数 值 分 析 (5)

Numerical Analysis

第五章 矩阵特征值问题

- 矩阵特征值与特征向量的计算：∈数值线性代数, 矩阵计算(matrix computation)的重要内容
- 矩阵特征值=特征多项式方程的解 ⟹ 一定是迭代解法
- 本章内容
  - 基本概念与特征值的分布
  - 幂法与反幂法
  - 矩阵的正交三角化, QR分解
  - 计算所有特征值的QR迭代算法

矩阵特征值的基本概念

Wenjian Yu

基本概念

- 矩阵$\boldsymbol{A}$的**特征值**与**特征向量**, ($\boldsymbol{A}$为方阵)

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$$

- 特征值λ是**特征方程**的根, 复数域内有**n**个(含重根)

$$\det(\lambda\boldsymbol{I} - \boldsymbol{A}) = \begin{vmatrix} \lambda - a_{11} & -a_{12} & \cdots & -a_{1n} \\ -a_{21} & \lambda - a_{22} & \cdots & -a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ -a_{n1} & -a_{n2} & \cdots & \lambda - a_{nn} \end{vmatrix} = 0$$

- **特征值谱**: $\lambda(\boldsymbol{A})$
- 给定一特征值λ, 特征向量是方程$(\lambda\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x} = \boldsymbol{0}$的非零解
- 对任一特征值, 特征向量都不唯一, 构成**特征子空间**
- 假设$\boldsymbol{A} \in \mathbb{R}^{n\times n}$, (1)特征值不一定是实数; (2)实特征值一定对应实特征向量; (3)非实特征值的共轭也是特征值, 其对应的特征向量一定不是实向量

特征值的有关性质

- 非奇异矩阵特征值均不为0; 0一定是奇异矩阵的特征值

Th5.2
- $\lambda(\boldsymbol{A}) = \lambda(\boldsymbol{A}^T)$ $\quad \det(\lambda\boldsymbol{I} - \boldsymbol{A}) = \det\left((\lambda\boldsymbol{I} - \boldsymbol{A})^T\right)$

Th5.3
- 若$\boldsymbol{A}$为对角阵或上(下)三角阵, 则其特征值为其对角元

Th5.4
- 若$\boldsymbol{A}$为分块对角阵或分块上(下)三角阵(对角块为方阵), 则 $\lambda(\boldsymbol{A}) = \bigcup_{j=1}^{m} \lambda\left(\boldsymbol{A}_{jj}\right)$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} & \cdots & \boldsymbol{A}_{1m} \\ & \boldsymbol{A}_{22} & \cdots & \boldsymbol{A}_{2m} \\ & & \ddots & \vdots \\ & & & \boldsymbol{A}_{mm} \end{bmatrix}$$

Th5.5
- 相似矩阵的特征值相等

$$\boldsymbol{B} = \boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$$

Th5.9
- 矩阵运算结果的特征值: 设λ_j为$\boldsymbol{A}$的特征值，则

$\lambda(c\boldsymbol{A}) = \{c\lambda_j\}$; $\lambda(\boldsymbol{A}+c\boldsymbol{I}) = \{\lambda_j + c\}$; $\lambda(\boldsymbol{A}^k) = \{\lambda_j^k\}$; $\lambda(\boldsymbol{A}^{-1}) = \{\lambda_j^{-1}\}$;

- **定义5.2** 设$\boldsymbol{A} \in \mathbb{R}^{n\times n}$有m个(m≤n)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, 则称$n_j$为$\tilde{\lambda}_j$的代数重数, 而$\tilde{\lambda}_j$的特征子空间的维数其几何重数.

特征值的有关性质

- 设n阶方阵$\boldsymbol{A}$的m个不同的特征值为$\tilde{\lambda}_1, \cdots, \tilde{\lambda}_m$，$\tilde{\lambda}_j$的代数重数为$n_j$, 几何重数为$k_j$, 则
  - Th5.6 $\sum_j n_j = n$; $\forall j, n_j \geq k_j$
  - 不同特征值的特征向量线性无关, 所有特征子空间的$\sum_{j=1}^{m} k_j$个基形成一组线性无关向量
  - 定义5.3 若$\forall j, n_j = k_j$, 这种矩阵$\boldsymbol{A}$为**非亏损阵**, 否则为**亏损阵**

    非亏损阵有n个特征向量构成全空间的基 $(\mathbb{C}^n)$ (特征值分解)
- **Th5.7**: $\exists X^{-1}AX = \Lambda$, Λ为对角阵 $\Leftrightarrow$ A为非亏损阵
- 实对称阵非亏损, 且特征值为实数 (可正交对角化)
- **Th5.8**(Jordan分解): $A = XJX^{-1}$,

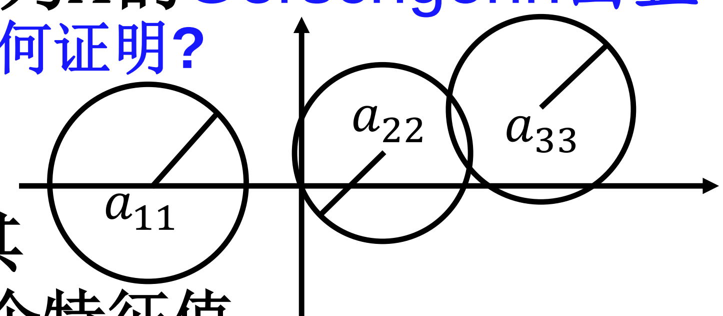
$\tilde{\lambda}_j$的几何重数为k_j, $p=\sum k_j$, $\tilde{\lambda}_j$对应k_j个约当块, 其阶数之和$= n_j$

$$J=\begin{bmatrix} J_1 & & \\ & \ddots & \\ & & J_p \end{bmatrix}, \quad J_s=\begin{bmatrix} \lambda_s & 1 & & \\ & \lambda_s & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_s \end{bmatrix}$$

特征值的分布

- **Motivation**
  - 迭代法$\boldsymbol{x}^{(k+1)}=\boldsymbol{B}\boldsymbol{x}^{(k)}+\boldsymbol{f}$的收敛性, 看$\rho(\boldsymbol{B})=\max_j|\lambda_j(\boldsymbol{B})|$
  - 矩阵的2-范数、2-条件数: $\mathrm{cond}(\boldsymbol{A})_2=\sqrt{\frac{\lambda_{\max}(\boldsymbol{A}^T\boldsymbol{A})}{\lambda_{\min}(\boldsymbol{A}^T\boldsymbol{A})}}$
- $\rho(\boldsymbol{A})\le\|\boldsymbol{A}\|$是关于特征值上界的重要结论
- **定义5.4** 对于复矩阵$\boldsymbol{A}$, 设$r_k=\sum_{j=1,j\neq k}^{n}|a_{kj}|$, 在复平面上以$a_{kk}$为圆心、$r_k$为半径的圆, 称为$\boldsymbol{A}$的Gerschgorin圆盘
- **Th5.10** (圆盘定理)
  - $\boldsymbol{A}$的特征值必在某个圆盘上
  - 若n个圆盘中有m个连通, 且与其他分离, 则这m个圆盘恰包含m个特征值

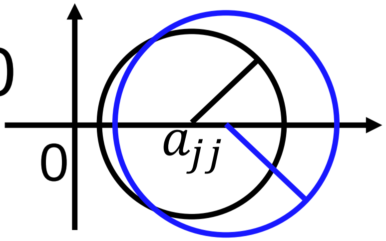
思考: 如何证明?



