Available online at www.sciencedirect.com
SCIENCE DIRECT®

Mathematical and Computer Modelling 44 (2006) 742–748

MATHEMATICAL AND COMPUTER MODELLING

www.elsevier.com/locate/mcm

An efficient algorithm for rank and subspace tracking☆

Hasan Erbay

Department of Mathematics, Kırıkkale University, Yahşihan, Kırıkkale, 71100, Turkey

Received 28 May 2005; received in revised form 6 February 2006; accepted 10 February 2006

Abstract

Traditionally, the singular value decomposition (SVD) has been used in rank and subspace tracking methods. However, the SVD is computationally costly, especially when the problem is recursive in nature and the size of the matrix is large.

The truncated ULV decomposition (TULV) is an alternative to the SVD. It provides a good approximation to subspaces for the data matrix and can be modified quickly to reflect changes in the data. It also reveals the rank of the matrix.

This paper presents a TULV updating algorithm. The algorithm is most efficient when the matrix is of low rank. Numerical results are presented that illustrate the accuracy of the algorithm.

Keywords: Singular value decomposition; Modifying decompositions; Subspace tracking; Rank estimation; ULV decomposition

1. Introduction

The singular value decomposition (SVD) has important applications in signal processing where reliable computation of the numerical rank, as well as the numerical range, null (noise) and dominant singular (signal) subspaces are required. Unfortunately, it is computationally expensive and difficult to modify [11]. This can be a drawback for recursive problems that require a simple matrix update when the new row is added to the matrix as happens in subspace-based signal processing and applications where the numerical rank and numerical subspaces are to be tracked. Therefore, alternative decompositions that enable the use of the existing results for incorporating changes in data are desired.

The rank-revealing QR decompositions (RRQR) [4,5], the rank-revealing URV decomposition (URV) [14] and the rank-revealing ULV decomposition (ULV) [15] have been considered as alternatives. Recently, Barlow and Erbay proposed the truncated ULV decomposition (TULV) [2].

The following notation is used throughout the paper. Matrices are represented by bold upper-case characters and column vectors by bold lower-case characters. $\mathbf{e}_i$ is the ith unit vector with the appropriate dimension. $\mathbf{I}$ is the identity matrix. $\|\cdot\|$ and $\|\cdot\|_F$ denote the two norm and the Frobenius norm, respectively.

The TULV is a special case of the two-sided orthogonal decompositions defined by Faddeev, Kublanovskaya, and Faddeeva [8] and Hanson and Lawson [12].

☆ Supported by DPT project under grant no. 2002K120430.
E-mail address: hxe68@yahoo.com.

0895-7177/$ - see front matter
doi:10.1016/j.mcm.2006.02.011

Given an $m \times n$ matrix $\mathbf{A}$, where $m \geq n$ and rank $(\mathbf{A}) = k$, the TULV of $\mathbf{A}$ is of the form

$$\mathbf{A} = \mathbf{U}_1 \mathbf{L} \mathbf{V}_1^{\mathrm{T}} + \mathbf{E}, \tag{1.1}$$

where $\mathbf{U}_1 \in \Re^{m\times k}$, $\mathbf{V}_1 \in \Re^{n\times k}$ are left orthogonal, that is, $\mathbf{U}_1^{\mathrm{T}}\mathbf{U}_1 = \mathbf{V}_1^{\mathrm{T}}\mathbf{V}_1 = \mathbf{I}$, $\mathbf{L} \in \Re^{k\times k}$ is nonsingular lower triangular, and $\mathbf{E} \in \Re^{m\times n}$ is an error matrix. The matrices $\mathbf{L}$ and $\mathbf{E}$ satisfy

$$\|\mathbf{L}^{-1}\| \leq \epsilon^{-1}, \qquad \|\mathbf{E}\| \leq \epsilon, \qquad \mathbf{U}_1^{\mathrm{T}}\mathbf{E} = 0. \tag{1.2}$$

The value ϵ is the tolerance or “noise level”. The value k is usually referred to as the ϵ-rank or the rank “revealed” by the TULV. It is not the numerical rank unless ϵ is very close to the machine precision times the norm of $\mathbf{A}$, and, in practice, ϵ will often be larger than that value.

