数值分析

第8章：矩阵的特征值(向量)计算

张亚楠[1]

苏州大学数学科学学院

May 14, 2020

[1]Email: ynzhang@suda.edu.cn

Contents

必要性

1. 特征值（向量）计算是数值分析中的重要内容，有广泛的应用：常微分方程组、矩阵乘幂、设计快速算法。。。
2. 了解矩阵的特征值和特征向量（特征对），也掌握了矩阵的几乎全部信息; 否则矩阵看起来只是一个“数表”

目标:

设 $A \in \boldsymbol{R}^{n\times n}$, 寻求 $\lambda \in \boldsymbol{C}$ 和非零向量 $x \in \boldsymbol{R}^n$, 使得:

$$Ax = \lambda x$$

想一想：特征值为何会出现复数？线性代数课程如何计算特征值？有哪些相应概念？ 该问题的计算复杂程度与求解线性方程组比起来，如何？

Theorem 1

若$Ax = \lambda x$ 则:

1. $\forall a \neq 0, \& \ a \in \mathbf{R}$, ax也是λ对应的特征向量
2. $A - \mu I$ 的特征值为 $\lambda - \mu$
3. A^{-1}的特征值为$\dfrac{1}{\lambda}$

这些定理可直接代入检验即可。通俗讲:

1. 特征向量的任何非零倍数仍是特征向量，特征向量只是一个方向的概念，无关长度
2. 任何非零向量均是单位矩阵的特征向量,
3. 若矩阵A可逆，则与其逆矩阵共享特征向量（好神奇！！）

Definition 2

设$A=(a_{ij})_{n\times n}$，令：

1) $r_i = \sum\limits_{j=1,j\neq i}^{n} |a_{ij}|,\ i=1,2,...,n$

2) 集合 $D_i = \left\{z \,\middle|\, |z-a_{ii}| \leq r_i,\ z \in \mathrm{C}\right\}$

所有 D_i 称之为 A的Gershgorin圆盘.

Theorem 3

设 $A=(a_{ij}) \in \boldsymbol{R}^{n\times n}$,

1) A 的每个特征值必属于下述某个圆盘之中

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

或者说，A的特征值都在复平面上n个圆盘的并集中.

2) 如果A有m个圆盘组成一个连通的并集S, 且 S 与余下 (n-m)个圆盘是分离的，则S包含A的m的特征值.

Theorem 4

设 $A = (a_{ij}) \in \boldsymbol{R}^{n\times n}$, A 的每个特征值必属于下述某个圆盘之中

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

Proof: 设 $Ax = \lambda x,\ x \neq 0\ \&\ x \in \mathbf{R}^n$

记 $|x_k| = \|x\|_\infty \neq 0$, 考虑$Ax = \lambda x$ 的第k个方程,

$$\sum_{j=1}^{n} a_{kj}x_j = a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n = \lambda x_k$$

上式改写为:

$$(\lambda - a_{kk})x_k = \sum_{j\neq k} a_{kj}x_j$$

进而

$$|\lambda - a_{kk}| \cdot |x_k| \leq \|x\|_\infty \cdot \sum_{j\neq k} |a_{kj}|$$

即

$$|\lambda - a_{kk}| \leq \sum_{j\neq k} |a_{kj}|$$

对称矩阵不但形式特殊，在实际问题中也经常出现；对称矩阵的特征值全是实数且特征向量的选择可以为彼此正交

$$A = S * \Lambda * S^{\mathrm{T}}, \quad S * S^{\mathrm{T}} = I$$

提示：（bar表示复数取共轭）

$$Ax = \lambda x \Rightarrow A\bar{x} = \bar{\lambda}\bar{x}$$

上述等式乘以转置x^{T}

$$x^{\mathrm{T}} A\bar{x} = x^{\mathrm{T}}\bar{\lambda}\bar{x} = \bar{\lambda}\|x\|^2$$

上式再取一次共轭（注意A是实、对称）并转置，得到

$$\bar{\lambda}\|x\|^2 = \lambda\|x\|^2 \quad \|x\| > 0; \quad \to \quad \lambda \in \boldsymbol{R}$$

假设

$$Ax = \lambda_1 x, \quad Ay = \lambda_2 y, \quad \lambda_1 \neq \lambda_2$$

证明两向量垂直

$$\left(x, y\right) = x^{\mathrm{T}} y = 0$$

提示：

$$\lambda_1\left(x, y\right) = \left(\lambda_1 x, y\right) = \left(Ax, y\right) = \left(x, Ay\right) = \left(x, \lambda_2 y\right) = \lambda_2\left(x, y\right) \to \left(x, y\right) = 0$$

Contents

选择题(写在正式介绍幂法和QR算法之前)

假如可以预先告知特征对的一个量，但是特征值和特征向量只能选一个，选哪个？WHY？

特征向量、向量、向量！！！

本小节只介绍幂法(乘幂方法). 并假定A有完备的特征向量组,

$$Ax_j = \lambda_j x_j, \quad 1 \le j \le n,$$

且

$$|\lambda_1| > |\lambda_2| \ge |\lambda_3| ... \ge |\lambda_n|$$

<u>该方法也适用于 $\lambda_1 = \lambda_2$, 但是不适用于$\lambda_1 = -\lambda_2$.</u>

回忆：线性方程组简单迭代法的收敛条件：迭代矩阵谱半径小于1；

Example 5

给定任意非零随机向量$v \in \boldsymbol{R}^2$

$$A = \begin{bmatrix} 1 & 2 \\ 4 & 8 \end{bmatrix}$$

想一想（猜一猜）：

$$Av, A^2v, ..., A^kv, ...$$

随着k变大，A^kv会变成什么样？2范数大小？ 已知$\lambda_A = 0$，9

WHY？

Matlab演示： power_meth_test.m

假定A有完备的特征向量组，

$$Ax_j = \lambda_j x_j, \quad 1 \le j \le n,$$

且

$$|\lambda_1| > |\lambda_2| \ge |\lambda_3| ... \ge |\lambda_n|$$

任给初始非零向量v_0，

$$v_0 = \sum_{j=1}^{n} \alpha_j x_j, \quad \alpha_1 \neq 0$$

因为$y_j = \alpha_j x_j$也是相应λ_j的特征向量，则上式可写为：

$$v_0 = \sum_{j=1}^{n} y_j = y_1 + y_2 + \cdots y_n$$

则

$$v_1 = Av_0 = Ay_1 + Ay_2 + ... + Ay_n = \boxed{\lambda_1} y_1 + \lambda_2 y_2 + ... + \lambda_n y_n$$

