A NEW MATRIX DECOMPOSITION FOR SIGNAL PROCESSING

FRANKLIN T. LUK[1] and SANZHENG QIAO[2]

SUBTITLE. Despite its important signal processing applications, the generalized singular value decomposition (GSVD) is under-utilized due to the high updating cost. In this paper, we introduce a new approximate GSVD that is easily amenable to updating.

RUNNING TITLE. New Matrix Decomposition

ABSTRACT. To solve the noise subspace problem, we extend the generalized singular value decomposition to a new decomposition that can be updated at a low cost. In addition, we show how a forgetting factor can be incorporated in our new decomposition.

KEYWORDS. ULV decomposition, generalized SVD, updating, forgetting factor, signal processing.

1. Problem Definition

A recurring matrix problem in signal processing concerns generalized eigenvalues:

$$A^H A x = \lambda B^H B x,$$

where A is $n \times p$, $n \geq p$, B is $m \times p$, and $m \geq p$. We assume further that the matrix B has full column rank. Often, the generalized eigenvalues, call them d_j^2's, satisfy this property:

$$d_1^2 \geq d_2^2 \geq \cdots \geq d_{p-k}^2 >> d_{p-k+1}^2 \approx \cdots \approx d_p^2 . \tag{1.1}$$

In signal processing applications, the matrix A is made up of signal, noise, and interference and the matrix B of noise and interference. Take an active system as an example; the matrix A represents data received from the sensors when the transmitter is on and B represents data collected when the transmitter is off. While the number $p-k$ of the largest generalized eigenvalues gives the number of signals, the k-dimensional subspace spanned by

[1]Department of Computer Science, Rensselaer Polytechnic Institute, Troy, New York 12180, USA
[2]Communications Research Laboratory, McMaster University, Hamilton, Ontario L8S 4K1, Canada

the eigenvectors corresponding to the k smallest generalized eigenvalues is called the *noise* subspace. We are interested in the following problem:

Noise Subspace Problem. Compute an orthonormal basis for the noise subspace.

2. ULLV Decomposition

The noise subspace problem has a known solution for the *special* case where $B = I_p$, where I_p denotes a $p \times p$ identity matrix. Compute a singular value decomposition (SVD) of A:

$$A = UD_AV^H,$$

where U is $n \times p$ and *orthonormal*, i.e., $U^HU = I_p$, V is $p \times p$ and unitary, D_A is diagonal and $D_A = \mathrm{diag}(d_1, \ldots, d_p)$. From (1.1) we get

$$d_1 \geq d_2 \geq \cdots \geq d_{p-k} >> d_{p-k+1} \approx \cdots \approx d_p \geq 0, \tag{2.1}$$

and the desired orthonormal basis is given by the last k columns of V. However, the SVD is not amenable to efficient updating when a new row is added to A. A clever procedure was devised by Stewart (1993) in the form of the ULV decomposition (ULVD).

Definition of ULVD. The ULV decomposition of an $n \times p$ $(n \geq p)$ matrix A is

$$A = UL_AV^H,$$

where U is $n \times p$ and orthonormal and V is $p \times p$ and unitary as in the SVD, but the middle matrix L_A is lower triangular and essentially block diagonal. In particular,

$$L_A = \begin{pmatrix} \check{L}_A & 0 \\ E & K \end{pmatrix}, \tag{2.2}$$

where

(i) $\check{L}_A$ and K are lower triangular and $\check{L}_A$ is $(p-k) \times (p-k)$;

(ii) $\sigma_{\min}(\check{L}_A) \approx d_{p-k}$ and $||E||^2 + ||K||^2 \approx d_{p-k+1}^2 + \cdots + d_p^2$. $\square$

By $\sigma_{\min}(M)$ we mean the smallest singular value of a matrix M, and by $||M||$ we refer to the Frobenius norm of the matrix M.

Essentially, Stewart (1993) showed that to separate the noise subspace from the signal subspace, it suffices to reduce A to the 2×2 block lower triangular form L_A, where both E and K are very small in norm. The last k columns of V provide an orthonormal basis for the noise subspace. By generalizing the diagonal matrix D_A in the SVD to the triangular L_A in the ULVD, the noise subspace can be efficiently updated (see Stewart (1993)).

