

第五章 矩阵计算问题

第 16 讲 特征值问题

黄定江

DaSE @ ECNU
djhuang@dase.ecnu.edu.cn

- ① 16.1 矩阵特征值分布范围的估计
- ② 16.2 幂法和反幂法
- ③ 16.3 幂法的应用

- ① 16.1 矩阵特征值分布范围的估计
- ② 16.2 幂法和反幂法
- ③ 16.3 幂法的应用

矩阵特征值的应用

工程中许多实际问题都归结为求某些矩阵的特征值和特征向量：

- 物理上：振动问题、稳定性问题
- 数据科学中：网页链接分析问题（Google PageRank）
- 机器学习中：如流形学习、谱聚类、线性判别分析、主成分分析等

16.1 矩阵特征值分布范围的估计

本讲我们首先讨论矩阵特征值的分布范围或它们的界，其在理论上或者实际中都有重要应用，比如在敏感性分析和迭代法计算中都需要对矩阵的特征值分布范围的了解：

- 计算矩阵的 2-条件数

$$\text{cond}(\mathbf{A})_2 = \sqrt{\frac{\lambda_{\max}(\mathbf{A}^T \mathbf{A})}{\lambda_{\min}(\mathbf{A}^T \mathbf{A})}}$$

- 考察一阶定常迭代法 $\mathbf{x}^{(k+1)} = \mathbf{B}\mathbf{x}^{(k)} + \mathbf{f}$ 的收敛性、收敛速度，收敛的判据是谱半径 $\rho(\mathbf{B}) = \max_{1 \leq j \leq n} |\lambda_j(\mathbf{B})| < 1$ ，收敛速度为 $R = -\log_{10} \rho(\mathbf{B})$

前面说明过谱半径的大小不超过任何一种算子范数，即

$$\rho(\mathbf{A}) \leq \|\mathbf{A}\|$$

这是关于特征值上界的一个重要结论。

Gerschgorin 圆盘

下面再介绍有关特征值界的另外一个重要结论。为了细致描述 n 阶矩阵的特征值在复平面的分布范围，首先引进 Gerschgorin 圆盘 (简称盖尔圆或盖氏圆)。本讲我们假设矩阵都是复矩阵。

定义 1

设 $A = (a_{kj}) \in \mathbb{C}^{n \times n}$ ，令 $R_k = \sum_{j=1, j \neq k}^n |a_{kj}|$ ，则称集合 $D_k = \{z | z \in \mathbb{C} : |z - a_{kk}| \leq R_k\}$ ， $k = 1, 2, \dots, n$ 为在复平面内以 a_{kk} 为圆心、 R_k 为半径的圆盘，称为 A 的第 k 个盖氏圆。

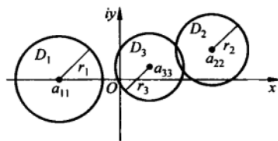


图 1: 复坐标平面，以及 3×3 复矩阵 A 的盖氏圆

圆盘定理

定理 1

设 $A = (a_{kj}) \in \mathbb{C}^{n \times n}$, 则:

(1) A 的每一个特征值必属于 A 的盖氏圆之中, 即对任一特征值 λ 必定存在 $k(1 \leq k \leq n)$, 使得

$$|\lambda - a_{kk}| \leq \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| \quad (1)$$

用集合的关系来说明, 这意味着 $\lambda(A) \subseteq \bigcup_{k=1}^n D_k$.