反复乘幂得到

$$v_k = A * v_{k-1} = A^k v_0 = \boxed{\lambda_1^k} y_1 + \lambda_2^k y_2 + ... + \lambda_n^k y_n$$

$$Ax_j = \lambda_j x_j, \quad 1 \le j \le n,$$

且

$$|\lambda_1| > |\lambda_2| \ge |\lambda_3| ... \ge |\lambda_n|$$

$$v_k = A^k v_0 = \lambda_1^k y_1 + ... + \lambda_n^k y_n = \lambda_1^k \left[y_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1} \right)^k y_j \right] \tag{1}$$

进而

$$\frac{v_k}{\underline{\underline{\lambda_1^k}}} = y_1 + \underbrace{\sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1} \right)^k y_j} = \underline{y_1} + \utilde{\varepsilon_k}$$

y_1是主特征向量，$v_k \approx \lambda_1^k * y_1$也是特征向量，不过可能数值比较大;

理论上

特征向量y_1或其同方向向量v_k找到了，如果顺带知道λ_1就更好了！！

$$v_k = A^k v_0 = \lambda_1^k \left[y_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1} \right)^k y_j \right] \tag{2}$$

得到(非正确表达)

$$\frac{v_{k+1}}{v_k} \quad \rightarrow \quad \frac{\lambda_1^{k+1} \left[y_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1} \right)^{k+1} y_j \right]}{\lambda_1^k \left[y_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1} \right)^k y_j \right]}$$

改正为

$$\frac{u^{\mathrm{T}} v_{k+1}}{u^{\mathrm{T}} v_k} = \frac{\cancel{\lambda_1^{k+1}} \left[u^{\mathrm{T}} y_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1} \right)^{k+1} u^{\mathrm{T}} y_j \right]}{\cancel{\lambda_1^k} \left[u^{\mathrm{T}} y_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1} \right)^k u^{\mathrm{T}} y_j \right]}$$

即

$$\frac{u^{\mathrm{T}} v_{k+1}}{u^{\mathrm{T}} v_k} = \lambda_1 \frac{\left[u^{\mathrm{T}} y_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1} \right)^{k+1} u^{\mathrm{T}} y_j \right]}{\left[u^{\mathrm{T}} y_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1} \right)^k u^{\mathrm{T}} y_j \right]} \rightarrow \lambda_1, \quad \text{as} \quad k \rightarrow \infty$$

两个问题：1. u^{T}如何选取？ 2. 收敛快慢和哪些因素有关？

$$\frac{u^{\mathrm{T}} v_{k+1}}{u^{\mathrm{T}} v_k} = \lambda_1 \frac{\left[u^{\mathrm{T}} y_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1}\right)^{k+1} u^{\mathrm{T}} y_j\right]}{\left[u^{\mathrm{T}} y_1 + \sum_{j=2}^{n} \left(\frac{\lambda_j}{\lambda_1}\right)^{k} u^{\mathrm{T}} y_j\right]} \to \lambda_1, \quad \text{as} \quad k \to \infty$$

$$|\lambda_1| > |\lambda_2| > |\lambda_3| > \ldots$$

$$\frac{u^{\mathrm{T}} v_{k+1}}{u^{\mathrm{T}} v_k} \approx \lambda_1 \frac{u^{\mathrm{T}} y_1 + (\lambda_2/\lambda_1)^{k+1} u^{\mathrm{T}} y_2}{u^{\mathrm{T}} y_1 + (\lambda_2/\lambda_1)^{k} u^{\mathrm{T}} y_2}$$

记$\alpha = (\lambda_2/\lambda_1)$，$U = \frac{u^{\mathrm{T}} y_2}{u^{\mathrm{T}} y_1}$, 注意到$|\alpha| < 1$,

$$\frac{u^{\mathrm{T}} v_{k+1}}{u^{\mathrm{T}} v_k} \approx \lambda_1 \frac{1+\alpha^{k+1} U}{1+\alpha^{k} U} \approx \lambda_1 * \left(1+\alpha^{k+1} U\right) * \left(1-\alpha^{k} U\right) = \lambda_1 * \left(1+\boldsymbol{O}(\alpha^k)\right)$$

收敛快慢与λ_2/λ_1 有关, 线性收敛！！u^{T}任意选取非零向量即可。

如果$A = A^{\mathrm{T}}$, 有无提高效率的方式？取$u^{\mathrm{T}} = v_k$试试看！（很自然的取法！！）

$$v_k^{\mathrm{T}} v_k = (A^k v_0)^{\mathrm{T}} (A^k v_0) = v_0^{\mathrm{T}} A^{2k} v_0$$

