Lecture 12: Singular Value Decomposition (SVD)

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

ECE 6435
Adv Numerical Methods in Sci Comp

Fall 2008
November 12, 2008

Singular Value Decomposition (SVD)

- What is Singular Value Decomposition (SVD)?
- Properties of SVD
- Computation of SVD
  - QR Decomposition $A = Q_1 R = Q_1 \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$
  - Bidiagonalization of $R_1 : R_1 = Q_2 B_1 V_B^T$
  - Implicit QR with Wilkinson shift to diagonalize B

$$B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} = U_\Sigma \Sigma V_\Sigma^T$$

$$\boxed{\text{Net Result: } A = U \Sigma V^T ;\ Q_1 Diag(Q_{2,} I_{m-n}) U_\Sigma ;\ V = V_B V_\Sigma}$$

- Sample Applications
  - Approximation of a noisy matrix by a matrix of lower rank.
  - Orthogonal procrustes problem
  - Intersection of null spaces
  - Robust Control

What is SVD?

What is Singular Value Decomposition (SVD)?

- Given an m x n matrix A $\exists$ orthogonal matrices U and V $\ni$

$$A = U\Sigma V^T$$

$$U = (\underline{u}_1 \underline{u}_2 \ldots \underline{u}_m) \in R^{mm} \;,\; V = (\underline{v}_1 \underline{v}_2 \ldots \underline{v}_n) \in R^{nn}$$

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}, \Sigma_r = Diag(\sigma_1 \sigma_2 \ldots \sigma_r 0 0 \ldots 0); \sigma_i \geq 0$$

$$UU^T = U^T U = I_m \,,\, VV^T = V^T V = I_n$$

- $\{\sigma_i\}$ are called singular values of A
- $\{\underline{u}_i\}$ are called left singular values of A

$$\Rightarrow A^T \underline{u}_i = \sigma_i \underline{v}_i \sim \text{analogous to left eigen vectors of } A$$

- $\{\underline{v}_i\}$ are called right singular values of A

$$\Rightarrow A^T \underline{v}_i = \sigma_i \underline{u}_i \sim \text{analogous to right eigen values of } A$$

Properties of SVD - 1

Relationship to AA^T and A^TA

- $AA^T = U\Sigma^2 U^T$

  $\Rightarrow AA^T U = U\Sigma^2$

  $\Rightarrow AA^T \underline{u}_i = \sigma_i^2 \underline{u}_i$

  $\Rightarrow \{u_i\}$ are eigen vectors of AA^T
- $A^T A = V\Sigma^2 V^T$

  $\Rightarrow A^T AV = V\Sigma^2$

  $\Rightarrow A^T A\underline{v}_i = \sigma_i^2 \underline{v}_i$

  $\Rightarrow \{v_i\}$ are eigen vectors of $A^T A$

Properties of SVD

- Rank $A = r$

  in practice $\sigma_{r+1} / \sigma_r \approx \varepsilon$, $\varepsilon \sim 10^{-3} - 10^{-10}$
- $A\underline{v}_{r+1} = A\underline{v}_{r+2} = \ldots = A\underline{v}_n = 0$

$\Rightarrow N(A) = \text{span } \{\underline{v}_{r+1}\ \underline{v}_{r+2} \ldots \underline{v}_n\}$

Properties of SVD - 2

$$R(A^T) = N(A)^{\perp} = \text{span}\{\underline{v}_1 \underline{v}_2 \ldots \underline{v}_r\}$$

- $R(A) = \{\underline{y} \in R^m \mid \sum_{i=1}^{n} \underline{a}_i x_i = \underline{y}\}$

  $= \text{span}\{\underline{u}_1 \underline{u}_2 \ldots \underline{u}_r\}$

  $N(A^T) = R(A)^{\perp} = \text{span}\{\underline{u}_{r+1} \underline{u}_{r+2} \ldots \underline{u}_m\}$

- $A = \sum_{i=1}^{r} \sigma_i \underline{u}_i \underline{v}_i^T = U_r \Sigma_r V_r^T$
  - one of the best methods of approximating a noisy matrix by a matrix of lower rank
  - useful in image compression and spectral estimation
- $\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)} = \sigma_{\max}$
- Frobenius norm: $\|A\|_F = \sqrt{\sum_i \sum_j a_{ij}^2} = \sqrt{tr(A^T A)} = \sqrt{\sum_{i=1}^{r} \sigma_i^2}$
- Condition number of $A = \kappa(A) = \sigma_{\max} / \sigma_{\min}$ for 2-norm