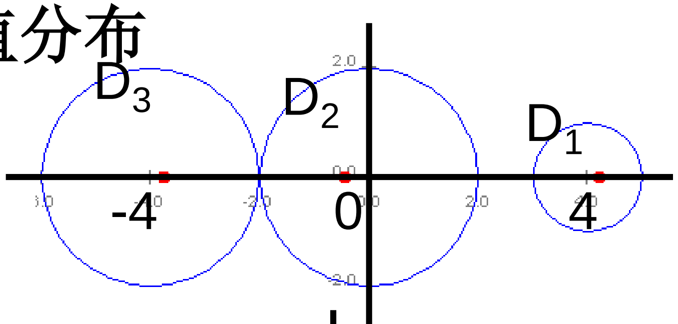
交互演示网站

圆盘定理的应用

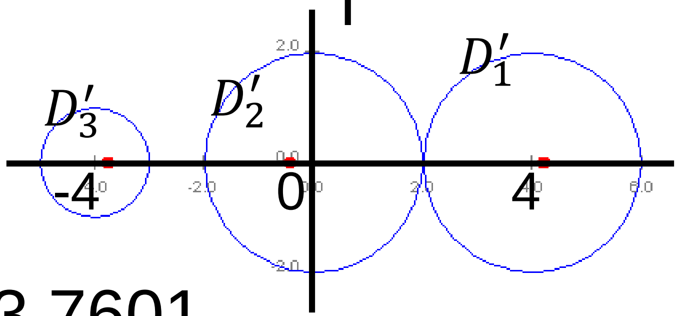
- **Th5.11** 严格对角占优阵, 正对角元, Re(λ)>0
  若**A**为对角占优的对称阵，则…



- 例: 估计矩阵$\boldsymbol{A}=\begin{bmatrix}4 & 1 & 0\\ 1 & 0 & -1\\ 1 & 1 & -4\end{bmatrix}$的特征值分布



- 直接应用圆盘定理, D_1与其他两个分离, 必含一个实特征值
- 再对$\boldsymbol{A}^T$应用圆盘定理, D_3'与其他圆盘分离, 必含一实特征值



- $\lambda_1 \in [3,5], \lambda_2 \in [-2,2], \lambda_3 \in [-5,-3]$
- 准确特征值为: 4.2030, -0.4429, -3.7601
- 还可先做简单的相似变换, 再估计分布, 见pp.153, 例5.4

幂法与反幂法

Wenjian Yu

计算最大的特征值、特征向量

- **定义5.5** 模最大的特征值称为主特征值, 也叫"第一特征值", 它对应的特征向量称为主特征向量
- 主特征值可能不唯一, 例如$5, -5, 3+4i, 3-4i$的模都是**5**
- 幂法(power iteration): 取任意非零向量$\boldsymbol{v}_0$, 计算$\boldsymbol{v}_k = \boldsymbol{A}\boldsymbol{v}_{k-1}, (k = 1,2,\cdots)$, 得到向量序列$\{\boldsymbol{v}_k\}$
- 看一个例子: $\boldsymbol{A} = \begin{bmatrix} 5 & -1 & -1 \\ 3 & 1 & -1 \\ 4 & -2 & 1 \end{bmatrix}$ (pp. 148, 例5.2, 其主特征值为3)
- 取$\boldsymbol{v}_0 = [1, 0, 0]^T$, 利用Matlab算向量序列$\{\boldsymbol{v}_k\}$ Matlab demo
- 通过实验看出: 相邻两个$\boldsymbol{v}_k$逐渐呈倍数关系, 倍数为**3**
- $\boldsymbol{v}_k$趋近于**3**对应的特征值向量

更多例子: 交互演示网站

幂法

- 幂法是否总能计算矩阵的主特征值?
- 定理5.12 若矩阵$\boldsymbol{A}$有唯一的主特征值λ_1, 向量$\boldsymbol{v}_0$非零, 计算$\boldsymbol{v}_k = \boldsymbol{A}\boldsymbol{v}_{k-1}, (k = 1,2,\cdots)$, 则一般有

$$\lim_{k\to\infty}\frac{(\boldsymbol{v}_{k+1})_j}{(\boldsymbol{v}_k)_j} = \lambda_1 \qquad \lim_{k\to\infty}\frac{\boldsymbol{v}_k}{\lambda_1^k} = \boldsymbol{x}_1$$

条件: λ_1的几何重数等于代数重数

$\boldsymbol{x}_1$为某个主特征向量, j的取值须保证有$\boldsymbol{x}_1$使$(\boldsymbol{x}_1)_j \neq 0$

- 证明: 设$\boldsymbol{A}$非亏损阵, 且λ_1不是重特征值, $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$

设$\boldsymbol{A}$的线性无关的单位特征向量为$\hat{\boldsymbol{x}}_1, \cdots, \hat{\boldsymbol{x}}_n$, $\boldsymbol{v}_0 = \alpha_1\hat{\boldsymbol{x}}_1 + \cdots + \alpha_n\hat{\boldsymbol{x}}_n$

$$\boldsymbol{v}_k = \boldsymbol{A}^k\boldsymbol{v}_0 = \alpha_1\lambda_1^k\hat{\boldsymbol{x}}_1 + \cdots + \alpha_n\lambda_n^k\hat{\boldsymbol{x}}_n = \lambda_1^k\left[\alpha_1\hat{\boldsymbol{x}}_1 + \sum_{j=2}^{n}\alpha_j\left(\frac{\lambda_j}{\lambda_1}\right)^k\hat{\boldsymbol{x}}_j\right]$$

思考: 若α_1=0呢?

$$\therefore \lim_{k\to\infty}\frac{\boldsymbol{v}_k}{\lambda_1^k} = \alpha_1\hat{\boldsymbol{x}}_1 \Longrightarrow \lim_{k\to\infty}\frac{(\boldsymbol{v}_{k+1})_j}{(\boldsymbol{v}_k)_j} = \lambda_1$$

对j的取值要求

幂法

$$\boldsymbol{v}_k = \boldsymbol{A}\boldsymbol{v}_{k-1}$$

特殊情况: $\boldsymbol{v}_0$不能为某特征向量!

- 关于幂法的说明
  - $\boldsymbol{A}$为亏损矩阵的情况, 用矩阵的Jordan标准型进行证明?
  - $\{\boldsymbol{v}_k\}$相邻项的第j分量的比值→主特征值, 选j为最大分量
  - 证明中假设了$\alpha_1 \neq 0$, 在实际应用时可随机选取$\boldsymbol{v}_0$
- 关于幂法的问题

  $\boldsymbol{v}_k \approx \lambda_1^k \boldsymbol{x}_1$, k很大时, 可能出现上溢或下溢

  $$\boldsymbol{v}_k = \lambda_1^k \left[\alpha_1 \hat{\boldsymbol{x}}_1 + \sum_{j=2}^{n} \alpha_j \left(\frac{\lambda_j}{\lambda_1} \right)^k \hat{\boldsymbol{x}}_j \right]$$, 收敛速度主要取决于$\left| \frac{\lambda_2}{\lambda_1} \right|$
- 实用的幂法: 用规格化向量的技术防止溢出
- 定义5.6 记$\overline{\max}(\boldsymbol{v})$为向量$\boldsymbol{v}$的绝对值最大分量, 若不唯一则取最小编号的那个. 称$\boldsymbol{u} = \boldsymbol{v} / \overline{\max}(\boldsymbol{v})$为向量$\boldsymbol{v}$的规格化向量

