高等数值算法与应用（四）

Advanced Numerical Algorithms & Applications

计算机科学与技术系 喻文健

内容概要

- 矩阵分解及其应用
  - 六大分解简介
  - 矩阵与线性方程组求解基本理论
  - LU分解, Cholesky分解及其应用
  - QR分解与线性最小二乘问题
  - 特征值分解、奇异值分解(SVD)及其应用
  - 稀疏矩阵的直接解法

Spectrum Decomposition and Algorithms

Wenjian Yu

- 本节主要内容
  - 矩阵的特征值与特征向量
  - 有关的计算问题
  - 特征值计算方法
    - 幂法、反幂法 (应用：**PageRank)**
    - 收缩技术
    - **QR**算法

演示程序**eigshow**

n×n矩阵A的特征值分解

- 基本概念与重要性质

$$Ax = \lambda x \;\Leftrightarrow\; \det(\lambda I - A) = \lambda^n - \left(\sum_{j=1}^{n} a_{jj}\right)\lambda^{n-1} + \cdots = 0$$

  - 特征值λ，特征向量**x**, 特征方程， 谱λ**(A)**, 特征子空间
  - 实矩阵：特征值可能是虚数，它一定与其共轭成对出现
  - 实特征值对应实特征向量，虚特征值对应非实特征向量
  - 实对称矩阵**A**：特征值均为 实数，可正交对角化

$$A = Q\Lambda Q^T \;\Leftrightarrow\; AQ = Q\Lambda$$

  - $\lambda(A) = \lambda\left(A^T\right)$, 同一特征值还有左特征向量 $y^T A = \lambda y^T$
  - 若为对角阵、上(下)三角阵，特征值为其对角线元素
  - 若为分块对角阵、分块上(下)三角阵，特征值为对角块矩阵特征值的合并

$$\lambda(A) = \bigcup_{j=1}^{m} \lambda\left(A_{jj}\right) \quad , \quad A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ & A_{22} & \cdots & A_{2m} \\ & & \ddots & \vdots \\ & & & A_{mm} \end{bmatrix}$$

特征值分解

- 基本概念与重要性质 (续)
  - “倍数”cA的特征值，“平移”A+pI的特征值
  - “幂”A^k的特征值, k为正整数
  - “多项式”P(A)的特征值，为特征值的多项式函数值
  - 若A非奇异, A^{-1}的特征值
  - 相似变换不改变特征值
  - $$B = X^{-1}AX,\ \lambda(A) = \lambda(B)$$
  - 设有m个不同的特征值 $\tilde{\lambda}_1, \cdots, \tilde{\lambda}_m$，若$\tilde{\lambda}_j$是特征方程的$n_j$重根，称$n_j$为其代数重数，其几何重数$d_j$为特征子空间维数
  - $\forall j,\ n_j \ge d_j$ ；若两者总相等，则有n个线性无关特征向量，这种矩阵称为非亏损矩阵，否则为亏损矩阵
  - 非亏损矩阵 $\Leftrightarrow$ A可对角化, $X^{-1}AX = \Lambda$

特征值分解

- 计算有关的问题
  - 矩阵**A**是实的，还是复的？规模大小？
  - 既要特征值，又要特征向量？
  - 要所有特征值，还是一部分（最大、最小）？
  - 特征值(向量)计算问题的敏感性
    - 同一个矩阵的不同特征值(向量)的敏感性可能不同
    - 对非亏损矩阵，设 $X^{-1}AX=\Lambda$, 矩阵**A**变化**ΔA, μ**为扰动后矩阵的特征值, λ_k为与它最接近的**A**的特征值，则

$$|\mu-\lambda_k(A)|\le \mathbf{cond}(X)_2\|\Delta A\|_2$$

    - 若特征向量接近线性相关，则特征值计算非常敏感
    - 对于正规阵**A** ($A^TA=AA^T$,特例是实对称阵)，有正交特征向量组，$\mathbf{cond}(X)_2=1$, 不敏感
    - 亏损矩阵一般分析：$|\Delta\lambda|\lesssim\frac{\|y\|_2\|x\|_2}{\|y^Tx\|_2}\|\Delta A\|_2=\frac{1}{\cos\theta}\|\Delta A\|_2$

y为左特征向量

特征值分解及其计算

特征值计算的方法

- 求解特征方程?
  - 计算特征多项式系数工作量很大，且对矩阵扰动敏感
  - 高阶多项式方程求根的工作量很大，两种误差累计使解不准
  - 例：$A = \begin{bmatrix} 1 & \epsilon \\ \epsilon & 1 \end{bmatrix}$, ϵ 比 $\sqrt{\epsilon_{\text{mach}}}$ 小一点，准确特征值为 **1**$\pm\epsilon$ **,** 但 $\det(A-\lambda I)=\lambda^2-2\lambda+1$ **,** 算出两个根均为**1**
  - 因此，反而是求矩阵特征值的方法被用于求解多项式方程

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

是**C_n**的特征多项式。因此，通过求**C_n**的特征值即求解了多项式方程。

$$C_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & -c_0 \\ 1 & 0 & \cdots & 0 & -c_1 \\ 0 & 1 & \cdots & 0 & -c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -c_{n-1} \end{bmatrix}$$