SVD & LS Solution

- Solves the least squares problem:

$$A\underline{x} = \underline{b}$$

$$\underline{x}_{LS} = V\Sigma^{\dagger}U^{T}\underline{b} = \left(\frac{\underline{v}_1}{\sigma_1} \ldots \frac{\underline{v}_r}{\sigma_r} 0..0\right)\begin{bmatrix} \underline{u}_1^T\underline{b} \\ \cdot \\ \cdot \\ \underline{u}_m^T\underline{b} \end{bmatrix} = \sum_{i=1}^{r}\left(\frac{\underline{u}_i^T\underline{b}}{\sigma_i}\right)\underline{v}_i$$

$$\text{Error}: \|\underline{b} - A\underline{x}_{LS}\|^2 = \sum_{i=r+1}^{n}(\underline{u}_i^T\underline{b})^2 \text{ since } \underline{b} = \sum_{i=1}^{m}(\underline{u}_i^T\underline{b})\underline{u}_i \Rightarrow \|\underline{b}\|^2 = \sum_{i=1}^{m}(\underline{u}_i^T\underline{b})^2$$

$$\text{and } A\underline{x}_{LS} = \sum_{i=1}^{r}(\underline{u}_i^T\underline{b})\underline{u}_i \Rightarrow \|A\underline{x}_{LS}\|^2 = \sum_{i=1}^{r}(\underline{u}_i^T\underline{b})^2$$

$$\underline{b}^T A\underline{x}_{LS} = \sum_{i=1}^{r}(\underline{u}_i^T\underline{b})^2$$

- Pseudo inverse: $A^{\dagger} = V\Sigma^{\dagger}U^{T}$ satisfies Moore-Penrose properties

References

❑ References:

1) G. H. Golub and W. Kahan, "Calculating the singular values and pseudo inverse of a matrix," *SIAM J.on NA*, 1965, 2, pp.205-224.

2) P. A. Businger and G. H. Golub, "Algorithm 358:SVD of a complex matrix," *CACM*, vol.12, 564-65, 1969.

3) G. H. Golub and C. Reinsch, "SVD and LS solutions," NM, 14, pp.403-420, 1970.

4) T. F. Chan, "An improved algorithm for computing the SVD," *ACM Trans.on Math software*, 8, pp.72-83, 1982, pp.84-88 same issue.

SVD Computation Steps

- The SVD computation consists of three steps:

1) Upper triangularize A via Householder or Givens transformation

$$A = Q_1 R = Q_1 \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \Rightarrow Q_1^T A = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

2) Bidiagonalize n x n matrix R_1

$$\Rightarrow Q_2^T R_1 V_B = \begin{bmatrix} d_1 & f_2 & 0 & 0 & . & . & . & 0 \\ 0 & d_2 & f_3 & . & 0 & . & . & 0 \\ 0 & 0 & d_3 & f_4 & . & . & . & 0 \\ 0 & . & . & . & . & 0 & d_{n-1} & f_n \\ 0 & . & . & . & . & . & 0 & d_n \end{bmatrix} \quad \text{upper bidiagonal}$$

– Define, $U_B^T = Diag(Q_2^T, I_{m-n}^T)Q_1^T \quad \Rightarrow U_B = Q_1 Diag(Q_2, I_{m-n})$

so,

$$U_B^T A V_B = \begin{array}{c} \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \end{array} \begin{array}{l} n \\ m-n \end{array} = B \text{ is a bidiagonalization of } A.$$

3) Use implicit QR with Wilkinson shift to diagonalize B

$$U_\Sigma^T B V_\Sigma = Diag(\sigma_1 \sigma_2 \ldots \sigma_n)$$

$\sigma_{r+1} = \ldots = \sigma_n = 0$ in theory, but small in practice. So, $U_\Sigma^T U_B^T A V_B V_\Sigma = \Sigma$

$$\Rightarrow U = U_B U_\Sigma = Q_1 Diag(Q_2, I_{m-n}) U_\Sigma, V = V_B V_\Sigma$$

