QR迭代法求解矩阵A的特征值

沈欢00986096
北京大学工学院，北京100871

2011 年 10 月 23 日

摘　要

本文用QR迭代法求解矩阵A的特征值:第一步先用豪斯荷尔德变换将矩阵A化为上海森伯格矩阵A_H，第二步再对A_H进行QR迭代(使用吉文斯变换)，当迭代满足精度要求时输出n个特征值。

1　问题描述

对于给定的矩阵A(存在n个不相等的实特征值),采用QR方法求其n个特征值。本文中的矩阵A由文件"$gr_900_900_crg.mm$"给出。并以$.mm$格式描述。

2　算法描述

2.1　得到实对称矩阵A

本文由文件"$gr_900_900_crg.mm$"得到了以$.mm$格式描述的系数矩阵A。A矩阵是900∗900的大型稀疏对称矩阵。于是，在matlaB中，使用"A=zeros(900,900)"语句生成900∗900的零矩阵。再按照.mm文件中的描述，分别对第i行、第j列的元素赋对应的值，就生成了系数矩阵A，并将A存为.mat文件以便之后应用。
由于A矩阵是实对称的900阶矩阵，所以A矩阵有900个实特征值，可以使用QR方法求解。下面描述具体算法。

2.2　QR算法

QR算法的核心思想是：对一个具有n个不相等的特征值的矩阵A，进行如下变换:

$$A = Q_1R_1; A_1 = R_1Q_1$$

$$A_1 = Q_2R_2; A_2 = R_2Q_2$$

$$\cdots\cdots\cdots$$

$$A_k = Q_{k+1} R_{k+1}; A_{k+1} = R_{k+1} Q_{k+1}$$

$$\cdots\cdots\cdots$$

其中Q_i是正交矩阵，R_i是可逆的上三角矩阵。由于:

$$A_{i+1} = Q_{i+1}^T (Q_{i+1} R_{i+1}) Q_{i+1} = Q_{i+1}^T A_i Q_{i+1}$$

所以A_{i+1}和A_i相似，具有相同的特征根。当i$\rightarrow \infty$时，A_i对角线以下元素趋于零。
利用这一原理，可以对给定A进行QR迭代，最终得到上三角矩阵(对角元素不一定非零)，输出对角元素可以作为A的特征值。
我们又发现，具有如下上海森伯格形式的矩阵A_H:

$$A_H = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n-1} & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n-1} & a_{2n} \\ 0 & a_{32} & a_{33} & a_{34} & \dots & \dots \\ & & & \dots & \dots & \\ & & & \dots & \dots & \\ 0 & 0 & \dots & \dots & a_{n,n-1} & a_{nn} \end{bmatrix} \tag{1}$$

A_H在QR迭代过程中可以保持上海森伯格形式不变。
所以，QR迭代算法分两步实现，第一步将矩阵A变换为上海森伯格矩阵A_H,第二步对此上海森伯格矩阵A_H进行QR迭代，直至满足精度为止，输出结果。
在第一步中，将矩阵A变换为上海森伯格矩阵A_H，为了保持变换后的A_H与A具有相同的特征值，这里采用豪斯荷尔德变换(正交变换)，本步在后文中详细说明；在第二步中，对上海森伯格矩阵A_H进行QR迭代，并不去直接找出Q矩阵和R矩阵，而是采用吉文斯变换达到相同的效果，本步在后文中也有详细说明；对于QR迭代的精度，在算法采用：当迭代到第k次时有:

$$\frac{|A_{i+1,i}|}{|A_{i,i}| + |A_{i+1,i+1}|} < accuracy$$

对于i=1,2,$\cdots\cdots$ n-1成立，其中accuracy是所设精度。认为此时的A_H次对角元素已经为零，迭代结束，输出该矩阵对角元素作为A的特征值。

QR算法主程序(见程序：function [eigenvalue,flag1]=QR(A,n))的程序流程图如图1所示。
对于方框中的两步：1、用豪斯荷尔德变换将A矩阵化为上海森伯格矩阵A_H; 2、用吉文斯变换实现一次等效的QR迭代；将分别描述。