- 幂法 **(power iteration)** $x_k = Ax_{k-1}$
  - 求最大的特征值**(**主特征值**)**、及对应的特征向量
  - 条件：主特征值是唯一的，对实矩阵它一定是实的

特征值、特征向量的计算

- 幂法与反幂法
  - 幂法的应用：**Google**的**PageRank**算法
  - 加速幂法的收敛**(**瑞利商、位移技术**)**
  - 反幂法
    - 求最小的特征值，对**A**$^{-1}$应用幂法
    - 与位移技术结合，即对 $A-\lambda I$ 使用, 精确求某一特征值
- 计算所有特征值的主要方法
  - **1.**若已知某个特征值、及特征向量，可采用收缩技术
  - **2. QR**算法
  - 3. 分治算法
  - 4. Krylov子空间迭代法: Lanczos算法，Arnoldi算法

计算所有特征值的方法

- 收缩技术

假设已经求出了一个特征值λ_1及相应的特征向量$\boldsymbol{x}_1$，则

$$\boldsymbol{A}\boldsymbol{x}_1 = \lambda_1 \boldsymbol{x}_1 .$$

首先构造一个 Householder 变换矩阵$\boldsymbol{H}$使

$$\boldsymbol{H}\boldsymbol{x}_1 = \sigma \boldsymbol{e}_1 ,$$

$$\Rightarrow \boldsymbol{H}\boldsymbol{A}\boldsymbol{H}^T\boldsymbol{e}_1 = \boldsymbol{H}\boldsymbol{A}\left(\frac{1}{\sigma}\boldsymbol{x}_1\right) = \frac{1}{\sigma}\boldsymbol{H}\boldsymbol{A}\boldsymbol{x}_1 = \frac{1}{\sigma}\boldsymbol{H}\lambda_1\boldsymbol{x}_1 = \frac{\lambda_1}{\sigma}(\sigma\boldsymbol{e}_1) = \lambda_1\boldsymbol{e}_1 .$$

由于$\boldsymbol{e}_1 = [1,0,\cdots,0]^T$，上式中$\boldsymbol{H}\boldsymbol{A}\boldsymbol{H}^T\boldsymbol{e}_1$即为矩阵$\boldsymbol{H}\boldsymbol{A}\boldsymbol{H}^T$的第一列。因此有

$$\boldsymbol{H}\boldsymbol{A}\boldsymbol{H}^T = \begin{bmatrix} \lambda_1 & \boldsymbol{r}_1^T \\ \boldsymbol{O} & \boldsymbol{A}_1 \end{bmatrix},$$

其中$\boldsymbol{A}_1 \in \mathbb{R}^{(n-1)\times(n-1)}$，$\boldsymbol{r}_1 \in \mathbb{R}^{(n-1)}$。因为正交相似变换不改变矩阵特征值，所以求矩阵$\boldsymbol{A}$的其余特征值变为求$n-1$阶矩阵$\boldsymbol{A}_1$的特征值。若$\lambda_2$是$\boldsymbol{A}_1$的一个特征值（假设$\lambda_2 \neq \lambda_1$），且对应的特征向量为$\boldsymbol{y}_2$，则可证明 **?**

$$\boldsymbol{x}_2 = \boldsymbol{H}\begin{bmatrix} \alpha \\ \boldsymbol{y}_2 \end{bmatrix}, \qquad \text{其中}\alpha = \frac{\boldsymbol{r}_1^T\boldsymbol{y}_2}{\lambda_2 - \lambda_1}$$

是$\boldsymbol{A}$的与λ_2对应的特征向量。

计算所有特征值的方法

- **例(收缩技术)：** 已知矩阵

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

的一个特征值和特征向量分别是$\lambda_1 = 2$，$\boldsymbol{x}_1 = [1,1,0]^T$，求它的其他特征值。

使用 Householder 变换对$\boldsymbol{x}_1$进行消元，相应的$\sigma = -\sqrt{2} = -1.4142$，所以构造Householder 矩阵$\boldsymbol{H}$所需的向量$\boldsymbol{v}$为：

$$\boldsymbol{v} = \boldsymbol{x}_1 - \sigma \boldsymbol{e}_1 = \begin{bmatrix} 2.4142 \\ 1 \\ 0 \end{bmatrix} \Rightarrow \boldsymbol{w} = \boldsymbol{v}/\|\boldsymbol{v}\|_2 = \begin{bmatrix} 0.9239 \\ 0.3827 \\ 0 \end{bmatrix}$$

$$\Rightarrow \boldsymbol{H} = \begin{bmatrix} -0.7072 & -0.7072 & 0 \\ -0.7072 & 0.7072 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \boldsymbol{HAH}^T = \begin{bmatrix} 2 & 3 & -1.4142 \\ 0 & 1 & 0 \\ 0 & -1.4142 & 2 \end{bmatrix}.$$

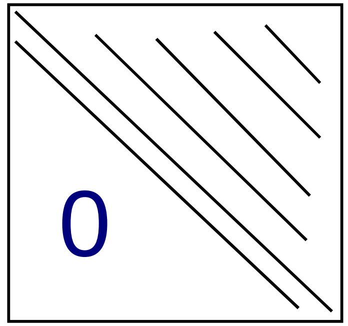
矩阵$\boldsymbol{A}$的其他特征值通过矩阵$\boldsymbol{A}_1 = \begin{bmatrix} 1 & 0 \\ -1.4142 & 2 \end{bmatrix}$求得，即$\lambda_2 = 1$，$\lambda_3 = 2$.