Step 1: QR Decomposition

- **Step 1 is Straightforward : involves min(m-1,n) steps of Householder**

$$\begin{bmatrix}\times&\times&\times\\\times&\times&\times\\\times&\times&\times\\\times&\times&\times\end{bmatrix}\xrightarrow{W_1}\begin{bmatrix}\times&\times&\times\\0&\times&\times\\0&\times&\times\\0&\times&\times\end{bmatrix}\xrightarrow{W_2}\begin{bmatrix}\times&\times&\times\\0&\times&\times\\0&0&\times\\0&0&\times\end{bmatrix}\xrightarrow{W_3}\begin{bmatrix}\times&\times&\times\\0&\times&\times\\0&0&\times\\0&0&0\end{bmatrix}=\begin{bmatrix}R_1\\0\end{bmatrix}$$

$$Q_1^T = W_m, W_{m-1}, \ldots, W_1; \quad m' = \min(m-1, n)$$

- **Bidiagonalization step involves alternate applications of $W_1V_1W_2V_2\ldots$**
  - Consider a full matrix case first:

$$\begin{bmatrix}\times&\times&\times&\times\\\times&\times&\times&\times\\\times&\times&\times&\times\\\times&\times&\times&\times\end{bmatrix}\xrightarrow[W_1]{\text{pivot on } a_{11}}\begin{bmatrix}\times&\times&\times&\times\\0&\times&\times&\times\\0&\times&\times&\times\\0&\times&\times&\times\end{bmatrix}\xrightarrow[V_1]{\text{pivot on } a_{12}}\begin{bmatrix}\times&\times&0&0\\0&\times&\times&\times\\0&\times&\times&\times\\0&\times&\times&\times\end{bmatrix}\xrightarrow[W_2]{\text{pivot on } a_{22}}$$

$$\begin{bmatrix}\times&\times&0&0\\0&\times&\times&\times\\0&0&\times&\times\\0&0&\times&\times\end{bmatrix}\xrightarrow[V_2]{\text{pivot on } a_{23}}\begin{bmatrix}\times&\times&0&0\\0&\times&\times&0\\0&0&\times&\times\\0&0&\times&\times\end{bmatrix}\xrightarrow[W_3]{\text{pivot on } a_{33}}\begin{bmatrix}\times&\times&0&0\\0&\times&\times&0\\0&0&\times&\times\\0&0&0&\times\end{bmatrix}$$

done!

$\Rightarrow$ so need: $W_1, W_2, \ldots, W_{n-1}$ on the left, and $V_1, V_2, \ldots, V_{n-2}$ on the right

Step 2: Bidiagonalization - 1

- **But, our matrix is upper Δ. Are there any simplifications?**
  Yes!!

• With an upper Δ matrix $W_1 = I \Rightarrow$ no need for first step

• Unfortunately V_1 destroys upper Δ nature of R_1

$$\text{Recall } V_1 = \begin{bmatrix} 1 & 0 \\ 0 & \Gamma \end{bmatrix} \Rightarrow R_1 V_1 = \begin{bmatrix} r_{11} & \underline{r}_1^T \\ 0 & \tilde{R} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \Gamma \end{bmatrix} = \begin{bmatrix} r_{11} & \underline{r}_1^T \Gamma \\ 0 & \tilde{R}\Gamma \end{bmatrix}$$

$$\text{where } \underline{r}_1^T = [r_{12} \ldots . r_{1n}]$$

• Apply $V_1 W_1 V_2 W_3 \ldots V_{n-2} W_{n-1}$ to the n x n matrix R_1 only

• So, the process is very good for $m >> n$.

This is typically the case in practice.

Step 2: Bidiagonalization - 2

- **The algorithm for Bidiagonalization of R_1 is based on Householder Transformation**

$\tilde{V}_B = I$

$Q_2 = I$

For $k = 1,\ 2,\ \ldots,\ n-2$ DO

Find Householder matrix $\tilde{V}$ of order $n-k$ ∋

$$(r_{kk+1}, r_{kk+2} \ldots, r_{kn})\tilde{V}_k = (*\ 0\ 0..0)$$

form $V_B = V_B.Diag(I_k\ \tilde{V}_k) = V_B V_K$; where $V_K = Diag(I_k\ \tilde{V}_k)$

Find Householder $\tilde{W}_{k+1}$ of order $n-k$ ∋

$$\tilde{W}_{k+1}\begin{pmatrix} r_{k+1,k+1} \\ . \\ . \\ r_{n,k+1} \end{pmatrix} = \begin{bmatrix} x \\ 0 \\ . \\ 0 \end{bmatrix}$$

$$Q_2 = Q_1 Diag(I_k, \tilde{W}_{k+1});\ \text{where } \tilde{W}_{k+1} = Diag(I_k, \tilde{W}_{k+1})$$

end

- Computational load: $O(4n^3/3)$ operations

Diagonalization of Bidiagonal - 1

- Storage considerations

  1) Store V_k in upper Δ form.

