lower computational load and performs better than LCSM-NLMS1 algorithm in sparse domains. The input signal is a zero-mean white Gaussian noise with unit variance. The additive noise has a zero-mean white Gaussian distribution with variance \(\sigma_n^2 = 0.01\). The threshold parameter \(\gamma\) is chosen as \(\sqrt{5\sigma_n^2}\). All algorithms are initialized with \(w(0) = 0.1 \times [1, \ldots, 1]^T\), and the regularization parameter is adopted as \(\delta = 10^{-12}\). For the LCSM-NLMS2 algorithm, \(\epsilon\) is selected as 0.0001. For the SM-PNLMS and the SM-\(l_0\)-NLMS algorithms, the parameters \(\alpha, \beta,\) and \(\epsilon\) are adopted as 0.005, 5, and 100, respectively. The learning curves are obtained by averaging the outcomes of 500 runs.

Figure I shows the MSE learning curves of the SM-PNLMS, the SM-\(l_0\)-NLMS, and the LCSM-NLMS2 algorithms when the system 1 is used as the unknown system. We can see that the LCSM-NLMS2 algorithm has competitive performance with other two algorithms. In other words, the LCSM-NLMS2 has
Fig. 1. The MSE learning curves of the SM-PNLMS, the SM-$l_0$-NLMS, and the LCSM-NLMS2 algorithms when the unknown system is System 1.

extremely similar performance to the SM-$l_0$-NLMS and the SM-PNLMS algorithms, whereas it requires very lower computational loads. Indeed, at each iteration during the steady-state, whenever an update is implemented, it updates only three coefficients, i.e., the coefficients greater than 0.0001. Moreover, the update rates of the SM-PNLMS, the SM-$l_0$-NLMS, and the LCSM-NLMS2 algorithms are 12.43%, 20.06%, and 13.35%, respectively. Therefore, the LCSM-NLMS2 algorithm can attain lower update rate as well.

Figure 2 depicts the MSE learning curves of the SM-PNLMS, the SM-$l_0$-NLMS, and the LCSM-NLMS2 algorithms when the system 2 is considered as the unknown system. In the case of block sparse system, we can observe that the LCSM-NLMS2 can attain similar steady-state MSE to the SM-PNLMS algorithm; however, the SM-$l_0$-NLMS algorithm has remarkably higher steady-state MSE. Also, in the steady-state, the LCSM-NLMS2 algorithm updates only four coefficients, when an update is executed. Furthermore, the update rate of the SM-PNLMS, the SM-$l_0$-NLMS, and the LCSM-NLMS2 algorithms are 10.70%, 19.93%, and 11.92%, respectively.

Figure 3 illustrates the MSE learning curves of the SM-PNLMS, the SM-$l_0$-NLMS, and the LCSM-NLMS2 algorithms when the system 3 is considered as the unknown system. In the case of symmetric block sparse system, we can see that the SM-PNLMS and the LCSM-NLMS2 algorithms have similar performance; however, the LCSM-NLMS2 requires extremely lower computational resources. Also, note that the MSE of the SM-$l_0$-NLMS algorithm started increasing after the iteration 1000. Moreover, the update rates of the SM-PNLMS, the SM-$l_0$-NLMS, and the LCSM-NLMS2 algorithms are 7.65%, 8.33%, and 8.04%, respectively. Therefore, the LCSM-NLMS2 attains the lower computational cost since it updates only five coefficients during the steady-state, whenever an update is implemented.

V. CONCLUSIONS

In this paper, the LCSM-NLMS1 and the LCSM-NLMS2 algorithms have been introduced to take benefit of sparsity in the signal systems and to reduce computational cost. For this purpose, a simple update
Fig. 2. The MSE learning curves of the SM-PNLMS, the SM-$I_0$-NLMS, and the LCSM-NLMS2 algorithms when the unknown system is System 2.

Fig. 3. The MSE learning curves of the SM-PNLMS, the SM-$I_0$-NLMS, and the LCSM-NLMS2 algorithms when the unknown system is System 3.
equation has been derived so that it only updates the coefficients whose magnitudes are greater than a pre-defined positive value. Moreover, this approach is used with the set-membership technique to attain even lower computational burden and a low update rate. The numerical results have presented the superior performance of the LCSM-NLMS2 algorithm to some other sparsity-aware set-membership adaptive filters regarding the computational resources. In other words, the LCSM-NLMS2 algorithm executed as well as the SM-PNLMS and the SM-$l_0$-NLMS algorithms, whereas requiring fewer arithmetic operations.