In this paper we consider the noise subspace problem for the *general* case where $B \neq I_p$. First, the problem may be solved via the generalized SVD (GSVD):

$$A = U_A D_A L V^H \quad \text{and} \quad B = U_B L V^H,$$

where U_A is $n \times p$ and orthonormal, U_B is $m \times p$ and orthonormal, V is $p \times p$ and unitary, L is $p \times p$ and lower triangular, and $D_A = \text{diag}(d_1, \ldots, d_p)$ where d_j's are the generalized singular values satisfying (2.1). Thus the last k columns of V provide a basis for the noise subspace. We propose here a generalized ULVD (ULLVD).

Definition of ULLVD. The ULLV decomposition of a pair of matrices A ($n \times p$, $n \geq p$) and B ($m \times p$, $m \geq p$) is

$$A = U_A L_A L V^H \quad \text{and} \quad B = U_B L V^H, \tag{2.3}$$

where U_A, U_B, V and L are just as in the GSVD. The new middle matrix L_A has the same form as in (2.2). $\square$

It follows that the desired orthonormal basis for the noise subspace is given by the last k columns of V.

3. Forgetting Factor and Updating the ULLVD

We examine how to update the ULLVD efficiently when new rows are added to A or B. As in signal processing, we incorporate a forgetting factor β, where $0 < \beta \leq 1$. Assume that we get the row data vectors x_1^T, x_2^T, ..., for A, and y_1^T, y_2^T, ..., for B. Adding a superscript for clarity, we get

$$A^{(n)} = \begin{pmatrix} \beta^{n-1} x_1^T \\ \vdots \\ \beta x_{n-1}^T \\ x_n^T \end{pmatrix} \quad \text{and} \quad B^{(m)} = \begin{pmatrix} \beta^{m-1} y_1^T \\ \vdots \\ \beta y_{m-1}^T \\ y_m^T \end{pmatrix}.$$

Thus,

$$A^{(n+1)} = \begin{pmatrix} \beta A^{(n)} \\ x_{n+1}^T \end{pmatrix} \quad \text{and} \quad B^{(m+1)} = \begin{pmatrix} \beta B^{(m)} \\ y_{m+1}^T \end{pmatrix}.$$

We consider the following problem:

Updating Problem. Given a ULLVD of A and B as in (2.3), and given $\hat{A}$ and $\hat{B}$ as *simple* updates of A and B, find a ULLVD of $\hat{A}$ and $\hat{B}$.

Case A. Add a Row to A: $A = A^{(n)}$, $\hat{A} = A^{(n+1)}$ and $B = \hat{B} = B^{(m)}$.

Case B. Add a Row to B: $A = \hat{A} = A^{(n)}$, $B = B^{(m)}$ and $\hat{B} = B^{(m+1)}$.

How do we begin? There are many possibilities. We propose to initialize A to a $p \times p$ zero matrix and B to a $p \times p$ matrix $B^{(p)}$; recall that B is required to have full column rank. Let $B = LV^H$ be an LQ decomposition, i.e., L is $p \times p$ and lower triangular and V is $p \times p$ and orthonormal. Then we get $U_A = I_p$, $L_A = 0$, and $U_B = I_p$.

3.1. Adding a Row to A

Step A0. Suppose we have a ULLVD of A and B, and a new row a^T is added to A. That is,

$$\hat{A} = \begin{pmatrix} \beta A \\ a^T \end{pmatrix} \quad \text{and} \quad \hat{B} = B.$$

We write down the initial decompositions as

$$\hat{A} = \begin{pmatrix} U_A & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \beta L_A & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} L \\ a^T V \end{pmatrix} V^H \quad \text{and} \quad \hat{B} = U_B \,(I_p \quad 0) \begin{pmatrix} L \\ a^T V \end{pmatrix} V^H. \tag{3.1}$$

Our goal is to transform the matrix factors of (3.1) into the proper forms of (2.3).