Throughout the rest of the paper “rank” will denote ϵ-rank.

In this paper we develop an $\mathcal{O}(nk)$ algorithm for the updating.

Updating. Given the TULV of $\mathbf{A}$, produce the TULV of

$$\mathbf{A}_u = \begin{pmatrix} \mathbf{a}^{\mathrm{T}} \\ \mathbf{A} \end{pmatrix}.$$

Due to memory size and computational complexity only $\mathbf{L}$, $\mathbf{U}_1$ and $\mathbf{V}_1$ are stored in some large scale applications [10] where $k \ll n$. Thus we need an efficient, stable update routine for these applications.

Updating routines given by Stewart [15], Yoon and Barlow [1,16] and Barlow, Erbay and Zhang [6] store and use the error matrix $\mathbf{E}$, so they are unsuitable for these applications.

Rabideau [13] developed a URV-based rank-adaptive subspace tracking algorithm with the same complexity, but the URV is less satisfactory for applications in which an approximate null space is needed [9,15].

In the next section we will present an updating algorithm to get the TULV of $\mathbf{A}_u$. Numerical results show that our algorithm reveals the rank and also produces approximations to subspaces of the matrix $\mathbf{A}_u$.

2. TULV updating

An important subproblem in the updating algorithm is that given $\mathbf{a} \in \Re^n$ and $\mathbf{V}_1 \in \Re^{n\times k}$ left orthogonal, find $\mathbf{v} \in \Re^n$ and $\mathbf{d} \in \Re^{k+1}$ such that

$$\|\mathbf{v}\| = 1, \qquad \mathbf{V}_1^{\mathrm{T}}\mathbf{v} = 0, \qquad \begin{pmatrix} \mathbf{V}_1 & \mathbf{v} \end{pmatrix} \mathbf{d} = \mathbf{a}.$$

Such an algorithm can be found in [3]. The algorithm runs in $O(nk)$ time.

Suppose that the new data $\mathbf{a}$ has arrived and we wish to determine the TULV of $\mathbf{A}_u$. Basic steps of the updating algorithm are summarized below.

Algorithm 2.1 (*TULV Updating*).

Input:

$\mathbf{L}$, $k \times k$ lower triangular matrix,

$\mathbf{U}_1$, $m \times k$ left orthogonal matrix,

$\mathbf{V}_1$, $n \times k$ left orthogonal matrix,

ϵ, a rank tolerance.

1. Compute $\mathbf{v} \in \Re^n$ *and* $\mathbf{d} \in \Re^{k+1}$ such that

$$\mathbf{a} = \begin{pmatrix} \mathbf{V}_1 & \mathbf{v} \end{pmatrix} \mathbf{d}, \quad \|\mathbf{v}\| = 1, \qquad V_1^{\mathrm{T}}\mathbf{v} = 0,$$

and partition $\mathbf{d}$ as

$$\mathbf{d} = \begin{array}{c} k \\ 1 \end{array} \begin{pmatrix} \mathbf{f} \\ \beta \end{pmatrix}.$$

Then

$$A_u = \begin{pmatrix} 0 & 1 \\ \mathbf{U}_1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{L} & 0 \\ \mathbf{f}^{\mathrm{T}} & \beta \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^{\mathrm{T}} \\ \mathbf{v}^{\mathrm{T}} \end{pmatrix} + \begin{matrix} 1 \\ m \end{matrix} \begin{pmatrix} 0 \\ \mathbf{E} \end{pmatrix}.$$

2. Find the smallest singular value σ_{k+1} of the matrix $\begin{pmatrix} \mathbf{L} & 0 \\ \mathbf{f}^{\mathrm{T}} & \beta \end{pmatrix}$ such that

$$\begin{pmatrix} \mathbf{L} & 0 \\ \mathbf{f}^{\mathrm{T}} & \beta \end{pmatrix} \mathbf{v}_{k+1} = \sigma_{k+1} \mathbf{u}_{k+1}.$$