  2) Store W_k in lowerΔ form.

  3) Store d_i and f_i of bidiagonal matrix as two vectors.

- Finally, compute $U_B = Q_1 Diag(Q_2\ I_{m-n})$

$$\text{Result } B = \begin{bmatrix} d_1 & f_2 & & & \\ & d_2 & f_3 & & \\ & & & d_{n-1} & f_n \\ & & & & d_n \end{bmatrix}$$

❑ Step 3 Involves diagonalization of the Bidiagonal

- Basically, there are two approaches

❑ Bad Approach

- Know that $T = B^T B$ is tridiagonal where $B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}$
- Use symmetric QR to diagonalize $B^T B = B_1^T B_1$

$$V_\Sigma^T B^T B V_\Sigma = \Sigma^2$$

- Upper triangularize BV_Σ via Householder:

$$U_\Sigma^T B V_\Sigma = R$$

- Note that $BV_\Sigma \Sigma^{-1}$ is orthogonal

$$\Rightarrow U_\Sigma^T B V_\Sigma = R \Rightarrow \text{ column of } R \text{ are orthogonal}$$

$$\Rightarrow R \text{ is diagonal } = \Sigma$$

Diagonalization of Bidiagonal - 2

So, $U_{\Sigma}^{T} B V_{\Sigma} = \Sigma$

- But, $\kappa(B^T B) = [k(B)]^2$ => No Good!

- **Good Approach:**
  - However, the bad approach provides us a direct method via the implicit Q-theorem.
  - Idea: we apply the implicit QR-step on T such that the first column of V_{Σ} is the same in both of the following decompositions:

$$V_{\Sigma}^{T} B^{T} B V_{\Sigma} = \Sigma^2 \qquad \Rightarrow V_{\Sigma} = \hat{V}_{\Sigma} \quad \text{if first column is the same}$$

$$U_{\Sigma}^{T} B \hat{V}_{\Sigma} = \Sigma = R \qquad \text{(from the implicit Q- theorem)}$$

- **So, the diagonalization of bidiagonal matrix consists of three steps:**
  - Find the Eigenvalue λ of 2x2 submatrix, i.e.,$(n-1, n)$ subblock of $T = B^T B$ that is closer to $d_n^2 + f_n^2 = t_{nn}$

$$\begin{bmatrix} d_1 & 0 & 0 & . & . & 0 & \\ f_2 & d_2 & . & . & 0 & & \\ & f_3 & & & & & \\ 0 & . & . & . & & d_{n-1} & 0 \\ & 0 & . & . & . & f_n & d_n \end{bmatrix} \begin{bmatrix} d_1 & f_2 & 0 & . & . & . & 0 \\ 0 & d_2 & f_3 & . & . & . & 0 \\ & & & & & d_{n-1} & f_n \\ & & & & & 0 & a_n \end{bmatrix} = \begin{bmatrix} d_1^2 & d_1 f_2 & & \\ d_1 f_2 & d_2^2 + f_2^2 & & \\ & & d_{n-1}^2 + f_{n-1}^2 & d_{n-1} f_n \\ & & d_{n-1} f_n & d_n^2 + f_n^2 \end{bmatrix}$$

Diagonalization of Bidiagonal - 3

$$\alpha = (d_{n-1}^2 + f_{n-1}^2 - d_n^2 - f_n^2)/2$$

$$\mu = d_n^2 + f_n^2 - (d_{n-1}^2 f_{n-1}^2)/[\alpha + sign(\alpha)\sqrt{\alpha^2 + d_{n-1}^2 f_{n-1}^2}]$$

- Compute

$$\begin{pmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{pmatrix} \begin{bmatrix} d_1^2 - \mu \\ d_1 f_2 \end{bmatrix} = \begin{pmatrix} * \\ 0 \end{pmatrix}$$