(2) 若 A 的盖氏圆中有 m 个圆盘组成一连通并集 S , 且 S 与余下的 $n - m$ 个圆盘分离, 则 S 内恰好包含 A 的 m 个特征值 (重特征值按重数计)。

对如图 1 所示的例子, 定理的第 (2) 个结论的含义是: D_1 中只包含一个特征值, 而另外两个特征值在 D_2 、 D_3 的并集中, 下面对定理的结论 (1) 进行证明。

证明:

设 λ 为 A 的任一特征值, 则有 $Ax = \lambda x$, x 为非零常量。设 x 中第 k 个分量最大, 即

$$|x_k| = \max_{1 \leq j \leq n} |x_j| > 0,$$

考虑线性方程中第 k 个方程

$$\sum_{j=1}^n a_{kj} x_j = \lambda x_k,$$

将其中与 x_k 有关的项移到等号左边, 其余移到右边, 再两边取模得

$$|\lambda - a_{kk}| |x_k| = \left| \sum_{\substack{j=1 \\ j \neq k}}^n a_{kj} x_j \right| \leq \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| |x_j| \leq |x_k| \sum_{\substack{j=1 \\ j \neq k}}^n |a_{kj}| \quad (2)$$

最后一个不等式的推导利用了“ x 中第 k 个分量最大”的假设。将不等式(2)除以 $|x_k|$, 即得到式(1), 因此证明了定理1的结论 (1)。上述证明过程还说明, 若某个特征向量的第 k 个分量的模最大, 则相应的特征值必定属于第 k 个圆盘中。

例 1

[圆盘定理应用] 试估计矩阵

$$\begin{pmatrix} 4 & 1 & 0 \\ 1 & 0 & -1 \\ 1 & 1 & -4 \end{pmatrix}$$

的特征值范围.

解：直接应用圆盘定理，该矩阵的三个圆盘如下：

$$D_1 : |\lambda - 4| \leq 1, \quad D_2 : |\lambda| \leq 2, \quad D_3 : |\lambda + 4| \leq 2.$$

D_1 与其他圆盘分离，则它仅含一个特征值，且必定为实数（若为虚数则其共轭也是特征值，这与 D_1 仅含一个特征值矛盾）。所以对矩阵特征值的范围的估计是

$$3 \leq \lambda_1 \leq 5, \quad \lambda_2, \lambda_3 \in D_2 \cup D_3.$$

再对矩阵 A^T 应用圆盘定理，则可以进一步优化上述结果。矩阵 A^T 对应的三个圆盘为

$$D'_1 : |\lambda - 4| \leq 2, \quad D'_2 : |\lambda| \leq 2, \quad D'_3 : |\lambda + 4| \leq 1.$$

这说明 D'_3 中存在一个特征值，且为实数，它属于区间 $[-5, -3]$ ，经过综合分析可知三个特征值均为实数，它们的范围是

$$\lambda_1 \in [3, 5], \quad \lambda_2 \in [-2, 2], \quad \lambda_3 \in [-5, -3].$$

事实上，可求出矩阵 A 的特征值为 4.2030, -0.4429, -3.7601.

还可以对矩阵 A 做简单的相似变换, 例如取 X 为对角阵, 然后再应用圆盘定理估计特征值的范围。

例 2

(特征值范围的估计): 选取适当的矩阵 X , 应用圆盘定理估计例1中矩阵的特征值范围.

$$\begin{pmatrix} 4 & 1 & 0 \\ 1 & 0 & -1 \\ 1 & 1 & -4 \end{pmatrix}$$

解：取

$$\mathbf{X}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.9 \end{pmatrix}$$

则

$$\mathbf{A}_1 = \mathbf{X}^{-1} \mathbf{A} \mathbf{X} = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 0 & -\frac{10}{9} \\ 0.9 & 0.9 & -4 \end{pmatrix}$$

的特征值与 \mathbf{A} 的相同. 对 \mathbf{A}_1 应用圆盘定理, 得到三个分离的圆盘, 它们分别包含一个实特征值, 由此得到特征值的范围估计

$$\lambda_1 \in [3, 5], \lambda_2 \in \left[-\frac{19}{9}, \frac{19}{9}\right], \lambda_3 \in [-5.8, -2.2].$$

此外, 还可以进一步估计矩阵的谱半径 $\rho(\mathbf{A})$ 的范围, 即 $3 \leq \rho(\mathbf{A}) \leq 5.8$.

