矩阵特征值的计算方式

幂法

设置方阵A的各特征值λ中仅有唯一的模长最大的值，且各特征矢u_i线性无关，即：

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

设$x^{(0)}$是任意不为零的n维向量：

由迭代公式：

$$x^{(k+1)} = Ax^{(k)} \tag{4.1}$$

可以证明，在k足够大时，矩阵特征值λ是向量$x^{(k+1)}$与向量x^k的任意两个对应分量之商$x_i^{(k+1)}/x_i^k$

证明：

由于u_i是n个n维向量且线性无关，故它是$n \times 1$空间中的一组基底，故无论$x^{(0)}$的取值，它总能被表示为$x^{(0)} = \sum_{i=1}^n a_i u_i$，其中，$a_i$不全为0. 即$x^{(0)}$是各个特征矢的线性组合.

则由迭代公式：

$$x^{(k+1)} = Ax^{(k)} = A^{k+1}x^{(0)} = \sum_{i=1}^{n} A^{k+1}a_i u_i = \sum_{i=1}^{n} \lambda_i^{k+1}a_i u_i$$

重写上述等式：

$$x^{(k+1)} = \lambda_1^{k+1}[a_1u_1 + (\frac{\lambda_2}{\lambda_1})^{k+1}a_2u_2 + (\frac{\lambda_3}{\lambda_1})^{k+1}a_3u_3 + ... + (\frac{\lambda_n}{\lambda_1})^{k+1}a_nu_n] \tag{4.2}$$

不妨假设λ_1是各个特征值中绝对值最大的一个，则在k充分大时，等式(4.2)可以截断为：

$$x^{(k+1)} \approxeq \lambda_1^{k+1}a_1u_1 \tag{4.3}$$

此时，$x^{(k+1)}$可以看作是方阵A对应于特征值λ_1的一个特性矢. 同时，在k充分大时，

$$x^{(k)} \approxeq \lambda_1^{k}a_1u_1 \tag{4.4}$$

比较式子(4.3)和(4.4)，可以知道，$x^{(k+1)}$和$x^{(k)}$线性相关，且线性系数之比为λ_1

故，可以将$x^{(k+1)}$和$x^{(k)}$的任一对应分量x_i作比，有：

$$\lambda_1 = x^{(k+1)}/x^{(k)} \tag{4.5}$$

■

但是精妙的数学逻辑应用到计算机上就会变得粗糙，由于我们需要重复计算若干多次式(4.1)，向量$x^{(k)}$中的分量可能出现溢出或舍入为0的情况. 故这时我们需要对每一次迭代出的向量$x^{(k)}$进行归一化处理：

$$\begin{cases} y^{(k)} = \dfrac{x^{(k)}}{\|x^{(k)}\|_\infty} \\ x^{(k+1)} = Ay^{(k)} \end{cases}$$

由上述原理，在实际计算中，在式子(4.5)的分量选择过程中，我们应当选择绝对值最大的分量以消除舍入误差.

有效性讨论

1. 若不满足"仅有一个模取得最大值"的情况，由上述推导过程易知算法仍有效
2. 特殊地，若有$\lambda_1 = -\lambda_2$时，迭代过程变为一摆动序列，则比较k足够大时的任意相邻三项即可得到求出λ_1和u_1与u_2
3. 易知，算法的收敛速度取决于$|\lambda_2|/|\lambda_1|$

迭代加速

原点移位法

设λ_i是A的特征值，则$\lambda_i - \lambda_0$是$A - \lambda_0 E$的特征值.

因此我们只需要找到合适的λ_0使得$\forall i \in \mathbb{N}^*, |\lambda_1 - \lambda_0| > |\lambda_i - \lambda_0|$且$max\frac{|\lambda_i - \lambda_0|}{|\lambda_1 - \lambda_0|} < |\frac{\lambda_2}{\lambda_1}|$，我们就可以用矩阵$A - \lambda_0 E$来代替矩阵$A$，从而获得更快的收敛过程.

但实践中我们不可能先验地知晓A的所有特征值，故此该方法在实际上难以精确地执行.