Step A1. We eliminate all but the first element of the row $a^T V$ using rotations from the right. For $i = p, p-1, \ldots, 2$, we annihilate the i-th element by postmultiplying with a rotation in the $(i-1, i)$ plane. This rotation will be incorporated into V. However, it introduces a nonzero $(i-1, i)$ entry in L, which is eliminated by applying from the left a rotation to rows $i-1$ and i of L. This eliminating scheme is illustrated as follows:

$$\begin{array}{cccc} & \downarrow & \downarrow & \\ \times & & & \\ \times & \times & & \\ \times & \times & \times & \\ \times & \times & \times & \times \\ \times & \times & \otimes & \end{array} \quad \rightsquigarrow \quad \begin{array}{ccccc} & \times & & & \\ \rightarrow & \times & \times & \otimes & \\ \rightarrow & \times & \times & \times & \\ & \times & \times & \times & \times \\ & \times & \times & & \end{array}$$

The rotation eliminating the $(i-1, i)$ element must be propagated to the left in both decompositions in (3.1). In the decomposition of B, this rotation can be directly incorporated into U_B because only the I_p part of the second factor is affected. In the decomposition of A, this rotation introduces a nonzero $(i-1, i)$ entry in the second factor, which is annihilated by applying a rotation from the left to its $(i-1)$-st and i-th rows. The following figure illustrates the effect of the two rotations:

$$\begin{array}{ccccc} & \downarrow & \downarrow & & \\ \times & & & & \\ \times & \times & & & \\ \times & \times & \times & & \\ \times & \times & \times & \times & \\ & & & & 1 \end{array} \quad \rightsquigarrow \quad \begin{array}{cccccc} & \times & & & & \\ \rightarrow & \times & \times & \otimes & & \\ \rightarrow & \times & \times & \times & & \\ & \times & \times & \times & \times & \\ & & & & & 1 \end{array}$$

The rotation annihilating the $(i-1, i)$ entry can be incorporated into U_A and consequently we get

$$\hat{A} = \begin{pmatrix} \bar{U}_A & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \bar{L}_A & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \bar{L} \\ \xi e_1^T \end{pmatrix} \bar{V}^H \quad \text{and} \quad \hat{B} = \bar{U}_B \,(I_p \quad 0) \begin{pmatrix} \bar{L} \\ \xi e_1^T \end{pmatrix} \bar{V}^H.$$

Step A2. We zero out the vector ξe_1^T using a transformation from the left. We apply a scaled rotation, say Y, of the form

$$Y = \begin{pmatrix} c^2 & & & & -c\bar{s} \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ cs & & & & c^2 \end{pmatrix},$$

from the left and rotate against the $(1,1)$ element of $\bar{L}$. Note that

$$Y^{-1} = \begin{pmatrix} 1 & & & & \bar{s}/c \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ -s/c & & & & 1 \end{pmatrix}.$$

When the transformation Y^{-1} is propagated to the left, it creates many nonzero elements in the last column of the second factor in both decompositions. The following figure shows the resultant nonzero structures:

$$\begin{pmatrix} \bar{L}_A & 0 \\ 0^T & 1 \end{pmatrix} Y^{-1} = \begin{pmatrix} \times & & & & \times \\ \times & \times & & & \times \\ \times & \times & \times & & \times \\ \times & \times & \times & \times & \times \\ \times & & & & \times \end{pmatrix} \quad \text{and} \quad (I_p \quad 0)\, Y^{-1} = \begin{pmatrix} 1 & & & & \times \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \end{pmatrix}.$$

Fortunately, the last column of these two matrices can be dropped because the third factor now has a zero row at the bottom. (The purpose of the scaled rotation is to keep the leading block of $(I_p \quad 0)Y^{-1}$ as an identity matrix.) Hence

$$\hat{A} = \begin{pmatrix} \tilde{U}_A & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \tilde{L}_A \\ \eta e_1^T \end{pmatrix} \hat{L}\hat{V}^H \quad \text{and} \quad B = \hat{U}_B \hat{L} \hat{V}^H.$$

Step A3. We zero out the vector ηe_1^T using a rotation from the left in the $(1, p+1)$ plane. Although the rotation creates many nonzeros in the last column of the first factor, this column can be dropped because the second factor now has a zero bottom row. Finally, we obtain the updated decompositions :

$$\hat{A} = \hat{U}_A \hat{L}_A \hat{L} \hat{V}^H \quad \text{and} \quad \hat{B} = \hat{U}_B \hat{L} \hat{V}^H.$$

Note that in Steps A1 to A3 we have taken special care to arrange the annihilation sequence so that $\hat{L}_A$ may satisfy (2.2) as much as possible. Specifically, the dimension of the block $\check{L}_A$ in (2.2) increases by at most one.

Step A4. The rank of $\hat{L}_A$ matrix may increase by one over that of L_A. A deflation followed by possibly an iterative refinement (Stewart, 1993) is applied to $\hat{L}_A$ to determine its rank.

