Efficient Algorithm for Minimal-Rank Matrix Approximations

Alle-Jan van der Veen

Abstract

For a given matrix H which has d singular values larger than ε, an expression for all rank-d approximants $\hat{H}$ such that $(H-\hat{H})$ has 2-norm less than ε is derived. These approximants have minimal rank, and the set includes the usual 'truncated SVD' low-rank approximation. The main step in the procedure is a generalized Schur algorithm, which requires only $O(1/2\, m^2 n)$ operations (for an $m \times n$ matrix H). The column span of the approximant is computed in this step, and updating and downdating of this space is straightforward. The algorithm is amenable to parallel implementation.

1 Introduction

Let H be a given $m \times n$ matrix, having d singular values larger than 1 and none equal to 1. Denote by $\|\cdot\|$ the matrix 2-norm. In this paper, we describe an algorithm to compute all possible matrices $\hat{H}$ such that

$$\begin{array}{ll} (a) & \| H - \hat{H} \| \leq 1\,, \\ (b) & \operatorname{rank}(\hat{H}) = d\,. \end{array}$$

Such a matrix $\hat{H}$ is a low-rank approximation of H in 2-norm. The problem can be generalized trivially by scaling H, in which case we compute $\hat{H}$ such that $\| H - \hat{H} \| \leq \varepsilon$ and such that the rank of $\hat{H}$ is equal to the number of singular values of H larger than ε.

One way to obtain an $\hat{H}$ which satisfies (a) and (b) is by computing a singular value decomposition (SVD) of H, and setting all singular values that are smaller than 1 equal to zero. This 'truncated SVD' approximant actually minimizes the approximation error: $\|H - \hat{H}\| = \sigma_{d+1} < 1$, and is optimal in Frobenius norm as well. However, the SVD is computationally expensive. We will describe a generalized Schur method which does not require knowledge of the singular values, but produces rank d 2-norm approximants using only $O(1/2\, m^2 n)$ operations. The column span of the approximant is obtained in the first phase of the algorithm, which is a Hyperbolic QR-factorization of the matrix $[I \;\; H]$. The computation of the approximant itself requires an additional matrix inversion.

The Schur method provides a general formula which describes the set of all possible 2-norm approximants of rank d. The Frobenius-norm approximant is also included in this set, and may in principle be obtained as the solution of a non-linear optimization problem over the parameters of the set.

The proposed Schur method for matrix approximation is a specialization of a recently developed extension of Hankel-norm model reduction theory to time-varying systems [3, 8, 7].

---

[0]Stanford University, Dept. Computer Science/SCCM. This work was supported in part by ARPA Contract No. F49620-91-C-0086, monitored by the AFOSR, and by the commission of the EC under the ESPRIT BRA program 6632 (NANA2). A large part was carried out while the author was affiliated with Delft University of Technology, Dept. EE, The Netherlands.

Proc. Fifth SIAM Conf. on Applied Linear Algebra, pp. 274-278, Snowbird, Utah, June 1994.

Other methods to alleviate the computational burden of the SVD while retaining important information such as rank and principal subspaces are the URV decomposition [5] and the rank revealing QR decomposition (RRQR) [2, 1]. The Schur method requires approximately the same number of operations, but has a simpler and more uniform dependence structure. No condition estimation or other global operations are needed, and the number of operations to determine the column space of the approximant is independent of the values of the entries of the matrix.

Notation The superscript $(\cdot)^*$ denotes complex conjugate transposition, $\mathcal{R}(A)$ is the column range (span) of the matrix A, I_m is the $m\times m$ identity matrix, and $0_{m\times n}$ is an $m\times n$ matrix with zero entries.