Aitken方法

我们称幂法中"k足够大时，$x^{(k+1)}$和$x^{(k)}$成比例"为一阶收敛. 若序列a_k一阶收敛至a，即：

$$\lim_{k\to\infty}\frac{a_{k+1} - a}{a_k - a} = c \neq 0$$

作近似：在k充分大时有：

$$\frac{a_{k+2} - a}{a_{k+1} - a} \approx \frac{a_{k+1} - a}{a_k - a}$$

解出a的表达式：

$$a' \approx a_k - \frac{(a_{k+1} - a_k)^2}{a_{k+2} - 2a_{k+1} + a_k}$$

每次进行幂法迭代之后，我们都依据上式计算一次λ'，若λ'与上一个λ'之间的误差满足条件，则输出结果.

函数插值法

已知平面上若干互不相同的点$(x_0,y_0),(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$，其关系属于某未知对应关系$f(x)$，如何构造一函数，使得该函数可以穿过这些点并刻画$f(x)$?

这种问题被称为插值问题，上述的函数被称为插值函数.

Lagrange多项式法

Lagrange多项式法使用一个n次多项式来拟合上述的$(n+1)$个点，且可以得出唯一的结果.

证明:

对于点集$\{(x_0,y_0),(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$，找到一$n$次多项式$f(x)=a_0+a_1x+a_2x^2+...+a_nx^n$使得$\forall x_i,f(x_i)=y_i$等价于方程组:

$$\begin{cases} f(x_0)=y_0 \\ f(x_1)=y_1 \\ ... \\ f(x_n)=y_n \end{cases} \tag{4.6}$$

展开之，可使其变为一个系数矩阵是范德蒙矩阵的关于多项式系数向量A的线性方程组，且范德蒙矩阵在x_i互不相同时满秩.

多项式系数向量A存在且唯一.

■

我们对Lagrange多项式（虽然还没有完全提出它的求解方式）进行误差估计:

设置有若干节点$(x_i,y_i),i\in[0,n]$是$[a,b]$上的$(n+1)$个互异节点，若待拟合函数在(a,b)上$(n+1)$阶可导，即$f(x)\in\mathbb{C}^{(n+1)}[a,b]$，若我们用Lagrange多项式$\phi(x)$拟合之，则其插值余项（截断误差）可以表述为:

$$R_n(x)=f(x)-\phi(x)=\frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x) \tag{4.7}$$

其中$\omega_{n+1}(x)=\Pi_{i=0}^n(x-x_i)$，$\xi\in(a,b)$

直球使用微分中值定理即可**证明:**

由定义，$R(x)=f(x)-\phi(x)$，再由Lagrange多项式的基本假设（式4.6），有$\forall i,R(x_i)=0$.

故可以等价地将$R(x)$转写为$R(x)=K(x)\omega_{n+1}(x)$，其中$\omega_{n+1}(x)=\Pi_{i=0}^n(x-x_i)$，$K(x)$是某未知的多项式，我们需要找到它:

使用惯用的套路构造辅助函数:

$$h(t) = R(t) - K(x)\omega_{n+1}(t)$$

观察到上述的函数有$(n+2)$个零点：$x, x_0, x_1, x_2, ..., x_n$，而我们事先约定这些零点都在$f(x)$的解析域$(a,b)$上（否则Lagrange多项式将失去意义）

对自变量t反复应用罗尔中值定理$(n+1)$次，则存在某一点$\xi \in (a,b)$，使得：

$$h^{(n+1)}(\xi) = R^{n+1}(\xi) - K(x)(n+1)! = 0$$

故得到$K(x) = \frac{R^{n+1}(\xi)}{(n+1)!}$，$\xi \in (a,b)$

而$R(x) = f(x) - \phi(x)$，$\phi(x)$是一个n次多项式，在$(n+1)$次求导后变为0，故

$$K(x) = \frac{f^{n+1}(\xi)}{(n+1)!}, \xi \in (a,b)$$

■

式4.7的意义在于，若记$M = \max_{(a,b)} |f^{(n+1)}(x)|$，则Lagrange插值法的截断误差$R(x) \leqslant \frac{M}{(n+1)!}\omega_{n+1}(x)$，这体现了对误差的一种控制.

Lagrange多项式的求解

最直观地，我们可以直接通过式(4.6)列方程组求解，但由于涉及到n次方运算，我们会得到一个范氏矩阵，而它往往是病态的，故考虑使用其它的方法求解.