计算所有特征值的方法

- **QR算法**
  - “二十世纪十大算法”之一，计算中小规模矩阵所有特征值的稳定、有效的方法
  - 实**Schur**分解: $A = QSQ^T \quad \Rightarrow \quad Q^T A Q = S$
  - S 为分块上三角阵，对角块为**1**阶或**2**阶方阵
  - **QR**算法就是不断做正交相似变换，得到**Schur**分解
    - $A_0 = A,\ Q_k R_k = A_k,\ A_{k+1} = R_k Q_k,\ k=0, 1, 2, \cdots$
    - $A_{k+1} = Q_k^T A_k Q_k$, 所以A_k的特征值与**A**的相同
    - 可证明，序列$\{A_k\}$“基本收敛”到拟上三角阵**S**, 对角元按绝对值从大到小排列
    - 条件：实矩阵**A**为非亏损阵，等模特征值只有实重特征值，或多重复的共轭特征值两种情况
  - 实际应用，先化简为上**Hessenberg**阵, 再使用位移技术

基本QR算法

计算所有特征值的方法



- 实用的QR算法
  - 对上Hessenberg矩阵A_k执行QR迭代
    - 对A_k进行QR分解: $P_{n-1}\cdots P_2P_1A_k = R_k \Rightarrow Q_k = (P_{n-1}\cdots P_2P_1)^T$
    - 算A_{k+1}: $A_{k+1} = R_kQ_k = P_{n-1}\cdots P_1A_k(P_{n-1}\cdots P_1)^T = P_{n-1}\cdots P_1A_kP_1^T\cdots P_{n-1}^T$
    - 采用Givens旋转，以4阶矩阵为例

$$\underset{A_k}{\begin{bmatrix}\times&\times&\times&\times\\\times&\times&\times&\times\\&\times&\times&\times\\&&\times&\times\end{bmatrix}} \rightarrow \underset{P_1A_k}{\begin{bmatrix}\times&\times&\times&\times\\&\times&\times&\times\\&\times&\times&\times\\&&\times&\times\end{bmatrix}} \rightarrow \underset{P_2P_1A_k}{\begin{bmatrix}\times&\times&\times&\times\\&\times&\times&\times\\&&\times&\times\\&&\times&\times\end{bmatrix}} \rightarrow \underset{P_3P_2P_1A_k}{\begin{bmatrix}\times&\times&\times&\times\\&\times&\times&\times\\&&\times&\times\\&&&\times\end{bmatrix}} \rightarrow$$

$$\rightarrow \underset{P_3P_2P_1A_kP_1^T}{\begin{bmatrix}\times&\times&\times&\times\\\times&\times&\times&\times\\&&\times&\times\\&&&\times\end{bmatrix}} \rightarrow \underset{P_3P_2P_1A_kP_1^TP_2^T}{\begin{bmatrix}\times&\times&\times&\times\\\times&\times&\times&\times\\&\times&\times&\times\\&&&\times\end{bmatrix}} \rightarrow \underset{A_{k+1}}{\begin{bmatrix}\times&\times&\times&\times\\\times&\times&\times&\times\\&\times&\times&\times\\&&\times&\times\end{bmatrix}}$$

  - A_{k+1}仍是上Hessenberg阵. 随k$\rightarrow$ 0, 下三角元素$\rightarrow$ 0

计算所有特征值的方法

- 实用的**QR**算法
  - 一般矩阵化为上**Hessenberg**阵, 使用**Householder**变换

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & \boldsymbol{r}_1^T \\ \boldsymbol{c}_1 & \boldsymbol{A}_{22}^{(1)} \end{bmatrix} \xrightarrow{\boldsymbol{H}_1=\begin{bmatrix} 1 & \boldsymbol{O}^T \\ \boldsymbol{O} & \boldsymbol{H}_1' \end{bmatrix}} \boldsymbol{H}_1\boldsymbol{A}=\begin{bmatrix} a_{11} & \boldsymbol{r}_1^T \\ \sigma_1\boldsymbol{e}_1 & \boldsymbol{H}_1'\boldsymbol{A}_{22}^{(1)} \end{bmatrix} \longrightarrow \boldsymbol{H}_1\boldsymbol{A}\boldsymbol{H}_1=\begin{bmatrix} a_{11} & \boldsymbol{r}_1^T\boldsymbol{H}_1' \\ \sigma_1\boldsymbol{e}_1 & \boldsymbol{H}_1'\boldsymbol{A}_{22}^{(1)}\boldsymbol{H}_1' \end{bmatrix}$$