实用的幂法

- 规格化向量
  - 例: $\boldsymbol{v}=[3,-5,0]^T, \overline{\max}(\boldsymbol{v})=-5$, 规格化向量为$\boldsymbol{u}=\left[-\frac{3}{5},1,0\right]^T$
  - 若$\boldsymbol{u}$为规格化向量, 则$\|\boldsymbol{u}\|_\infty = 1$, $\quad \overline{\max}(\boldsymbol{u}) =?$
  - 向量$\boldsymbol{v}_1$,$\boldsymbol{v}_2$的规格化向量分别为$\boldsymbol{u}_1$,$\boldsymbol{u}_2$, 若$\boldsymbol{v}_1=\alpha\boldsymbol{v}_2$, 则$\boldsymbol{u}_1=\boldsymbol{u}_2$
- 在幂法的每步增加向量规格化操作

$$\boldsymbol{v}_1 = \boldsymbol{A}\boldsymbol{v}_0 \Longrightarrow \text{规格化向量}\boldsymbol{u}_1 = \frac{\boldsymbol{v}_1}{\overline{\max}(\boldsymbol{v}_1)} = \frac{\boldsymbol{A}\boldsymbol{v}_0}{\overline{\max}(\boldsymbol{A}\boldsymbol{v}_0)}$$

$$\boldsymbol{v}_2 = \boldsymbol{A}\boldsymbol{u}_1 = \frac{\boldsymbol{A}^2\boldsymbol{v}_0}{\overline{\max}(\boldsymbol{A}\boldsymbol{v}_0)} \Longrightarrow \text{规格化向量}\boldsymbol{u}_2 = \frac{\boldsymbol{v}_2}{\overline{\max}(\boldsymbol{v}_2)} = \frac{\boldsymbol{A}^2\boldsymbol{v}_0}{\overline{\max}(\boldsymbol{A}^2\boldsymbol{v}_0)}$$

$$\begin{cases} \boldsymbol{v}_k = \boldsymbol{A}\boldsymbol{u}_{k-1} = \dfrac{\boldsymbol{A}^k\boldsymbol{v}_0}{\overline{\max}(\boldsymbol{A}^{k-1}\boldsymbol{v}_0)} \\ \boldsymbol{u}_k = \dfrac{\boldsymbol{v}_k}{\overline{\max}(\boldsymbol{v}_k)} = \dfrac{\boldsymbol{A}^k\boldsymbol{v}_0}{\overline{\max}(\boldsymbol{A}^k\boldsymbol{v}_0)} \end{cases}, k = 1,2,\cdots \Longrightarrow \begin{array}{l} \lim\limits_{k\to\infty} \overline{\max}(\boldsymbol{v}_k) = \lambda_1 \\ \lim\limits_{k\to\infty} \boldsymbol{u}_k = \dfrac{\boldsymbol{x}_1}{\overline{\max}(\boldsymbol{x}_1)} \end{array}$$

定理5.14 (只需主特征值唯一)

实用的幂法

- **算法5.1**:计算主特征值λ_1和主特征向量$\boldsymbol{x}_1$的实用幂法

输入： $\boldsymbol{v}, \boldsymbol{A}$; 输出： $\boldsymbol{x}_1, \lambda_1$.
$\boldsymbol{u} := \boldsymbol{v}$;
While 不满足判停准则 do
  $\boldsymbol{v} := \boldsymbol{A}\boldsymbol{u}$;
  $\lambda_1 := \overline{\max}(\boldsymbol{v})$; {主特征值近似值}
  $\boldsymbol{u} := \boldsymbol{v}/\lambda_1$; {规格化}
End
$\boldsymbol{x}_1 := \boldsymbol{u}$. {规格化的主特征向量}

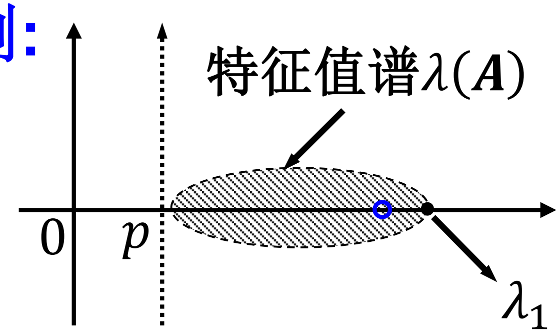
每步的主要计算是算一次矩阵与向量乘法

若$\boldsymbol{A}$为稀疏矩阵, 很容易提高效率

- 用相邻两步特征值近似值之差作为判停准则
- 保证向量的值不溢出, 绝对值最大分量收敛到主特征值
- 适用范围: 主特征值唯一

加速幂法的收敛

例:



- 原点位移技术
  - $\boldsymbol{B}=\boldsymbol{A}-p\boldsymbol{I}$的特征值为$\boldsymbol{A}$的特征值$-p$
    对$\boldsymbol{B}$应用幂法可能加快收敛
  - 如图中$\boldsymbol{A}$的特征值分布, $\left|\frac{\lambda_2(\boldsymbol{B})}{\lambda_1-p}\right|$较小, 因此更快算出$\lambda_1$
- 瑞利商(Rayleigh quotient)加速
  - 实对称矩阵$\boldsymbol{A}$的瑞利商: $R(\boldsymbol{x})=\frac{\langle \boldsymbol{A}\boldsymbol{x},\boldsymbol{x}\rangle}{\langle \boldsymbol{x},\boldsymbol{x}\rangle}=\frac{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}}$

Th5.15
  - 对实对称阵$\boldsymbol{A}$, $\lambda_n\le R(\boldsymbol{x})\le\lambda_1$, $\lambda_1(\lambda_n)$为最大(最小)特征值; 若$\boldsymbol{x}$为相应特征向量, 上式取等号

证明中利用"实对称阵可正交对角化"

Th5.16
  - 结合幂法, 加速特征值的收敛: 每步对$\boldsymbol{u}_k$计算$R(\boldsymbol{u}_k)$
  - 每步仅多计算两次向量内积

自己看课本pp. 158-159

反幂法

- A^{-1}的特征值是A特征值的倒数, 对A^{-1}应用幂法得最小特征值

算法5.2 计算最小特征值与特征向量的反幂法
输入： $\boldsymbol{v}, \boldsymbol{A}$；输出： $\boldsymbol{x}_n, \lambda_n$.
$\boldsymbol{u} := \boldsymbol{v}$；
While 不满足判停准则 do
    $\boldsymbol{v} := \boldsymbol{A}^{-1}\boldsymbol{u}$；　　　　{求解线性方程组}
    $\lambda_n := 1/\overline{\max}(\boldsymbol{v})$；　{最小特征值的近似值}
    $\boldsymbol{u} := \lambda_n \boldsymbol{v}$；　　　　　{规格化}
End
$\boldsymbol{x}_n := \boldsymbol{u}$.　　　　　　　{规格化的特征向量}

求解线性方程组, 计算量可能比幂法大很多

- 适用范围: A按模最小的特征值唯一
- 与原点位移技术结合: 若已知某个特征值$\lambda_j \approx p$, 则$\lambda_j - p$是$B = A - pI$按模最小的特征值, 对B使用反幂法 可用瑞利商加速