则

$$\begin{aligned}
\frac{v_k^{\mathrm{T}} v_{k+1}}{v_k^{\mathrm{T}} v_k} = \frac{v_0^{\mathrm{T}} A^{2k+1} v_0}{v_0^{\mathrm{T}} A^{2k} v_0} &= \frac{v_0^{\mathrm{T}} \cdot \left(\sum_{j=1}^{n} \lambda_j^{2k+1} y_j \right)}{v_0^{\mathrm{T}} \cdot \left(\sum_{j=1}^{n} \lambda_j^{2k} y_j \right)} \\
&= \frac{\lambda_1^{2k+1} \left(v_0^{\mathrm{T}} \cdot y_1 + \sum_{j=2}^{n} (\lambda_j/\lambda_1)^{2k+1} v_0^{\mathrm{T}} \cdot y_j \right)}{\lambda_1^{2k} \left(v_0^{\mathrm{T}} \cdot y_1 + \sum_{j=2}^{n} (\lambda_j/\lambda_1)^{2k} v_0^{\mathrm{T}} \cdot y_j \right)} \\
&\approx \lambda_1 \cdot \frac{1 + (\lambda_2/\lambda_1)^{2k+1} (v_0^{\mathrm{T}} \cdot y_2 / v_0^{\mathrm{T}} \cdot y_1)}{1 + (\lambda_2/\lambda_1)^{2k} (v_0^{\mathrm{T}} \cdot y_2 / v_0^{\mathrm{T}} \cdot y_1)} \\
&= \lambda_1 \left[1 + \boldsymbol{O}\left(\left(\frac{\lambda_2}{\lambda_1}\right)^{2k} \right) \right]
\end{aligned}$$

对称矩阵平方收敛！！对称矩阵A的Rayleigh 商 (一个n元函数)

$$R(u) = \frac{u^{\mathrm{T}} A u}{u^{\mathrm{T}} u}, \quad u \neq \vec{0}$$

A的Rayleigh 商 (一个n元函数)

$$R(u) = \frac{u^{\mathrm{T}}Au}{u^{\mathrm{T}}u}, \quad u \neq \vec{0}$$

当u 是A的特征向量， R(u) 即是特征值； 若取$\|u\| = 1$, 形式简化为

$$R(u) = u^{\mathrm{T}}Au = u^{\mathrm{T}} * \lambda * u = \lambda * u^{\mathrm{T}}u = \lambda * \|u\|^2 = \lambda, \quad \text{if u is eigenvector}$$

一句话：一旦得到特征向量，可得相应特征值！！

幂法两步走:

1. 反复乘幂得到特征向量
2. Rayleigh商得到相应特征值

理论上知道特征对的一个，可求出另一个。

$$Ax = \lambda x$$

计算复杂度

1. 已知λ, 求 x, 求解

$$\Big(A - \lambda I\Big)x = 0, \quad \text{Computational cost} \quad \boldsymbol{O}(n^3)$$

2. 已知x, 求 λ

$$\lambda = R(x) = \frac{x^{\mathrm{T}} * A * x}{x^{\mathrm{T}} * x}, \quad \boldsymbol{O}(n^2)$$

幂法即是：先特征向量x, 后特征值 λ

根据前文推导，给定一个方阵A, 非零向量v_0

$$\begin{cases} v_{k+1} = Av_k, & k \geq 0 \\ v_0 = \text{ initial guess} \end{cases}$$

得到主特征值和特征向量

$$\lambda_1 \leftarrow \frac{v_k^T v_{k+1}}{v_k^T v_k}, \qquad v_k$$

Example 6

$$\begin{bmatrix} 1 & 2 \\ 4 & 8 \end{bmatrix} \quad \begin{bmatrix} 0.3 & 0.2 \\ 0.7 & 0.8 \end{bmatrix} \quad \begin{bmatrix} 0.3 & 0.4 \\ 0.4 & 0.3 \end{bmatrix}$$

这三个矩阵都可以利用上述算法计算出有效特征值; 但也有明显瑕疵；WHY?

Matlab : power_meth_test.m

数值不稳定原因：

1. 若$\rho(A) > 1$, 则 $A^k \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \to \infty$
2. 若 $\rho(A) < 1$, 则 $A^k \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \to 0$

幂法算法(微调)

$$\begin{cases} v_{k+1} = Au_k, & u_{k+1} = \frac{v_{k+1}}{\|v_{k+1}\|}, \quad k \geq 0 \\ v_0 = \text{ initial guess }, & u_0 = \frac{v_0}{\|v_0\|} \end{cases}$$

$$\lambda = v_{k+1}^{\mathrm{T}} \cdot u_k = u_k^{\mathrm{T}} A u_k = R(u_k)$$

Example 7

用幂法计算特征对(教材248页: 教材方法迭代20次，$\lambda = 2.5365323$)

$$A = \begin{bmatrix} 1 & 1 & 1/2 \\ 1 & 1 & 1/4 \\ 1/2 & 1/4 & 2 \end{bmatrix} \quad 2.536525860417181$$

Matlab 演示 power_meth_test.m 加个10阶的随机对称矩阵试验

Contents

理论基础：可逆矩阵与其逆矩阵共享特征向量

$$Ax = \lambda x \Rightarrow A^{-1}Ax = \lambda A^{-1}x \Rightarrow A^{-1}x = \frac{1}{\lambda}x$$

想一想：若对A^{-1}作用幂法，得到的主特征值$\frac{1}{\lambda}$是什么? 其倒数λ和A 有何关系?

$$\begin{cases} v_{k+1} = A^{-1}u_k, & u_{k+1} = \frac{v_{k+1}}{\|v_{k+1}\|}, \quad k \geq 0 \\ v_0 = \text{initial guess}, & u_0 = \frac{v_0}{\|v_0\|} \end{cases}$$

输出

$$\frac{1}{\lambda} = u_k^{\mathrm{T}} * v_{k+1} = u_k^{\mathrm{T}} * A^{-1} * u_k$$