$$\boldsymbol{A}^{(2)}=\boldsymbol{H}_1\boldsymbol{A}\boldsymbol{H}_1=\begin{bmatrix} \boldsymbol{A}_{11}^{(2)} & \boldsymbol{A}_{12}^{(2)} \\ \boldsymbol{A}_{21}^{(2)} & \boldsymbol{A}_{22}^{(2)} \end{bmatrix} \xrightarrow{\boldsymbol{H}_2=\begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{O}^T \\ \boldsymbol{O} & \boldsymbol{H}_2' \end{bmatrix}} \boldsymbol{H}_2\boldsymbol{A}^{(2)}\boldsymbol{H}_2=\left[\begin{array}{cc:cccc} a_{11} & a_{12}^{(2)} & a_{13}^{(3)} & \cdots & \cdots & a_{1n}^{(3)} \\ \sigma_1 & a_{22}^{(2)} & a_{23}^{(3)} & \cdots & \cdots & a_{2n}^{(3)} \\ \hdashline 0 & \sigma_2 & a_{33}^{(3)} & \cdots & \cdots & a_{3n}^{(3)} \\ 0 & 0 & \vdots & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & \cdots & a_{nn}^{(3)} \end{array}\right]$$

…… ……

  - 最终$\boldsymbol{H}_{n-1}\cdots\boldsymbol{H}_1\boldsymbol{A}\boldsymbol{H}_1\cdots\boldsymbol{H}_{n-1}$为上**Hessenberg**阵
  - 好处：减小后续**QR**迭代每步计算量
    加速收敛过程

演示程序**eigsvdgui**

矩阵元素颜色什么意思？

计算所有特征值的方法

- 实用的**QR**算法
  - 带位移的**QR**迭代

$$\begin{cases} Q_k R_k = A_k - s_k I\,, & \text{(作 QR 分解)} \\ A_{k+1} = R_k Q_k + s_k I, & k = 1, 2, \cdots \end{cases} \longrightarrow \begin{array}{l} A_{k+1} = R_k Q_k + s_k I = \\ Q_k^T (A_k - s_k I) Q_k + s_k I = Q_k^T A_k Q_k \end{array}$$

  - 迭代矩阵序列仍正交相似，加快收敛速度，使迭代过程对更多的矩阵能收敛
  - 简单策略取s_k为$A_k(n, n)$, 还有更复杂的策略和双位移技术
  - 实对称矩阵, **Wilkinson**证明了单位移的**QR**迭代过程收敛; 实非对称矩阵, 还没有这样的证明, 理论上可能不收敛

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \\ & & & & \times \end{bmatrix}$$

非对称矩阵

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & & & & \\ & \times & & & \\ & & \times & & \\ & & & \times & \\ & & & & \times \end{bmatrix}$$

对称矩阵

计算所有特征值的方法

- 实用的**QR**算法的算法描述

输入：A, n；输出：$\lambda_1, \cdots, \lambda_n$.

利用 Householder 变换将矩阵A化简为上 Hessenberg 阵;

While A不是拟上三角矩阵 do

 $s := a_{nn}$;

 For j=1, 2, ..., n   {计算$A - sI$ }

  $a_{jj} := a_{jj} - s$;

 End

 使用 Givens 旋转将A化为上三角阵，得到旋转阵G_j $(j = 1, \cdots, n-1)$的参数c_j, s_j ;

 For j=1, 2, ..., n-1   {计算$RQ = RG_1^T \cdots G_{n-1}^T$}

  $A := AG_j^T$;   {执行列的 Givens 旋转}

 End

 For j=1, 2, ..., n   {计算$A + sI$ }

  $a_{jj} := a_{jj} + s$;

 End

End

Matlab中有关命令及其他

- **eig命令** - 主要对稠密矩阵
  - d=eig(A); 返回所有特征值，使用**qr**算法
  - [V, D]=eig(A); 返回特征值和完备特征向量组**, qr**算法
  - 若**A**稀疏且对称，也可返回所有特征值，对其他稀疏矩阵**,** 或要算特征向量，则需使用**eigs**命令
  - 可能还会对矩阵元素进行比例调整，保证数值稳定性
- **eigs命令**
  - 针对稀疏矩阵**, Lanczos**算法**, Arnoldi**算法
  - d = eigs(A, k); 返回最大的**k**个特征值**, k**缺省值为**6**
  - [V, D]=eigs(A, k); 返回最大的**k**个特征值与相应特征向量
- **hess, schur命令**

期末Project安排

- 要求
  - 根据所学内容编写一个**Matlab**演示软件
  - 形象地表现概念、算法、或应用，利于使用者学习巩固知识，中文界面且交互性、效率较好
  - 选题可根据**Heath**网站改编，或根据教学内容自编 **http://www.cse.illinois.edu/iem/**
  - 由单独的**.m**程序文件实现**,** 设计使用者的思考编程题
  - 特别欢迎有独特的创意 **(**若改编**NCM**需明确说明修改计划**)**