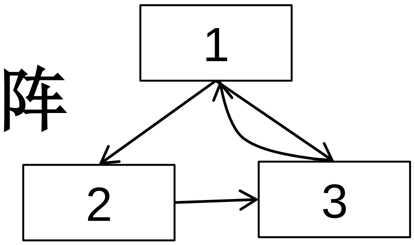
小结

- 实用幂法可能失败的情况
  - 矩阵A的主特征值不唯一, 例如实矩阵, 模最大的特征值不是实数
  - 初始向量在主特征向量方向没有分量 (通常没问题)
  - 反幂法的情况类似
- 说明几点
  - 按幂法迭代计算, 若前后两次迭代向量成比例, 则它一定就是特征向量, 也相应求出特征值
  - 加速幂法的方法还有Aitken外推等算法
  - 反幂法结合位移技术, 以及瑞利商加速, 也很有用

应用实例: PageRank™的计算

- **Google网络搜索**
  - PageRank技术是其创立之初的关键创新之一
  - 搜索分两步: **找到匹配关键词的网址, 排序显示**
  - L. Page和S. Brin于1998年提出PageRank算法
  - 给出网页信息可靠/重要性的指标(PageRank)
  - 怎样的网页PageRank高? **被推荐, 被链接到**
- 数学模型
  - n: 网页的总数, $\boldsymbol{G}=\left(g_{ij}\right)_{n\times n}$: 网页链接矩阵
  - 若网页j链接到网页i, 则 $g_{ij}=1$, 否则$g_{ij}=0$

$$\boldsymbol{G}=\begin{bmatrix}0 & 0 & 1\\ 1 & 0 & 0\\ 1 & 1 & 0\end{bmatrix}$$



大规模、非对称、稀疏阵

应用实例: PageRank™的计算

- 数学模型 (续)
  - $\boldsymbol{G}$的列元素之和$c_j = \sum_i g_{ij}$, 等于网页j的"出度"
  - 访问网页的**Markov**过程, 到达网页的极限概率定义为它的PageRank (按概率p沿链接跳, 按概率1-p随便跳)
  - 设当前在网页j、下一步到网页i的条件概率为a_{ij}:

i在j的链接上: $p \cdot 1/c_j + (1-p) \cdot 1/n$ 乘g_{ij}

i不在j的链接上: $(1-p) \cdot 1/n$ 乘(1-g_{ij})

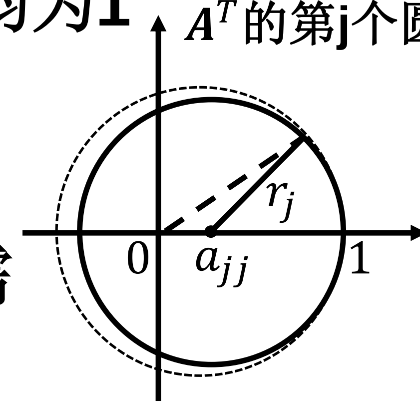
$$a_{ij} = \frac{p g_{ij}}{c_j} + \frac{1-p}{n}$$

  - a_{ij}组成转移概率矩阵$\boldsymbol{A}$: $\boldsymbol{A} = p\boldsymbol{G}\boldsymbol{D} + \frac{1-p}{n}\boldsymbol{e}\boldsymbol{e}^T$ (若不存在$c_j = 0$情况)
  - 若有$c_j = 0$呢? $a_{ij} = 1/n$, 修改$\boldsymbol{A}$相应的列
  - 设$\boldsymbol{x}^{(k)}$为第k次跳转后在各个网页的概率, 则$\boldsymbol{x}^{(k+1)} = \boldsymbol{A}\boldsymbol{x}^{(k)}$为再跳一次后在各网页的概率

应用实例: PageRank™的计算

- 数学模型(续)
  - PageRank: 随机"冲浪"过程访问网页的极限概率

    $\boldsymbol{x}^{(k+1)} = \boldsymbol{A}\boldsymbol{x}^{(k)}$　　　$\lim_{k\to\infty} \boldsymbol{x}^{(k)} = ?$　　它存在吗?
- 上述算法(幂法)的理论分析
  - 数学本质: 求1对应的特征向量$\boldsymbol{x}$　　$\begin{cases} \boldsymbol{A}\boldsymbol{x} = \boldsymbol{x} \\ \sum_{i=1}^{n} x_i = 1 \end{cases}$
  - 矩阵$\boldsymbol{A}$的特点: $a_{ij} > 0$, 每列元素之和均为1
  - $\rho(\boldsymbol{A}) \le 1$; $\boldsymbol{A}^T\boldsymbol{e} = \boldsymbol{e}$ ⟹ 1是主特征值
  - 可放心地用幂法　　唯一主特征值



- PageRank反映超链接结构, 隔一段时间需重新计算
- 计算中不需对$\boldsymbol{x}^{(k)}$规格化; 实际编程时不形成矩阵$\boldsymbol{A}$

矩阵的QR分解

Wenjian Yu

Householder变换

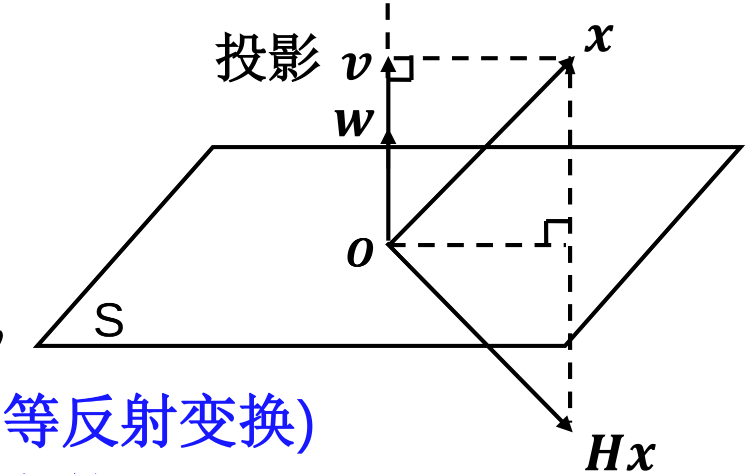
- 矩阵的正交三角化
  - 高斯消去过程
  - 可用正交阵来乘吗?
  - 是计算所有特征值的算法的基础

- Householder矩阵
  - $\|\boldsymbol{w}\|_2 = 1$
  - **定义5.8** $\boldsymbol{w} \in \mathbb{R}^n$且$\boldsymbol{w}^T\boldsymbol{w} = 1$, 称$\boldsymbol{H}(\boldsymbol{w}) = \boldsymbol{I} - 2\boldsymbol{w}\boldsymbol{w}^T$为Householder矩阵 (初等反射阵)
  - $\boldsymbol{H}(\boldsymbol{w}) = \boldsymbol{H}(-\boldsymbol{w})$
  - $\boldsymbol{H}$为对称阵、正交阵
  - $\boldsymbol{H}\boldsymbol{x}$实现Householder变换

$$\boldsymbol{H} = \begin{bmatrix} 1-2w_1^2 & -2w_1w_2 & \cdots & -2w_1w_n \\ -2w_2w_1 & 1-2w_2^2 & \cdots & -2w_2w_n \\ \vdots & \vdots & \cdots & \vdots \\ -2w_nw_1 & -2w_nw_2 & \cdots & 1-2w_n^2 \end{bmatrix}$$

Householder变换

交互演示5.4



- Householder变换的几何意义
  - $\boldsymbol{Hx}$: 以$\boldsymbol{w}$为法向画出超平面S

$$\boldsymbol{Hx} = (\boldsymbol{I} - 2\boldsymbol{w}\boldsymbol{w}^T)\boldsymbol{x} = \boldsymbol{x} - 2\boldsymbol{w}\boldsymbol{w}^T\boldsymbol{x}$$

$$\boldsymbol{w}\boldsymbol{w}^T\boldsymbol{x} = (\boldsymbol{w}^T\boldsymbol{x})\boldsymbol{w} = \boldsymbol{v} \Longrightarrow \boldsymbol{Hx} = \boldsymbol{x} - 2\boldsymbol{v}$$

