Generalized URV Subspace Tracking LMS Algorithm [1]

S. Hosur and A. H. Tewfik and D. Boley

Dept. of Electrical Engineering and Computer Science, University of Minnesota, Minneapolis, MN 55455

Abstract

The convergence rate of the Least Mean Squares (LMS) algorithm is poor whenever the adaptive filter input auto-correlation matrix is ill-conditioned. In this paper we propose a new LMS algorithm to alleviate this problem. It uses a data dependent signal transformation. The algorithm tracks the subspaces corresponding to clusters of eigenvalues of the auto-correlation matrix of the input to the adaptive filter, which have the *same order of magnitude.* The algorithm up-dates the projection of the tap weights of the adaptive filter onto each subspace using LMS algorithms with *different* step sizes. The technique also permits adaptation only in those subspaces, which contain strong signal components leading to a lower excess Mean Squared Error (MSE) as compared to traditional algorithms.

1 Introduction

The LMS adaptive algorithm is the most popular algorithm for adaptive filtering because of its simplicity and robustness. However, its main drawback is slow convergence whenever the adaptive filter input auto-correlation matrix is ill-conditioned i.e. the eigenvalue spread of this matrix is large [1], [2]. A class of adaptive filters known as the transform domain filters have been developed for the purpose of convergence rate improvement [1]. All transform domain adaptive filters try to approximately de-correlate and scale the input to the adaptive filter in the transform domain, in order to obtain an autocorrelation matrix with zero eigen value spread in that domain.

The use of fixed parameter orthogonal transforms will not result in optimal convergence rates for all types of input signals. A transform which achieves optimal convergence rates for all inputs is the Karhounen-Loéuve Transform (KLT), which is data dependent. The KLT forms the inverse of the input autocorrelation matrix and uses this inverse to obtain a zero eigenvalue spread in the transform domain autocorrelation matrix. However when the input is very ill-conditioned, forming the inverse is not recommended. Also the excess MSE of the LMS adaptive filter is proportional to its step size. Increasing the step size tends to increase the rate of convergence of the algorithm but at the same time increases the excess MSE. In the case of transform domain filters, to achieve zero eigenvalue spread and hence increase convergence speed, the de-correlated modes are scaled in the transform domain. This scaling process is equivalent to using a larger step size for modes which contain little or no signal energy. As these modes essentially contain noise, the excess MSE increases.

Our goal in this paper is to develop an adaptive signal transformation which can be used to speed up the convergence rate of the LMS algorithm, and at the same time provide a way of adapting only to the strong signal modes, in order to decrease the excess MSE. The transform should also be able to track the signal behavior in a non-stationary environment. We develop such a data adaptive transform domain LMS algorithm, using a generalization of the rank revealing URV decomposition, first introduced by Stewart [8].

In the next section, we introduce the rank revealing URV decomposition and the idea of tracking subspaces corresponding to clusters of singular values. Section 3 generalizes the URV decomposition and introduces the idea of subspace domain LMS filtering. Simulation results are given in Section 4. Conclusions and future work are presented in the final section.

2 Subspace Tracking

Recently, some subspace updating techniques have been suggested in the context of total least squares (TLS) [3] - [7]. A Kalman filter was used to update the eigenvector corresponding to the smallest eigenvalue in [3]. However it was not suggested how to modify the algorithm in case of multiple eigenvalues corresponding to noise. [4] discusses a strategy to obtain a fast eigen-decomposition of a covariance matrix. The eigenvalues in the noise subspace are replaced by their average value and the same is done to the signal eigenvalues. This technique could work well if the exact eigenvalues could be grouped together in two tight clusters. In [5], [6], the eigen-problem on the covariance matrix is replaced by the singular problem, thereby reducing the condition numbers to their square roots and increasing numerical accuracy, and using the averaging technique of [4]. Again the assumption that the eigenvalues could be grouped into two tight clusters is made. In normal signal scenarios this assumption is generally not valid.

The URV decomposition was first introduced by Stewart [8], to break the eigenspace of $\mathbf{R}^N$, where N is the length of the impulse response of the adaptive filter, into two subspaces, one corresponding to the cluster of largest singular values and the other corresponding to the null subspace.

The URV decomposition of a rank k matrix $\mathbf{A}$ is a decomposition of the form

$$\mathbf{A} = \mathbf{U} \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{V}^H \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{R}$ is an upper triangular matrix of order k. The singular value decomposition can be considered as a special case of the URV decomposition where the upper triangular matrix $\mathbf{R}$ is diagonal.