例: **floatshow.m, qformula.m, revste.m, matrix_cond.m**

- 时间安排
  - 选题：第**12**周之前**(12**月**3**日**deadline), **在**“**课程讨论**”**提交选题内容，获得同意，各人之间不要重复
  - 完成**/**演示阶段：第**16**周，做课堂演示，交实验报告
  - 占成绩比重：**40%**

Singular Value Decomposition (SVD)

Wenjian Yu

Singular Value Decomposition (SVD)

- 矩阵的奇异值分解
  - A为m×n矩阵，则 $A = U\Sigma V^T$
  - U, V均为正交阵, $\Sigma=(\sigma_{ij})$为m×n的对角阵, 且对角元素≥0，称为奇异值
  - 记Σ的第i个对角元为σ_i, 通常调整U, V列的顺序使得奇异值按降序排列：$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} = U\Sigma V^T = \begin{bmatrix} .141 & .825 & -.420 & -.351 \\ .344 & .426 & .298 & .782 \\ .547 & .0278 & .664 & -.509 \\ .750 & -.371 & -.542 & .0790 \end{bmatrix}$$

只讨论m≥n的情况

Why?

$$\begin{bmatrix} 25.5 & 0 & 0 \\ 0 & 1.29 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} .504 & .574 & .644 \\ -.761 & -.057 & .646 \\ .408 & -.816 & .408 \end{bmatrix}$$

Singular Value Decomposition (SVD)

- 奇异值分解有关性质
  - A^T的分解：$A = U\Sigma V^T \Rightarrow A^T = V\Sigma^T U^T$
  - U, V列向量为u_i, v_i, 称为左、右奇异向量
  - $Av_i = \sigma_i u_i$，即$Av_1 = \sigma_1 u_1, Av_2 = \sigma_2 u_2, \cdots$ 演示程序**eigshow**
  - 假设m≥n, 有n个奇异值

$$A = U\Sigma V^T \Rightarrow \underbrace{A^T A}_{n\times n} = V\Sigma^T\Sigma V^T = V\begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix} V^T$$

  - 所以奇异值是$A^T A$特征值的算术平方根，是唯一的
  - 存在性的证明：设$A^T A$非零特征值为$\sigma_1^2, \sigma_2^2, \cdots, \sigma_r^2$形成$\Sigma_r^2$

$$A^T A = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} \Sigma_r^2 & O \\ O & O \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \Longrightarrow \begin{matrix} AV_2 = O \\ \Sigma_r^{-1} V_1^T A^T A V_1 \Sigma_r^{-1} = I_r \end{matrix}$$

令$U_1 = AV_1\Sigma_r^{-1}$, 则$U_1^T U_1 = I_r$, 则可将U_1看作正交阵的r列

Singular Value Decomposition (SVD)

- 奇异值分解
  - 存在性的证明 $A = U\Sigma V^T$

$$A^T A = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} \Sigma_r^2 & O \\ O & O \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \Longrightarrow \begin{array}{l} A V_2 = O \\ U_1 = AV_1\Sigma_r^{-1} \text{为正交阵r列} \end{array}$$

根据 U_1 补齐正交单位向量得到正交阵 $U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}$

$$U^T AV = \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} A \begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} U_1^T AV_1 & U_1^T AV_2 \\ U_2^T AV_1 & U_2^T AV_2 \end{bmatrix} = \begin{bmatrix} U_1^T AV_1 & O \\ U_2^T AV_1 & O \end{bmatrix}$$

$$U_1 = AV_1\Sigma_r^{-1} \Longrightarrow \begin{array}{l} U_1^T AV_1 = \Sigma_r^{-1} V_1^T A^T AV_1 = \Sigma_r \\ U_2^T AV_1 = U_2^T U_1 \Sigma_r = O \end{array}$$

所以，$U^T AV = \begin{bmatrix} \Sigma_r & O \\ O & O \end{bmatrix} = \Sigma$ **SVD的存在性得证!**

Singular Value Decomposition (SVD)

- 奇异向量的意义
  - $A = U\Sigma V^T \Rightarrow AV = U\Sigma$
  - 对应于非零奇异值的矩阵**U**的列, $u_1, \dots, u_r$是**span(A)**的单位正交基, **U**的剩下的列是**span(A)**补空间的单位正交基
  - **0**奇异值对应的矩阵**V**的列, $v_{r+1}, \dots, v_n$是**A**的核空间 $\{x \in \mathbb{R}^n : Ax = O\}$ 的单位正交基
- 与特征值分解的比较**(m=n)** $A = X\Lambda X^{-1} \Rightarrow AX = X\Lambda$
  - 特征值分解仅对非亏损阵存在，非零特征值对应的矩阵**X**的列向量为**span(A)**的基, 但一般不正交
  - 当**A**为对称半正定时，其特征值分解存在，特征值**≥0** $A = Q\Lambda Q^T$也是**SVD**, 若正定, $\mathrm{cond}(A)_2 = \lambda_{\max}/\lambda_{\min} = \sigma_{\max}/\sigma_{\min}$
  - 其他实对称矩阵, $A = Q\Lambda Q^T$适当修改即得**SVD**分解

Singular Value Decomposition (SVD)