- 1 16.1 矩阵特征值分布范围的估计
- 2 16.2 幂法和反幂法**
- 3 16.3 幂法的应用

16.2.1 幂法

幂法是一种计算矩阵最大的特征值及其对应特征向量的方法。本节介绍幂法、反幂法以及加快幂法迭代收敛的技术。

定义 2

在矩阵 A 的特征值中，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。

应注意的是，主特征值有可能不唯一，因为模相同的复数可以有很多，例如模为 5 的特征值可能是 $5, -5, 3 + 4i, 3 - 4i$ 等等。另外注意谱半径和主特征值的区别。

如果矩阵 A 有唯一的主特征值，则一般通过幂法能够方便地计算出主特征值及其对应的特征向量。对于实矩阵，这个主特征值显然是实数，但不排除它是重特征值的情况。幂法的计算过程是，首先任取一非零向量 $\mathbf{x}_0 \in \mathbb{R}^n$ ，再进行迭代计算

$$\mathbf{x}_k = A\mathbf{x}_{k-1}, k = 1, 2, \dots$$

得到向量序列 $\{\mathbf{x}_k\}$ ，根据它即可求出主特征值与特征向量。下面我们来看一下具体的计算过程。

假设 A 的特征值可按模的大小排列为 $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$, 且其对应特征向量 $\xi_1, \xi_2, \dots, \xi_n$ 线性无关。此时, 任意非零向量 $x^{(0)}$ 均可用 $\xi_1, \xi_2, \dots, \xi_n$ 线性表示, 即

$$x^{(0)} = \alpha_1 \xi_1 + \alpha_2 \xi_2 + \cdots + \alpha_n \xi_n$$

且 $\alpha_1, \alpha_2, \dots, \alpha_n$ 不全为零。做向量序列 $x^{(k)} = A^k x^{(0)}$, 则

$$\begin{aligned} x^{(k)} &= A^k x^{(0)} = \alpha_1 A^k \xi_1 + \alpha_2 A^k \xi_2 + \cdots + \alpha_n A^k \xi_n \\ &= \alpha_1 \lambda_1^k \xi_1 + \alpha_2 \lambda_2^k \xi_2 + \cdots + \alpha_n \lambda_n^k \xi_n \\ &= \lambda_1^k \left(\alpha_1 \xi_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k \xi_2 + \cdots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^k \xi_n \right) \end{aligned}$$

由此可见, 若 $\alpha_1 \neq 0$, 则有

$$\lim_{k \rightarrow \infty} \left(\frac{\lambda_i}{\lambda_1} \right)^k = 0, i = 2, \dots, n$$

故当 k 充分大的时候, 必有

$$x^{(k)} \approx \lambda_1^k \alpha_1 \xi_1$$

即 $\mathbf{x}^{(k)}$ 可以近似看成 λ_1 对应的特征向量, 而 $\mathbf{x}^{(k)}$ 与 $\mathbf{x}^{(k-1)}$ 分量之比为

$$\frac{\mathbf{x}^{(k)}}{\mathbf{x}^{(k-1)}} \approx \frac{\lambda_1^k \alpha_1 \boldsymbol{\xi}_1}{\lambda_1^{k-1} \alpha_1 \boldsymbol{\xi}_1} = \lambda_1$$

于是利用向量序列 $\{\mathbf{x}^{(k)}\}$ 即可求出按模最大的特征值 λ_1 , 又可以求出对应的特征向量 $\boldsymbol{\xi}_1$ 。

在实际计算中, 考虑到当 $|\lambda_1| > 1$ 时, $\lambda_1^k \rightarrow \infty$; $|\lambda_1| < 1$ 时, $\lambda_1^k \rightarrow 0$, 因而计算 $\mathbf{x}^{(k)}$ 时可能会发生上溢或者下溢, 故每一步将 $\mathbf{x}^{(k)}$ 归一化处理, 即将 $\mathbf{x}^{(k)}$ 的各分量都除以模最大的分量, 使 $\|\mathbf{x}^{(k)}\| = 1$, 于是求 A 按模最大的特征值 λ_1 和对应的特征向量 ξ_1 的算法, 可归纳为如下步骤。