The URV decomposition is *rank revealing* in the following sense. If $\mathbf{A}$ is nearly of rank k, i.e., if the first k singular values of $\mathbf{A}$ are much larger than the remaining singular values, then it can be shown that there exists a URV decomposition of $\mathbf{A}$ of the form

$$\mathbf{A} = \mathbf{U} \begin{pmatrix} \mathbf{R} & \mathbf{F} \\ \mathbf{0} & \mathbf{G} \end{pmatrix} \mathbf{V}^H \tag{2}$$

where $\mathbf{R}$ and $\mathbf{G}$ are upper triangular, the smallest entry in $\mathbf{R}$ is on the order of the k^{th} singular value of $\mathbf{A}$ and the sum of squares of the Frobenius norms of $\mathbf{F}$ and $\mathbf{G}$ is on the order of the sum of squares of the $(k+1)^{\text{st}}$ through the last singular values of $\mathbf{A}$. Hence, if the singular values of $\mathbf{A}$ satisfy

$$\sigma_1 \geq \cdots \geq \sigma_k > \sigma_{k+1} \geq \cdots \geq \sigma_N$$

where σ_k is large compared to σ_{k+1}, then

$$\inf(\mathbf{R}) \cong \sigma_k \sqrt{\|\mathbf{F}\|^2 + \|\mathbf{G}\|^2} \cong \sqrt{\sigma_{k+1}^2 + \cdots + \sigma_N^2}.$$

Thus, this decomposition immediately provides us with the sub-spaces corresponding to a group of largest singular values and another corresponding to the group of smallest singular values.

The URV updating procedure updates the URV decomposition of the data matrix corresponding to the input process, as additional data vectors become available. In essence, it updates the subspaces corresponding to the group of large eigenvalues and that of small eigenvalues of the correlation matrix of the input to the adaptive filter.

The URV updating algorithm consists of a series of plane (Givens) rotations [9], which are multiplications by orthogonal matrices of the form

$$\mathbf{Q} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & c & \mathbf{0} & s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -s & \mathbf{0} & c & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix}$$

where $c^2 + s^2 = 1$ and the $\mathbf{I}$'s represent identity matrices of appropriate dimensions. Pre-multiplication by a plane rotation operates on the rows of the matrix while post-multiplication operates on its columns. By using a sequence of plane rotations, appropriately constructed, in a special order, we can annihilate desired entries while filling in as few zero entries as possible, and then restoring the few zeros filled in. Each rotation applied from the right is accumulated in $\mathbf{V}$, to maintain $\mathbf{A} = \mathbf{URV}^H$, where $\mathbf{U}$ is not saved.

The URV updating procedure used is similar to that used in [7] and can be described as follows :-

- *Absorb a new row* : The matrix $\mathbf{A}$ is replaced by $(\beta \mathbf{A}^H \quad \mathbf{x})^H$ i.e a new row is augmented to the matrix $\mathbf{A}$ and $\beta < 1$ is the forgetting factor that damps out the effect of previous data. This problem can be rewritten as the problem of updating the matrix

  $$\begin{pmatrix} \mathbf{R} & \mathbf{F} \\ \mathbf{0} & \mathbf{G} \\ \mathbf{y}^H & \mathbf{z}^H \end{pmatrix}$$

  where $[\mathbf{y}^H \mathbf{z}^H] = \mathbf{x}^H \mathbf{V}$. When absorbing the row, the matrices $\mathbf{R}$ and $\mathbf{V}$ are updated. This is done by reducing $\mathbf{y}^H$ so that only one nonzero component and $\mathbf{G}$ remain upper triangular by applying plane rotations from the right. The entire matrix is then reduced to an upper triangular form. The rank is automatically incremented by one. No determination is made if the rank is actually increased.

- *Deflate and Refine* : It is possible for the matrix not to change rank or the rank could decrease. The Frobenius norms of $\mathbf{F}$ and $\mathbf{G}$ and an approximation of the last singular value of $\mathbf{R}$ and the corresponding singular vector are estimated using a *condition number estimator* [9]. One can also use any of the many condition number estimators proposed in literature [10]. The URV decomposition is then deflated by one, i.e., transformations are applied to decrease the rank index by one so that the smallest singular value of $\mathbf{R}$ is moved into the trailing columns. This is done by isolating the smallest singular value of $\mathbf{R}$ into its last column. A heuristic is used to determine if the rank boundary is in the middle of a cluster of singular values. If the answer to the heuristic is negative, deflation is done repeatedly till the heuristic finds a gap in the singular values, marking the end of the cluster. If s be the smallest singular value of $\mathbf{R}$, and f is the Frobenius norm of the trailing part, then according to the heuristic, a gap exists if $s > df$, where d is a user chosen *spread.* In order to allow for round off or other small noise, we pretend that the trailing part has an extra singular value equal to the user chosen *zero tolerance* b. The heuristic actually used is

  $$s^2 > d^2(f^2 + b^2)$$

  Thus any singular value below b or within a cluster of b will be treated as part of the trailing part. The user defined parameters are only the *spread* d and the *zero tolerance* b.