$\Rightarrow$ pivot on $d_1^2 - \mu$ and zero out $d_1 f_2$

$\Rightarrow$ apply J_{21} to B on the right where $J_{21}^T = \begin{pmatrix} c_1 & s_1 & & \\ -s_1 & c_1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix}$

$$B \leftarrow BJ_{21} = \begin{bmatrix} d_1 & f_2 & & & \\ & d_2 & f_3 & & \\ & & & d_{n-1} & f_n \\ & & & & d_n \end{bmatrix} \begin{bmatrix} c_1 & -s_1 & & \\ s_1 & c_1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

$$= \begin{bmatrix} d_1c_1 + f_2s_1 & -d_1s_1 + f_2c_1 & & & \\ \boxed{d_2s_1} & d_2c_1 & f_3 & & \\ & & d_4 & & \\ & & & d_{n-1} & f_n \\ & & & & d_n \end{bmatrix}$$

where $\boxed{} \Rightarrow$ unwanted element

- Chase away unwanted element via $U_1^T = J_{21}^T$ etc...

Diagonalization of Bidiagonal - 4

- To illustrate the idea, consider the following 4x4 example

Example:

$$\begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow[\text{step } V_1]{\text{implicit Q}} \begin{bmatrix} \times & \times & 0 & 0 \\ \boxed{+} & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{U_1^T} \begin{bmatrix} \times & \times & \boxed{+} & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{V_2} \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & \boxed{+} & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}$$

$$\xrightarrow{U_2^T} \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & \boxed{+} \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \xrightarrow{V_3} \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & \boxed{+} & \times \end{bmatrix} \xrightarrow{U_3^T} \begin{bmatrix} \times & \times & 0 & 0 \\ 0 & \times & \times & 0 \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} \quad \textbf{done!}$$

- **In general:**

$$U_{n-1}^T \dots U_1^T B V_1 \dots V_{n-1} = \overline{B}$$

where $\overline{B}$ is the new bidiagonal matrix and $\widetilde{V}_\Sigma = V_1 \dots V_{n-1}$

$V_1 =$ implicit Q-step

since $\widetilde{V}_\Sigma \underline{e}_1 = V_\Sigma \underline{e}_1$ where $V_\Sigma^T B^T B V_\Sigma = \Sigma$

$\Rightarrow \widetilde{V}_\Sigma = V_\Sigma$ from the implicit Q-theorem

What Happens If?

- **Q: What happens when d_k or $f_{k+1} = 0$?**

$$f_{k+1} = 0 \Rightarrow B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}$$

$\Rightarrow$ so work with subproblems

$d_k = 0 \Rightarrow$ apply a series of Given rotations to zero out f_{k+1}

$$\begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 \\ 0 & 0 & 0 & x & 0 \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \xrightarrow{J_{34}^T} \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 \\ 0 & 0 & 0 & 0 & x \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix} \xrightarrow{J_{35}^T} \begin{bmatrix} x & x & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x \\ 0 & 0 & 0 & 0 & x \end{bmatrix}$$

Overall Diagonalization Process

- **The above ideas can be summarized as the following:**
  - Golub-Kahan SVD step

$$t_{nn} = d_n^2 + f_n^2$$

$$\mu = t_{nn} + \alpha + sign(\alpha)\sqrt{\alpha^2 + d_{n-1}^2 f_{n-1}^2} \quad = t_{nn} - (d_{n-1}^2 f_n^2)/[\alpha + sign(\alpha)\sqrt{\alpha^2 + d_{n-1}^2 f_{n-1}^2}]$$

$$y = d_i^2 - \mu, \quad z = d_1 f_2$$

For $k = 1, 2, \ldots, n-1$ DO

- Determine $(c_k\, s_k)$ ∋

$$\begin{pmatrix} y & z \end{pmatrix} \begin{pmatrix} c_k & -s_k \\ s_k & c_k \end{pmatrix} = \begin{pmatrix} * & 0 \end{pmatrix} \quad , B \leftarrow BV_k \quad y = b_{kk}, \; z = b_{k+1,k}$$

- Determine $(c_k\, s_k)$ ∋

$$\begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} * \\ 0 \end{pmatrix} \qquad B \leftarrow U_k^T B$$

if $k < n-1$ then

$$y = b_{k,k+1}, \; z = b_{k,k+2}$$

end if

end DO

Putting it all Together

The overall SVD algorithm

- Compute QR factorization of A $\quad A = Q_1 \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$
- Compute bidiagonalization of R_1

$$\Rightarrow Q_1^T R_1 V_B = B_1 \quad \text{upper bidiagonal}$$

$$U_B^T = Diag(Q_2^T, I_{m-n}) Q_1^T \quad \text{or } U_B = Q_1 Diag(Q_2, I_{m-n}) \qquad U_B^T A V_B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} = B$$