- (1) 输入矩阵 A , 初始向量 $\mathbf{v}^{(0)}$, 误差限 ϵ , 最大迭代次数 N 。记 m_0 是 $\mathbf{v}^{(0)}$ 按模最大的分量, $\mathbf{x}^{(0)} = \mathbf{v}^{(0)} / m_0$ 。置 $k := 0$
- (2) 计算 $\mathbf{v}^{(k+1)} = A\mathbf{x}^{(k)}$ 。记 m_{k+1} 是 $\mathbf{v}^{(k+1)}$ 按模最大的分量, $\mathbf{x}^{(k+1)} = \mathbf{v}^{(k+1)} / m_{k+1}$
- (3) 若 $|m_{k+1} - m_k| < \epsilon$, 停算, 输出近似特征值 m_{k+1} 和近似特征向量 $\mathbf{x}^{(k+1)}$ 否则转 (4)
- (4) 若 $k < N$ 置 $k := k + 1$ 转 (2) 否则输出计算失败信息, 停算

上述算法我们称为幂法。

算法 1 幂法

```
1:  $k = 0; \mathbf{x}^{(k)} = \mathbf{x}$   
2: repeat  
3:    $\mathbf{y}^{(k+1)} = \mathbf{A}\mathbf{x}^{(k)}$   
4:    $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|_\infty$   
5:    $\lambda^{(k+1)} = \mathbf{x}^{(k+1)^T} \mathbf{A} \mathbf{x}^{(k+1)}$   
6:    $k = k + 1$   
7: until 收敛
```

定理 2

设矩阵 A 的特征值可按模大小排列为 $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$, 且其对应特征向量 $\xi_1, \xi_2, \dots, \xi_n$ 线性无关, 序列 $\{x^{(k)}\}$ 由幂法产生, 则有

$$\lim_{k \rightarrow \infty} x^{(k)} = \frac{\xi_1}{\max\{\xi_1\}} := \xi_1^0, \quad \lim_{k \rightarrow \infty} m_k = \lambda_1$$

式中 ξ_1^0 为将 ξ_1 归一化后得到的向量, $\max\{\xi_1\}$ 为向量 ξ_1 模最大的分量。

16.2.2 幂法加速收敛的方法

加速幂法迭代收敛过程的方法主要有两种：

- 原点位移法
- 瑞利商加速

原点位移法

幂法的收敛速度与比值 $|\lambda_2/\lambda_1|$ 的大小有关, $|\lambda_2/\lambda_1|$ 越小, 收敛速度越快, 当此比值接近于 1 时, 收敛速度是非常缓慢的。因此可以对原矩阵作一原点位移, 令

$$B = A - \alpha I$$

式中: α 为参数。选择此参数可使矩阵 B 的上述比值更小, 以加快幂法的收敛速度。设矩阵 A 的特征值为 $\lambda_1, \lambda_2, \dots, \lambda_n$, 对应的特征向量为 $\xi_1, \xi_2, \dots, \xi_n$, 则矩阵 B 的特征值为 $\lambda_1 - \alpha, \lambda_2 - \alpha, \dots, \lambda_n - \alpha$, B 的特征向量与 A 的特征向量相同。假设原点位移后, B 的特征值 $\lambda_1 - \alpha$ 仍为模最大的特征值, 选择 α 的目的是使

$$\max_{2 \leq i \leq n} \frac{\lambda_i - \alpha}{\lambda_1 - \alpha} < \frac{\lambda_2}{\lambda_1}$$

适当地选择 α 可使幂法的收敛速度得到加速。此时 $m_k \rightarrow \lambda_1 - \alpha, m_k + \alpha \rightarrow \lambda_1$, 而 $x^{(k)}$ 仍然收敛于 A 的特征向量 ξ_1^0 。这种加速方法叫做原点位移法。