3. If $(\sigma_{k+1} \geq \epsilon)$ then set

$$\bar{\mathbf{L}} = \begin{pmatrix} \mathbf{L} & 0 \\ \mathbf{f}^{\mathrm{T}} & \alpha \end{pmatrix}, \qquad \bar{\mathbf{U}}_1 = \begin{pmatrix} 0 & 1 \\ \mathbf{U}_1 & 0 \end{pmatrix}, \qquad \bar{\mathbf{V}}_1 = \begin{pmatrix} \mathbf{V}_1^{\mathrm{T}} \\ \mathbf{v}^{\mathrm{T}} \end{pmatrix}$$

where

$$\bar{\mathbf{E}} = \left(\mathbf{I} - \bar{\mathbf{U}}\bar{\mathbf{U}}^{\mathrm{T}}\right) \mathbf{A}_u,$$

else if $(\sigma_{k+1} < \epsilon)$ then find orthogonal matrices $\mathbf{P}$, $\mathbf{Q}$ with $\mathbf{P}$ such that

$$\mathbf{P}^{\mathrm{T}} \mathbf{u}_{k+1} = \mathbf{e}_{k+1}$$

and with $\mathbf{Q}$ such that

$$\mathbf{P}^{\mathrm{T}} \begin{pmatrix} \mathbf{L} & 0 \\ \mathbf{f}^{\mathrm{T}} & \alpha \end{pmatrix} \mathbf{Q} = \begin{pmatrix} \bar{\mathbf{L}} & 0 \\ \bar{\mathbf{f}}^{\mathrm{T}} & \bar{\beta} \end{pmatrix}$$

remains lower triangular.
Take

$$\begin{pmatrix} \bar{\mathbf{U}}_1 & \bar{\mathbf{u}}_0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \mathbf{U}_1 & 0 \end{pmatrix} \mathbf{P}$$

$$\begin{pmatrix} \bar{\mathbf{V}}_1 & \bar{\mathbf{v}}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{V}_1 & \mathbf{v}_{k+1} \end{pmatrix} \mathbf{Q}.$$

4. Finally,

$$\mathbf{A}_u = \bar{\mathbf{U}}_1 \bar{\mathbf{L}} \bar{\mathbf{V}}_1^{\mathrm{T}} + \bar{\mathbf{E}}, \qquad \bar{\mathbf{E}} = \left(\mathbf{I} - \bar{\mathbf{U}}_1 \bar{\mathbf{U}}_1^{\mathrm{T}}\right) \mathbf{A}_u.$$

The updating algorithm runs in $\mathcal{O}(nk)$ time if only the subspace $\mathbf{V}_1$ is required, but if the subspace $\mathbf{U}_1$ is also required, the complexity becomes $\mathcal{O}(mk)$.

Theorem 2.2. *Let* $\mathbf{A} = \mathbf{U}_1 \mathbf{L} \mathbf{V}_1^{\mathrm{T}} + \mathbf{E}$ *be the TULV of* $\mathbf{A}$. *Then*

$$\mathbf{E} = (\mathbf{I} - \mathbf{U}_1 \mathbf{U}_1^{\mathrm{T}}) \mathbf{A}.$$

Proof. Since $\mathbf{U}_1^{\mathrm{T}} \mathbf{E} = 0$ and $\mathbf{U}_1^{\mathrm{T}} \mathbf{U}_1 = \mathbf{I}$, we have

$$\mathbf{U}_1^{\mathrm{T}} \mathbf{A} = \mathbf{L} \mathbf{V}_1^{\mathrm{T}}$$

from which the theorem follows immediately. □

Proposition 2.3. *Let* $\mathbf{A} = \mathbf{U}_1 \mathbf{L} \mathbf{V}_1^{\mathrm{T}} + \mathbf{E}$ *be the TULV of* $\mathbf{A}$. *Let* $\mathbf{A}_u = \bar{\mathbf{U}}_1 \bar{\mathbf{L}} \bar{\mathbf{V}}_1^{\mathrm{T}} + \bar{\mathbf{E}}$ *be the TULV of* $\mathbf{A}_u = \begin{pmatrix} \mathbf{a}^{\mathrm{T}} \\ \mathbf{A} \end{pmatrix}$ *obtained by Algorithm* 2.1. *If* $\operatorname{rank}(\mathbf{A}_u) > \operatorname{rank}(\mathbf{A})$ *then*