2.3 用豪斯荷尔德变换将A矩阵化为上海森伯格矩阵A_H

豪斯荷尔德矩阵

$$H = I - 2ww^T$$

其中w是单位向量。所以H矩阵是正交、对称矩阵。

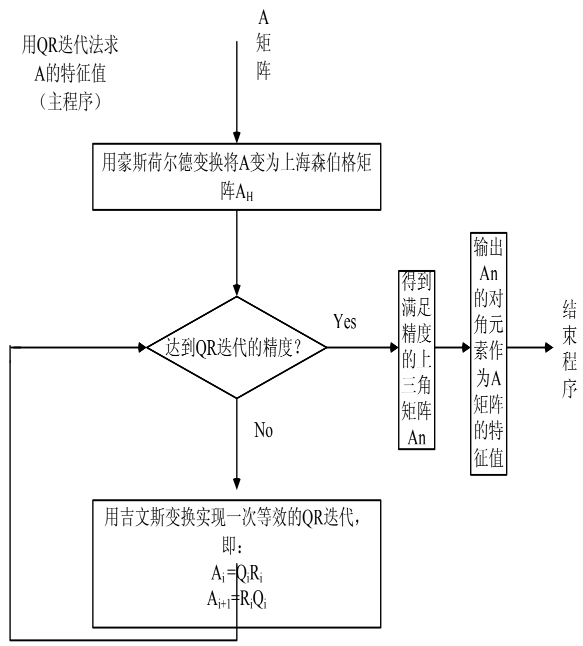

图 1: 主程序(QR迭代法求矩阵特征值)算法

用豪斯荷尔德变换将A矩阵化为上海森伯格矩阵A_H是实现下述过程:

$$A_1 = H_1 A H_1$$

$$A_2 = H_2 A_1 H_2$$

$$\cdots\cdots$$

$$A_{n-2} = H_{n-2} A_{n-3} H_{n-2}$$

其中H_1、$H_2\cdots\cdots$ 满足:

$$H_1 \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} a_{11}^{'} \\ a_{21}^{'} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{2}$$

$$H_2\begin{bmatrix} a_{12}^{'} \\ a_{22}^{'} \\ a_{32}^{'} \\ a_{42}^{'} \\ \vdots \\ a_{n2}^{'} \end{bmatrix}=\begin{bmatrix} a_{12}^{''} \\ a_{22}^{''} \\ a_{32}^{''} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3}$$

$$\cdots\cdots$$

于是构造豪斯荷尔德矩阵H_i成为此函数的核心。对于H_i，目的是将之前得到的A_{i-1}中的第i列实现如下变换:

$$H_i\begin{bmatrix} a_{1i}^{'} \\ a_{2i}^{'} \\ a_{3i}^{'} \\ \cdots \\ a_{ii}^{'} \\ a_{i+1,i} \\ a_{i+2,i} \\ \vdots \\ a_{ni}^{'} \end{bmatrix}=\begin{bmatrix} a_{1i}^{''} \\ a_{2i}^{''} \\ a_{3i}^{''} \\ \cdots \\ a_{ii}^{''} \\ a_{i+1,i}^{''} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{4}$$

于是从A_{i-1}中取出$n-i$维列向量

$$\overrightarrow{x}=[a_{i+1,i}^{'},a_{i+2,i}^{'},\cdots,a_{ni}^{'}]^T$$

取

$$\alpha=\|\overrightarrow{x}\|_2;$$

和n-i维单位向量

$$\overrightarrow{e}=[1,0,0,\cdots,0]^T$$

构造

$$\omega=\frac{\overrightarrow{x}-\alpha\overrightarrow{e}}{\|\overrightarrow{x}-\alpha\overrightarrow{e}\|_2}$$

于是就得到了$(n-i)*(n-i)$的豪斯荷尔德矩阵$H_i^{'}$, 再将其扩充:

$$H_i=\begin{bmatrix} I_{i*i} & 0 \\ 0 & H_i^{'} \end{bmatrix} \tag{5}$$

得到H_i。实现:

$$A_i=H_iA_{i-1}H_i$$

将此步依次从$i=1$至$i=n-2$操作n-2次，得到上海森伯格矩阵A_H,且与原矩阵A相似。
本函数(function AH=A-to-AHessenberg(A,n))流程图如图二所示。

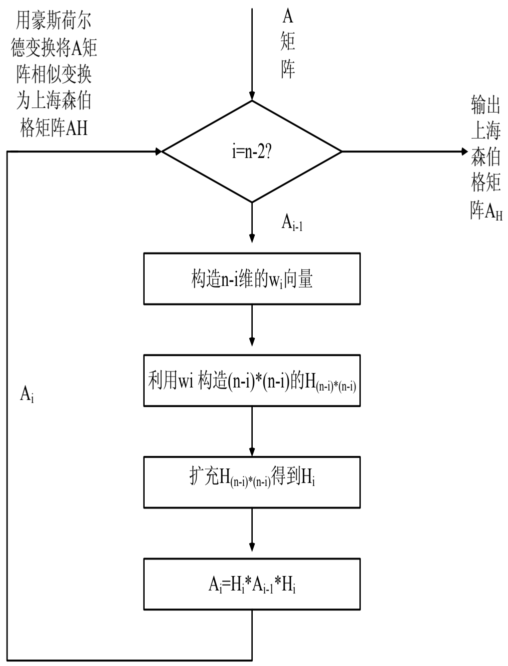

图 2: 用豪斯荷尔德变换将矩阵A化为上海森伯格矩阵AH算法

2.4 用吉文斯变换实现一次等效的QR迭代

在上文，已用豪斯荷尔德变换将矩阵A化为上海森伯格矩阵A_H. 对于一次QR迭代而言，相当于用吉文斯变换实现等效地实现以下过程:

$$G(n-1,n;\theta_{n-1})G(n-2,n-1;\theta_{n-1})\cdots\cdots G(1,2;\theta_1)A_H=R$$