- 奇异值与矩阵的2-范数、条件数
  - $A=U\Sigma V^T$，其中Σ=diag($\sigma_1,\sigma_2,\cdots,\sigma_n$)

最大奇异值

$$\|A\|_2=\max_{x\neq O}\frac{\|Ax\|_2}{\|x\|_2}=\max_{x\neq O}\frac{\|U\Sigma V^Tx\|_2}{\|x\|_2}=\max_{x\neq O}\frac{\|\Sigma V^Tx\|_2}{\|V^Tx\|_2}=\max_{y\neq O}\frac{\|\Sigma y\|_2}{\|y\|_2}=\sigma_1$$

若 A 非奇异(m=n)，其奇异值均>0

$$\|A^{-1}\|_2=\left(\min_{x\neq O}\frac{\|Ax\|_2}{\|x\|_2}\right)^{-1}=\left(\min_{y\neq O}\frac{\|\Sigma y\|_2}{\|y\|_2}\right)^{-1}=1/\sigma_n$$

$$\Longrightarrow \quad \mathrm{cond}(A)_2=\sigma_1/\sigma_n$$

若m>n，仍有 $\|A\|_2=\sigma_1$. 若列满秩(奇异值>0)，

$$\begin{bmatrix}\sigma_1^{-1}&&\\&\ddots&\\&&\sigma_n^{-1}\end{bmatrix}$$

$$A^+=(A^TA)^{-1}A^T=(V\Sigma^T\Sigma V^T)^{-1}V\Sigma^TU^T=V\underline{(\Sigma^T\Sigma)^{-1}\Sigma^T}U^T$$

$$\Longrightarrow \quad \|A^+\|_2=1/\sigma_n \quad \Longrightarrow \quad \mathrm{cond}(A)_2=\sigma_1/\sigma_n$$

其他情况，规定 $\mathrm{cond}(A)_2=\infty$

如何计算奇异值分解

- 利用求特征值的QR算法
  - 等效于求A^TA矩阵的特征值
  - 实际上不直接计算A^TA, 因为对A施加左、右正交变换不改变它的奇异值
  - 所以，先将A化简(Householder变换)为简单的B, 再对B^TB执行QR迭代算法

$$
A=\begin{bmatrix} a_1 & \cdots\cdots & a_n \end{bmatrix}
\xrightarrow[\boldsymbol{H}_1\boldsymbol{a}_1=\alpha_1\boldsymbol{e}_1^{(m)}]{\boldsymbol{H}_1\boldsymbol{A}}
\left[\begin{array}{c|c} \alpha_1 & \boldsymbol{b}_1^T \\ \hline 0 & \end{array}\right]
\xrightarrow[\boldsymbol{H}_2\boldsymbol{b}_1=\beta_1\boldsymbol{e}_1^{(n-1)}]{\boldsymbol{H}_1\boldsymbol{A}\begin{bmatrix}1 & 0\\ 0 & \boldsymbol{H}_2\end{bmatrix}=\boldsymbol{H}_1\boldsymbol{A}\boldsymbol{T}_1}
\left[\begin{array}{c|cc} \alpha_1 & \beta_1 & \\ \hline 0 & & \end{array}\right]
$$

$$
\xrightarrow{\cdots}
\begin{bmatrix} \alpha_1 & \beta_1 & & \\ & \ddots & \ddots & \\ & & \ddots & \beta_{n-1} \\ & & & \alpha_n \end{bmatrix}
=\boldsymbol{H}_n\cdots\boldsymbol{H}_1\boldsymbol{A}\boldsymbol{T}_1\cdots\boldsymbol{T}_{n-1}=\boldsymbol{B}
$$

QR迭代过程：

$\boldsymbol{B}_k^T\boldsymbol{B}_k$为三对角矩阵，迭代中保持其形状

演示程序: **eigsvdgui**

SVD的应用

- 任意矩阵的2-范数, 2-条件数

$$\|A\|_2 = \max_{x \neq o} \frac{\|Ax\|_2}{\|x\|_2} = \sigma_{\max} \ , \quad \mathrm{cond}(A)_2 = \sigma_{\max} / \sigma_{\min}$$

- 确定矩阵的秩: 不为零的奇异值的数目 $\boxed{AV = U\Sigma}$ span(A)的维数
- 解线性最小二乘问题 $Ax \cong b$
  - A列满秩, 奇异值均>0

$$U\Sigma V^T x \cong b \longrightarrow \Sigma V^T x \cong U^T b \longrightarrow \Sigma_1 V^T x = U_1^T b \longrightarrow x = V\Sigma_1^{-1} U_1^T b$$

  - A列不满秩, 存在0奇异值

$$\min \left\| U^T b - \Sigma V^T x \right\|_2 \xrightarrow{\text{令}\, y = V^T x} \min \left\| U^T b - \Sigma y \right\|_2$$

取最小值要求 $\sigma_i y_i = u_i^T b \ , \ i = 1, \cdots, r$

若y的其他分量=0, 是2-范数最小的解. 由于 $\|y\|_2 = \|x\|_2$,

得到原问题最小范数解: $x = Vy = \sum_{i=1}^{r} v_i y_i = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i$