$$\|\bar{\mathbf{E}}\|_F^2 = \|\mathbf{E}\|_F^2.$$

Proof. By Theorem 2.2,

$$\begin{aligned}
\bar{\mathbf{E}} &= \left(\mathbf{I} - \bar{\mathbf{U}}_1\bar{\mathbf{U}}_1^{\mathrm{T}}\right)\mathbf{A}_u \\
&= \left(\mathbf{I} - \begin{pmatrix} 0 & 1 \\ \mathbf{U}_1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ \mathbf{U}_1 & 0 \end{pmatrix}^{\mathrm{T}}\right)\begin{pmatrix} \mathbf{a}^{\mathrm{T}} \\ \mathbf{A} \end{pmatrix} \\
&= \left(\mathbf{I} - \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{U}_1\mathbf{U}_1^{\mathrm{T}} \end{pmatrix}\right)\begin{pmatrix} \mathbf{a}^{\mathrm{T}} \\ \mathbf{A} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{a}^{\mathrm{T}} \\ \mathbf{A} \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & \mathbf{U}_1\mathbf{U}_1^{\mathrm{T}} \end{pmatrix}\begin{pmatrix} \mathbf{a}^{\mathrm{T}} \\ \mathbf{A} \end{pmatrix} \\
&= \begin{pmatrix} 0 \\ (\mathbf{I} - \mathbf{U}_1\mathbf{U}_1^{\mathrm{T}})\mathbf{A} \end{pmatrix} \\
&= \begin{pmatrix} 0 \\ \mathbf{E} \end{pmatrix}
\end{aligned}$$

which completes the proof. □

Proposition 2.4. *Let* $\mathbf{A} = \mathbf{U}_1\mathbf{L}\mathbf{V}_1^{\mathrm{T}} + \mathbf{E}$ *be the TULV of* $\mathbf{A}$ *having rank* k. *Let* $\mathbf{A}_u = \bar{\mathbf{U}}_1\bar{\mathbf{L}}\bar{\mathbf{V}}_1^{\mathrm{T}} + \bar{\mathbf{E}}$ *be the TULV of* $\mathbf{A}_u = \begin{pmatrix} \mathbf{a}^{\mathrm{T}} \\ \mathbf{A} \end{pmatrix}$ *obtained by Algorithm 2.1. If* $\operatorname{rank}(\mathbf{A}_u) \le \operatorname{rank}(\mathbf{A})$ *then*

$$\|\bar{\mathbf{E}}\|_F^2 = \|\mathbf{E}\|_F^2 + \sigma_{k+1}^2$$

where σ_{k+1} *is the smallest singular value of* $\begin{pmatrix} \mathbf{L} & 0 \\ \mathbf{f}^{\mathrm{T}} & \alpha \end{pmatrix}$ *with* $\|\mathbf{v}\| = 1$, $\mathbf{V}_1^{\mathrm{T}}\mathbf{v} = 0$, $\mathbf{a} = \begin{pmatrix} \mathbf{V}_1 & \mathbf{v} \end{pmatrix}\mathbf{d}$ *and* $\mathbf{d} = \begin{pmatrix} \mathbf{f} \\ \alpha \end{pmatrix}$.