A matrix Θ is J-unitary if it satisfies

$$\Theta^* J\Theta = J\,, \qquad \Theta J\Theta^* = J\,, \qquad J = \begin{bmatrix} I & \\ & -I \end{bmatrix}. \tag{1}$$

J is a signature matrix; the identity matrices need not have equal sizes. We partition Θ according to J as

$$\Theta = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix}. \tag{2}$$

The J-unitarity of Θ implies $\Theta_{22}^*\Theta_{22} = I + \Theta_{12}^*\Theta_{12}$ and $\Theta_{22}\Theta_{22}^* = I + \Theta_{21}\Theta_{21}^*$. Hence, Θ_{22} is invertible, and

$$\|\,\Theta_{22}^{-1}\,\| \;\le\; 1\,, \qquad \|\,\Theta_{12}\Theta_{22}^{-1}\,\| \;<\; 1\,. \tag{3}$$

2 Basic Approximation Theorem

THEOREM 2.1. *Let H be an $m\times n$ matrix with d singular values larger than 1 and none equal to 1. Then there exists a J-unitary matrix Θ such that*

$$[I \quad H]\Theta \;=\; [A' \quad B']\,, \qquad A' = \begin{array}{c} \\ m \end{array}\!\!\begin{array}{c} \begin{array}{cc} m-d & d \end{array} \\ \begin{bmatrix} A & 0 \end{bmatrix} \end{array}, \qquad B' = \begin{array}{c} \\ m \end{array}\!\!\begin{array}{c} \begin{array}{cc} d & n-d \end{array} \\ \begin{bmatrix} B & 0 \end{bmatrix} \end{array}. \tag{4}$$

Partition Θ into 2×2 blocks as in (2), and define

$$\hat{H} = [B \;\; 0]\Theta_{22}^{-1}\,. \tag{5}$$

$\hat{H}$ is a rank d approximant such that $\|\,H - \hat{H}\,\| \le 1$. The column span of $\hat{H}$ is equal to that of B, which is of full rank d.

Proof. Consider $I - HH^*$. It is non-singular by assumption, and hence there is a J-Cholesky factorization such that

$$I - HH^* = XJ'X^*\,,$$

where X is an $m\times m$ factor, and has full rank m. Put $X = [A \;\; B]$, partitioned according to J', so that $XJ'X^* = AA^* - BB^*$. Since $[I \;\; H]$ has full range, there must be an $n\times m$ matrix, T say, mapping it to X, i.e. $[I \;\; H]T = X$. Since X is also of full rank, it follows that $TJ'T^* = J$. T can be extended to a square invertible J-unitary matrix Θ such that (4) holds [7].

Let $H = U\Sigma V^*$ be an SVD of H. Then $(I - \Sigma^2)$ has the same signature as $AA^* - BB^*$: d negative entries, and $m-d$ positive entries. Hence, A has $m-d$ columns and is of full rank, while B has d columns and is of full rank.

By equation (4), $[B \;\; 0] = I\Theta_{12} + H\Theta_{22}$, so that $H - \hat{H} = -\Theta_{12}\Theta_{22}^{-1}$, which is contractive (equation (3)). Hence $\|\,H - \hat{H}\,\| \le 1$. □

3 Parametrization of All 2-Norm Approximants

Let H be a given matrix, and let Θ, A and B be as defined before in equation (4). Theorem 3.1 gives a chain fraction formula of all possible 2-norm approximants of H, of rank equal to d (there are no approximants of rank less than d). The parametrization is in terms of an $m \times n$ matrix S_L, which has the following 2×2 block partitioning:

$$S_L = \begin{array}{c c} & \begin{array}{cc} d & n-d \end{array} \\ \begin{array}{c} m-d \\ d \end{array} & \begin{bmatrix} (S_L)_{11} & (S_L)_{12} \\ (S_L)_{21} & (S_L)_{22} \end{bmatrix} \end{array} . \tag{6}$$

THEOREM 3.1. *Let $H : m \times n$ be a given matrix, with d singular values larger than 1 and none equal to 1. Define Θ, A', B' as in equation (4). Suppose that a matrix $\hat{H}$ satisfies*

$$\begin{array}{ll} (a) & \| H - \hat{H} \| \le 1 \,, \\ (b) & \operatorname{rank}(\hat{H}) \le d \,. \end{array}$$