是A^{-1}的最大特征值；其倒数

$$\frac{1}{u_k^{\mathrm{T}} * v_{k+1}} \approx u_k^{\mathrm{T}} * A * u_k$$

是A的最小特征值(取瑞丽商即可，特征值和特征向量是配对的！)。

思考

之前反复强调：数值算法应当避免矩阵求逆！！现在如何改进反幂法的算法?

$$\begin{cases} v_{k+1} = A^{-1} u_k, & u_{k+1} = \frac{v_{k+1}}{\|v_{k+1}\|}, \quad k \geq 0 \\ v_0 = \text{ initial guess }, & u_0 = \frac{v_0}{\|v_0\|} \end{cases}$$

逆矩阵乘向量等价于解线性方程组

$$Av_{k+1} = u_k$$

因为迭代更新过程中需要反复求解，则应当预先对矩阵A进行LU或者cholesky分解; 例如

$$P * A = L * U$$

进而求解两个三角形方程

$$\begin{cases} Ly = P * u_k \\ Uv_{k+1} = y \end{cases}$$

如此处理，反幂法与幂法的计算量相当（不包含预处理LU分解部分）； 实际运算迭代每一步，反幂法是解两个三角形方程， 比矩阵乘向量略慢一些。如果Matlab版本较新，也可采用傻瓜式分解命令decomposition, Matlab会记录矩阵的信息，并根据矩阵 的信息选择分解为LU, LDL, cholesky, qr

。。。

Matlab 举例(之前例子)： inv_power_eig.m

Tips： 反幂法收敛很快，迭代步数极少，为了写代码简单， 不预先对A分解, 直接解方程也可接受！不接受对A求逆 ！！！

思考：刚刚的例子，为何反幂法效果明显优于幂法？反幂法的收敛速度有哪些因素相关？

$$\frac{\lambda_n}{\lambda_{n-1}} \quad \text{绝对值越小越好}$$

Example 8

通过幂法的3、5迭代，已知矩阵的主特征值大约2.536

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

能否通过反幂法加速，尽快得到高精度的、靠近2.536特征值?

更一般问题

假如已知矩阵A的某个特征值的近似值μ, 现在需要计算μ附近的高精度特征值！！如何有效利用近似值?

问题

假如已知矩阵A的某个特征值的近似值μ, 现在需要计算μ附近的高精度特征值！！如何有效利用近似值？可行！理论基础: 单位矩阵的特征值任意可选！！！

构造

$$B = A - \mu I$$

B的最小特征值以及对应特征向量可否计算？反幂法！！若计算出 B的最小特征值 λ_B 和相应的特征向量u, 则 Rayleigh商

$$\lambda_A = u^{\mathrm{T}} A u$$

即是靠近 μ的特征值。

Example 9

利用该方法计算 $\mu = 2.5$ 附近的特征值和特征向量

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

若近似值稍差，$\mu = 2.3$, 2.2 , 效果有无影响？若有，如何改进？

想一想：

既然反幂法收敛快慢与μ的近似程度有关，能否每算一次也顺带更新一下μ 的值？岂不更快？

假设σ_0 是矩阵A的某个近似特征值(不需要特别精确)，模1向量u_0; 按照以下方式进行幂法迭代

$$\begin{cases} v_{k+1} = \left(A - \sigma_k I\right)^{-1} * u_k \\ u_{k+1} = \frac{v_{k+1}}{\|v_{k+1}\|} \\ \sigma_{k+1} = u_{k+1}^{\mathrm{T}} * A * u_{k+1} \end{cases}$$

该方法收敛很快，对称矩阵是立方收敛，一般只要迭代几步即可。

若σ_k 接近特征值时，

$$\left(A - \sigma_k I\right)$$

接近奇异，系数矩阵条件数大，数值求解不准确；编程时应当做好判断，和迭代停止标准。

不要被"特征值计算"标题误导，我们要：特征向量、特征向量、特征向量！！！

1. 幂法：计算最大（绝对值）特征值（向量）基本方法
2. 幂法应用（反幂法）：计算最小特征值（向量）基于数学理论的幂法推广
3. 反幂法应用：已知粗略近似值μ, 快速计算靠近μ的精确特征值瑞丽商反幂法

想一想：之前有哪些知识点现在可以完善?

1. 矩阵的谱半径、谱条件数
2. 多项式求根是病态敏感问题，假如利用二分法已经得到了某个近似根，现在能否利用反幂法提高求根的准确度和稳定性?

1. 优点：简单，大型稀疏矩阵也可用；
2. 缺点：只能计算一个最大或者最小的特征值，其它的特征值需要知道其粗略近似值?

幂法：简单、有效 ！！！ 变化多端：反幂法， Rayleigh商反幂法。。。

Contents

由线性代数知识，线性无关的一组向量可以选为线性空间的一组基， 而通过Gram-Schmidt正交化方法后可以得到该空间的一组标准正交基。

给定非奇异矩阵$A \in \boldsymbol{R}^{n\times n}$, $rank(A) = n$, 通过Gram-Schmidt正交化步骤可得正交矩阵 Q, 即Q的各个列向量彼此正交。从A 到 Q 是如何变化的? 二者有无数学关系式之类?

Example 10

若满秩方阵A, Q有相同的列空间，则必存在非奇异矩阵B, 使得

$$A = Q * B$$

Hint: 检验$B = Q^{-1} * A$非奇异即可。

对A的列向量进行正交化的过程也是QR分解过程， 即存在正交矩阵 $Q \in \boldsymbol{R}^{n\times n}$, 及上三角矩阵$R$,满足

$$A = Q * R, \quad Q^{\mathrm{T}} * Q = I.$$

思考: R 为何是上三角矩阵? 这个过程如何实现?

