Block Bidiagonal Decomposition and Least Squares Problems

Åke Björck
Department of Mathematics
Linköping University

Perspectives in Numerical Analysis, Helsinki, May 27–29, 2008

Outline

The Bidiagonal Decomposition

In 1965 Golub and Kahan gave two algorithms for computing a bidiagonal decomposition (BD)

$$A = U \begin{pmatrix} L \\ 0 \end{pmatrix} V^T \in \mathbf{R}^{m \times n},$$

where B is *lower* bidiagonal,

$$B = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_n \\ & & & \beta_{n+1} \end{pmatrix} \in \mathbf{R}^{(n+1)\times n},$$

and U and V are square orthogonal matrices. Given $u_1 = Q_1 e_1$, the decomposition is uniquely determined if $\alpha_k \beta_{k+1} \neq 0$, $k = 1 : n$,

The Bidiagonal Decomposition

The Householder Algorithm

For given Householder matrix Q_1, set $A_0 = Q_1 A$, and apply Householder transformations alternately from right and left:

$$A_k = Q_{k+1}(A_{k-1} P_k), \quad k = 1, 2, \ldots .$$

P_k zeros $n - k$ elements in kth row;
Q_{k+1} zeros $m - k - 1$ elements in the kth column.
If Q_1 is chosen so that $Q_1 b = \beta_1 e_1$, then

$$U^T \begin{pmatrix} b & AV \end{pmatrix} = \begin{pmatrix} \beta_1 e_1 & B \\ 0 & 0 \end{pmatrix},$$

where B is lower bidiagonal and

$$U = Q_1 Q_2 \cdots Q_{n+1}, \qquad V = P_1 P_2 \cdots P_{n-1}.$$

The Bidiagonal Decomposition

Lanczos algorithm

In the Lanczos process, successive columns of

$$U = (u_1, u_2, \dots, u_{n+1}), \quad V = (v_1, v_2, \dots, v_n),$$

with $\beta_1 u_1 = b$, $\beta_1 = \|b\|_2$, are generated recursively. Equating columns in

$$A^T U = VB^T \quad \text{and} \quad AV = UB,$$

gives the coupled two-term recurrence relations ($v_0 \equiv 0$):

$$\begin{aligned} \alpha_k v_k &= A^T u_k - \beta_k v_{k-1}, \\ \beta_{k+1} u_{k+1} &= Av_k - \alpha_k u_k, \quad k = 1 : n. \end{aligned}$$

Here α_k and β_{k+1} are determined by the condition $\|v_k\|_2 = \|u_k\|_2 = 1$.

The Bidiagonal Decomposition

Since the BD is unique, *both algorithms generate the same decomposition in exact arithmetic.*

From the Lanczos recurrences it follows that

$$U_k = (u_1, \ldots, u_k) \text{ and } V_k = (v_1, \ldots, v_k),$$

form orthonormal bases for the left and right
Krylov subspaces

$$\mathcal{K}_k(AA^T, u_1) = \text{span}\left[b, AA^Tb, \ldots, (AA^T)^{k-1}b\right],$$

$$\mathcal{K}_k(A^TA, v_1) = \text{span}\left[A^Tb, \ldots, (A^TA)^{k-1}A^Tb\right].$$

If either $\alpha_k = 0$ or $\beta_{k+1} = 0$, the process terminates. Then the maximal dimensioned Krylov space has been reached.

Least Squares and LSQR

The LSQR algorithm (Paige and Saunders 1982) is a Krylov subspace method for the LS problem

$$\min_x \|b - Ax\|_2.$$

After k steps, one seeks an approximate solution

$$x_k = V_k y_k \in \mathcal{K}_k(A^T A, A^T b).$$

From the recurrence formulas it follows that $AV_k = U_{k+1} B_k$. Thus

$$b - AV_k y_k = U_{k+1}(\beta_1 e_1 - B_k y_k),$$

where B_k is lower bidiagonal. By the orthogonality of U_{k+1} the optimal y_k is obtained by solving a bidiagonal subproblem

$$\min_y \|\beta_1 e_1 - B_k y\|_2,$$

Least Squares and LSQR

This is solved using Givens rotations to transform B_k into upper bidiagonal form

$$G(\beta_1 e_1 \mid B_k) = \left(\begin{array}{c|c} f_k & R_k \\ \bar{\phi}_{k+1} & 0 \end{array} \right).$$

giving

$$x_k = V R_k^{-1} f_k, \qquad \|b - A x_k\|_2 = |\bar{\phi}_{k+1}|.$$

Then x_k, $k = 1, 2, 3, \ldots$ gives approximations on a nested sequence of Krylov subspaces. Often superior to truncated SVD.

LSQR uses the Lanczos recursion and interleaves the solution of the LS subproblems. A Householder implementation could be used for dense A.