Then $\operatorname{rank}(\hat{H}) = d$, *and* $\hat{H} = H - S$ *where*

$$S = (\Theta_{11} S_L - \Theta_{12})(\Theta_{22} - \Theta_{21} S_L)^{-1} \,, \tag{7}$$

for some S_L *with* $\| S_L \| \le 1$ *and* $(S_L)_{12} = 0$. $\hat{H}$ *satisfies*

$$\hat{H} = (B' - A' S_L)(\Theta_{22} - \Theta_{21} S_L)^{-1} \,. \tag{8}$$

Proof. $\| S \| = \| H - \hat{H} \| \le 1$. Define $[G_1^* \quad G_2^*] := [S^* \quad I]\Theta$. Using $\Theta^{-1} = J\Theta^* J$ gives

$$\begin{bmatrix} -S \\ I \end{bmatrix} = \Theta \begin{bmatrix} -G_1 \\ G_2 \end{bmatrix} . \tag{9}$$

Because $\| \Theta_{22}^{-1} \Theta_{21} \| < 1$ and $\| S \| \le 1$, G_2 is invertible. The J-unitarity of Θ and the contractiveness of S implies $G_1^* G_1 \le G_2^* G_2$. Hence $S_L := G_1 G_2^{-1}$ is well defined and contractive, and (9) yields (7). It remains to show that $(S_L)_{12} = 0$. Make the partitionings

$$G_1 = \begin{array}{c} m-d \\ d \end{array} \begin{bmatrix} G_{11} \\ G_{12} \end{bmatrix} , \qquad G_2 = \begin{array}{c} d \\ n-d \end{array} \begin{bmatrix} G_{21} \\ G_{22} \end{bmatrix} , \qquad G_2^{-1} = \begin{array}{c} \begin{array}{cc} d & n-d \end{array} \\ \begin{bmatrix} (G_2^{-1})_1 & (G_2^{-1})_2 \end{bmatrix} \end{array} ,$$

which are conform the partitionings of A' and B'. Then $(S_L)_{12} = 0 \iff G_{11}(G_2^{-1})_2 = 0$. The proof that $G_{11}(G_2^{-1})_2 = 0$ consists of 4 steps.

1. $[H^* \;\; I_n]\Theta = \left[(0_{n \times (m-d)} \quad *) \quad (0_{n \times d} \quad *)\right]$, where '$*$' stands for any matrix.

   Proof:

$$\begin{array}{rrcl} & [I \quad H] & = & [A' \quad B']\Theta^{-1} \\ \Leftrightarrow & \begin{bmatrix} I \\ H^* \end{bmatrix} & = & \Theta^{-*} \begin{bmatrix} A'^* \\ B'^* \end{bmatrix} = J\Theta J \begin{bmatrix} A'^* \\ B'^* \end{bmatrix} \\ \Rightarrow & 0 & = & [H^* \;\; I_n] J \begin{bmatrix} I \\ H^* \end{bmatrix} = [H^* \;\; I_n]\Theta J \begin{bmatrix} A'^* \\ B'^* \end{bmatrix} \\ \Rightarrow & [H^* \;\; I_n]\Theta & = & [(0 \;\; *) \quad (0 \;\; *)] \,. \end{array}$$

   (In the last step, we used the fact that $[A \;\; B]$ is of full rank.)

2. $[G_1^* \;\; G_2^*] \;=\; -\hat{H}^*[I_m \;\; 0]\Theta \;+\; [H^* \;\; I_n]\Theta$.

   Proof: $[G_1^* \;\; G_2^*] \;=\; [S^* \;\; I]\Theta \;=\; [-\hat{H}^* \;\; 0]\Theta \;+\; [H^* \;\; I]\Theta$.

3. $\mathcal{R}(G_{11}^*) \subset \mathcal{R}(\hat{H}^*)$, $\mathcal{R}(G_{21}^*) = \mathcal{R}(\hat{H}^*)$, $\hat{H}$ has rank d.