算法 2 原点位移法

- 1: $k = 0; \mathbf{x}^{(k)} = \mathbf{x}$
 - 2: **repeat**
 - 3: $\mathbf{y}^{(k+1)} = (\mathbf{A} - \alpha \mathbf{I})^{-1} \mathbf{x}^{(k)}$
 - 4: $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|_{\infty}$
 - 5: $\lambda(k+1) = \mathbf{x}^{(k+1)T} \mathbf{A} \mathbf{x}^{(k+1)}$
 - 6: $k = k + 1$
 - 7: **until** 收敛
-

瑞利商加速

假设在原点位移法的某个步骤中，我们有一个近似特征向量 $\mathbf{x}^{(k)} \neq 0$ 。然后，我们寻找近似特征值 λ_k ，也就是满足下列方程的特征值和特征向量

$$\mathbf{x}^{(k)} \lambda_k = \mathbf{A} \mathbf{x}^{(k)}$$

我们寻找特征值 λ_k ，就是要使得方程残差的平方范数最小，即 $\min \|\mathbf{x}^{(k)} \lambda_k - \mathbf{A} \mathbf{x}^{(k)}\|_2^2$ 。通过令导数为 0 得到

$$\lambda_k = \frac{\mathbf{x}^{(k)T} \mathbf{A} \mathbf{x}^{(k)}}{\mathbf{x}^{(k)T} \mathbf{x}^{(k)}}$$

这个量称为瑞利商。

我们如果在原点位移法中根据瑞利商来选择位移，则可以得到瑞利商迭代算法。可以证明瑞利商迭代算法具有局部二次收敛性，即经过一定次数的迭代后，迭代 $k+1$ 次时运行解的收敛间隙与迭代 k 次时该解的间隙平方成正比。

算法 3 瑞利商加速

- 1: $k = 0; \mathbf{x}^{(k)} = \mathbf{y}$
 - 2: **repeat**
 - 3: $\lambda_k = \frac{\mathbf{x}^{(k)T} \mathbf{A} \mathbf{x}^{(k)}}{\mathbf{x}^{(k)T} \mathbf{x}^{(k)}}$
 - 4: $\mathbf{y}^{(k+1)} = (\mathbf{A} - \lambda_k \mathbf{I})^{-1} \mathbf{x}^{(k)}$
 - 5: $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|_\infty$
 - 6: $k = k + 1$
 - 7: **until** 收敛
-

16.2.3 反幂法

反幂法 (inverse iteration) 基于幂法, 可看成是幂法的一种应用, 它能够求矩阵 A 按模最小的特征值及其特征向量。对于一个非奇异矩阵 A , A^{-1} 的特征值为矩阵 A 的特征值的倒数, A^{-1} 的主特征值便是 A 按模最小的特征值的倒数。因此, 可对 A^{-1} 应用幂法求出矩阵 A 的最小特征值。这就是反幂法的基本思想。

与幂法相对应, 反幂法的适用条件是: 矩阵 A 按模最小的特征值唯一, 且几何重数等于代数重数。对于实矩阵, 满足此条件时这个最小特征值一定是实数, 相应的特征向量也为实向量。算法过程描述如下:

设 A 可逆, 由于 $A\xi_i = \lambda_i\xi_i$ 时, 成立 $A^{-1}\xi_i = \lambda_i^{-1}\xi_i$ 。因此, 若 $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_{n-1}| > |\lambda_n|$, 则 λ_n^{-1} 是 A^{-1} 按模最大的特征值, 此时按反幂法, 必有

$$m_k \rightarrow \lambda_n^{-1}, \mathbf{x}^{(k)} \rightarrow \xi_n^0$$

其收敛率为 $|\lambda_n/\lambda_{n-1}|$ 。任取初始向量 $\mathbf{x}^{(0)}$, 构造向量序列

$$\mathbf{x}^{(k+1)} = A^{-1}\mathbf{x}^{(k)}, k = 0, 1, 2, \dots$$

按幂法计算即可。

但用上述式子计算，首先要求 A^{-1} ，这比较麻烦而且是不经济的，实际计算中通常用解方程组的办法，即用

$$A\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}, k = 0, 1, 2, \dots$$

求 $\mathbf{x}^{(k+1)}$ 。为防止计算机溢出，实际计算时所用公式为

$$\mathbf{v}^{(k)} = \mathbf{x}^{(k)} / \max(\mathbf{x}^{(k)}), k = 0, 1, 2, \dots$$
$$A\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)},$$