记

$$A = \begin{bmatrix} | & | & \cdots & | \\ a_1 & a_2 & \cdots & a_n \\ | & | & \cdots & | \end{bmatrix}, \quad Q = \begin{bmatrix} | & | & \cdots & | \\ q_1 & q_2 & \cdots & q_n \\ | & | & \cdots & | \end{bmatrix}$$

算法实现

1. 取$q_1 = a_1/\|a_1\|$
2. a_k依次往q_j，$1 \le j \le k-1$ 做投影，得到q_k

$$q_k = a_k - \sum_{j=1}^{k-1} \left(a_k, q_j\right) * q_j$$

Matlab 演示：myqr1.m 结论：

1. 直接一次性投影，稳定性差
2. 依次投影，效果略好，也不理想，Hilbert矩阵为例
3. 可行性没问题，有更稳定的方法：正交变换

from SIAM News, Volume 33, Number 4

The Best of the 20th Century: Editors Name Top 10 Algorithms

By Barry A. Cipra

The 10 Algorithms with the Greatest Influence on the Development and Practice of Science and Engineering in the 20th Century

1. Metropolis Algorithm for Monte Carlo
2. Simplex Method for Linear Programming
3. Krylov Subspace Iteration Methods
4. The Decompositional Approach to Matrix Computations: LU, LDL。。
5. The Fortran Optimizing Compiler
6. QR Algorithm for Computing Eigenvalues
7. Quicksort Algorithm for Sorting
8. Fast Fourier Transform
9. Integer Relation Detection
10. Fast Multipole Method

给定非奇异矩阵 $A = A_0$,按照如下迭代格式

$$\begin{cases} A_k = Q_k \cdot R_k \\ A_{k+1} = R_k \cdot Q_k \end{cases}$$

其中Q是正交矩阵，R是上三角矩阵

可得到相似矩阵序列 A_k $(k = 0, 1, 2, ...)$, 即

$$A_{k+1} = Q_k^{\mathrm{T}} \cdot A_k \cdot Q_k = \left(Q_k^{\mathrm{T}} \cdots Q_0^{\mathrm{T}}\right) \cdot A_0 \cdot \left(Q_0 \cdots Q_k\right) = \hat{Q_k}^{\mathrm{T}} * A_0 * \hat{Q_k}$$

在一定假设条件下，A_k 收敛到上三角矩阵，其对角线即是特征值。若A是对称矩阵，则收敛到对角矩阵。

问题:

WHY? 为什么会收敛? 难以置信！！

$$\begin{cases} A_k = Q_k \cdot R_k \\ A_{k+1} = R_k \cdot Q_k \end{cases}$$

$$A_{k+1} = Q_k^{\mathrm{T}} \cdot A_k \cdot Q_k = \left(Q_k^{\mathrm{T}} \cdots Q_0^{\mathrm{T}}\right) \cdot A_0 \cdot \left(Q_0 \cdots Q_k\right) = \hat{Q_k}^{\mathrm{T}} * A_0 * \hat{Q_k}$$

记录QR迭代产生的 Q_k 和 R_k

$$\begin{array}{ll} \boldsymbol{A}_0 = \boldsymbol{Q}_0 \boldsymbol{R}_0 & \boldsymbol{A}_1 = \boldsymbol{R}_0 \boldsymbol{Q}_0 \\ \boldsymbol{A}_1 = \boldsymbol{Q}_1 \boldsymbol{R}_1 & \boldsymbol{A}_2 = \boldsymbol{R}_1 \boldsymbol{Q}_1 \\ \boldsymbol{A}_2 = \boldsymbol{Q}_2 \boldsymbol{R}_2 & \boldsymbol{A}_3 = \boldsymbol{R}_2 \boldsymbol{Q}_2 \end{array}$$

检验:

$$\begin{aligned} \left(Q_0 Q_1 Q_2\right) * \left(R_2 R_1 R_0\right) &= Q_0 Q_1 A_2 R_1 R_0 \\ &= Q_0 Q_1 R_1 Q_1 R_1 R_0 \\ &= Q_0 R_0 Q_0 R_0 Q_0 R_0 \\ &= A^3 := \hat{Q}_2 \hat{R}_2 \end{aligned}$$

归纳法可得

$$A^{k+1} = \hat{Q}_k \hat{R}_k$$

两个重要等式 $A_{k+1} = \hat{Q}_k^{\mathrm{T}} A_0 \hat{Q}_k$， $A^{k+1} = \hat{Q}_k \hat{R}_k$

$$A_{k+1} = \hat{Q}_k^{\mathrm{T}} A_0 \hat{Q}_k, \quad A^{k+1} = \hat{Q}_k \hat{R}_k$$

假设A有完备的特征向量空间，且特征值彼此分离 $|\lambda_1| > |\lambda_2| > ... > |\lambda_n| > 0$, 为了简化推导和解释，进一步假设 $A = A^{\mathrm{T}}$. 则

$$A = S\Lambda S^{\mathrm{T}}, \quad S * S^{\mathrm{T}} = I. \quad S^{\mathrm{T}} = L * U$$

$$A^{k+1} = S * \Lambda^{k+1} * S^{\mathrm{T}} = S * \Lambda^{k+1} * L * U = S * \Lambda^{k+1} * L\Lambda^{-(k+1)} * \underline{\underline{\Lambda^{(k+1)} * U}}$$

现在说明三个事实

1. $\Lambda^{k+1} * L\Lambda^{-(k+1)} \to I$, 检验 $\lambda_i^{k+1} * l_{ij} * \lambda_j^{-(k+1)} = l_{ij} * (\lambda_i/\lambda_j)^{k+1} \to 0, \quad \text{as } i > j$
2. $\Lambda^{k+1} * L\Lambda^{-(k+1)} * \underline{\underline{\Lambda^{(k+1)} * U}} := bigU$是上三角
3. 根据QR分解的唯一性(R的对角元符号确定的前提下);

$$A^{k+1} = S * bigU = \hat{Q}_k \hat{R}_k \quad \to \quad S = \hat{Q}_k$$

结论:

$$A_{k+1} = \hat{Q}_k^{\mathrm{T}} * A_0 * \hat{Q}_k \approx S^{\mathrm{T}} * A * S$$