3.2. Adding a Row to B

Step B0. When a new row b^T is added to B, i.e.,

$$\hat{A} = A \quad \text{and} \quad \hat{B} = \begin{pmatrix} \beta B \\ b^T \end{pmatrix},$$

we write

$$\hat{A} = U_A \begin{pmatrix} \beta^{-1} L_A & 0 \end{pmatrix} \begin{pmatrix} \beta L \\ b^T V \end{pmatrix} V^H \quad \text{and} \quad \hat{B} = \begin{pmatrix} U_B & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \beta L \\ b^T V \end{pmatrix} V^H. \tag{3.2}$$

Again, our goal is to transform the matrix factors of (3.2) into the proper forms of (2.3). Unlike (3.1), here we need to associate the forgetting factor β with L.

Step B1. We eliminate all but the first element of the row $b^T V$ using rotations from the right. Proceeding as in Step A1, we get

$$\hat{A} = \hat{U}_A \begin{pmatrix} \bar{L}_A & 0 \end{pmatrix} \begin{pmatrix} \bar{L} \\ \xi e_1^T \end{pmatrix} \hat{V}^H \quad \text{and} \quad \hat{B} = \begin{pmatrix} \bar{U}_B & 0 \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \bar{L} \\ \xi e_1^T \end{pmatrix} \hat{V}^H.$$

Step B2. We zero out the vector ξe_1^T using a rotation from the left in the $(1, p+1)$ plane. When the rotation is propagated to the left, it creates a nonzero vector right after $\bar{L}_A$ (and also $\bar{U}_B$). Fortunately, this new vector can be dropped since vector ξe_1^T has been annihilated. Consequently,

$$\hat{A} = \hat{U}_A \hat{L}_A \hat{L} \hat{V}^H \quad \text{and} \quad \hat{B} = \hat{U}_B \hat{L} \hat{V}^H.$$

As in Case A, we have arranged the sequence of transformations so that $\hat{L}_A$ will satisfy (2.2) as much as possible.

Step B3. The rank of $\hat{L}_A$ matrix may increase by one over that of L_A. A deflation (possibly followed by iterative refinement, cf. Stewart (1993)) is applied to $\hat{L}_A$ to determine its rank.

3.3. Computational Cost

We are only interested in an orthonormal basis for the noise subspace, and so we do not need the matrices U_A and U_B. That is, we update only the three $p \times p$ matrices L_A, L and V, deriving a very efficient method. First, consider Case A. The initialization step costs $1.5p^2$ multiplies for forming βL_A and $a^T V$. Step A1 requires approximately $12p^2$ multiplies. Step A2 costs $O(p)$ multiplies, and Step A3 $O(1)$ multiplies. The cost of Step A4 depends on the rank of $\hat{L}_A$ (see Stewart (1993)), which is usually less than p. Second, for Case B, the initialization step costs $2p^2$ multiplies because we need to form $\beta^{-1} L_A$, βL and $b^T V$. Step B1 requires approximately $12p^2$ multiplies, and Step B2 $O(p)$ multiplies. The cost of Step B3 is dependent on the rank of $\hat{L}_A$.

4. Example

Our new ULLVD algorithm was programmed in MATLAB and run on a SUN3/80 at McMaster University; the machine precision equals 2×10^{-16}. The results show that this adaptive algorithm computes the noise subspace just as accurately as a stable GSVD algorithm implemented along the lines described in Van Loan (1985) and Luk and Qiao (1986). In our example, B is a random 9×5 matrix. We let

$$A^{(140)} = \begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{pmatrix},$$

where A_0 is a 5×5 null matrix, A_1 a 20×5 random matrix of rank 3, A_2 a 60×5 random matrix of rank 2, and A_3 a 60×5 random matrix of rank 3. Hence $A^{(140)}$ is a 145×5 matrix whose top five rows are zeros. Define $A^{(n)}$ by

$$A^{(n)} = \begin{pmatrix} A_0 \\ \beta^{n-1} x_1^T \\ \vdots \\ \beta x_{n-1}^T \\ x_n^T \end{pmatrix},$$

for $n = 1, 2, \ldots, 140$. In order to avoid a lengthy presentation of the results, the forgetting factor β is set to 0.7; in practice β is often very close to 1, e.g., 0.95. The relative accuracy of the noise subspace computed by the adaptive ULLVD algorithm is determined as follows. Let