式中： $\max(\mathbf{x}^{(k)})$ 为 $\mathbf{x}^{(k)}$ 模最大的分量。

在实际计算中, 若知道某个矩阵特征值的估计值, 常利用反幂法结合原点位移技术来求其精确值和对应的特征向量。

若 A 的特征值是 λ , 则 $\lambda - \alpha$ 是 $A - \alpha I$ 的特征值。因此反幂法可以用于已知矩阵的近似特征值为 α 时, 求矩阵的特征向量并且提高特征值精度。

此时, 可以用原点位移法来加速迭代过程, 于是上式相应为

$$(A - \alpha I)x^{(k+1)} = x^{(k)}, k = 0, 1, 2, \dots$$

求 $x^{(k+1)}$ 。为防止计算机溢出, 实际计算时所用公式为

$$\begin{aligned} v^{(k)} &= x^{(k)} / \max(x^{(k)}), \\ (A - \alpha I)x^{(k+1)} &= x^{(k)}, \end{aligned} \quad k = 0, 1, 2, \dots$$

反幂法

算法（反幂法）

- (1) 选取初值 $\mathbf{x}^{(0)}$, 近似值 α , 误差限 ϵ , 最大迭代次数 N 。记 m_0 为 $\mathbf{x}^{(0)}$ 中按模最大的分量, $\mathbf{v}^{(0)} = \mathbf{x}^{(0)} / m_0$ 。置 $k := 0$
- (2) 解方程组 $(\mathbf{A} - \alpha \mathbf{I})\mathbf{x}^{(k+1)} = \mathbf{v}^{(k)}$ 得 $\mathbf{x}^{(k+1)}$ 。
- (3) 记 m_{k+1} 为 $\mathbf{x}^{(k+1)}$ 中按模最大的分量, $\mathbf{v}^{(k+1)} = \mathbf{x}^{(k+1)} / m_{k+1}$ 。
- (4) 若 $|m_{k+1}^{-1} - m_k^{-1}| < \epsilon$, 则置 $\lambda := m_{k+1}^{-1} + \alpha$, 输出 λ 和 $\mathbf{x}^{(k+1)}$, 停算; 否则, 转 (5)
- (5) 若 $k < N$, 置 $k := k + 1$, 转 (2), 否则输出计算失败信息, 停算。

算法 4 反幂法

- 1: $k = 0; \mathbf{x}^{(k)} = \mathbf{x}$
 - 2: **repeat**
 - 3: $\mathbf{y}^{(k+1)} = (\mathbf{A} - \alpha \mathbf{I})^{-1} \mathbf{x}^{(k)}$
 - 4: $\mathbf{x}^{(k+1)} = \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|_{\infty}$
 - 5: $\lambda^{(k+1)} = \mathbf{x}^{(k+1)^T} \mathbf{A} \mathbf{x}^{(k+1)} + \alpha$
 - 6: $k = k + 1$
 - 7: **until** 收敛
-

- 1 16.1 矩阵特征值分布范围的估计
- 2 16.2 幂法和反幂法
- 3 16.3 幂法的应用**

16.3.1 SVD 的计算：奇异值分解 (回顾)

对矩阵 $A \in \mathbb{R}^{m \times n}$ 计算 SVD, 我们需要求得 $A^T A$ 或 AA^T 的特征值和特征向量:

$$A^T A = V \Lambda_n V^T, \quad AA^T = U \Lambda_m U^T$$

其中 Λ_n 和 Λ_m 是对角阵并且最大的 r 个对角元是 A 奇异值的平方 σ_i^2 , $i = 1, \dots, r$, 并且其余对角元为 0, 有

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

我们将展示如何通过幂法计算左右奇异向量以及对应的奇异值。

基本的想法是将幂法应用到矩阵 $A^T A$ 上, 从而求得 $A^T A$ 的特征值和特征向量。但是这样我们需要先计算 $A^T A$, 这需要计算 mn^2 次乘法。
因此考虑以下迭代:

$$\begin{aligned} \mathbf{u}^{(k+1)} &= \frac{\mathbf{A} \mathbf{v}^{(k)}}{\|\mathbf{A} \mathbf{v}^{(k)}\|_2} \\ \mathbf{v}^{(k+1)} &= \frac{\mathbf{A}^T \mathbf{u}^{(k+1)}}{\|\mathbf{A}^T \mathbf{u}^{(k+1)}\|_2} \end{aligned}$$

消去 $\mathbf{u}^{(k+1)}$ 并由 $\mathbf{v}^{(k+1)}$ 是单位向量得到:

$$\mathbf{v}^{(k+1)} = \frac{\mathbf{A}^T \mathbf{A} \mathbf{v}^{(k)}}{\|\mathbf{A}^T \mathbf{A} \mathbf{v}^{(k)}\|_2}$$

这样我们就避免了计算 $A^T A$, 并且同时得到了最大奇异值 σ_1 的左、右奇异向量 \mathbf{u}_1 和 \mathbf{v}_1 。

为了计算其余的左右奇异向量以及奇异值, 我们计算

$$\mathbf{A}_i = \mathbf{A}_{i-1} - \sigma_i \mathbf{u} \mathbf{v}_i^T, \quad i = 1, \dots, r;$$

其中 $\mathbf{A}_0 = \mathbf{A}, \sigma_0 = 0, r = \text{rank}(\mathbf{A})$, 并对 $\mathbf{A}_i, i = 1, \dots, r$ 应用右侧的算法, 如果所有的奇异值都不相等, 我们就可以计算出 \mathbf{A} 的 SVD:

Require: $\mathbf{A} \in \mathbb{R}^{m \times n}, \sigma_1 > \sigma_2, \mathbf{v} \in \mathbb{R}^n, \|\mathbf{v}\|_2 =$

1

1: $k = 0, \mathbf{v}(k) = \mathbf{v}$

2: **repeat**

3: $\mathbf{y}(k+1) = \mathbf{A} \mathbf{v}(k)$

4: $\mathbf{u}(k+1) = \mathbf{y}(k+1) / \|\mathbf{y}(k+1)\|_2$

5: $\mathbf{z}(k+1) = \mathbf{A}^T \mathbf{u}(k+1)$

6: $\mathbf{v}(k+1) = \mathbf{z}(k+1) / \|\mathbf{v}(k+1)\|_2$

7: $k = k + 1$

8: **until** 收敛

16.3 PageRank

搜索引擎与 PageRank

互联网 (Internet) 的使用已经深入到人们的日常生活中, 其巨大的信息量和强大的功能给生产、生活带来了很大的便利. 随着网络的信息量越来越庞大, 如何有效地搜索出用户真正需要的信息变得十分重要. 自 1998 年搜索引擎网站 Google 创立以来, 网络搜索引擎成为解决上述问题的重要手段。

1998 年, 美国斯坦福大学的博士生 Larry Page 和 Sergey Brin 创立了 Google 公司, 他们的核心技术就是通过 Pagerank 技术对海量的网页进行重要性分析. 该技术利用网页相互链接的关系对网页进行组织, 确定出每个网页的重要级别 (Pagerank). 当用户进行搜索时, Google 找出符合搜索要求的网页, 并按它们的 Pagerank 大小依次列出. 这样, 用户一般显示结果的第一页或者前几页就能找到真正有用的结果。