  - $\boldsymbol{Hx}$为$\boldsymbol{x}$关于平面S的镜像 (初等反射变换)
  - $\boldsymbol{Hx}$与$\boldsymbol{x}$的2-范数相等, 属于正交变换
- **Th5.18** 设$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \boldsymbol{x} \neq \boldsymbol{y}, \|\boldsymbol{x}\|_2 = \|\boldsymbol{y}\|_2$, 则存在Householder矩阵$\boldsymbol{H}$, 使$\boldsymbol{Hx} = \boldsymbol{y}$
- 几何的启示: $\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{y}$, $\boldsymbol{w} = \boldsymbol{v}/\|\boldsymbol{v}\|_2$，构造矩阵$\boldsymbol{H}$
- **Th5.19** 可将Th5.18中的$\boldsymbol{y}$设为 $\begin{bmatrix} \pm\|\boldsymbol{x}\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \equiv -\sigma \boldsymbol{e}_1$ 用正交变换实现消元！

  构造$\boldsymbol{H}$时, $\boldsymbol{v} = \boldsymbol{x} + \sigma \boldsymbol{e}_1$

  $\therefore \sigma = \mathrm{sign}(x_1)\|\boldsymbol{x}\|_2$

Householder变换

- Householder变换
  - 对向量做正交变换实现消元, 结果$-\sigma \boldsymbol{e}_1$中的负号是为了数值稳定
  - 例: 确定一个Householder变换, 对向量实现消元操作

$$\boldsymbol{a} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} \Longrightarrow \begin{bmatrix} * \\ 0 \\ 0 \end{bmatrix}$$

解: $\sigma = \operatorname{sign}(a_1)\|\boldsymbol{a}\|_2 = 3$, 构造$\boldsymbol{v} = \boldsymbol{a} + \sigma \boldsymbol{e}_1 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix}$

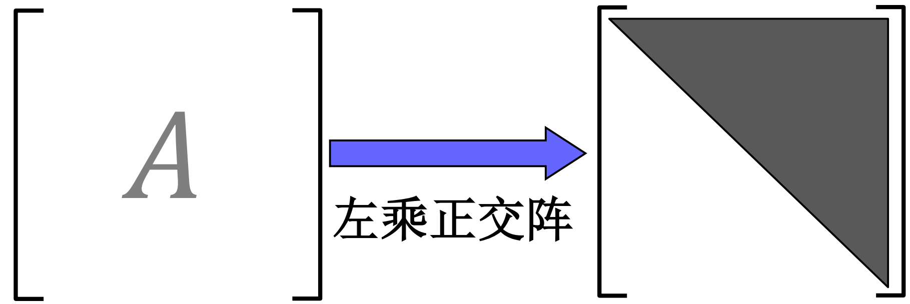
取$\boldsymbol{w} = \boldsymbol{v}/\|\boldsymbol{v}\|_2$, 则实现变换的矩阵为$\boldsymbol{H} = \boldsymbol{I} - 2\boldsymbol{w}\boldsymbol{w}^T$

验证: $\boldsymbol{H}\boldsymbol{a} = \boldsymbol{a} - 2(\boldsymbol{w}^T\boldsymbol{a})\boldsymbol{w} = \boldsymbol{a} - 2\dfrac{\boldsymbol{v}^T\boldsymbol{a}}{\boldsymbol{v}^T\boldsymbol{v}}\boldsymbol{v} = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix} - 2 \times \dfrac{15}{30} \times \begin{bmatrix} 5 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -3 \\ 0 \\ 0 \end{bmatrix}$

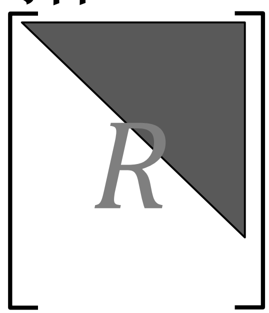
很重要! 用向量$\boldsymbol{v}$或$\boldsymbol{w}$表示矩阵$\boldsymbol{H}$, 只算向量内积

矩阵的QR分解

- 用Householder变换实现正交三角化？
  - $\boldsymbol{H}_k \cdots \boldsymbol{H}_2 \boldsymbol{H}_1 \boldsymbol{A} = \boldsymbol{R}$
  - $\boldsymbol{A} \in \mathbb{R}^{m \times n}$不一定是方阵，上三角阵$\boldsymbol{R}$也同样



$m > n$的情况：



  - $\boldsymbol{A} = \boldsymbol{H}_1 \boldsymbol{H}_2 \cdots \boldsymbol{H}_k \boldsymbol{R} = \boldsymbol{Q}\boldsymbol{R}$，其中$\boldsymbol{Q}$为正交阵，称为**QR分解**
  - 还有其他正交变换消元的手段：Givens旋转变换，后面将介绍。

Th5.20 对任意实矩阵$\boldsymbol{A}$，一定存在QR分解；
若$\boldsymbol{A}$为方阵，且要求$\boldsymbol{R}$的对角元都>0，此分解唯一。

矩阵的QR分解

交互演示5.5

- 用Householder变换实现: $\boldsymbol{H}_k\cdots\boldsymbol{H}_2\boldsymbol{H}_1\boldsymbol{A}=\boldsymbol{R}$
  - 设$\boldsymbol{A}\in\mathbb{R}^{m\times n}, m\geq n$, $\boldsymbol{A}=[\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n]$, 其中$\boldsymbol{a}_j$为$m$维向量

构造m阶反射阵$\boldsymbol{H}_1$消$\boldsymbol{A}$的第1列:

$$\boldsymbol{A}^{(2)}=\boldsymbol{H}_1\boldsymbol{A}=\begin{bmatrix}-\sigma_1 & a_{12}^{(2)} & \cdots & a_{1n}^{(2)}\\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)}\\ \vdots & \vdots & & \vdots\\ 0 & a_{m2}^{(2)} & \cdots & a_{mn}^{(2)}\end{bmatrix}=\begin{bmatrix}-\sigma_1 & | & & |\\ 0 & \boldsymbol{H}_1\boldsymbol{a}_2 & \cdots & \boldsymbol{H}_1\boldsymbol{a}_n\\ \vdots & | & & |\\ 0 & | & & |\end{bmatrix}$$

用$\boldsymbol{H}_2$实现$\boldsymbol{A}^{(2)}$第2列的消元, 同时不影响第1列

令$\boldsymbol{H}_2=\begin{bmatrix}1 & \boldsymbol{0}^T\\ \boldsymbol{0} & \boldsymbol{H}_2'\end{bmatrix}$

记$\boldsymbol{A}^{(2)}=\begin{bmatrix}-\sigma_1 & \boldsymbol{r}_1^T\\ \boldsymbol{0} & \boldsymbol{A}'^{(2)}\end{bmatrix}$

$$\boldsymbol{H}_2\boldsymbol{A}^{(2)}=\begin{bmatrix}-\sigma_1 & \boldsymbol{r}_1^T\\ \boldsymbol{0} & \boldsymbol{H}_2'\boldsymbol{A}'^{(2)}\end{bmatrix}=\left[\begin{array}{c|cccc}-\sigma_1 & & \boldsymbol{r}_1^T & & \\ \hline 0 & -\sigma_2 & | & & |\\ 0 & 0 & \boldsymbol{H}_2'\boldsymbol{a}_2'^{(2)} & \cdots & \boldsymbol{H}_2'\boldsymbol{a}_{n-1}'^{(2)}\\ \vdots & \vdots & | & & |\\ 0 & 0 & | & & |\end{array}\right]$$

构造$\boldsymbol{H}_2'$

$\boldsymbol{H}_2$也是Householder阵(其向量$\boldsymbol{v}$的第一个分量为0).后续$\boldsymbol{H}_j$可类似构造

算法5.3：基于Householder变换的矩阵正交三角化

输入: $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_n]$; **输出**: $\boldsymbol{A}; \boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_n$.

