Lecture: Background – Norm, Matrix Inverse

Shaoshuai Mou

PURDUE
UNIVERSITY

Review for Eigenvalues

Eigenvalues of a square matrix $M \in \mathbb{R}^{n\times n}$

(Right) Eigenvector: $Mv = \lambda v \qquad v \neq 0$

Left-Eigenvector: $w'M = \lambda w' \quad w \neq 0$

- The **characteristic polynomial** of M is

$$\det\ (\lambda I - M) = \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_1\lambda + c_0$$

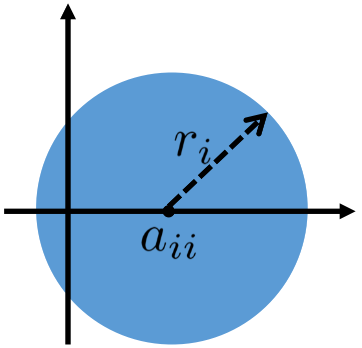
- **Cayley-Hamilton Theorem.** $M^n + c_{n-1}M^{n-1} + \cdots + c_1 M + c_0 I = 0$

- **Gershgorin Circle Theorem:** For **any** square matrix $A \in \mathbb{R}^{n\times n}$, let λ denote any of its eigenvalue.

  Then there exists i such that

$$|\lambda - a_{ii}| \leq r_i \quad \text{where} \quad r_i = \sum_{j=1, j\neq i}^{n} |a_{ij}|$$

  In other words, any eigenvalue must lies in at least one Gershgorin Circle.



  Gershgorin Circle Theorem provides a way to roughly estimate eigenvalues of any square matrices from their entries.

- ***Lemma:*** Inner product of a left eigenvector and a right eigenvector corresponding to different eigenvalues of the same matrix is 0, namely, $\lambda \neq \bar{\lambda} \rightarrow w'v = 0.$

- ***Lemma:*** Nonzero eigenvalues of *AB* are the same as those of *BA*.

- For a symmetric matrix, all its eigenvalues are real $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$

  and with corresponding orthonormal eigenvectors $v_1, v_2, \ldots, v_n$

$$Mv_i = \lambda_i v_i, \quad i = 1, 2, \ldots, n \qquad v_i' v_j = 0, \quad j \neq i \qquad v_i' v_i = 1$$

- **Min-Max Theorem**: If M is symmetric, $\lambda_{\min} x'x \leq x'Mx \leq \lambda_{\max} x'x$

Singular Values

- **Singular values** of a matrix $M \in \mathbb{C}^{m \times n}$ are the square root of eigenvalues of M^*M

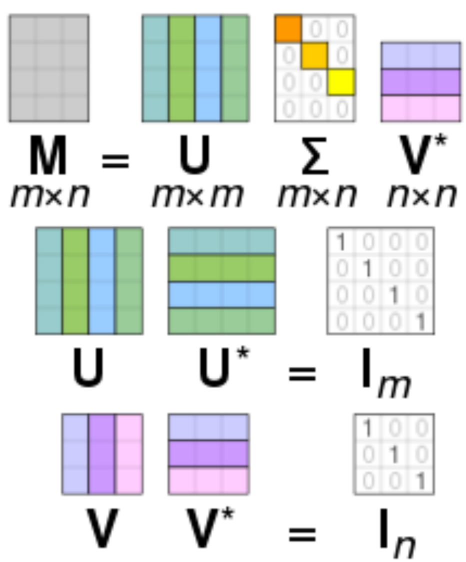
- **Singular Value Decomposition** of a matrix $M \in \mathbb{C}^{m \times n}$ is a factorization of the form

$$M = U\Sigma V^*$$

U,V: unitary matrix $UU^* = I, VV^* = I$

$\sum$:rectangular diagonal matrix with diagonal entries equal to singular values of M

- Figure Explanation



- How to achieve SVD?

Matlab: [U,S, V]=svd (M)

Let's try M=[1,0,0,0,2;0,0,3,0,0;0,0,0,0,0;0,2,0,0,0] in Matlab

- SVD has extensive applications such as total least squares problem in regression, low-rank matrix approximation, signal processing, and so on.

Research Topic: Distributed Algorithm for SVD?