是对角矩阵。再次提醒：此处只是粗略的解释，不是严格数学证明！！

给定$A_0 = A$, 基本QR算法

$$\begin{cases} A_k = Q_k \cdot R_k \\ A_{k+1} = R_k \cdot Q_k \end{cases}$$

计算的难点和复杂度如何？ QR 分解如何实现？

1. Gram-schimdt方法理论可行，但是数值不稳定！
2. 哪些矩阵或者变换数值稳定呢？正交矩阵、变换！

换个观点：

正交化的过程是 A 到 Q；现在给一个A，能否通过正交变换把A变成上三角R 呢？

Householder 变换！！

Example 11

对任意模 1向量$w \in \boldsymbol{R}^n$, 检验 反射矩阵$H = I - 2ww^{\mathrm{T}}$,满足

$$H = H^{\mathrm{T}}, \quad H * H = I.$$

即H为对称正交矩阵。

Hint:

$$\left(I - 2ww^{\mathrm{T}}\right) * \left(I - 2ww^{\mathrm{T}}\right) = I - 4 * ww^{\mathrm{T}} + 4 * w(w^{\mathrm{T}}w)w^{\mathrm{T}} = I$$

记超平面（过圆点以w为法向）

$$S = \{x \mid w^{\mathrm{T}}x = 0, \quad \forall x \in \boldsymbol{R}^n\},$$

知S是$\boldsymbol{R}^n$子空间，维数或者rank为$(n-1)$. 则任意向量$z \in \boldsymbol{R}^n$可以在子空间S和$span\{w\}$做正交分解。

$$z = x + y, \quad x \in S, \ \& \ y = \alpha \cdot w, \quad \alpha \in \boldsymbol{R}$$

进而

$$Hz = Hx + Hy = x + (I - 2ww^{\mathrm{T}})y = x + y - 2ww^{\mathrm{T}} * \alpha w = x - y$$

几何上，上述反射矩阵可以看作是以S为镜面的反射。

Theorem 12

给定两个模长相等的向量$x, y \in \boldsymbol{R}^n$, 则存在反射变换$H$, 使得 $Hx = y$.

提示：取$w = \frac{x-y}{\|x-y\|}$, 检验$H = I - 2ww^{\mathrm{T}}$满足要求。

利用正交变换进行A = QR分解，数值稳定性良好。 考虑向量$\forall x \in \boldsymbol{R}^n$, & $\|x\| = \rho$ 可通过Householder变换P得到

$$Px = [\pm\rho, 0, ..., 0]^{\mathrm{T}}$$

给定方阵，可通过反射变换实现如下效果

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{P_1 = I - 2w * w^{\mathrm{T}}} \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}$$

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{P_1 = I - 2w * w^{\mathrm{T}}} \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}$$

对小一阶规模的矩阵重复上述过程

$$\begin{bmatrix} I & 0 \\ 0 & P_2 \end{bmatrix} \cdot \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & P_2 \cdot A_{22} \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}$$

注意到$P_1, diag(I, P_2), \ldots$是正交矩阵，其乘积也是正交矩阵，则上述过程

$$P_{n-1} P_2 P_1 \cdot A = R, \quad P * A = R, \quad A = P^{\mathrm{T}} \cdot R = QR$$

$$P_{n-1}P_2P_1 \cdot A = R$$

算法： 给定初始n阶矩阵矩阵$P = I$，$R = A$;

1. 对 k = 1 : n, 需要处理的矩阵规模为 $(n-k+1)$,. 对应原始矩阵A的指标($j1 = k : n$) 计算

$$v = A(j1, k), \quad \rho = \|v\|, \quad w = v - \rho * e_1,\ w = w/\|w\|$$

反射矩阵

$$P_k = I - 2w * w^{\mathrm{T}}$$

2. 反射矩阵作用到$R(j1, j1)$上, 同时也作用到$P = P_{k-1} * \ldots * P_1$;

$$R(j1, j1) = R(j1, j1) - 2 * w * \left(w^{\mathrm{T}} * R(j1, j1)\right)$$

$$P(j1, :) = P(j1, :) - 2 * w * \left(w^{\mathrm{T}} * P(j1, :)\right)$$

上述过程的每一步计算量主要在圆括号，整体计算量是 $\boldsymbol{O}(n^3)$.

Matlab : myhousQR.m 对比下稳定性；并和Matlab命令qr对比：计算结果一样稳定！

有了QR分解的稳定算法： householder变换

给定矩阵$A = A_0$, 基本QR算法可按照如下进行

1. $A_k = Q_k * R_k$
2. $A_{k+1} = R_k * Q_k$
3. $\hat{Q}_k = Q_0 * Q_1 * \ldots * Q_k$

1. 当$A = A^{\mathrm{T}}$, 一步到位(特征对)
$$A_k \to \Lambda = \hat{Q}_k^{\mathrm{T}} * A_0 * \hat{Q}_k$$
2. 当$A \neq A^{\mathrm{T}}$, 只得到特征值上三角(A_{k+1})
$$A_0 = \hat{Q}_k * \underset{\sim\sim\sim}{A_{k+1}} * \hat{Q}_k^{\mathrm{T}}$$

如果矩阵非对称，则特征向量还需要进一步计算。 Matlab : myqrTest.m

Example 13

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

1. 对称矩阵比较方便：直接得到特征对! 好消息！
2. 非对称矩阵最终得到上三角形式：仍然需要进一步计算特征向量! 差强人意！
3. 如果矩阵是稠密的，对称不对称，计算量都很大！！！需要改进！

逐一解决后两个问题

三角形矩阵的特征对?