For k=1, 2, …, n

  $\sigma_k := \mathrm{sign}(a_{kk})\sqrt{\sum_{j=k}^{m} a_{jk}^2}$; {下三角部分第k列的2-范数 }

  If $\sigma_k = a_{kk}$ **then** {第k列对角线下方已经全为0 }

    Continue with next k ;

  End

  $\boldsymbol{v}_k := [0, \cdots, 0, a_{kk}, \cdots, a_{mk}]^T + \sigma_k \boldsymbol{e}_k$; {构造$\boldsymbol{H}_k$的向量}

  $\beta_k := \boldsymbol{v}_k^T \boldsymbol{v}_k$;

  For j=k, k+1, …, n {对剩余各列作Householder变换 }

    $\gamma_j := \boldsymbol{v}_k^T \boldsymbol{a}_j$;

    $\boldsymbol{a}_j := \boldsymbol{a}_j - (2\gamma_j/\beta_k)\boldsymbol{v}_k$; { $\boldsymbol{H}_k \boldsymbol{a}_j^{(k)} = \boldsymbol{a}_j^{(k)} - 2\left(\boldsymbol{v}_k^T \boldsymbol{a}_j^{(k)} / \boldsymbol{v}_k^T \boldsymbol{v}_k\right)\boldsymbol{v}_k$ }

  End

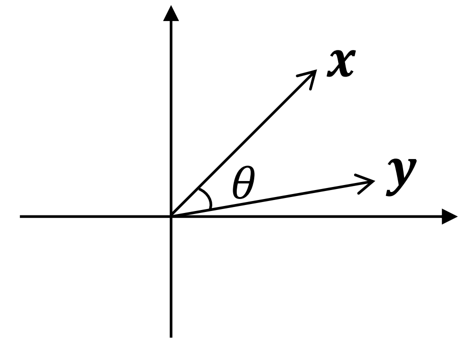
End

Matlab: [Q,R] = qr(A);
R = qr(A)

- 算法执行完, $\boldsymbol{A}$变成$\boldsymbol{R}$, 得到构造$\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_k$所需的$\boldsymbol{v}$向量
- 总的乘法次数: $(2n+1)\cdot m + (2n-1)\cdot m + \cdots + 3m \approx mn^2$
- 若考虑$\boldsymbol{v}$向量的稀疏性，总的乘法次数为$mn^2 - n^3/3$ ？思考

Givens旋转变换

- 二维平面旋转变换
  - 将向量$\boldsymbol{x}$顺时针旋转θ角度后得到$\boldsymbol{y}$



$$\boldsymbol{y}=\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\boldsymbol{x}$$

  - 定义二阶Givens矩阵为: $\boldsymbol{G}=\begin{bmatrix}c & s\\ -s & c\end{bmatrix}$, 其中$c=\cos\theta, s=\sin\theta$
  - $\boldsymbol{G}$是正交阵 (几何解释)
  - 对$\boldsymbol{x}$做Givens旋转得$\boldsymbol{Gx}$, 选择参数c, s, 可使$\boldsymbol{Gx}=\begin{bmatrix}*\\0\end{bmatrix}$
- n阶Givens旋转阵

  将二阶Givens阵嵌入n阶单位阵:

$$\underbrace{\begin{bmatrix}1&0&0&0&0\\0&\boldsymbol{c}&0&\boldsymbol{s}&0\\0&0&1&0&0\\0&-\boldsymbol{s}&0&\boldsymbol{c}&0\\0&0&0&0&1\end{bmatrix}}_{G}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\end{bmatrix}=\begin{bmatrix}x_1\\ \boldsymbol{\alpha}\\x_3\\ \boldsymbol{0}\\x_5\end{bmatrix}$$

$$c=\frac{x_2}{\sqrt{x_2^2+x_4^2}},\quad s=\frac{x_4}{\sqrt{x_2^2+x_4^2}}$$

  - $\boldsymbol{G}$仍是正交阵

Givens旋转变换

- **例:** 通过Givens旋转变换进行消元 $\boldsymbol{a}=\begin{bmatrix}2\\0\\1\\2\end{bmatrix} \longrightarrow \begin{bmatrix}*\\0\\0\\0\end{bmatrix}$

  解: 先针对第1, 3分量构造二阶旋转矩阵,

  $$\boldsymbol{G}_1'=\begin{bmatrix}c_1 & s_1\\ -s_1 & c_1\end{bmatrix} \qquad c_1=\frac{2}{\sqrt{2^2+1}}=2/\sqrt{5}, \qquad s_1=1/\sqrt{5}$$

  则: $$\boldsymbol{G}_1\boldsymbol{a}=\begin{bmatrix}c_1 & 0 & s_1 & 0\\ 0 & 1 & 0 & 0\\ -s_1 & 0 & c_1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}2\\0\\1\\2\end{bmatrix}=\begin{bmatrix}\sqrt{5}\\0\\0\\2\end{bmatrix}$$

  接着对第1, 4分量旋转

  求出$c_2=\sqrt{5}/3$, $s_2=2/3$,
  $$\boldsymbol{G}_2\boldsymbol{G}_1\boldsymbol{a}=\begin{bmatrix}c_2 & 0 & 0 & s_2\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ -s_2 & 0 & 0 & c_2\end{bmatrix}\begin{bmatrix}\sqrt{5}\\0\\0\\2\end{bmatrix}=\begin{bmatrix}3\\0\\0\\0\end{bmatrix}$$

- 每个Givens旋转阵用参数c, s刻画
- 每次旋转变换仅影响向量两个元素, 仅影响矩阵的两行
- 也可实现矩阵的QR分解, 适合于稀疏矩阵

QR迭代算法

Wenjian Yu

计算矩阵的所有特征值

- 两个问题
  - 什么样的矩阵易于求所有特征值?
  - 对矩阵做怎样的变换能保持特征值不变?

$$\boldsymbol{X}^{-1}\boldsymbol{A}\boldsymbol{X}$$
$$\boldsymbol{Q}^{T}\boldsymbol{A}\boldsymbol{Q}$$

- 思路：用正交相似变换化矩阵为三角阵或分块三角阵
- 收缩技术
  - 用幂法/反幂法已求出$\boldsymbol{A}$的一个特征值λ_1, 特征向量$\boldsymbol{x}_1$
  - 构造变换对$\boldsymbol{x}_1$消元:$\boldsymbol{H}\boldsymbol{x}_1=\sigma\boldsymbol{e}_1$, 再对$\boldsymbol{A}$做正交相似变换

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

$$\boldsymbol{HAH}^T$$

$$\boldsymbol{HAH}^T=\begin{bmatrix}\lambda_1 & \boldsymbol{r}_1^T\\ \boldsymbol{0} & \boldsymbol{A}_1\end{bmatrix}$$