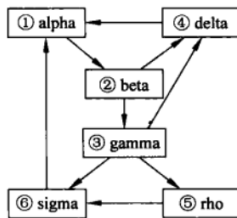
16.3 PageRank

总的来说, 所要求解的问题可归结为: 给定 $n \times n$ 的网页连接矩阵 G , 以及选择当前网页链接的概率 p 时, 计算主特征值 1 对应的特征向量 \mathbf{x} 。

$$\begin{cases} A\mathbf{x} = \mathbf{x} \\ \sum_{i=1}^n x_i = 1 \end{cases}$$

对于实际的大规模稀疏矩阵 A , 幂法是求其主特征向量的可靠的、唯一的选择。

用一个只有 6 个网页的微型网络作为例子，其网页链接关系如图 (B) 所示。



图(B) 网页超链接关系

通过下述 MATLAB 命令可生成矩阵 G

$i = [2 \ 3 \ 4 \ 4 \ 5 \ 6 \ 1 \ 6 \ 1];$ //网页 i

$j = [1 \ 2 \ 2 \ 3 \ 3 \ 3 \ 4 \ 5 \ 6];$ //网页 j

$n = 6;$

$G = \text{sparse}(i, j, 1, n, n);$ //生成 $j \rightarrow i$ 的网络连接稀疏矩阵

再使用下述命令得到矩阵 A

$c = \text{full}(\text{sum}(G));$

$D = \text{spdiags}(1./c', 0, n, n);$

$e = \text{ones}(n, 1);$

$p = .85; \text{delta} = (1 - p)/n;$

$A = p * G * D + \text{delta} * e * e';$

也可以使用 Python 中的 NumPy 库生成 G

```
 $i = np.array([2, 3, 4, 4, 5, 6, 1, 6, 1]) - 1$ 
```

```
 $j = np.array([1, 2, 2, 3, 3, 3, 4, 5, 6]) - 1$ 
```

```
 $data = np.ones(len(i))$ 
```

```
 $n = 6;$ 
```

```
 $G = csr\_matrix((data, (i, j)), shape = (n, n)).toarray()$ 
```

再使用下述命令得到矩阵 A

```
 $c = np.sum(G, axis = 0)$ 
```

```
 $D = np.diag(1/c)$ 
```

```
 $e = np.ones((n, n))$ 
```

```
 $p = .85; delta = (1 - p)/n$ 
```

```
 $A = p * np.matmul(G, D) + delta * e$ 
```

得到的矩阵 A 为

$$\begin{pmatrix} 0.025 & 0.025 & 0.025 & 0.875 & 0.025 & 0.875 \\ 0.875 & 0.025 & 0.025 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.45 & 0.025 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.45 & 0.3083 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.025 & 0.3083 & 0.025 & 0.025 & 0.025 \\ 0.025 & 0.025 & 0.3083 & 0.025 & 0.875 & 0.025 \end{pmatrix}$$

使用幂方法可求出其主特征向量，其步骤如下：

(1) 给出初始向量 $\mathbf{x}_0 = [1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$

(2) $\mathbf{x}^{k+1} = \mathbf{A}\mathbf{x}^k$

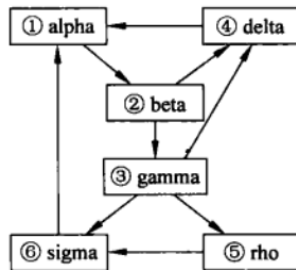
(3) 归一化：

$$\mathbf{x}^{k+1} = \frac{\mathbf{x}^{k+1}}{\sum_{i=1}^n x_i^{k+1}}$$

(4) 当 $\mathbf{x}^{k+1} - \mathbf{x}^k > \varepsilon$ ，重复计算 (2)(3)

最后可得到 PageRank 为

$$\mathbf{x} = [0.2675 \ 0.2524 \ 0.1323 \ 0.1697 \ 0.0625 \ 0.1156]^T$$



图(B) 网页超链接关系



本讲小结

特征值的求解方法

- 幂法
- 幂法加速收敛的方法：原点位移法、瑞利商加速
- 反幂法

幂法的应用

- 计算 SVD
- Google 的 PageRank 算法
- ...

计算一般矩阵的全部特征值和特征向量的最有效方法之一：QR 方法！
对称矩阵的特征值问题的计算：对称 QR 方法、Jacobi 方法等！