SVD的应用

- 更一般的矩阵伪逆 (推广到列不满秩矩阵)
  - 标量σ的"伪逆": 1/σ, 或0
  - 对角阵的伪逆 Σ^+: 矩阵转置，然后每个对角元求"伪逆"
  - 一般的矩阵A：

    $A = U\Sigma V^T$ ，则 $A^+ = V\Sigma^+U^T$
  - 易证明，它与A列满秩时伪逆定义兼容 $\boxed{A^+ = (A^TA)^{-1}A^T,}$
  - 任意情况下，$Ax \cong b$的最小2-范数解为 $x = A^+b$
  - 其他伪逆的性质：

$$\left(A^+\right)^+ = A$$

$$\left(A^T\right)^+ = \left(A^+\right)^T$$

$$\left\|A^+\right\|_2 = 1/\sigma_r$$

SVD的应用

- 低秩矩阵近似 (Low rank matrix approximation)

$$A=[u_1\cdots u_m]\begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & & \end{bmatrix}\begin{bmatrix}v_1^T\\ \vdots \\ v_n^T\end{bmatrix}=[\sigma_1u_1\cdots\sigma_nu_n]\begin{bmatrix}v_1^T\\ \vdots \\ v_n^T\end{bmatrix}=\sum_{i=1}^{n}\sigma_iE_i$$

其中 $E_i=u_iv_i^T$, $i=1,\cdots,n$, 它秩为1, 奇异值 $\sigma_1=1$, 其余=0

存储 E_i 只要m+n个单元，计算 E_ix 只需m+n次乘法

$\|E_i\|_2=1$

忽略小奇异值对应求和项，称 $\tilde{A}=\sum_{i=1}^{k}\sigma_iE_i$

为A的秩k近似，可证明, 以Frobenius范数度量，在所有秩为k的矩阵中，它最接近A

演示程序svd_appl.m

在图像处理、数据压缩、信息抽取等方面都有应用

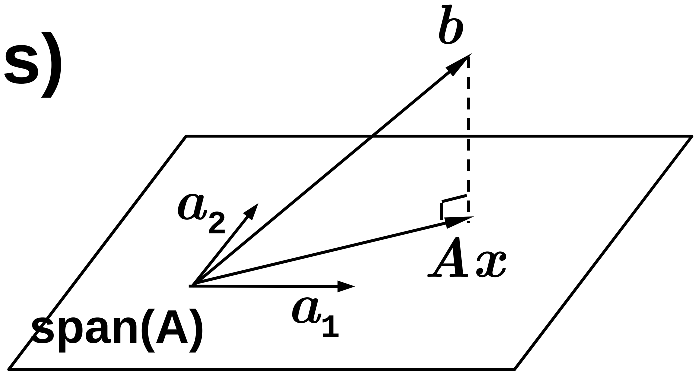
若A实对称, 从合同标准型导出类似公式 , 其中 E_i 更简单

PCA, PFA

SVD的应用

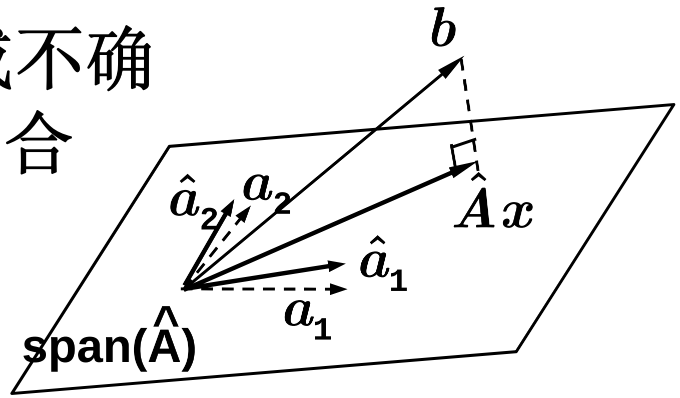
- 完全最小二乘问题 (Total least squares)

普通线性最小二乘问题：$[a_1 \cdots a_n]\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = Ax \cong b$



假设 b 的数据有误差, 而模型数据, 即 A 准确, 数据点到曲线竖直距离最小

完全最小二乘问题：所有数据都有误差或不确定性，包括 A 中的数据, 要求 b 和 A 到拟合"平面"总的2-范数距离最小



设 $y = \hat{A}x$, x 为待求的参数, 则

$$\begin{bmatrix} \hat{A} & y \end{bmatrix} \approx \begin{bmatrix} A & b \end{bmatrix}$$

列秩： n (n+1)

解法：用SVD求[A b]的秩n近似

若 $\begin{bmatrix} A & b \end{bmatrix} = U\Sigma V^T$, 则 $\begin{bmatrix} \hat{A} & y \end{bmatrix} = U\hat{\Sigma} V^T$,

$\hat{\Sigma}$ 由将 Σ 中 σ_{n+1} 变为0得到

$$\begin{bmatrix} \hat{A} & y \end{bmatrix}\begin{bmatrix} x \\ -1 \end{bmatrix} = O$$