Norm

- A **norm** on a vector space is a non-negative, real, scalar function, which satisfies

$$||A|| \geq 0 \text{ with } ||A|| = 0 \Leftrightarrow A = 0$$ Positivity

$$||cA|| = |c| \cdot ||A||$$ Homogeneity

$$||A + B|| \leq ||A|| + ||B||$$ Triangle Inequality

$$||AB|| = ||A|| \cdot ||B||$$ Sub-multiplicativity

Some commonly used p-matrix norm: $$||A||_p = \sup_{x \neq 0} \frac{||Ax||_p}{||x||}$$

$$||A||_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} a_{ij}$$

Maximum column sum

$$||A||_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} a_{ij}$$

Maximum row sum

$$||A||_2 = \sigma_{\max}$$

$$||A||_2 \leq \sqrt{||A||_1 ||A||_\infty}$$

$$A = \begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 0 & 1/3 & 1/3 & 1/2 \\ 0 & 1/3 & 1/3 & 1/3 \end{bmatrix}$$

Try computing norms in Matlab
norm (X,p)

❖ Spectral radius is not a norm.

$$\rho(A) \leq ||A||$$

$$\rho(AB) \cancel{\leq} \rho(A)\rho(B)$$

$$A = \begin{bmatrix} 0.1 & 10 \\ 0 & 0.1 \end{bmatrix} \qquad B = \begin{bmatrix} 0.8 & 0 \\ 10 & 0 \end{bmatrix} \qquad AB = \begin{bmatrix} 100.08 & 0 \\ 1 & 0 \end{bmatrix}$$

$$\rho(A) = 0.1 \qquad \rho(B) = 0.8 \qquad \rho(AB) = 100.08$$

Question: Given two LTI systems x(k+1)=Ax(k) and x(k+1)=Bx(k), which are exponentially stable. Is the system x(k+1)=ABx(k) also exponentially stable?

Here, by exponentially stable is meant: For any x(0), one has x(t) converges exponentially fast to 0.

Matrix Inverse

For a square matrix $M \in \mathbb{R}^{n \times n}$, if there exists another matrix $\bar{M}$ such that

$$M\bar{M} = \bar{M}M = I_n$$

M is also called *non-singular*.

Then $\bar{M}$ is called the **inverse** of matrix M, and it is usually written as M^{-1}

- **Gauss–Jordan elimination** for computing matrix inverse

$$(AB)^{-1} = B^{-1}A^{-1}$$

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

elementary row operations

- Interchange two rows
- Scale one row by a non-zero constant
- Add one row to another row

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 3/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 3/4 \end{bmatrix} = A^{-1}$$

Achieving the inverse of large-scale matrix is with computationally expensive.

(complexity $O(n^3)$)

So is solving large-scale linear equations $Ax = b$

- **Lower-Upper (LU) Decomposition**.

  Decompose M as a product of a lower-triangular and upper-triangular matrix.

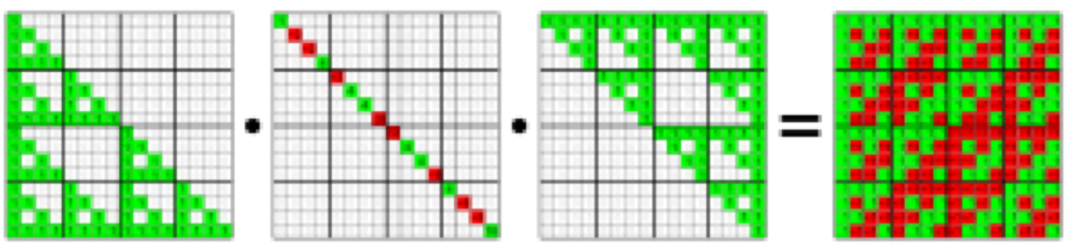
$$A = L \cdot U$$

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} L_{11} & & \\ L_{21} & L_{22} & \\ L_{31} & L_{32} & L_{33} \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ & U_{22} & U_{23} \\ & & U_{33} \end{pmatrix}$$

$$A^{-1} = U^{-1} L^{-1}$$

Try in Matlab [L,U]=lu(A)

- **LDU Decomposition**.



Example of LDU decomposition of a ***Walsh matrix***
(entries either 1 or -1; rows/columns are orthogonal)

Two analytical ways for matrix inverse

- **Schur complement**

Partition a large matrix in to blocks, $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ $\quad A \in \mathbb{R}^{p\times p}, D \in \mathbb{R}^{q\times q}$

✓ If D^{-1} is known or easily to compute,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I_p & BD^{-1} \\ 0 & I_q \end{bmatrix} \begin{bmatrix} A - BD^{-1}C & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} I_p & 0 \\ D^{-1}C & I_q \end{bmatrix}$$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} I_p & 0 \\ -D^{-1}C & I_q \end{bmatrix} \begin{bmatrix} (A - BD^{-1}C)^{-1} & 0 \\ 0 & D^{-1} \end{bmatrix} \begin{bmatrix} I_p & -BD^{-1} \\ 0 & I_q \end{bmatrix}$$