计算矩阵的所有特征值

- 收缩技术的例子: 已知 $\boldsymbol{A}=\begin{bmatrix}3 & -1 & 1\\ 2 & 0 & 1\\ 1 & -1 & 2\end{bmatrix}$的一个特征值为 $\lambda_1=2$, 对应特征向量为$\boldsymbol{x}_1=[1,1,0]^T$, 求其他特征值

解：用Householder变换对x_1消元, 相应的$\sigma=\sqrt{2}=1.4142$

构造矩阵$\boldsymbol{H}$的$\boldsymbol{v}$向量为: $\boldsymbol{v}=\boldsymbol{x}_1+\sigma\boldsymbol{e}_1=\begin{bmatrix}2.4142\\ 1\\ 0\end{bmatrix}$

$$\Rightarrow \boldsymbol{H}=\begin{bmatrix}-0.7072 & -0.7072 & 0\\ -0.7072 & 0.7072 & 0\\ 0 & 0 & 1\end{bmatrix},\ \boldsymbol{HAH}^T=\begin{bmatrix}2 & 3 & -1.4142\\ 0 & 1 & 0\\ 0 & -1.4142 & 2\end{bmatrix}$$

$\boldsymbol{A}_1$

易知$\boldsymbol{A}_1$的特征值为1和2, 是$\boldsymbol{A}$的其他特征值

- 根据的$\boldsymbol{A}_1$特征向量，还可求$\boldsymbol{A}$的特征向量
- 不形成$\boldsymbol{H}$计算$\boldsymbol{HAH}^T$: 先算$\boldsymbol{B}=\boldsymbol{HA}^T$, 再$\boldsymbol{HAH}^T=\boldsymbol{HB}^T$

QR迭代算法

- 理论基础
  - 用一系列正交相似变换$\boldsymbol{B} = \boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{Q}$, 逐渐将矩阵$\boldsymbol{A}$化为上三角或对角块阶数很小的分块上三角矩阵
  - 设$\boldsymbol{A} \in \mathbb{R}^{n\times n}$，若$\boldsymbol{A}$为分块上三角阵，且对角块为1阶或2阶矩阵，则称$\boldsymbol{A}$为拟上三角阵，也叫实Schur型
  - **Th5.21**(Schur分解): $\forall \boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\exists$正交阵$\boldsymbol{Q}$使$\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{Q} = \boldsymbol{S}$, 其中$\boldsymbol{S}$为拟上三角阵, 其一阶对角块是$\boldsymbol{A}$的实特征值, 二阶对角块的特征值是$\boldsymbol{A}$的两个共轭复特征值
- QR算法 *(“二十世纪十大算法”之一)*

$$\begin{cases} \text{对}\boldsymbol{A}_k\text{做QR分解: } \boldsymbol{A}_k = \boldsymbol{Q}_k\boldsymbol{R}_k \\ \boldsymbol{A}_{k+1} = \boldsymbol{R}_k\boldsymbol{Q}_k, (k = 0,1,\cdots) \end{cases}$$ 生成正交相似矩阵序列$\{\boldsymbol{A}_k\}$

QR迭代算法

- **Th5.22**: 收敛定理
  - 矩阵$\boldsymbol{A}$满足一定条件, 则QR迭代所得矩阵序列$\{\boldsymbol{A}_k\}$基本收敛于拟上三角阵

拟上三角阵例子? “基本收敛”

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix} \text{ or } \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}$$

算法5.4: 计算矩阵特征值的QR算法
输入: $\boldsymbol{A}$; 输出: $\lambda_1, \cdots, \lambda_n$.
While $\boldsymbol{A}$不是拟上三角阵 **do**
　计算$\boldsymbol{A}$的QR分解, 得到矩阵$\boldsymbol{Q}$和$\boldsymbol{R}$;
　$\boldsymbol{A} := \boldsymbol{RQ}$;
End
根据$\boldsymbol{A}$的对角块求特征值$\lambda_1, \cdots, \lambda_n$.

课本pp.173的例5.13

正交相似变换是保对称性的, $\boldsymbol{A}$对称, 则$\boldsymbol{Q}^T\boldsymbol{A}\boldsymbol{Q}$也对称

对对称阵做QR迭代, 若收敛, 极限为块对角阵(块阶≤2)

实用的QR迭代技术

- QR迭代法的不足之处
  - 每步迭代的计算量很大 *将矩阵化简为上Hessenberg型*
  - 可能不收敛, 或收敛很慢 ***带原点位移的QR算法***
- 上Hessenberg约化
  - 上Hessenberg阵: $\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix}$
  - 将A正交相似约化为上Hessenberg阵, 再做**QR**分解(迭代), 用Givens旋转, 每步计算由$O(n^3)$降为$O(n^2)$

设$\boldsymbol{P}_1, \boldsymbol{P}_2, \cdots, \boldsymbol{P}_{n-1}$为Givens旋转阵, $\boldsymbol{A}_k$为上Hessenberg阵

$$\boldsymbol{P}_{n-1} \cdots \boldsymbol{P}_2 \boldsymbol{P}_1 \boldsymbol{A}_k = \boldsymbol{R}_k \Longrightarrow \boldsymbol{Q}_k = (\boldsymbol{P}_{n-1} \cdots \boldsymbol{P}_2 \boldsymbol{P}_1)^T$$

$$\boldsymbol{A}_{k+1} = \boldsymbol{R}_k \boldsymbol{Q}_k = \boldsymbol{P}_{n-1} \cdots \boldsymbol{P}_2 \boldsymbol{P}_1 \boldsymbol{A}_k (\boldsymbol{P}_{n-1} \cdots \boldsymbol{P}_2 \boldsymbol{P}_1)^T$$

$$= \boldsymbol{P}_{n-1} \cdots \boldsymbol{P}_2 \boldsymbol{P}_1 \boldsymbol{A}_k \boldsymbol{P}_1^T \boldsymbol{P}_2^T \cdots \boldsymbol{P}_{n-1}^T$$

思考: 证明$\boldsymbol{A}_{k+1}$仍是上Hessenberg阵

实用的QR迭代技术

- 将$\boldsymbol{A}$正交相似约化为上Hessengberg
  - 用Householder变换

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

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \boldsymbol{r}_1^T \\ \boldsymbol{c}_1 & \boldsymbol{A}_{22}^{(1)} \end{bmatrix} \xrightarrow{H_1 = \begin{bmatrix} 1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{H}_1' \end{bmatrix}} \boldsymbol{H_1A} = \begin{bmatrix} a_{11} & \boldsymbol{r}_1^T \\ \sigma_1 \boldsymbol{e}_1 & \boldsymbol{H}_1'\boldsymbol{A}_{22}^{(1)} \end{bmatrix} \longrightarrow \boldsymbol{H_1AH_1} = \begin{bmatrix} a_{11} & \boldsymbol{r}_1^T\boldsymbol{H}_1' \\ \sigma_1 \boldsymbol{e}_1 & \boldsymbol{H}_1'\boldsymbol{A}_{22}^{(1)}\boldsymbol{H}_1' \end{bmatrix}$$

$$\boldsymbol{A}^{(2)} = \boldsymbol{H}_1\boldsymbol{A}\boldsymbol{H_1} = \begin{bmatrix} \boldsymbol{A}_{11}^{(2)} & \boldsymbol{A}_{12}^{(2)} \\ \boldsymbol{A}_{21}^{(2)} & \boldsymbol{A}_{22}^{(2)} \end{bmatrix} \xrightarrow{\boldsymbol{H}_2 = \begin{bmatrix} \boldsymbol{I}_2 & \boldsymbol{O}^T \\ \boldsymbol{O} & \boldsymbol{H}_2' \end{bmatrix}} \boldsymbol{H}_2\boldsymbol{A}^{(2)}\boldsymbol{H}_2 = \left[\begin{array}{cc:cccc} a_{11} & a_{12}^{(2)} & a_{13}^{(3)} & \cdots & \cdots & a_{1n}^{(3)} \\ \sigma_1 & a_{22}^{(2)} & a_{23}^{(3)} & \cdots & \cdots & a_{2n}^{(3)} \\ \hdashline 0 & \sigma_2 & a_{33}^{(3)} & \cdots & \cdots & a_{3n}^{(3)} \\ 0 & 0 & \vdots & \ddots & & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & a_{n3}^{(3)} & \cdots & \cdots & a_{nn}^{(3)} \end{array}\right]$$