- DO for ever

set $a_{i,i+1}$ to zero if $|a_{i,i+1}| \le \varepsilon\left(|a_{ii}| + |a_{i+1,i+1}|\right)$ for any $i = 1, \ldots n-1$

– Find the largest q and smallest p ∋

$$A = \begin{array}{c} \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & A_{22} & 0 \\ 0 & 0 & A_{33} \\ 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} p \\ n-p-q \\ q \\ m-n \end{array}$$ where A_{33} is diagonal and A_{22} has nonzero super diagonal,

if q = n, quit

else, if any diagonal entry of A_{22} is zero, zero the super diagonal entry in the same row

else Apply Golub-Kahan SVD step to A_{22}

$$A \leftarrow Diag(I_p, U^T, I_{q+m+n}) A Diag(I_p, V, I_q) \quad \text{end if}$$

end if

end DO

Matrix Approximation

SVD Properties and Applications:

- Approximation of a noisy matrix

– Suppose we have a noisy A matrix (e.g., an image) and want to select a filtered A

– The need for such filtering occurs in image restoration, feature selection (also called subset selection problem)

– Consider $A_k = \sum_{i=1}^{k} \sigma_i \underline{u}_i \underline{v}_i^T$

$$\sigma_{k+1} = \|A - A_k\|_2 \leq \|A - B\|_2 \text{ for any } B \ni \; rank(B) = k$$

$\Rightarrow$ among all matrices of rank k, A_k is the closest to A.

– Proof : suppose B has rank k.

then can find $\underline{x}_{k+1} \ldots \underline{x}_n \ni N(B) = \text{span}\{\underline{x}_{k+1} \ldots \underline{x}_n\}$

$\text{span}\{\underline{x}_{k+1} \ldots \underline{x}_n\} \cap \text{span}\{\underline{v}_1 \ldots \underline{v}_{k+1}\} \neq 0$

suppose $\underline{z} \ni \text{span}\{\underline{x}_{k+1} \ldots \underline{x}_n\} \cap \text{span}\{\underline{v}_1 \ldots \underline{v}_{k+1}\}$

and $\|\underline{z}_2\|_2 = 1$

then $B\underline{z} = 0$

Orthogonal Procrustes Problem - 1

$$A\underline{z}=\sum_{i=1}^{k+1}\sigma_i(\underline{v}_i^T\underline{z})\underline{u}_i$$

$$\|A-B\|_2^2\ge\|(A-B)\underline{z}\|_2^2=\|A\underline{z}\|_2^2=\sum_{i=1}^{k+1}\sigma_i^2(\underline{v}_i^T\underline{z})^2\ge\sigma_{k+1}^2$$

$\Rightarrow$ minimum value when $B=\sum_{i=1}^{k}\sigma_i\underline{u}_i\underline{v}_i^T$

- Q: how to pick k?

  we pick k ∍

  $A=\sum_{\mathrm{i=1}}^{\mathrm{k}}\sigma_{\mathrm{i}}\underline{u}_i\underline{v}_i^T$, where k ∍ $\dfrac{\sigma_k}{\sigma_{k+1}}$ is a large quantity

  $\Rightarrow$ a sudden drop in singular values signifies rank deficiency

❑ Orthogonal “Procrustes” problem

- Suppose have an m x p matrix A from experiment 1 and have another m x p matrix B from experiment 2
- Q: Can I get A from B by rotation, i.e., is $A = BQ$?

  $\min \|A-BQ\|_F$ s.t $Q^TQ=I_n$

  $\min\ \mathrm{tr}(A^TA)+\mathrm{tr}(B^TB)-2\mathrm{tr}(Q^TB^TA)$ s.t $Q^TQ=I_n$

Orthogonal Procrustes Problem - 2

$$\Rightarrow \max Q^T B^T A$$
$$s.t \quad Q^T Q = I_n$$

- One way of finding Q involves letting U and V be orthogonal matrices such that

$$U^T B^T A V = \Sigma = diag(\sigma_1 \sigma_2 \ldots \sigma_n)$$
$$\mathrm{tr}(Q^T B^T A) = \mathrm{tr}(Q^T U \Sigma V^T) = \mathrm{tr}(V^T Q^T U \Sigma) = \mathrm{tr}(Z\Sigma)$$
$$= \sum_{i=1}^{p} \sigma_i z_{ii} \le \sum_{i=1}^{p} \sigma_i$$