事实上凡是要解方程求Lagrange多项式的，都会面临病态的问题，因此我们从方程形式上来考察它：

Lagrange多项式可以认为是一组线性无关的基$1, x, x^2, x^3, ..., x^n$线性组合而成的，因此我们可以构造另外一组与其同构的基来解决这一问题. 我们构造这样的一组基底：

$$l_i(x), i \in [0,n], s.t: l_i(x) = \frac{\prod_{j=0, j\neq i}^{n}(x - x_j)}{\prod_{j=0, j\neq i}^{n}(x_i - x_j)}$$

这组基满足一个特殊的性质：对于点集中的若干离散x，当且仅当$x = x_i$时$l_i(x)$为1，否则为0，故我们可以构造插值函数如下：

$$f(x) = \sum_{i=0}^{n} y_i l_i(x) \tag{4.8}$$

容易得知上述的函数满足Lagrange插值的基本假设（式4.6）

但是基于4.8的Lagrange插值法具有一明显的弊端：即一旦点集发生改变，各个基均需要进行修改，故我们希望找到另外一组基，使得即使点集发生了增减，我们也只需简单地增加或删除一项即可：

Newton插值法

课本中对Newton插值的提法太过啰嗦，故给出一种额外的提法：

我们以一种逐步构建的方式来提出Newton插值.

1 最初，我们的点集中只有(x_0, y_0)，因此我们给出插值式$f_0(x) = y_0$来拟合之.

2 随后我们加入了点(x_1, y_1)，此时插值公式需要额外满足$f(x_1) = y_1$，根据Newton插值的思想，我们通过为$f_0(x)$附加一项来达成此目的：（$s.t.$在数学符号中指"subject to"，"使得"）

$$f_1(x) = y_0 + R_1(x),\ s.t. f_1(x_0) = y_0, f_1(x_1) = y_1$$

故$R_1(x)$一定满足：$R_1(x_0) = 0, R_1(x_1) = y_1 - y_0$

故可以提出满足上述条件的$R_1(x)$较为简单形式：$R_1(x) = b_1(x - x_0)$，代入解出

$$b_1 = \frac{y_1 - y_0}{x_1 - x_0}$$

此时插值公式变为

$$f_1(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$$

3 继续加入点(x_2, y_2)，此时插值公式需要额外满足$f(x_2) = y_2$，故仍使用上述的方式：

$$f_2(x) = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) + R_2(x)$$

同理，设置$R_2(x) = b(x - x_0)(x - x_1)$，解出

$$b_2 = \frac{\frac{y_2 - y_0}{x_2 - x_0} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_1}$$

这个形式不够简洁，**经过简单的数学变换之后**（这种变换蕴藏着“均差(差商)的轮换对称性”，后述），可以化为

$$b_2 = \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0}$$

我们发现它有着某种对称性：我们定义$f[a, b] = \frac{f(a) - f(b)}{a - b}$，则$b_2$可以转写为：

$$b_2 = \frac{f[x_2, x_1] - f_[x_1, x_0]}{x_2 - x_0}$$

这样的形式仍符合着某种更"高级"的对称性，故我们将b_2记为$f[x_0, x_1, x_2]$，表示先求"较为低级的"两个$f[x_2, x_1]$和$f[x_1, x_0]$，再将其作差，与外层的两个值之差$x_2 - x_0$相除. 这样，我们递归地提出了均差(差商)的概念：

1. $f[a, b] = \frac{f(a) - f(b)}{a - b}$
2. $f[a, a_1, ..., b_1, b] = \frac{f[a, ..., b_1] - f[a_1, ..., b]}{a - b}$，"..."可以为空且$a_1, b_1$可以相同(即本式支持三个及以上项目)

容易证明，均差具有轮换对称性，即只要参数的组合一致，参数的排列与均差的值无关.

故Newton插值法可以直观地表述为:

$$f^*(x) = f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \ldots$$

分析地描述为:

$$f^*(x) = \sum_{i=0}^{n} l_i(x) \tag{4.9-1}$$

$$l_i(x) = f[x_0\ to.\ x_i] \prod_{j=0}^{i-1}(x - x_0) \tag{4.9-2}$$

请务必注意上述式子中连乘符号上的$(i-1)$.

