GSVD Progress

Ji Wang

1 Week of May 31, 2019

1.1 Quick exit from GSVD main routine

Given that $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{p\times n}$, $\text{rank}(B) = l$ and $\text{rank}([A; B]) = k + l = r$, if $m \leq k$, we can exit from main routine once preprocessing is done. Justification is the following:

In 2.1, after preprocessing, we obtain R_1 and R_2. Let's take a close look at the structure of R_1 and R_2, respectively.

$$R_1 = \begin{array}{c c} & \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{array}{c} k \\ l \\ m-k-l \end{array} & \begin{bmatrix} 0 & R_{12}^{(a)} & R_{13}^{(a)} \\ 0 & 0 & R_{23}^{(a)} \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$R_2 = \begin{array}{c c} & \begin{array}{ccc} n-k-l & k & l \end{array} \\ \begin{array}{c} l \\ p-l \end{array} & \begin{bmatrix} 0 & 0 & R_{13}^{(b)} \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

As is shown, if $m \leq k$, $R_{23}^{(a)}$ will no longer exist. Further, we have:

$$\begin{array}{c c} & n \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} \end{array} = \begin{array}{c c} & \begin{array}{cccc} n-k-l & m & k-m & l \end{array} \\ \begin{array}{c} m \\ l \\ p-l \end{array} & \begin{bmatrix} 0 & R & T_1 & T_2 \\ 0 & 0 & 0 & R_{13}^{(b)} \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

$$\begin{array}{c c} & \begin{array}{ccc} m & k-m & l \end{array} \\ \begin{array}{c} m \\ l \end{array} & \begin{bmatrix} R & T_1 & T_2 \\ 0 & 0 & R_{13}^{(b)} \end{bmatrix} \end{array} = \begin{array}{c c} & \begin{array}{cc} m & l \end{array} \\ \begin{array}{c} m \\ l \end{array} & \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \end{array} \begin{array}{c c} & \begin{array}{ccc} m & k-m & l \end{array} \\ \begin{array}{c} m \\ l \end{array} & \begin{bmatrix} R & T_1 & T_2 \\ 0 & 0 & R_{13}^{(b)} \end{bmatrix} \end{array}$$

2 Week of May 6, 2019

2.1 Derivation of GSVD from ([A;B]) without rank determination

$$
\begin{aligned}
\begin{array}{cc} & n \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} A \\ B \end{bmatrix} \end{array}
&\overset{\text{preprocessing}}{=}
\begin{array}{cc} & \begin{array}{cc} m & p \end{array} \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} U_1 & 0 \\ 0 & V_1 \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} Q_1^T \end{bmatrix} \end{array} \\
&\overset{\text{split QR}}{=}
\begin{array}{cc} & \begin{array}{cc} m & p \end{array} \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} U_1 & 0 \\ 0 & V_1 \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} R_3 \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} Q_1^T \end{bmatrix} \end{array} \\
&\overset{\text{CSD}}{=}
\begin{array}{cc} & \begin{array}{cc} m & p \end{array} \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} U_1 & 0 \\ 0 & V_1 \end{bmatrix} \end{array}
\begin{array}{cc} & \begin{array}{cc} m & p \end{array} \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} U_2 & 0 \\ 0 & V_2 \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} C \\ S \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} Q_2^T \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} R_3 \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} Q_1^T \end{bmatrix} \end{array} \\
&=
\begin{array}{cc} & \begin{array}{cc} m & p \end{array} \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} U_1U_2 & 0 \\ 0 & V_1V_2 \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} C \\ S \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} Q_2^T R_3 \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} Q_1^T \end{bmatrix} \end{array} \\
&\overset{\text{RQ}}{=}
\begin{array}{cc} & \begin{array}{cc} m & p \end{array} \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} C \\ S \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} R \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} Q_3 \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} Q_1^T \end{bmatrix} \end{array} \\
&=
\begin{array}{cc} & \begin{array}{cc} m & p \end{array} \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ \begin{array}{c} m \\ p \end{array} & \begin{bmatrix} C \\ S \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} R \end{bmatrix} \end{array}
\begin{array}{cc} & n \\ n & \begin{bmatrix} Q^T \end{bmatrix} \end{array}
\end{aligned}
$$

2.2 Golub & Van Loan's definition, proof, algorithm of GSVD (adopted by MatLab)

3 Week of April 29, 2019

3.1 Ordering of $\cos$ and $\sin$

3.1.1 LAPACK 3.6.0

The generalized (or quotient) singular value decomposition of an m-by-n matrix A and a p-by-n matrix B is given by the pair of factorizations:

$$A = U\Sigma_1[0, R]Q^T \quad \text{and} \quad B = V\Sigma_2[0, R]Q^T$$

Σ_1 is m-by-r, Σ_2 is p-by-r. (The integer r is the rank of $\begin{pmatrix} A \\ B \end{pmatrix}$, and satisfies $r \leq n$.) Both are real, non-negative and diagonal, and $\Sigma_1^T\Sigma_1 + \Sigma_2^T\Sigma_2 = I$. Write $\Sigma_1^T\Sigma_1 = diag(\alpha_1^2, ..., \alpha_r^2)$ and $\Sigma_2^T\Sigma_2 = diag(\beta_1^2, ..., \beta_r^2)$, where α_i and β_i lie in the interval from 0 to 1. The ratios α_1/β_1,...,α_r/β_r are called the generalized singular values of the pair A, B. If $\beta_i = 0$, then the generalized singular value α_i/β_i is infinite.
There is **no finite ordering** of α and β in GSVD in LAPACK.