$$A^{(n)} = U^{(n)} L_A^{(n)} L^{(n)} V^{(n)H}$$

be a ULLVD of $A^{(n)}$, and

$$A^{(n)} = \bar{U}^{(n)} D_A^{(n)} \bar{L}^{(n)} \bar{V}^{(n)H}$$

be a GSVD of $A^{(n)}$. Also, partition

$$V^{(n)} = \begin{pmatrix} V_S^{(n)} & V_N^{(n)} \end{pmatrix} \quad \text{and} \quad \bar{V}^{(n)} = \begin{pmatrix} \bar{V}_S^{(n)} & \bar{V}_N^{(n)} \end{pmatrix},$$

so that $V_N^{(n)}$ and $\bar{V}_N^{(n)}$ represent the bases for the noise subspaces computed by the two different algorithms. The parameter $\theta^{(n)} = \|(V_N^{(n)})^H \bar{V}_S^{(n)}\|_2$ approximates the angle between the two noise subspaces. Our experimental results are presented in Table 1. First, we initialized $A^{(0)} = A_0$ and $L_A^{(0)} = 0$, and performed an LQ decomposition on the top five rows of B to get $L^{(0)}$ and $V^{(0)}$. Next, we added the first five data rows x_1^T to x_5^T. As a little detour, we increased the row dimension of B from five to nine; the detected rank of $L_A^{(n)}$ stayed at three and $\theta^{(n)}$ varied between 1.4×10^{-15} and 3.7×10^{-15}. Finally, we added the remaining data rows x_6^T to x_{140}^T. We were pleased that our ULLVD algorithm computed a noise subspace very close to the one given by the GSVD. However, a minor loss in accuracy occurred in steps 52 to 58 (i.e., right after the rank of $L_A^{(n)}$ decreased from four to three) because the condition estimator got confused when the smallest singular values were closely clustered. When $L_A^{(n)}$ has full column rank, the noise spaces are null and $\theta^{(n)}$ is indicated by NM (not meaningful) in Table 1.

n	1	2	3	4	5	6	15	20	21	22	$\cdots$	41	42
$\mathrm{rank}(L_A^{(n)})$	1	2	3	3	3	3	3	3	4	5	$\cdots$	5	4
$\theta^{(n)}(\times 10^{-15})$	2.1	2.4	4.2	2.4	1.9	3.5	1.8	2.4	1.5	NM	$\cdots$	NM	2.0

n	50	51	52	53	54	55	56	57	58	59
$\mathrm{rank}(L_A^{(n)})$	4	3	3	3	3	3	3	3	3	2
$\theta^{(n)}(\times 10^{-12})$	1.6×10^{-3}	2.4×10^{-3}	2.6	2.8	2.6	3.4	3.1	3.3	3.1	1.5×10^{-3}

n	70	80	81	82	83	$\cdots$	103	104	108	109	125	140
$\mathrm{rank}(L_A^{(n)})$	2	2	3	4	5	$\cdots$	5	4	4	3	3	3
$\theta^{(n)}(\times 10^{-15})$	1.3	0.8	1.9	1.4	NM	$\cdots$	NM	1.4	2.6	3.0	1.2	1.8

Table 1: Accuracy of ULLVD Algorithm in Noise Subspace Estimation

5. Conclusion

In this paper, we consider the problem of computing an orthonormal basis for the null subspace; in particular, we are interested in efficient updating when new data arrive. For a matrix pair A and B with p columns each, the traditional GSVD technique requires $O(p^3)$ flops, whereas our new ULLV approach updates the basis vectors using only $O(p^2)$ flops.

Acknowledgements

F. T. Luk was supported in part by the Office of Naval Research under contract N00014-93-1-0268 and S. Qiao by the Natural Sciences and Engineering Research Council of Canada under grant OGP0046301.

References

[1] Luk, F. T. and S. Qiao (1986). Computing the CS-decomposition on systolic arrays. *SIAM J. Sci. Statist. Comput.*, **7**, 1121-1125.

[2] Stewart, G. W. (1993). Updating a rank-revealing ULV decomposition. *SIAM J. Matrix Anal. Appl.*, **14**, 494-499.

[3] Van Loan, C. F. (1985). Computing the CS and generalized singular value decomposition. *Numer. Math.*, **46**, 479-492.