1. 三角形矩阵特征值在对角线上
2. 找方程 $\boxed{\left(\lambda I - A\right)x = 0}$ 的非零解，得到特征向量

稠密矩阵太复杂?

1. 一般矩阵变成Hessenberg矩阵
2. 对称矩阵变成三对角

Example 14

给定如下形式的矩阵

$$A = \begin{bmatrix} a & + & + \\ 0 & b & + \\ 0 & 0 & c \end{bmatrix}$$

由特征多项式知道 特征值即是对角元素

$$\lambda_A = a,\ b,\ c$$

如何求出每个特征值对应的特征向量？代价多大？

$$B = A - c * I = \begin{bmatrix} * & + & + \\ 0 & * & + \\ 0 & 0 & 0 \end{bmatrix}$$

求出B的非零解，也即是零空间Null Space的一组基。一般情况下一个特征值对应一个，具体可按照如下方式倒着依次计算（matlab: eigTriangle.m）

$$x(1:n-1) = \Big(B(1:n-1, 1:n-1)\Big)^{-1} * \Big(-B(1:n-1, n)\Big)$$

即为所求。$x(n) = 1$; 可单位化处理。

tips： Matlab有命令rref可用于计算Null space. 此处是三角形矩阵，计算量为 $n^2/2$; 总计算量$n^3/6$。

Contents

一般矩阵进行QR分解，太过复杂；先从一类简单的、特殊的、但又普遍适用、的矩阵出发。

Hessenberg矩阵是近似三角矩阵，其中下三角部分只有次对角线非零。

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

回忆：Hessenberg矩阵是三角形和上三角的结合；若进行LU分解，运算量多大?

Hessenberg矩阵进行QR分解，计算复杂度可降低为$\boldsymbol{O}(n^2)$. 如下所示，反射变换作用第一列时，定义的w 只有两个非零元素。 每次反射变换只需要处理矩阵的两行即可。

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} \xrightarrow{H = I - 2w * w^T} \begin{bmatrix} * & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}$$

每次乘除法次数为$2n$，$k = 1 \to n$, 总的工作量为$\boldsymbol{O}(n^2)$.

一般矩阵的QR分解运算量n^3, Hessenberg矩阵是n^2, 次数的降低是大大降低！！

Matlab 举例： myHessQRDecomp.m 并与Matlab qr 对比

Hessenberg矩阵

$$H=\begin{bmatrix}\times&\times&\times&\times\\\times&\times&\times&\times\\0&\times&\times&\times\\0&0&\times&\times\end{bmatrix}=Q*R=\begin{bmatrix}\times&\times&\times&\times\\\times&\times&\times&\times\\0&\times&\times&\times\\0&0&\times&\times\end{bmatrix}*\begin{bmatrix}\times&\times&\times&\times\\0&\times&\times&\times\\0&0&\times&\times\\0&0&0&\times\end{bmatrix}$$

R是上三角； Q 是正交矩阵；Q的形状如何？ R*Q 形状如何？

简单而重要

1. Q 也是Hessenberg矩阵；
2. R*Q 也是Hessenberg矩阵

按照上图检验即可。

对Hessenberg矩阵进行QR算法

1. $A_k = Q_k * R_k$
2. $A_{k+1} = R_k * Q_k$

QR 分解似乎计算量降低了很多！！ R*Q怎么样？

对Hessenberg矩阵 A 进行QR算法

1. $A_k = Q_k * R_k$
2. $A_{k+1} = R_k * Q_k$

每步迭代 QR 分解计算量$\boldsymbol{O}(n^2)$ ！！ R*Q本质上是对R 做列变换； 不要直接矩阵乘法，用列变换同样可将这一步降低到$\boldsymbol{O}(n^2)$;

至此对Hessenberg的QR算法，每步迭代的复杂度降低到$\boldsymbol{O}(n^2)$, 可以接受！！

Example 15

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 4 & 4 & 4 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 2 & 3 \end{bmatrix}$$

基本QR算法计算上Hessenberg矩阵的特征值！不要忘记 特征向量！！！, 三角形矩阵的特征向量

Matlab演示：myBasicQR_hess.m

实用中，针对Hessenberg矩阵多采用带 位移的QR算法(如: Rayleigh quotient shift， Wilkinson shift)，

$$\begin{cases} A_k - \sigma * I = Q_k * R_k \\ A_{k+1} = R_k * Q_k + \sigma * I \end{cases}$$

记$\hat{Q}_k = Q_0 * Q_1 * \ldots * Q_k$ Rayleigh商加速取 $\sigma = H_{nn}$, (如下H 是QR迭代中的某一步A_k)

$$H = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \alpha & \lambda_n \end{bmatrix}$$

由特征多项式知道, 当$\alpha = 0$ 时，λ_n是H的一个特征值， 实际计算时，当$\alpha < \epsilon$, λ_n 即所求特征值。 进而目标矩阵变为$(n-1)$阶的。

Matlab myHessQRshift.m 对比不带位移的算法，收敛更快！！

问题

为何带位移的处理，α很快趋于0？类似于瑞丽商反幂法！！

提示：假设$\sigma = H_{nn} = \lambda_n$ 是(近似)特征向量，则针对(近似)奇异矩阵$\left(A_k - \sigma * I\right)$的 QR分解得到的$Q$，$R$ 形状如何？ $R_{nn} = \approx 0$;, 进而新的A_{k+1}

$$H_{n,n-1} = R(n,:) * Q(:, n-1) = \approx 0$$

$$A_{new} = R * Q = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \varepsilon \approx 0 \end{bmatrix} * \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \alpha \approx 0 & \lambda_n \end{bmatrix}$$