Observe that if the new data increases the rank then the conditions in (1.2) hold. However, no increase in rank destroys the rank-revealing character of the matrix. In this case one or repeated use of the refinement algorithm [3,15] restores it. This algorithm takes the TULV of $\mathbf{A}$ and produces a new TULV such that

$$\mathbf{A} = \bar{\mathbf{U}}_1\bar{\mathbf{L}}\bar{\mathbf{V}}_1^{\mathrm{T}} + \bar{\mathbf{E}}$$

where $\bar{\mathbf{U}}_1 \in \Re^{m\times s}$, $\bar{\mathbf{V}}_1 \in \Re^{n\times s}$, $\bar{\mathbf{L}} \in \Re^{s\times s}$ for $s \le k$ such that

$$\|\bar{\mathbf{E}}\|_F < \|\mathbf{E}\|_F.$$

3. Numerical experiments

In this section we analyze the performance of the TULV updating algorithm. We tested the algorithm in the context of the exponential window method. In this method new data $\mathbf{a}(t) \in \Re^n$ is continuously added to the data matrix $\mathbf{A}(t) \in \Re^{(m_0+t)\times n}$ with $m_0 \ge n$ at time t. For each t, $t = 0, 1, 2, \ldots$,

$$\mathbf{A}(t) = \begin{pmatrix} \mathbf{a}(t)^{\mathrm{T}} \\ \alpha\mathbf{a}(t-1)^{\mathrm{T}} \\ \alpha^2\mathbf{a}(t-2)^{\mathrm{T}} \\ \vdots \\ \alpha^t\mathbf{A}(0) \end{pmatrix}$$

where $\mathbf{A}(0) \in \Re^{m_0\times n}$ is an initial matrix and $\alpha < 1$ is a positive forgetting factor.

Given the TULV of $\mathbf{A}(t)$, it is desirable to obtain the TULV of the matrix

$$\mathbf{A}(t+1) = \begin{pmatrix} \mathbf{a}(t+1)^{\mathrm{T}} \\ \alpha\mathbf{A}(t) \end{pmatrix}.$$

Recall that after an update in the exponential window method, the numerical rank can increase, decrease or stay the same, due to the forgetting factor.

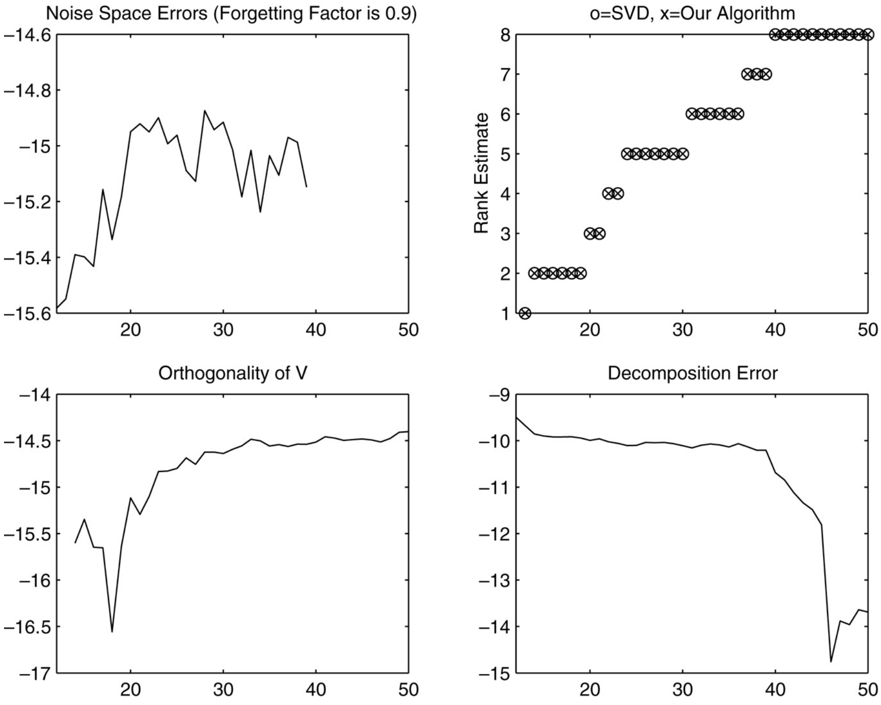

Fig. 3.1. The plots show some results obtained by Example 3.1.