3 Generalized URV-LMS Algorithm

The URV updating procedure developed in the previous section, maintains and updates only two groups of singular values. If the input autocorrelation matrix has eigenvalues which could be so classified.

One can easily generalize the URV updating procedure described in the previous section to track subspaces corresponding to more than two singular value clusters. Each step of the generalized procedure may be viewed as a recursive application of the URV decomposition on the **R** matrix computed at the previous stage within the same step.

3.1 Generalized URV Updation

The idea of a generalized URV decomposition, which divides the singular values into more than two clusters can be introduced with the simple example where there are three groups of singular values. Now there are two singular value boundaries which have to be maintained and updated properly. When a new row is added and the updating is done, the procedure just pretends that we have an ordinary URV decomposition using the second singular value boundary. Because of the way the row is absorbed in the first step, the rank boundaries are automatically incremented by one. During the deflation step, the second boundary is moved left by one. This increases the dimension of the first boundary by one. Therefore a second deflation step is carried out to restore the separation of the first and the second cluster of singular values. This step does not affect the second boundary. In cases where one needs to cluster the eigenvalues into more than three groups, a similar procedure is followed.

Using the generalized URV decomposition, we can group the singular values of any matrix **A** into an arbitrary number of groups. The number of groups or clusters is determined automatically by the largest condition number that can be tolerated in each cluster. This implies that if one chooses the clustering to be done in such a way that each cluster has singular values of the same order of magnitude, the condition number in each cluster is improved which in turn implies a faster convergence of the LMS filter applied to a projection of weights in the corresponding subspace. The largest condition number is the maximum of the ratio of the largest singular value in each cluster to its smallest singular value. This value depends on the spread and zero tolerance, specified by the user.

3.2 The URV-LMS Algorithm

Let the input signal vector at time n be given as

$$\mathbf{x}_n = [x(n), x(n-1), \cdots, x(n-N+1)]^T \tag{3}$$

and let the weight vector at this time be h_n. The corresponding filter output is

$$z_n = \mathbf{x}_n^T \mathbf{h}_n \tag{4}$$

The output error e_n is the difference between the desired response $d(n)$ and the output of the adaptive filter at time n, z_n.

$$e_n = d(n) - z_n \tag{5}$$

The LMS algorithm tries to minimize the mean squared value of the error e_n, updating the weight vector $\mathbf{h}_n$ with each new data sample received as

$$\mathbf{h}_{n+1} = \mathbf{h}_n + 2\mu \mathbf{x}_n e_n \tag{6}$$

where the step size μ is a positive constant.

The convergence of the LMS algorithm depends on the condition number of the input autocorrelation matrix $\mathbf{R}_x \overset{\triangle}{=} E[\mathbf{x}_n \mathbf{x}_n^T]$. If the input vector is transformed as $\mathbf{x}_n$ to $\mathbf{u}_n = \mathbf{E}^H \mathbf{x}_n$, where E is the unitary eigenvector matrix of $\mathbf{R}_x$, then the output process $\mathbf{z}_n$ would be de-correlated. However, this implies that we need to perform an eigen decomposition of the autocorrelation matrix or a singular value decomposition of the data matrix at every adaptation, implying a computational computational complexity of $O(N^3)$ for every adaptation.

Instead of transforming the input using the eigen matrix, we could transform the input using the unitary matrix **V** obtained by the generalized URV decomposition. This would imply a savings in the computational costs as the URV decomposition can be updated with each new data at a relatively low computational cost. The input data vector $\mathbf{x}_n$ is transformed into the vector $\mathbf{y}_n$,

$$\mathbf{y}_n = \mathbf{V}^H \mathbf{x}_n \tag{7}$$

These transformed coefficients are then weighed using the subspace domain adaptive filter coefficient vector $\mathbf{g}_n$. The output signal z_n is given as

$$z_n = \mathbf{g}_n^T \mathbf{y}_n \tag{8}$$

The corresponding error signal e_n is given as

$$e_n = d(n) - z_n$$

The weight update equation is given by

$$\mathbf{g}_{n+1} = \mathbf{g}_n + 2\mathbf{M} e_n \mathbf{y}_n \tag{9}$$

where $\mathbf{M}$ is a diagonal matrix of step sizes used. The diagonal elements of $\mathbf{M}$ can usually be clustered into values of equal step sizes, corresponding to the subspaces isolated using the generalized URV. This clustering is due to the fact that each subspace is selected to minimize the condition number in that subspace. Hence adaptation of all the projected tap weights within each subspace has nearly the same convergence speed and one only needs to match the convergence speeds of the slow converging subspace projections of the tap weights to those of the fast converging subspace projections. This can be done by using larger step sizes for those subspace projections of the tap weights which converge slowly, to increase their convergence speed. Also the clusters obtained using the generalized URV are very well organized, with the largest singular value cluster first, making construction of $\mathbf{M}$ is very straightforward. The diagonal values of the upper triangular matrix generated in the generalized URV decomposition reflect the average magnitude of the singular values in each cluster. This information can also be used in the selection of the step sizes and hence in the construction of $\mathbf{M}$

An increase in step size usually implies an increase in the misadjustment error. The subspaces which belong to small singular values are dominated by noise and would tend to increase the noise in the solution. Thus by not adapting in those subspaces, we can reduce the misadjustment error. This can be simply done by setting those diagonal entries of $\mathbf{M}$, which correspond to projections of the tap weights onto these subspaces, to zero.