3.2 Features of my Julia version

3.2.1 Mathematical Definition

The generalized singular value decomposition (GSVD) of an m-by-n matrix A and p-by-n matrix B is the following, where $m + p \geq n$:

$$A = UD_1RQ^T \quad \text{and} \quad B = VD_2RQ^T$$

where U, V and Q are orthogonal matrices. Let $k+l$ be the effective numerical rank of the matrix $\begin{pmatrix} A \\ B \end{pmatrix}$, then R is a $(k+l)$-by-n matrix of structure $[0\ R_0]$ where R_0 is $(k+l)$-by-$(k+l)$ and is nonsingular upper triangular matrix, $D1$ and $D2$ are m-by-$(k+l)$ and p-by-$(k+l)$ non-negative "diagonal" matrices and satisfy $D_1^T D_1 + D_2^T D_2 = I$. The nonzero elements of D_1 are in **non-increasing** order while the nonzero elements of D_2 are in **non-decreasing** order.
(TO BE REVEALED)Detailed structure of D_1 and D_2:

3.2.2 API design

1. $U, V, Q, alpha, beta, R, k, l = gsvd(A, B, 0)$
2. $U, V, Q, D1, D2, R, k, l = gsvd(A, B, 1)$ or $F = gsvd(A, B)$

Notice that both A and B are overwritten.

3.3 Issues to be resolved

4 Week of April 22, 2019

4.1 Ordering of cos and sin

4.1.1 LAPACK 3.6.0

The generalized (or quotient) singular value decomposition of an m-by-n matrix A and a p-by-n matrix B is given by the pair of factorizations:

$$A = U\Sigma_1[0, R]Q^T \quad \text{and} \quad B = V\Sigma_2[0, R]Q^T$$

Σ_1 is m-by-r, Σ_2 is p-by-r. (The integer r is the rank of $\begin{pmatrix} A \\ B \end{pmatrix}$, and satisfies $r \leq n$.) Both are real, non-negative and diagonal, and $\Sigma_1^T\Sigma_1 + \Sigma_2^T\Sigma_2 = I$. Write $\Sigma_1^T\Sigma_1 = diag(\alpha_1^2, ..., \alpha_r^2)$ and $\Sigma_2^T\Sigma_2 = diag(\beta_1^2, ..., \beta_r^2)$, where α_i and β_i lie in the interval from 0 to 1. The ratios α_1/β_1,...,α_r/β_r are called the generalized singular values of the pair A, B. If $\beta_i = 0$, then the generalized singular value α_i/β_i is infinite.

4.1.2 MATLAB 2019b

$[U, V, X, C, S] = gsvd(A, B)$ returns unitary matrices U and V, a (usually) square matrix X, and non-negative diagonal matrices C and S so that

$$\begin{aligned} A &= U * C * X' \\ B &= V * S * X' \\ C' * C + S' * S &= I \end{aligned}$$

A and B must have the same number of columns, but may have different numbers of rows. If A is m-by-p and B is n-by-p, then U is m-by-m, V is n-by-n, X is p-by-q, C is m-by-q and S is n-by-q, where $q = min(m+n, p)$.

The nonzero elements of S are always on its main diagonal. The nonzero elements of C are on the diagonal $diag(C, max(0, q-m))$. If $m \geq q$, this is the main diagonal of C.

$sigma = gsvd(A, B)$ returns the vector of generalized singular values, $sqrt(diag(C'*C)./diag(S'*S))$. The vector sigma has length q and is in **non-decreasing** order, where $\mathbf{q = min(m{+}n,p)}$. In other words, The nonzero elements of S are in **non-increasing** order while the nonzero elements of C are in **non-decreasing** order.

It's interesting to notice that MATLAB has a compact form of products for gsvd.

4.1.3 Julia 1.10

Compute the generalized SVD of A and B, returning a GeneralizedSVD factorization object F, such that $A = F.U * F.D_1 * F.R_0 * F.Q'$ and $B = F.V * F.D_2 * F.R_0 * F.Q'$.

The entries of $F.D1$ and $F.D2$ are related, as explained in the LAPACK documentation for the generalized SVD and the xGGSVD3 routine which is called underneath (in LAPACK 3.6.0 and newer).

4.2 Features of my Julia version

4.2.1 Definition

The generalized singular value decomposition (GSVD) of an m-by-n matrix A and p-by-n matrix B is the following, where $m + p \geq n$:

$$A = UD_1RQ^T \quad \text{and} \quad B = VD_2RQ^T$$

where U, V and Q are orthogonal matrices. Let $k + l$ be the effective numerical rank of the matrix $\begin{pmatrix} A \\ B \end{pmatrix}$, then R is a $(k + l)$-by-n matrix of structure $[0 \ R_0]$ where R_0 is $(k + l)$-by-$(k + l)$ and is nonsingular upper triangular matrix, $D1$ and $D2$ are m-by-$(k+l)$ and p-by-$(k+l)$ non-negative "diagonal" matrices and satisfy $D_1^T D_1 + D_2^T D_2 = I$. The nonzero elements of D_1 are in **non-increasing** order while the nonzero elements of D_2 are in **non-decreasing** order.

4.2.2 2 APIs

1. LAPACK-style: $U, V, Q, alpha, beta, R, k, l = gsvd(A, B, 0)$
2. Julia-style: $U, V, Q, D1, D2, R, k, l = gsvd(A, B, 1)$

4.3 Testing

4.3.1 Alan testing

Still suffers some subtlety in the dimension of D_1 and D_2.

4.3.2 Spatial matrices testing

Tested example matrices pair from documentation, and publication.

4.3.3 General testing

Tested function correctness, time performance and stability.