   Proof: From the preceding two items, it follows directly that

$$\mathcal{R}(G_{11}^*) \subset \mathcal{R}(\hat{H}^*)\,, \qquad \mathcal{R}(G_{21}^*) \subset \mathcal{R}(\hat{H}^*)\,.$$

   G_2 is invertible, hence $\mathcal{R}(G_{21}^*)$ is of full dimension d. Since the rank of $\hat{H}$ is less than or equal to d, it follows that the rank of $\hat{H}$ is precisely equal to d, and hence actually $\mathcal{R}(G_{21}^*) = \mathcal{R}(\hat{H}^*)$.

4. $G_{11}(G_2^{-1})_2 = 0$.

   Proof: From the preceding item, we have $G_{11} = \Delta G_{21}$ (some matrix Δ). Hence

$$\begin{array}{rrcl} & G_2(G_2)^{-1} & = & I \\ \Leftrightarrow & \begin{bmatrix} G_{21} \\ G_{22} \end{bmatrix} [(G_2^{-1})_1 \quad (G_2^{-1})_2] & = & \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \\ \Rightarrow & G_{21}(G_2^{-1})_2 & = & 0 \\ \Rightarrow & G_{11}(G_2^{-1})_2 & = & \Delta G_{21}(G_2^{-1})_2 = 0\,. \end{array}$$

$\square$

4 Computation of Θ

We indicate a Schur algorithm for computing the factorization $[I \quad H]\,\Theta \;=\; [A' \quad B']$ (viz. [8]). The computations consist of elementary (Givens) hyperbolic rotations which recursively create zero entries at selected positions: it may be viewed as a Hyperbolic QR factorization. The basic operations are J-unitary elementary rotations of up to six different types, and we have to keep track of signatures to determine which type to use.

Using the elementary rotations, we compute Θ in two steps: $\Theta = \tilde{\Theta}\Pi$, where $\tilde{\Theta}$ is a J-unitary matrix with respect to an *unsorted* signature matrix, and Π is a permutation matrix which sorts the signature matrix of $\tilde{\Theta}$.

Let $\tilde{\theta}$ be some elementary rotation, such that $[a \quad b]\tilde{\theta} = [* \quad 0]$, and such that $\tilde{\theta}\tilde{j}_2\tilde{\theta}^* = \tilde{j}_1$, $\tilde{\theta}^*\tilde{j}_1\tilde{\theta} = \tilde{j}_2$, for unsorted 2×2 signature matrices $\tilde{j}_1, \tilde{j}_2$. There are six possibilities; we omit the details (see [6]). For a given elementary rotation $\tilde{\theta}$, let $\tilde{\Theta}_{(i,k)}$ be the embedding of this rotation into an $(m+n)\times(m+n)$ J-unitary matrix, so that a plane rotation in the i-th and $m+k$-th planes is obtained. $\tilde{\Theta}$ consists of a series of such embedded rotations,

$$\tilde{\Theta} \;=\; \tilde{\Theta}_{(m,1)}\tilde{\Theta}_{(m-1,1)}\cdots\tilde{\Theta}_{(1,1)}\cdot\tilde{\Theta}_{(m,2)}\cdots\tilde{\Theta}_{(1,2)}\cdot\;\cdots\;\cdot\tilde{\Theta}_{(m,n)}\cdots\tilde{\Theta}_{(1,n)}\,.$$

where $\tilde{\Theta}_{(i,k)}$ is such that it produces a zero at entry $(i, m+k)$, viz.

$$\left[\begin{array}{ccc|ccc} 1 & & & \times & \times & \times \\ & 1 & & \times & \times & \times \\ & & 1 & \times & \times & \times \end{array}\right] \overset{\tilde{\Theta}_{(m,1)}}{\longrightarrow} \left[\begin{array}{ccc|ccc} 1 & & \times & \times & \times & \times \\ & 1 & \times & \times & \times & \times \\ & & \times & 0 & \times & \times \end{array}\right] \overset{\tilde{\Theta}_{(m-1,1)}}{\longrightarrow}$$
$$\left[\begin{array}{ccc|ccc} 1 & \times & \times & \times & \times & \times \\ & \times & \times & 0 & \times & \times \\ & & \times & 0 & \times & \times \end{array}\right] \longrightarrow \;\cdots\; \overset{\tilde{\Theta}_{(1,n)}}{\longrightarrow} \left[\begin{array}{ccc|ccc} \times & \times & \times & 0 & 0 & 0 \\ & \times & \times & 0 & 0 & 0 \\ & & \times & 0 & 0 & 0 \end{array}\right].$$

This scheme ensures that $[I \;\; H]\tilde{\Theta} = [\tilde{X} \;\; 0]$, where $\tilde{X}$ is a resulting upper triangular invertible matrix; it contains the columns of A and B in some permuted order.