Least Squares and LSQR

Bidiagonalization can also be used for total least squares (TLS) problems. Here the error function to be minimized is

$$\frac{\|b - Ax_k\|_2^2}{\|x_k\|_2^2 + 1} = \frac{\|\beta_1 e_1 - B_k y_k\|_2^2}{\|y_k\|_2^2 + 1}.$$

The solution of the lower bidiagonal TLS subproblems are obtained from the SVD of matrix

$$\begin{pmatrix} \beta_1 e_1 & B_k \end{pmatrix}, \quad k = 1, 2, 3 \ldots.$$

Partial least squares (PLS) is widely used in data analysis and statistics. PLS is mathematically equivalent to LSQR but differently implemented. The predictive variables tend to be many and are often highly correlated.

Outline

Block Bidiagonalization Methods

Let Q_1 be a given $m \times m$ orthogonal matrix and set

$$U_1 = Q_1 \begin{pmatrix} I_p \\ 0 \end{pmatrix} \in \mathbf{R}^{m \times p}.$$

Next form $Q_1^T A$ and compute the LQ factorization of its first p rows

$$(I_p \;\; 0)(Q_1^T A)P_1 = (L_1 \; 0),$$

where L_1 is lower triangular.

Proceed by alternately performing QR factorizations of blocks of p columns and LQ factorizations of blocks of p rows.

Block Bidiagonalization Methods

After k steps we have computed a block bidiagonal matrix

$$T_k = \begin{pmatrix} L_1 & & & \\ R_2 & L_2 & & \\ & \ddots & \ddots & \\ & & R_k & L_k \\ & & & R_{k+1} \end{pmatrix} \in \mathbf{R}^{(k+1)p \times kp},$$

and two block matrices with orthogonal columns

$$(U_1, U_2, \dots, U_{k+1}) = Q_{k+1} \cdots Q_2 Q_1 \begin{pmatrix} I_{(k+1)p} \\ 0 \end{pmatrix},$$

$$(V_1, V_2, \dots, V_k) = P_k \cdots P_2 P_1 \begin{pmatrix} I_{kp} \\ 0 \end{pmatrix}.$$

The matrix T_k is a banded lower triangular matrix with $(p+1)$ nonzero diagonals.

Block Bidiagonalization Methods

The Householder algorithm gives a constructive proof of the existence of such a block bidiagonal decomposition.

If the LQ and QR factorizations have full rank then the decomposition is uniquely determined by U_1.

To derive a block Lanczos algorithm we use the identities

$$\begin{aligned} A(V_1 \; V_2 \cdots \; V_n) &= (U_1 \; U_2 \cdots \; U_{n+1})T_n \\ A^T(U_1 \; U_2 \cdots \; U_{n+1}) &= (V_1 \; V_2 \cdots \; V_n)T_n^T. \end{aligned}$$

Start by forming $Z_1 = A^T U_1$ and compute its thin QR factorization

$$Z_1 = V_1 L_1^T.$$

Block Bidiagonalization Methods

For $k = 1 : n$, we then do: Compute the residuals

$$W_k = AV_k - U_k L_k \in \mathbf{R}^{m \times p},$$

and compute the QR factorization $W_k = U_{k+1} R_{k+1}$.
Compute the residuals

$$Z_{k+1} = A^T U_{k+1} - V_k R_{k+1}^T \in \mathbf{R}^{n \times p},$$

and compute the QR factorization $Z_{k+1} = V_{k+1} L_{k+1}^T$;

Householder QR factorizations can be used to guarantee orthogonality within each block U_k and V_k.

Block Bidiagonalization Methods

The block Lanczos bidiagonalization was given by Golub, Luk, and Overton in 1981.

From the Lanczos recurrence relations it follows by induction that for $k = 1, 2, 3, \ldots$, the block algorithm generates orthogonal bases for the Krylov spaces

$$\begin{aligned} \mathrm{span}(U_1, \ldots, U_k) &= \mathcal{K}_k(AA^T, U_1), \\ \mathrm{span}(V_1, \ldots, V_k) &= \mathcal{K}_k(A^T A, A^T U_1), \end{aligned}$$

The process can be continued as long as the residual matrices have full column rank. Rank deficiency will be considered later.

Outline

Multiple Right-Hand Sides in LS and TLS

In many applications one needs to solve least squares problems with multiple right-hand sides $B = (b_1, \dots, b_p)$.

$$B = (b_1, \dots, b_d), \quad d \geq 2.$$

There are two possible approaches;

• Select a seed right-hand side and use the Krylov subspace generated to start up the solution of the second, etc.

• Use a block Krylov solver where all right-hand sides are treated simultaneously.