$\begin{bmatrix} x \\ -1 \end{bmatrix}$ 与 v_{n+1} 成比例

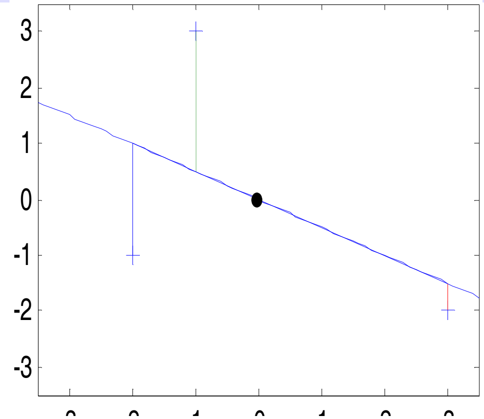
SVD的应用

- 完全最小二乘问题的例子

用模型函数 $f(t) = xt$

拟合数据：

t	-2	-1	3
y	-1	3	-2

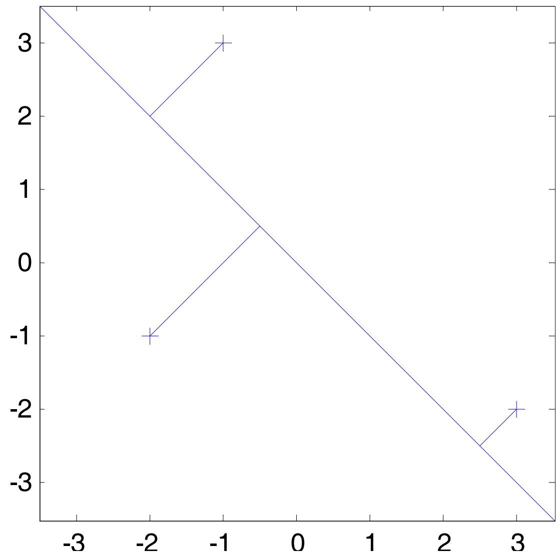


常规最小二乘问题拟合结果: $x = -0.5$

完全最小二乘拟合:

$$[A \quad b] = \begin{bmatrix} -2 & -1 \\ -1 & 3 \\ 3 & -2 \end{bmatrix} = \begin{bmatrix} -0.154 & 0.802 & 0.577 \\ -0.617 & -0.533 & 0.577 \\ 0.772 & -0.267 & 0.577 \end{bmatrix}$$

$$\begin{bmatrix} 4.583 & 0 \\ 0 & 2.646 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0.707 & -0.707 \\ -0.707 & -0.707 \end{bmatrix} \Rightarrow x = \frac{v_{n+1}(1:n)}{-v_{n+1}(n+1)} = -1$$



$$[\hat{A} \quad \hat{y}] \approx [A \quad b]$$

$$\begin{matrix} \downarrow & \downarrow & & \downarrow & \downarrow \\ \hat{t} & \hat{y} & & t & y \end{matrix}$$

所有过原点直线中，数据点到该线垂线距离平方和最小!

Matlab命令与方法比较

- **Matlab中svd命令**
  - s = svd(X); [U,S,V] = svd(X); 返回奇异值，或完整分解
  - [U,S,V] = svd(X,0); [U,S,V] = svd(X,‘econ’); 分解的简化形式, **S**为方阵，**U**或**V**为正交阵的一部分
- 方法间比较
  - 解线性最小二乘问题, 正交变换法最稳定、准确，但计算量可能比法方程方法大(**Householder**变换, **m>>n)**
  - 对**(**近似**)**秩亏损问题，只能用列主元正交变换法**,** 或基于**SVD**的方法, 后者更可靠，计算量则更大
  - 奇异值、特征值分解联系紧密, 前者似乎更易用一些

Some useful Matlab commands (2)

- **Constants:** realmin, realmax, eps, Inf, -Inf, NaN
- **Logical:** isinf, isfinite, isnan
- **Floating point arithmetic:** single, double, vpa
- **Operators:** \ (back slash)
- **Matrices:** zeros, ones, eye, diag, rand, cond, condest, rank, det, inv, eig, tril, triu, sparse, issparse, speye, spones, spdiags, sprand, nnz, full, eigs
- **Graphs:** spy
- **Algorithms:** lu, chol, qr, eig, svd, hess, schur, polyfit

- 本节内容小结
  - 矩阵的奇异值分解的概念
  - 分解的存在性, 唯一性
  - 奇异向量的意义
  - 奇异值与**2-**范数, **2-**条件数
  - 计算奇异值的算法
  - 奇异值分解的应用
    - **2-**范数、条件数，矩阵的秩
    - 解秩亏损的线性最小二乘问题, 最小**2-**范数解
    - 一般的伪逆定义
    - 低秩矩阵近似
    - 完全最小二乘问题的求解

Assignment

- 阅读课本第三章, 第四章有关内容
- 阅读文献: **(“教学资源”--” top 10 algorithm”)**
- B. N. Parlett, “The QR Algorithm,” Computing in Science and Engineering, Vol. 2, No. 1, pp. 38-44, 2000
- 作业题见网络学堂