待求解矩阵阶数降低一个。逐次降低，直至2阶、1阶；完毕。

为了保留完整的矩阵信息，删去的最后一列要保留，为最后计算特征向量准备。

1. 依次降低矩阵的规模: 逐个击破
2. 注意保留特征向量的信息：虚拟n阶上三角矩阵 T（同基本QR算法）

$$H_n = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & 0 & \lambda_n \end{bmatrix} = \begin{bmatrix} H_{n-1} & \boxed{\mid} \\ 0 & \lambda_n \end{bmatrix}$$

不要忘记虚拟的n阶的上三角矩阵T；保留H的最后一列 $\boxed{\mid}$ 并参与后面的qr迭代！！！

$$Q_{n-1}^{\mathrm{T}} H_{n-1} Q_{n-1} \rightarrow \begin{bmatrix} Q_{n-1}^{\mathrm{T}} & \\ & I \end{bmatrix} \begin{bmatrix} H_{n-1} & \boxed{\mid} \\ 0 & \lambda_n \end{bmatrix} \begin{bmatrix} Q_{n-1} & \\ & I \end{bmatrix} = \begin{bmatrix} Q_{n-1}^{\mathrm{T}} H_{n-1} Q_{n-1} & Q_{n-1}^{\mathrm{T}} \boxed{\mid} \\ 0 & \lambda_n \end{bmatrix}$$

同时保留正交变换中的$\hat{Q} = Q_0 Q_1 \ldots Q_k$, 最终输出的结果：

$$A_0 = \hat{Q} * T * \hat{Q}^{\mathrm{T}}$$

重复以上过程直至$n = 2$, 停止计算，求出2阶矩阵的特征对即可。

1. Matlab : myHessQRshift.m （只要特征值，忽略其它）
2. myHessQRshift211.m (保留一切：正交阵Q，上三角 T)

对Hessnberg矩阵进行shift- QR算法，得到

$$A = A_0 = \hat{Q} * T * \hat{Q}^{\mathrm{T}} = \hat{Q} \begin{bmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \end{bmatrix} \hat{Q}^{\mathrm{T}}$$

对三角形矩阵T 计算其特征对，

$$T * S = S * \Lambda, \quad \Lambda = diag(T)$$

代入

$$T = \hat{Q}^{\mathrm{T}} * A * \hat{Q} \rightarrow \hat{Q}^{\mathrm{T}} * A * \hat{Q} S = S \Lambda$$

得到

$$A * (\hat{Q} S) = (\hat{Q} S) * \Lambda$$

至此，对Hessenberg矩阵，有了比较理想的算法：Rayleigh quotient shift QR method !!

一般的矩阵如何处理？对称矩阵如何处理？

Contents

Theorem 16

给定矩阵A, 存在正交矩阵P, 上Hessenberg矩阵H, 使得

$$A = P * H * P^{\mathrm{T}}$$

一般矩阵可通过正交变换变为Hessenberg 矩阵； 对称矩阵可变为对称三对角.

假设上述结论正确，给定稠密矩阵A，计算其所有特征对，分几步走?

1. 变换一般矩阵为 Hessenberg矩阵

$$A = P * H * P^{\mathrm{T}}$$

2. shift QR算法变H 为上三角

$$H = Q * T * Q^{\mathrm{T}}$$

3. 计算上三角的特征向量

$$T * S = S * \Lambda$$

完美！！！ $\boxed{A * (PQS) = (PQS) * \Lambda}$

Houserholder变换可以用来对矩阵进行相似变换

$$\left[\begin{array}{cccc} \times & \times & \times & \times \\ \hline \color{red}{\times} & \times & \times & \times \\ \color{red}{\times} & \times & \times & \times \\ \color{red}{\times} & \times & \times & \times \end{array}\right] \xrightarrow[P_1 \cdot A]{\text{Householder trans}} \left[\begin{array}{cccc} \times & \times & \times & \times \\ \hline \color{red}{\times} & \times & \times & \times \\ \color{red}{0} & \times & \times & \times \\ \color{red}{0} & \times & \times & \times \end{array}\right] \xrightarrow{(P_1 A) \cdot P_1} \left[\begin{array}{c|ccc} \times & * & * & * \\ \hline \color{red}{\times} & * & * & * \\ \color{red}{0} & * & * & * \\ \color{red}{0} & * & * & * \end{array}\right]$$

对小规模的$(n-1)$阶矩阵 重复以上过程可得

$$A \rightarrow P_1 A P_1 \rightarrow P_2 P_1 A P_1 P_2 \rightarrow \cdots \rightarrow \text{Hessenberg form}$$

注意到

$$P_k = P_k^{\mathrm{T}} = P_k^{-1}$$

得到

$$\boxed{H = P^{\mathrm{T}} * A * P, \qquad P = P_1 * P_2 \ldots P_{n-2}}$$

检验整个过程的计算量是$\boldsymbol{O}(n^3)$. 如果A本身是对称矩阵，则最终得到的Hessenberg矩阵为三对角。

特征对计算三步走：A$\rightarrow H \rightarrow T \rightarrow eigenpairs$!!!

matlab: myhess1.m ; 与命令 hess 对比！

小结

若已知某个特征值的近似值（或者只要最大、最小特征值）

1. 反幂法（ 即使只要最大特征值、也可选择Rayleish商反幂法）
2. 幂法、反幂法只涉及矩阵乘法（或者解线性方程组），大型稀疏矩阵也适用

若对矩阵A特征值的信息未知、且矩阵规模不大；分三步走

1. $A \to H$,
$$A = P * H * P^{\mathrm{T}}$$
2. $H \to T$,
$$H = Q * T * Q^{\mathrm{T}}$$
3. 求解三角形矩阵T 的特征向量

Tips : 若矩阵是对称的，则上述过程将大大简化，读者可推导并编程检验！