Newton插值的截断误差

在非节点的任意点$(x, f(x))$上，Newton插值的截断误差可以表述为:

$$R(x) = f^*(x) - f(x)$$

而$f(x)$不可直接求知，因此采用一巧妙的解法：以点$(x, f(x))$为第$(n+1)$个节点构造一新的Newton插值式$f_{(n+1)}(x)$，而此式可以精确反映$f(x)$的值. 则易知其差值:

$$R(x) = f[x_0\ to.\ x_n, x] \prod_{j=0}^{n}(x - x_0)$$

即为Newton插值的截断误差. 而对于一个点系的插值式是唯一的，因此结合Lagrange插值法可以导出:

$$f[x_0\ to.\ x_n, x]\omega_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\omega_{n+1}(x)$$

其中ξ在点集(x_i, y_i)连成的开区域内部，即

$$f[x_0\ to.\ x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

等距均差和等距节点的Newton插值

如果上述过程中的节点都是等距的，即$\forall i \in [0, n-1], x_i - x_{i+1} \equiv h$，则均差可以改写为如下的形式(证明是简单的):

$$f[x_a, x_{a+1}, \ldots, x_{a+n}] = \frac{\Delta^n_{a+n,a} f}{n!h^n}$$

其中$\Delta^n f$被称为n阶差分，根据差分与均差在内容上的一致性，我们可以递归地描述差分:

1. $\Delta^1_{a+1,a} f = f(a+1) - f(a)$
2. $\Delta^n_{a+n,a} f = \Delta^{n-1}_{a+n-1,a} f - \Delta^{n-1}_{a+n,a+1} f$

至于所谓向前差分或向后差分，我们可以通过互换下角标顺序来描述(反正阶数又不会变)，又何必创造两个不同的符号呢？数学本来是简洁明快又可爱的，为什么要弄出这么多分裂的概念来让她变得啰嗦而丑陋呢？如是，我们即可重新以差分来改写节点等距情况下的Newton插值式：

$$f^*(x) = \sum_{i=0}^{n} l_i(x) \tag{4.10-1}$$

$$l_i(x) = \frac{\Delta_{i,0}^{i} f}{i! h^i} \prod_{j=0}^{i-1} (x - x_0) \tag{4.10-2}$$

直观地表述为

$$f^*(x) = f(x_0) + \frac{\Delta_{1,0}^{1} f}{1! h^1}(x - x_0) + \frac{\Delta_{2,0}^{2} f}{2! h^2}(x - x_0)(x - x_1) + \frac{\Delta_{3,0}^{3} f}{3! h^3}(x - x_0)(x - x_1)(x - x_2)$$

$$\ldots$$

而在某些情况下，x会使用类似于等差数列的形式表示：$x = x_0 + ht$，这时我们可以使用t作自变量，将上述的式子再度改写：

$$f^*(x_0 + ht) = \sum_{i=0}^{n} l_i(t) \tag{4.11-1}$$

$$l_i(t) = \frac{\Delta_{i,0}^{i} f}{i! h^i} \prod_{j=0}^{i-1} (x_0 + th - (x_0 + jh)) = \Delta_{i,0}^{i} f \frac{\prod_{j=0}^{i-1}(t - j)}{i!} \tag{4.10-2}$$

直观地表述为

$$f^*(x_0 + ht) = f(x_0) + t\Delta_{1,0}^{1} f + \frac{t(t-1)}{2!}\Delta_{2,0}^{2} f + \frac{t(t-1)(t-2)}{3!}\Delta_{3,0}^{3} f + \ldots$$

这一式子被称为向前插值公式. 代入求值时应注意自变量是t，若输入的是x则需相应的线性转化.

同理，若将x表示为$x = x_n + ht$，也可以得出类似的结果：

$$f^*(x_n + ht) = \sum_{i=0}^{n} l_i(t) \tag{4.11-1}$$

$$l_i(t) = \frac{\Delta_{0,i}^{i} f}{i! h^i} \prod_{j=0}^{i-1} (x_0 + th - (x_0 + jh)) = \Delta_{0,i}^{i} f \frac{\prod_{j=0}^{i-1}(t - j)}{i!} \tag{4.10-2}$$

直观地表述为

$$f^*(x_0 + ht) = f(x_0) + t\Delta_{0,1}^{1} f + \frac{t(t-1)}{2!}\Delta_{0,2}^{2} f + \frac{t(t-1)(t-2)}{3!}\Delta_{0,3}^{3} f + \ldots$$

向前和向后插值都是基于式子(4.10)的，只不过对于x的等差数列表示不同.