To compute this ordering, and to compute the rotations $\tilde{\theta}$, we have to keep track of the signature of each column. Initially, all columns of I have positive signature, and all columns of H have negative signature. At the (i,k)-th step, the signature $\tilde{j}_1$ of the elementary rotation $\tilde{\theta}$ is set equal to the signature of the columns of the entries on which this rotation acts. From $\tilde{j}_1$ and these entries, $\tilde{\theta}$ and $\tilde{j}_2$ are computed; the latter becomes the new signature of these columns, after the rotation. By preservation of inertia, these columns may have the same signature, or have their signatures reversed. Having performed $m \times n$ such steps, we have obtained $[\tilde{X} \;\; 0]$, and the signature of its columns is given by an unsorted signature matrix $\tilde{J}$. Let Π be a permutation matrix such that $J = \Pi^* \tilde{J} \Pi$. Then $\Theta = \tilde{\Theta}\Pi$, $[A' \;\; B'] = [\tilde{X} \;\; 0]\Pi$, and $[I \;\; H]\Theta = [A' \;\; B']$ is the required factorization.

5 Concluding Remarks

The recursive construction of Θ using the Schur method is not always possible, unless extra conditions on the singular values of certain submatrices of H are posed [6]. This is a well-known complication from which all indefinite Schur methods suffer and that can be treated only by global matrix operations (as in [4]).

In many applications, only the column span of the approximant is needed. This space is equal to the column span of B (in (5)) or of $B - A(S_L)_{11}$ (in (8)), and is obtained as a result of the Schur algorithm. Θ needs to be stored only if $\hat{H}$ is to be computed.

The Schur algorithm is amenable to parallel implementations. Because of the uni-directional recursive structure of the computations, updating and downdating the factorization is straightforward. Updating (adding new columns of H) corresponds to augmenting $[\tilde{X} \;\; 0]$ at the right with new columns and continuing the factorization, whereas downdating (removing columns of H) can effectively be carried out by augmenting $[\tilde{X} \;\; 0]$ at the right with these old columns, but now giving them a positive signature [6].

References

[1] C.H. BISCHOF AND G.M. SHROFF, *On updating signal subspaces*, IEEE Trans. Signal Processing, 40 (1992), pp. 96–105.

[2] T.F. CHAN, *Rank revealing QR factorizations*, Lin. Alg. Appl., 88/89 (1987), pp. 67–82.

[3] P.M. DEWILDE AND A.J. VAN DER VEEN, *On the Hankel-norm approximation of upper-triangular operators and matrices*, Integral Equations and Operator Theory, 17 (1993), pp. 1–45.

[4] K. DIEPOLD AND R. PAULI, *Schur parametrization of symmetric indefinite matrices based on a network theoretic model*, Archiv für Elektronik und Übertragungstechnik AEÜ, 44 (1990), pp. 92–96.

[5] G.W. STEWART, *An updating algorithm for subspace tracking*, IEEE Trans. Signal Processing, 40 (1992), pp. 1535–1541.

[6] A.J. VAN DER VEEN, *A Schur method for low-rank matrix approximation*, submitted for publication, (1993).

[7] ———, *Time-Varying System Theory and Computational Modeling: Realization, Approximation, and Factorization*, PhD thesis, Delft University of Technology, Delft, The Netherlands, June 1993.

[8] A.J. VAN DER VEEN AND P.M. DEWILDE, *On low-complexity approximation of matrices*, to appear in Linear Algebra and its Applications, special issue on systems and control, (1994).