Block Krylov methods:

• use a much larger search space from the start
• introduce matrix–matrix multiplies into the algorithm.

Multiple Right-Hand Sides in LS and TLS

A natural generalization of LSQR is to compute a sequence of approximate solutions of the form

$$X_k = V_k Y_k, \quad k = 1, 2, 3, \dots,$$

are determined. That is, X_k is restricted to lie in a Krylov subspace $\mathcal{K}_k(A^T A, A^T B)$. It follows that

$$B - AX_k = B - AV_k Y_k = U_{k+1}(R_1 E_1 - T_k Y_k).$$

Using the orthogonality of the columns of U_{k+1} gives

$$\|B - AX_k\|_F = \|R_1 E_1 - T_k Y_k\|_F.$$

Block LSQR

Hence $\|B - AX_k\|_F$ is minimized for $X_k \in \mathcal{R}(V_k)$ by taking $X_k = V_k Y_k$, where Y_k solves

$$\min_{Y_k} \|R_1 E_1 - T_k Y_k\|_F.$$

The approximations X_k are the optimal solutions on the nested sequence of block Krylov subspaces.

$$\mathcal{K}_k(A^T A, A^T B), \quad k = 1, 2, 3, \ldots.$$

The block bidiagonal LS subproblems are solved by using orthogonal transformations to bring the lower triangular banded matrix T_k into upper triangular banded form.

Multiple Right-Hand Sides in LS and TLS

As in LSQR the solution can be interleaved with the block bidiagonalization. For example, ($p = 2$)

$$
\left(\begin{array}{cc|cccc|cccc}
\times & \times & \times & * & * & & & & & \\
* & \times & \otimes & \times & * & * & & & & \\
* & * & \otimes & \otimes & \times & * & * & & & \\
* & * & & \otimes & \otimes & \times & * & * & & \\
\hline
* & * & & & \otimes & \otimes & \times & * & & \\
* & * & & & & \otimes & \times & \times & & \\
 & & & & & & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times
\end{array}\right)
= \begin{pmatrix} C_1 & R_X & S_1 \\ C_2 & 0 & S_2 \end{pmatrix}.
$$

At this step the approximate solution to $\min \|AX - B\|_F$, is

$$X = V_k(R_X^{-1}C_1), \qquad \|B - AX\|_F = \|C_2\|_f.$$

Multiple Right-Hand Sides in LS and TLS

The block algorithm can be used also for TLS problems with multiple right-hand sides which is

$$\min_{E,\ F} \| \begin{pmatrix} F & E \end{pmatrix} \|_F, \quad (A+E)X = B+F,$$

This cannot, as the LS problem, be reduced to p separate problems.
Using the orthogonal invariance, the error function to be minimized is

$$\frac{\|B-AX\|_F^2}{\|X\|_F^2+1} = \frac{\|R_1E_1 - TY\|_F^2}{\|Y\|_F^2+1}.$$

Multiple Right-Hand Sides in LS and TLS

The TLS solution can be expressed in terms of the SVD

$$\begin{pmatrix} B & A \end{pmatrix} = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1^T \\ V_2^T \end{pmatrix},$$

where $\Sigma_1 = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$.

The solution of the lower triangular block bidiagonal TLS problem can be constructed from the SVD of the block bidiagonal matrix.

Outline

Rank Deficient Blocks and Deflation

When rank deficient blocks occur the bidiagonalization must be modified. For example ($p = 1$), at a particular step

$$Q_3 Q_2 Q_1 (b,\ AP_1 P_2) = \left(\begin{array}{c|cc|ccc} \beta_1 & \alpha_1 & & & & \\ & \beta_2 & \alpha_2 & & & \\ & & \beta_3 & \times & \otimes & \otimes \\ \hline & & & \times & \times & \times \\ & & & \otimes & \times & \times \\ & & & \otimes & \times & \times \\ & & & \otimes & \times & \times \end{array}\right),$$

If $\alpha_3 = 0$ or $\beta_4 = 0$, the reduced matrix has into block diagonal form. Then the LS problem decomposes and the bidiagonalization can be terminated.

Rank Deficient Blocks and Deflation

If $\alpha_k = 0$, then

$$U_k^T (b,\ AV_k) = \begin{pmatrix} \beta_1 e_1 & T_k & 0 \\ 0 & 0 & A_k \end{pmatrix}$$

and the problem is reduced to

$$\min_y \|\beta_1 e_1 - T_k y\|_2.$$

Then the right Krylov vectors

$$A^T b, (A^T A) A^T b, \ldots, (A^T A)^{k-1} A^T b$$

are linearly dependent.