Schur complement

✓ If A^{-1} is known or easily to compute,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I_p & 0 \\ CA^{-1} & I_q \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & D - CA^{-1}B \end{bmatrix} \begin{bmatrix} I_p & A^{-1}B \\ 0 & I_q \end{bmatrix}$$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} I_p & -A^{-1}B \\ 0 & I_q \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & (D - CA^{-1}B)^{-1} \end{bmatrix} \begin{bmatrix} I_p & 0 \\ -CA^{-1} & I_q \end{bmatrix}$$

Motivation Example: Suppose we have already achieved

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} 3/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 3/4 \end{bmatrix}$$

Compute the inverse of

$$M = \left[\begin{array}{ccc:c} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 1 \\ 0 & -1 & 2 & 0 \\ \hdashline 1 & 2 & 3 & -4 \end{array}\right]$$

Only one row/column is added.
Any method to utilize A^{-1} *?*

$$B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \qquad D = -4$$

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} I_p & -A^{-1}B \\ 0 & I_q \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & (D - CA^{-1}B)^{-1} \end{bmatrix} \begin{bmatrix} I_p & 0 \\ -CA^{-1} & I_q \end{bmatrix}$$

scalar

- **Woodbury Matrix Identity**

$M = A + UV$ where A^{-1} is known or easily to compute

$$(A+UV)^{-1} = A^{-1} - A^{-1}U(I + VA^{-1}U)^{-1}VA^{-1}$$

scalar

Motivation Example: Suppose we have already achieved

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} 3/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 3/4 \end{bmatrix}$$

Compute the inverse of

$$M = \begin{bmatrix} 2 & -1 & 1 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} = A + UV \qquad U = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad V = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

➢ Application into a Distributed Algorithm for Least Square Solutions:

X. Wang, J. Zhou, S. Mou, M. J. Corless. IEEE Transactions on Automatic Control, 64(10),2019

$$\begin{bmatrix} x_i(t+1) \\ z_i(t+1) \end{bmatrix} = Q_i^{-1} \begin{bmatrix} x_i(t) + \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} z_j(t) + \frac{1}{d_i} A'_i b_i \\ z_i(t) - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} x_j(t) \end{bmatrix}$$

$A_i \in \mathbb{R}^{m_i \times n}$

$$Q_i^{-1} = \begin{bmatrix} I_n + \frac{1}{d_i} A'_i A_i & I_n \\ -I_n & I_n \end{bmatrix}^{-1} \in \mathbb{R}^{2n \times 2n}$$

$$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{bmatrix} x = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

n is usually large although m_i is small.

Schur Complement

$\in \mathbb{R}^{n \times n}$

$$= \begin{bmatrix} I_n & 0 \\ I_n & I_n \end{bmatrix} \begin{bmatrix} (2I_n + \frac{1}{d_i} A'_i A_i)^{-1} & 0 \\ 0 & I_n \end{bmatrix} \begin{bmatrix} I_n & -I_n \\ 0 & I_n \end{bmatrix}$$

by the Woodbury Matrix Identity

$$= \frac{1}{2} I_n - \frac{1}{4} A'_i (d_i I_{n_i} + A_i A'_i)^{-1} A_i$$

$\in \mathbb{R}^{m_i \times n_i}$

m_i is usually very small (=1)

Pseudo-inverse of a matrix

- The inverse of a full rank matrix $M \in \mathbb{R}^{n\times m}$ is defined as

$M^{-1}M = I_m$ left inverse

may not exist L

$MM^{-1} = I_n$ right inverse

- For $M \in \mathbb{R}^{n\times m}$, its **pseudo**-inverse $M^\dagger$

is the unique $m \times n$ matrix such that
$$\begin{cases} MM^\dagger M = M \\ M^\dagger M M^\dagger = M^\dagger \\ MM^\dagger, \quad M^\dagger M \text{ are both symmetric} \end{cases}$$

Matrix inverse is always a pseudo-inverse!

❖ ***For an undirected connected graph, what is the pseudo-inverse of its Laplacian?***

L is symmetric $L = U\mathrm{diag}\{0, \lambda_2, ..., \lambda_n\}U'$ Columns of U are **orthonormal eigenvectors** of L.

$U'U = I$

$$\boxed{L^\dagger = U\mathrm{diag}\{0, \frac{1}{\lambda_2}, ..., \frac{1}{\lambda_n}\}U'}$$

$$\boxed{L^\dagger \mathbf{1} = 0 \qquad LL^\dagger = L^\dagger L = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n}$$

$$= U(I - e_1 e_1')U'$$

Verify the three conditions

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$= I - (Ue_1)(Ue_1)'$$

$$= I - \frac{\mathbf{1}}{\sqrt{n}}\frac{\mathbf{1}'}{\sqrt{n}}$$

$$Ue_1 = \frac{1}{\sqrt{n}}\mathbf{1}$$