$\Rightarrow$ We achieve the upper bound when z_{ii}=1,i=1,2,...n

$$\Rightarrow \quad V^T Q^T U = I_n \text{ or } Q = UV^T$$

- Alternatively, form the Lagrangian function

$$\Rightarrow \quad \mathrm{tr}(Q^T U \Sigma V^T) + \mathrm{tr}(\Lambda (I_n - Q^T Q))/2$$
$$U \Sigma V^T - Q\Lambda = 0$$
$$Q^T U \Sigma V^T = \Lambda \qquad \text{or } (I_n - QQ^T) U \Sigma V^T = 0$$

solution: $Q = UV^T$, since $U\Sigma V^T - UV^T V U^T U \Sigma V^T = 0$

$$\Lambda = V \Sigma V^T > 0$$

Hessian = - $\Lambda \Rightarrow$ maximum at $Q = UV^T$

Intersection of Null Spaces - 1

❑ Intersection of Null Spaces

- Recall that $N(A) = \{\underline{x} \mid A\underline{x} = 0\}$
- Hence, finding a basis for $N(A) \cap N(B)$ implies

$$\begin{array}{c} m \\ p \end{array}\begin{pmatrix} A \\ B \end{pmatrix}\underline{x} = 0 \Leftrightarrow \underline{x} \in N(A) \cap N(B)$$

- One way of finding such a basis is:
  - Perform SVD of $\begin{pmatrix} A \\ B \end{pmatrix} = U_{AB}\Sigma V_{AB}^T$
  - $\begin{pmatrix} A \\ B \end{pmatrix}\underline{v}_k = 0$ for $k = r+1,\ldots,n$
- Another way involves simpler steps
  - Let $Z = (\underline{z}_1 \ldots \underline{z}_{n\text{-}r}) = (\underline{v}_{r+1} \ldots \underline{v}_n)$ be an orthonormal basis for $N(A)$
  - Find an orthonormal basis for $N(BZ) = (\underline{w}_1 \ldots \underline{w}_q) = W$

    where $W = n\text{-}r \times q$ matrix
  - Then, columns of (ZW) form an orthonormal basis or $N(A) \cap N(B)$

Intersection of Null Spaces - 2

- Proof: Since $AZ = 0$ and $BZW = 0$

  $R(ZW) \in N(A) \cap N(B)$

  suppose $\underline{x} \in N(A) \cap N(B)$

  $$\Rightarrow \underline{x} = Z\underline{\alpha} = \sum_{i=1}^{n} \alpha_i \underline{z}_i \text{ for some nonzero } \alpha_i$$

  since $\underline{0} = B\underline{x} = BZ\underline{\alpha}$

  $$\Rightarrow \underline{\alpha} = \sum_{i=1}^{q} b_i \underline{w}_i$$

  then

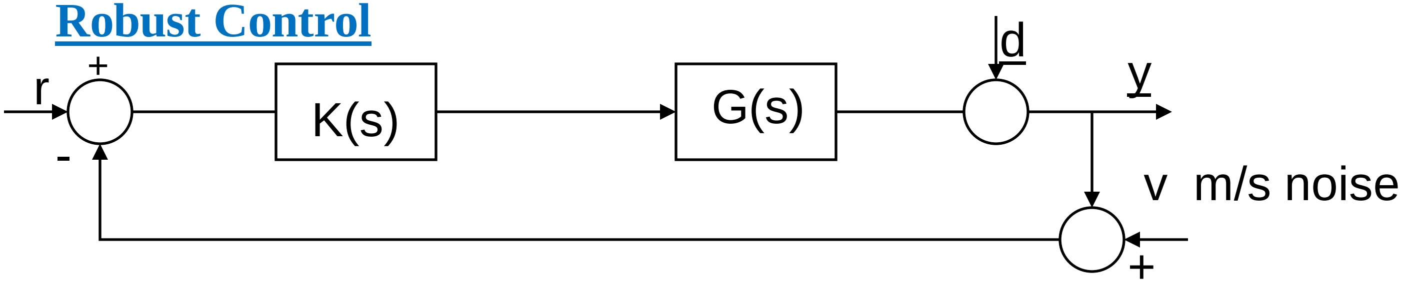
  $$\underline{x} = Z\underline{\alpha} = ZW\underline{b} \in R(ZW)$$