4 Simulation Results

We illustrate the performance of our procedure with a simple example in which a white noise random sequence $a(n)$ that can take the values ±1 with equal probability is filtered with a 3 tap FIR filter whose impulse response is a raised cosine $h(n) = (1 + \cos(2\pi(n-2)/W))/2$, $n = 1, 2, 3$. White Gaussian noise is added to the output and an 11 tap equalizer is adaptively constructed using the LMS and URV LMS algorithms (Figs. 4 and 4). Note that whereas the speed of convergence of the traditional LMS algorithm depends heavily on the eigenvalue spread of the input covariance matrix as determined by W, the URV LMS algorithm has no problem adapting to the environment even when W is large ($W = 3.5$) and the condition number of the input covariance matrix is correspondingly large ($\lambda_{\max}/\lambda_{\min} = 47.4592$).

An Adaptive Line Enhancer (ALE) experiment was also conducted to illustrate the performance of the algorithm when the adaptation is done only in the signal subspaces. The input to the ALE was chosen to be $0.1\cos\left(\frac{\pi}{15}n\right) + \cos\left(\frac{5\pi}{16}n\right)$ corrupted by white Gaussian noise of variance 0.0001. The autocorrelation matrix of the input to the ALE has only four significant eigenvalues, which could be grouped into two clusters. The ALE was adapted using both the LMS and the URV-LMS algorithms. The URV-LMS algorithm was adapted only in the subspaces corresponding to the two large singular value clusters. The superior performance of the URV-LMS algorithm can be seen from the learning curves are plotted in Fig. 4.

5 Conclusions

In this paper we developed a new LMS algorithm to improve the convergence rate whenever the input autocorrelation matrix is ill-conditioned. The algorithm is data adaptive and uses a generalization of the URV updation procedure of Stewart. The rate of convergence depends on the worst conditioned subspace, whose condition number depends on the user defined *spread* and *zero tolerances*. However, there is a tradeoff between the convergence speed, which depends on the condition number of each subspace and the computational expense of the generalized URV algorithm. As we noted in Section 3 and demonstrated using simulations, this algorithm can be used to adapt only in subspaces which contain signal components of significant strength, thus lowering the excess MSE. Our present work involves investigating the tradeoffs between error due to non-adaptation in noisy subspaces and the reduction in the excess MSE. Work on reducing the computational complexity and improving the quality of the subspaces estimated using the generalized URV decomposition is also in progress.

References

[1] D. F. Marshall, W. K. Jenkins and J. J. Murphy, “The Use of Orthogonal Transforms for Improving Performance of Adaptive Filters,” IEEE Trans. Circuits and Systems, Vol. 36, No. 4, pp. 474-483, April 1989.

[2] S. Haykin, *Adaptive Filter Theory*, 2nd ed., Prentice Hall, 1991.

[3] C. E. Davila, “Recursive Total Least Squares Algorithm for adaptive filtering,” Proceedings of ICASSP-91, ICASSP, pp 1853-1856, May 1991.

[4] K. B. Yu, “Recursive updating the eigenvalue decomposition of a covariance matrix,” IEEE Trans. Signal Proc., Vol. 39, pp. 1136-1145, 1991.

[5] E. M. Dowling and R. D. DeGroat, “Recursive Total Least Squares Adaptive Filtering,” Proceedings of the SPIE Conf. on Adaptive Signal Proc., vol. 1565, SPIE, pp. 35-46, July 1991.

[6] R. D. DeGroat, “Noniterative subspace tracking,” IEEE Trans. on Sig. Proc., Vol. 40, pp. 571-577, 1992.

[7] D. L. Boley and K. T. Sutherland, “Recursive Total Least Squares : An Alternative to the Discrete Kalman Filter,” Technical Report, Dept. of Comp. Sci., Univ. of Minn., TR 93-92, April 1993.

[8] G. W. Stewart, “An Updating Algorithm for Subspace Tracking,” IEEE Trans. Signal Proc., vol. 40, No. 6, pp. 1535-1541, June 1992.

[9] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., Baltimore MD, Johns Hopkins University Press, 1989.

[10] N. J. Higham, “A Survey of Condition Number Estimators For Triangular Matrices,” SIAM Rev. No. 29, pp. 575-596, 1987.