The examples were run in Matlab 6.5 on a personal computer.
Matlab's SVD of the window matrix $\mathbf{A}(t)$ was used as a reference in checking the accuracy.
Let

$$\mathbf{A}(t) = \mathbf{Y}(t)\ \boldsymbol{\Sigma}(t)\ \mathbf{Z}(t)^{\mathrm{T}}, \qquad \mathbf{Z}(t) = \begin{pmatrix}\mathbf{Z}_1(t) & \mathbf{Z}_2(t)\end{pmatrix}$$

be the SVD of $\mathbf{A}(t)$ obtained by Matlab's SVD algorithm at step t. Also, let

$$\mathbf{A}(t) = \mathbf{U}_1(t)\mathbf{L}(t)\mathbf{V}_1(t)^{\mathrm{T}} + \mathbf{E}(t)$$

be the TULV of $\mathbf{A}(t)$ obtained by our algorithm. In the Davis–Kahan [7] framework, the accuracy of the subspace errors is characterized by

$$\sin\theta(t) = \|\mathbf{V}_1^{\mathrm{T}}(t)\mathbf{Z}_2(t)\|. \tag{3.1}$$

We plotted $\sin\theta$ on the log 10 scale to have a better view of the errors. We tracked the rank of the matrix $\mathbf{A}(t)$ and compared it to Matlab's SVD. We checked the orthogonality of $\mathbf{V}_1\ (t)$ by computing

$$\|\mathbf{I}_k - \mathbf{V}_1^{\mathrm{T}}(t)\mathbf{V}_1(t)\|$$

and plotting it. We also computed the decomposition error

$$\|\mathbf{A}(t) - \mathbf{U}_1(t)\mathbf{L}(t)\mathbf{V}_1^{\mathrm{T}}(t)\|$$

at each step t.

Example 3.1. The data matrix $\mathbf{A}$ is a 50-by-8 random matrix with elements generated from on the interval (0, 1). To vary the numerical rank of the matrix, 42 randomly chosen rows of $\mathbf{A}$ were multiplied by $\eta = 10^{-9}$. The tolerance value ϵ for determining the numerical rank was 10^{-8}. The initial matrix $\mathbf{A}(0)$ was of size 12-by-8 and the forgetting factor α was 0.9.

Example 3.2. The data matrix $\mathbf{A}$ is a 50-by-8 random matrix with elements generated from on the interval (0, 1). To vary the numerical rank of the matrix, 42 randomly chosen rows of $\mathbf{A}$ were multiplied by $\eta = 10^{-9}$. The tolerance

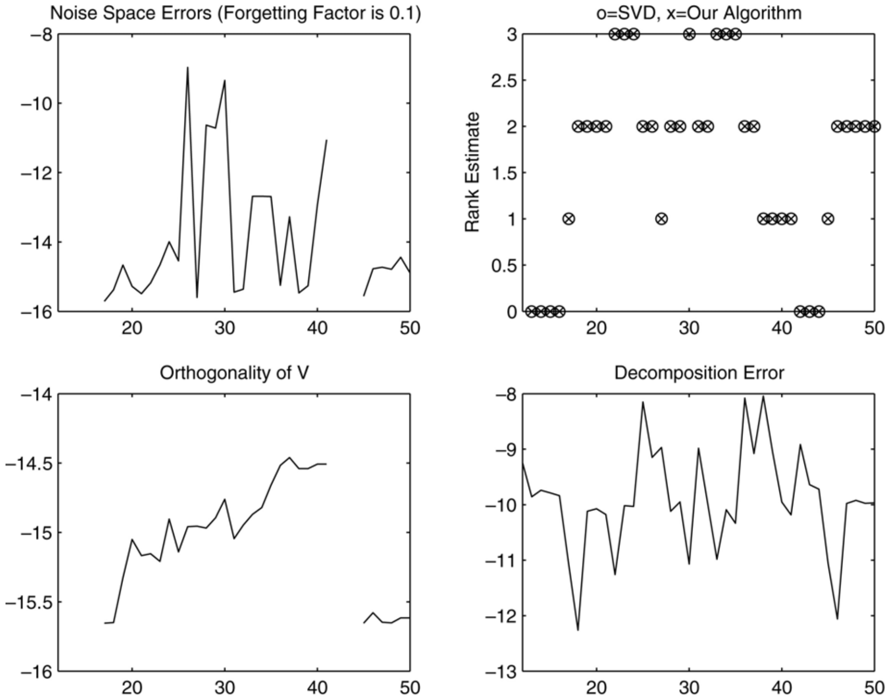

Fig. 3.2. The plots show some results obtained for Example 3.2.

value ϵ for determining the numerical rank was 10^{-8}. The initial matrix $\mathbf{A}(0)$ was of size 12-by-8 and the forgetting factor α was 0.1.