  So, found a basis for $N(A) \cap N(B)$

- To find basis $N(A) \cap N(B)$: find $N(A)$ and $N[B\, span(\underline{v}_{r+1} \ldots \underline{v}_n)]$

Robust Control

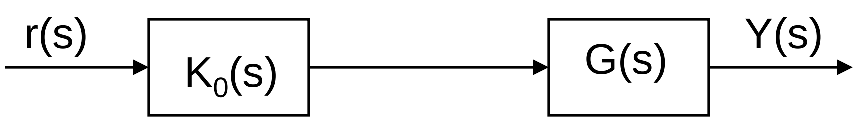
Robust Control



- $U(s) = K(s)[\underline{r} - \underline{v} - \underline{y}]$
- $Y(s) = G(s)K(s)[\underline{r} - \underline{v} - \underline{y}] + \underline{d}$

  $\Rightarrow Y(s) = [I + G(s)K(s)]^{-1} G(s)K(s)(\underline{R}(s) - \underline{V}(s)) + [I + G(s)K(s)]^{-1} \underline{D}(s)$
- Assume for simplicity that $\underline{d} = \underline{v} = \underline{0}$

  $\Rightarrow Y(s) = [I + G(s)K(s)]^{-1} G(s)K(s)R(s) = T_{CL}(s)R(s)$
- Consider an equivalent open loop system:



  - $K_o(s) = (I + KG)^{-1} K$
  - $use\ (I + GK)^{-1} GK = (I + KG)^{-1} KG$
  - $K_o(s)G(s) = (I + GK)^{-1} GK = I - (I + GK)^{-1} = (I + KG)^{-1} KG \Rightarrow K_o(s) = (I + KG)^{-1} K$
  - $Y(s) = T_{OL} R(s)$

- Let us consider the sentivity of open-loop and closed-loop systems to changes in G(s)

Robust Control & SVD

- $G(s) \rightarrow G(s) + \Delta G(s) = G'(s)$
- $\Delta T_{CL}(s) = (I + G'K)^{-1} \Delta T_{OL}(s)$
- $(I + G'K)^{-1}$ is called inverse return difference operator
- For robustness, we want $(I + G'K)^{-1}$ "small"

- Single-input single-output (SISO) systems
  - $|1 + G(j\omega)K(j\omega)|$ large for all $\omega \leq \omega_0$

    where ω_0 is active frequency range
  - Larger return difference (RD)

    => greater robustness to parameter variations
- Multi-input multi-output (MIMO) systems
  - Make $\sigma_{\max}[I + G(j\omega)K(j\omega)]^{-1}$ small (or)

    Make $\sigma_{\min}[I + G(j\omega)K(j\omega)]$ large

Large RD => large loop gains GK => tight loops =>good performance

---

Key Tradeoffs in Robust Control

- Note: Output due to $\underline{v} \Rightarrow (I + GK)^{-1} GK V(s)$

  $\Rightarrow$ errors due to $\underline{v}$ pass through, if GK is large

- Note: $U(s) = K(s)(I + GK)^{-1}(R(s) - V(s) - Y(s))$

  $\approx G^{-1}(\underline{r} - \underline{v} - \underline{y})$

  $\Rightarrow$ so, for some $\omega \ni \sigma_{\max}(G(j\omega)) << 1$, we get amplification of command inputs and measurement noise.

- Key robust control system design tradeoffs

  Control saturation

  Sensor noise amplification

  Disturbance reduction

❑ **For more information read:**

Doyle and Stein, *T-AC* Feb.1981 and papers on robustness and H_∞- designs in *Transactions on Automatic Control*.

Summary

- ❑ **Properties of SVD**
- ❑ **Computation of SVD**
  - QR Decomposition $A = Q_1 R = Q_1 \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$
  - Bidiagonalization of $R_1 : R_1 = Q_2 B_1 V_B^T$
  - Implicit QR with Wilkinson shift to diagonalize B

$$B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} = U_\Sigma \Sigma V_\Sigma^T$$

$$\boxed{\text{Net Result: } A = U\Sigma V^T ;\, Q_1 Diag(Q_{2,} I_{m-n}) U_\Sigma ;\, V = V_B V_\Sigma}$$

- ❑ **Applications**
  - Approximation of a noisy matrix by a matrix of lower rank.
  - Orthogonal procrustes problem
  - Intersection of null spaces
  - Robust Control