REFERENCES

[1] D.L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancelers,” IEEE Transactions on Speech and Audio Processing, vol. 8, no. 5, pp. 508–518, Sept. 2000.
[2] J. Benesty and S.L. Gay, “An improved PNLS algorithm,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002), Dallas, USA, May 2002, vol. 2, pp. 1881–1884.
[3] S. Mula, V.C. Gogineni, and A.S. Dhar, “Robust proportionate adaptive filter architectures under impulsive noise,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 5, pp. 1223–1227, May 2019.
[4] S. Werner, J.A. Apolinario Jr., and P.S.R. Diniz, “Set-membership proportionate affine projection algorithms,” EURASIP Journal on Audio, Speech, and Music Processing, vol. 2007, no. 1, pp. 1–10, Jan. 2007.
[5] M.B. Salman and T. Çiloğlu, “Gradient controlled improved proportionate affine projection sign algorithm,” in 2018 26th Signal Processing and Communications Applications Conference (SIU 2018), May 2018, pp. 1–4.
[6] Y. Kopsinis, K. Slavakis, and S. Theodoridis, “Online sparse system identification and signal reconstruction using projections onto weighted $l_1$ balls,” IEEE Transactions on Signal Processing, vol. 59, no. 3, pp. 936–952, March 2011.
[7] Y. Gu, J. Jin, and S. Mei, “$l_0$ norm constraint LMS algorithm for sparse system identification,” IEEE Signal Processing Letters, vol. 16, no. 9, pp. 774–777, Sept. 2009.
[8] H. Yazdanpanah, J.A. Apolinário, P.S.R. Diniz, and M.V.S. Lima, “$l_0$-norm feature LMS algorithms,” in 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP 2018), Anaheim, CA, USA, Nov. 2018, pp. 311–315.
[9] H. Yazdanpanah and P.S.R. Diniz, “Recursive least-squares algorithms for sparse system modeling,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017), New Orleans, LA, USA, March 2017, pp. 3879–3883.
[10] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, “A sparsity promoting adaptive algorithm for distributed learning,” IEEE Transactions on Signal Processing, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.
[11] H. Yazdanpanah, M.V.S. Lima, and P.S.R. Diniz, “On the robustness of set-membership adaptive filtering algorithms,” EURASIP Journal on Advances in Signal Processing, vol. 2017, no. 1, pp. 1–12, Oct. 2017.
[12] P.S.R. Diniz, Adaptive Filtering: Algorithms and Practical Implementation, Springer, New York, USA, 4th edition, 2013.
[13] Z. Zheng, Z. Liu, H Zhao, Y. Yu, and L. Lu, “Robust set-membership normalized subband adaptive filtering algorithms and their application to acoustic echo cancellation,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 8, pp. 2098–2111, Aug. 2017.
[14] M.Z.A. Bhotto and A. Antoniou, “A robust constrained set-membership affine-projection adaptive-filtering algorithm,” in 2012 5th International Symposium on Communications, Control and Signal Processing, May 2012, pp. 1–4.
[15] H. Yazdanpanah and P.S.R. Diniz, “New trinion and quaternion set-membership affine projection algorithms,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 64, no. 2, pp. 216–220, Feb. 2017.
[16] C. Liu and Z. Zhang, “Set-membership normalised least M-estimate spline adaptive filtering algorithm in impulsive noise,” Electronics Letters, vol. 54, no. 6, pp. 393–395, 2018.
[17] J. Sharafi and A. Maarefparvar, “Robustness analysis of the data-selective Volterra NLMS algorithm,” arXiv preprint arXiv:2003.11514, 2020.
[18] S. Zhang and J. Zhang, “Set-membership NLMS algorithm with robust error bound,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 61, no. 7, pp. 536–540, July 2014.
[19] H. Yazdanpanah, P.S.R. Diniz, and M.V.S. Lima, “Improved simple set-membership affine projection algorithm for sparse system modeling: Analysis and implementation,” IET Signal Processing, vol. 14, no. 2, pp. 81–88, 2020.
[20] H. Yazdanpanah, P.S.R. Diniz, and M.V.S. Lima, “A simple set-membership affine projection algorithm for sparse system modeling,” in 2016 24th European Signal Processing Conference (EUSIPCO 2016), Budapest, Hungary, Aug. 2016, pp. 1798–1802.
[21] X. Wang and H.V. Poor, Wireless Communication Systems: Advanced Techniques for Signal Reception, Prentice Hall, Upper Saddle River, NJ, 2004.
[22] M.V.S. Lima, T.N. Ferreira, W.A. Martins, and P.S.R. Diniz, “Sparsity-aware data-selective adaptive filters,” IEEE Transactions on Signal Processing, vol. 62, no. 17, pp. 4557–4572, Sept. 2014.