Rank Deficient Blocks and Deflation

If $\beta_{k+1} = 0$, then

$$U_{k+1}^T(b,\ AV_k) = \begin{pmatrix} \beta_1 e_1 & \tilde{T}_k & 0 \\ 0 & 0 & A_k \end{pmatrix}$$

where $\tilde{T}_k = (T_k\ \alpha_k e_k)$ is square. Then the original system is consistent and the solution satisfies

$$\tilde{T}_k y = \beta_1 e_1.$$

Then the left Krylov vectors

$$b, (AA^T)b, \ldots, (AA^T)^k b$$

are linearly dependent.

Rank Deficient Blocks and Deflation

From well known properties of tridiagonal matrices it follows:

• The matrix T_k has full column rank and its singular values are simple.

• The right-hand side βe_1 has nonzero components along each left singular vector of T_k.

Paige and Strakoš call this a core subproblem and shows that it is minimally dimensioned.

If we scale $b := \gamma b$, then only β_1 changes. Thus, we have a core subproblem for any weighted TLS problem.

$$\min \|(\gamma r, E)\|_F \text{ s.t. } (A + E)x = b + r.$$

Rank Deficient Blocks and Deflation

How do we proceed in the block algorithm if a rank deficient block occurs in an LQ or QR factorization?

The triangular factor then typically has the form ($p = 4$ and $r = \mathrm{rank}(R_k) = 3$):

$$R_k = \begin{pmatrix} \times & \times & \times & \times \\ & 0 & \times & \times \\ & & 0 & \times \\ & & & 0 \end{pmatrix} \in \mathbf{R}^{r \times p}.$$

(Recall that the factorizations are performed without pivoting).

All following elements in this diagonal of T_k will be zero.
The bandwidth and block size are reduced by one.

Rank Deficient Blocks and Deflation

Example $p = 2$: Assume that the first element in the diagonal of the block R_2 becomes zero.

$$
\left(\begin{array}{cc|cccc|cccc}
\times & \times & \times & & & & & & & \\
 & \times & \times & \times & & & & & & \\
 & & \times & \times & \times & & & & & \\
 & & & \otimes & \times & \times & & & & \\
 & & & & \otimes & \times & \otimes & & & \\
\hline
 & & & & & \otimes & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times \\
 & & & & & & \times & \times & \times & \times
\end{array}\right).
$$

If a diagonal element (say in L_3) becomes zero the problem decomposes.

In general, the reduction can be terminated when the bandwidth has been reduced p times.

Rank Deficient Blocks and Deflation

Deflation is related to linear dependencies in the associated Krylov subspaces:

If $(AA^T)^k b_j$ is linearly dependent on previous vectors, then all left and right Krylov vectors of the form

$$(AA^T)^p b_j, \qquad (A^T A)^p A^T b_j, \quad p \geq k,$$

should be removed.

If $(A^T A)^k A^T b_j$ is linearly dependent on previous vectors, then all vectors of the form

$$(A^T A)^p A^T b_j, \qquad (AA^T)^{p+1} b_j, \quad p \geq k,$$

should be removed.

When the reduction terminates when both left and right Krylov subspaces have reached their maximal dimensions.

Rank Deficient Blocks and Deflation

The block Lanczos algorithm is modified similarly when a rank deficient block occurs. Recall the block Lanczos recurrence:
For $k = 1 : n$,

$$V_k L_k^T = A^T U_k - V_{k-1} R_k^T,$$
$$U_{k+1} R_{k+1} = A V_k - U_k L_k.$$

Suppose the first rank deficient block is

$$L_k \in \mathbf{R}^{p \times r}, \quad r < p.$$

Then V_k is $n \times r$, i.e. only $r < p$ vectors are determined. In the next step

$$A V_k - U_k L_k \in \mathbf{R}^{m \times r}.$$

Recurrence still works! Block size has been reduced to r.

Outline

• We have emphasized the similarity and (mathematical) equivalence of the Householder and Lanczos algorithms for block bidiagonalization.

For dense problems in data analysis and statistics the Householder algorithm should be used because of its backward stability. Today quite large dense problems can be handled in seconds on a desktop computer!

The relationship of block LSQR to multivariate PLS needs further investigation.

G. H. Golub and W. Kahan.
Calculating the singular values and pseudo-inverse of a matrix.
SIAM J. Numer. Anal. Ser. B, 2:205–224, 1965.

G. H. Golub, F. T. Luk, and M. L. Overton.
A block Lanczos method for computing the singular values and corresponding singular vectors of a matrix.
ACM Trans. Math. Software, 7:149–169, 1981.

S. Karimi and F. Toutounian.
The block least squares method for solving nonsymmetric linear systems with multiple right-hand sides.
Appl. Math. Comput., 177:852–862, 2006.

C. C. Paige and M. A. Saunders.
LSQR. An algorithm for sparse linear equations and sparse least squares.
ACM Trans. Math. Software, 8:43–71, 1982.