…… ……

  - 最终$\boldsymbol{H}_{n-2}\cdots\boldsymbol{H}_1\boldsymbol{A}\boldsymbol{H}_1\cdots\boldsymbol{H}_{n-2}$为上**Hessenberg**阵

实用的QR迭代技术

- 实对称矩阵$\boldsymbol{A}$

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

- 带原点位移的QR算法

位移因子

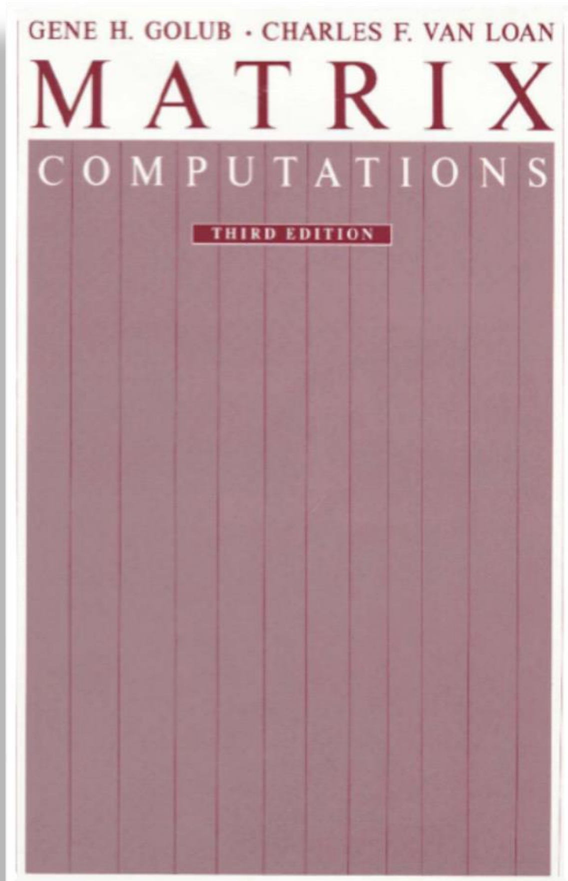
- □ 单位移技术 $\begin{cases} \boldsymbol{Q}_k\boldsymbol{R}_k = \boldsymbol{A}_k - s_k\boldsymbol{I} , & \text{(作QR分解)} \\ \boldsymbol{A}_{k+1} = \boldsymbol{R}_k\boldsymbol{Q}_k + s_k\boldsymbol{I} , & k = 1, 2, \cdots \end{cases}$

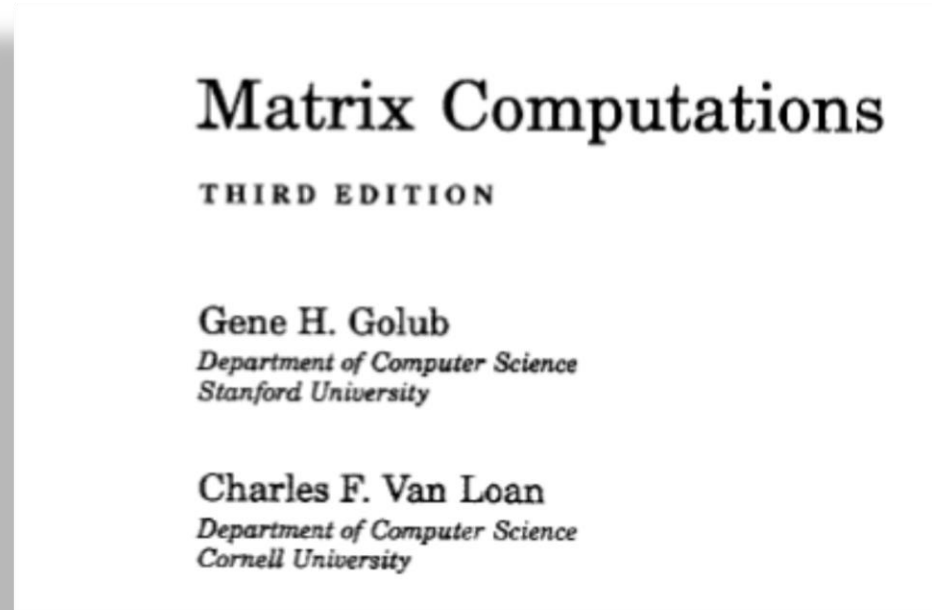
$$\boldsymbol{A}_{k+1} = \boldsymbol{R}_k\boldsymbol{Q}_k + s_k\boldsymbol{I}$$
$$= \boldsymbol{Q}_k^T(\boldsymbol{A}_k - s_k\boldsymbol{I})\boldsymbol{Q}_k + s_k\boldsymbol{I} = \boldsymbol{Q}_k^T\boldsymbol{A}_k\boldsymbol{Q}_k \Longrightarrow \{\boldsymbol{A}_k\}\text{仍两两正交相似}$$

- □ 简单的Rayleigh策略: 取$s_k = \boldsymbol{A}_k(n, n)$, 加速收敛
- □ 非对称阵有复特征值, 采用双位移
- □ **Matlab命令: qr, eig, eigs, planerot**

Matlab演示, eiggui

- G. H. Golub, C. F. Van Loan. *Matrix Computations* (3rd edition), Johns Hopkins University Press, 1996.



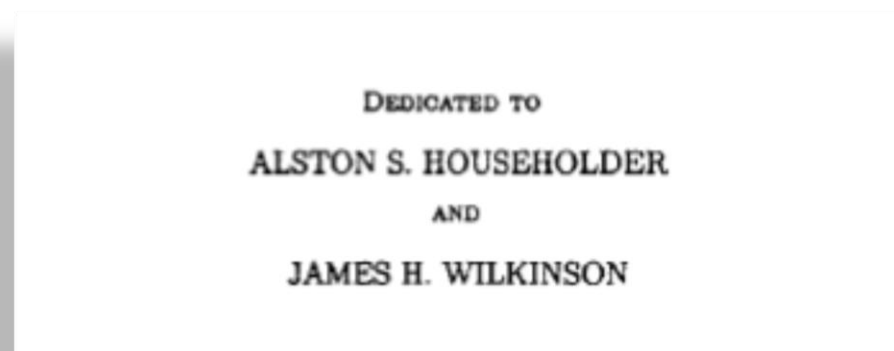

Matrix Computations

THIRD EDITION

Gene H. Golub
Department of Computer Science
Stanford University

Charles F. Van Loan
Department of Computer Science
Cornell University

- iii -


DEDICATED TO

ALSTON S. HOUSEHOLDER

AND

JAMES H. WILKINSON

- vi -

http://www.cs.cornell.edu/cv/

label: Hollywood