The first plot in Figs. 3.1 and 3.2 shows the subspace errors obtained by our algorithm.

The second plot in Figs. 3.1 and 3.2 shows the ability of the algorithm to track the numerical rank in spite of frequent rank changes. The cross '$\times$' and the circle '$\circ$' represent the numerical rank estimates from the TULV updating algorithm and Matlab's SVD algorithm, respectively. The horizontal axes represent the window steps.

The third plot in Figs. 3.1 and 3.2 shows that $\mathbf{V}_1(t)$ is numerically orthogonal at each window step t.

The fourth plot shows the accuracy of the decomposition obtained from the TULV updating algorithm.

References

[1] J.L. Barlow, P.A. Yoon, Solving recursive TLS problems using rank-revealing ULV decomposition, in: S. Van Huffel (Ed.), Proc. Workshop on TLS and Errors-In-Variables, SIAM Publications, Philadelphia, PA, 1997, pp. 117–126.
[2] J.L. Barlow, H. Erbay, Modifiable low-rank approximation to a matrix (in preparation).
[3] J.L. Barlow, H. Erbay, I. Slapničar, Alternative algorithms for construction and refinement of ULV decompositions, SIAM J. Matrix Anal. Appl. 27 (2005) 198–211.
[4] A. Björck, Numerical methods for least squares problems, SIAM Publications, Philadelphia, PA, 1996.
[5] T.F. Chan, Rank revealing QR factorizations, Linear Algebra Appl. 88 (1987) 67–82.
[6] H. Erbay, J.L. Barlow, Z. Zhang, A modified Gram–Schmidt-based downdating technique for the ULV decompositions with applications to recursive TLS problems, Comput. Statist. Data Anal. 41 (1) (2002) 195–209.
[7] C. Davis, W.M. Kahan, The rotation of eigenvectors by a perturbation III, SIAM J. Numer. Anal. 7 (1970) 1–46.
[8] D.K. Faddeev, V.N. Kublanovskaya, V.N. Faddeeva, Solution of linear algebraic systems with rectangular matrices, Proc. Steklov Inst. Math. 96 (1968) 93–111.
[9] R.D. Fierro, J.R. Bunch, Bounding the subspaces from rank revealing two sided orthogonal decompositions, SIAM J. Matrix Anal. Appl. 16 (3) (1995) 743–759.
[10] R. Fierro, L. Vanhamme, S. Van Huffel, Total Least Squares Algorithms Based on Rank Revealing Complete Orthogonal Decompositions, SIAM Publications, 1997, pp. 99–116.
[11] M. Gu, S. Eisenstat, Downdating the singular value decomposition, SIAM J. Matrix Anal. Appl. 16 (1995) 793–810.
[12] R.J. Hanson, C.L. Lawson, Extensions and applications of the Householder algorithm for solving linear least squares problems, Math. Comp. 23 (1969) 787–812.

[13] D.J. Rabideau, Fast, rank adaptive subspace tracking and applications, IEEE Trans. Signal Process. 44 (1996) 2229–2244.
[14] G.W. Stewart, An updating algorithm for subspace tracking, IEEE Trans. Signal Process. 40 (1992) 1535–1541.
[15] G.W. Stewart, Updating a rank-revealing ULV decomposition, SIAM J. Matrix Anal. Appl. 14 (1993) 494–499.
[16] P.A. Yoon, J.L. Barlow, An efficient rank detection procedure for modifying the ULV decomposition, BIT 38 (1998) 781–801.