其中$G(i-1,i;\theta_{i-1})$是吉文斯扩充矩阵，是正交矩阵。
该过程等效于:

$$AH=QR$$

其中

$$Q=G(1,2;\theta_1)^T\cdots\cdots G(n-2,n-1;\theta_{n-1})^T G(n-1,n;\theta_{n-1})^T$$

得到上三角矩阵R后，再进行:

$$AH_{new}=RQ=RG(1,2;\theta_1)^T\cdots\cdots G(n-2,n-1;\theta_{n-1})^T G(n-1,n;\theta_{n-1})^T$$

但是如果真的将Q矩阵找出，并如上述作用，那么计算开销会非常大。所以我们考虑一个等效的过程。对于第一个过程，实际上是依次将A_H矩阵的第i行和第i+1行进行变换(总共进行n-1次变换)，目的是使得

$$AH_{i+1,i} = 0$$

。
于是，在第一个过程中就只进行第i行和第i+1行的变换。取:

$$c_i = \frac{AH_{i,i}}{\sqrt{AH_{i,i}^2 + AH_{i+1,i}^2}}$$

$$s_i = \frac{AH_{i+1,i}}{\sqrt{AH_{i,i}^2 + AH_{i+1,i}^2}}$$

于是，第i行和第i+1行的元素变换为:

$$AH_{i,j}^{'} = c_i * AH_{i,j} + s_i * AH_{i+1,j}$$

$$AH_{i+1,j}^{'} = c_i * AH_{i+1,j} - s_i * AH_{i,j}$$

$$j = 1, 2, \cdots\cdots, n$$

.
对i从1至n-1,重复以上过程,则等效实现QR分解，并得到上三角矩阵R。
在第二个过程中，R矩阵右乘上Q矩阵，即:

$$AH_{new} = RQ = RG(1,2;\theta_1)^T \cdots\cdots G(n-2, n-1; \theta_{n-1})^T G(n-1, n; \theta_{n-1})^T$$

等效于依次对R矩阵的第i列和第i+1列进行列变换。变换后的元素为:

$$R_{j,i}^{'} = c_i * R_{j,i} + s_i * R_{j,i+1}$$

$$R_{j,i+1}^{'} = c_i * R_{j,i+1} - s_i * R_{j,i}$$

$$j = 1, 2, \cdots\cdots, n$$

.
对i从1至n-1,重复以上过程,则等效地实现$AH_{new} = RQ$过程。 至此，完成了一次QR迭代。
本函数(AHnew=QR-iteration-once(AH,n))的流程图如图三所示。

3 结果输出

在$function[eigenvalue, flag-1] = QR(A, n)$函数中，设置$accuracy = 10^{-3}$,取n=900,A为本文2.1所得的矩阵，运行该函数。结果显示，$flag_1 = 9344$。也就是，经过9344次迭代，上海森伯格矩阵次A_H对角元素接近0，可由对角元素得到A的特征值。所得结果在同一文件夹中的$eigenvalue.mat$中

图四显示了900个特征值从小到大的排列。
表一给出了900个特征值中最大的八个和最小的八个特征值。

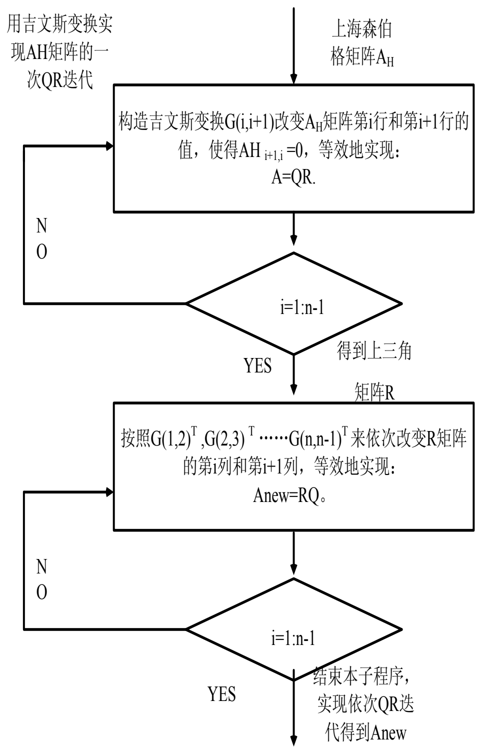

图 3: 用吉文斯变换等效实现一次QR迭代算法

表 1: 八个最大的和最小的特征值

最大特征值	11.9591	11.9591	11.9287	11.9287	11.8784	11.8783	11.8673	11.8668
最小特征值	0.0615	0.1532	0.1532	0.2440	0.3050	0.3050	0.3942	0.3942

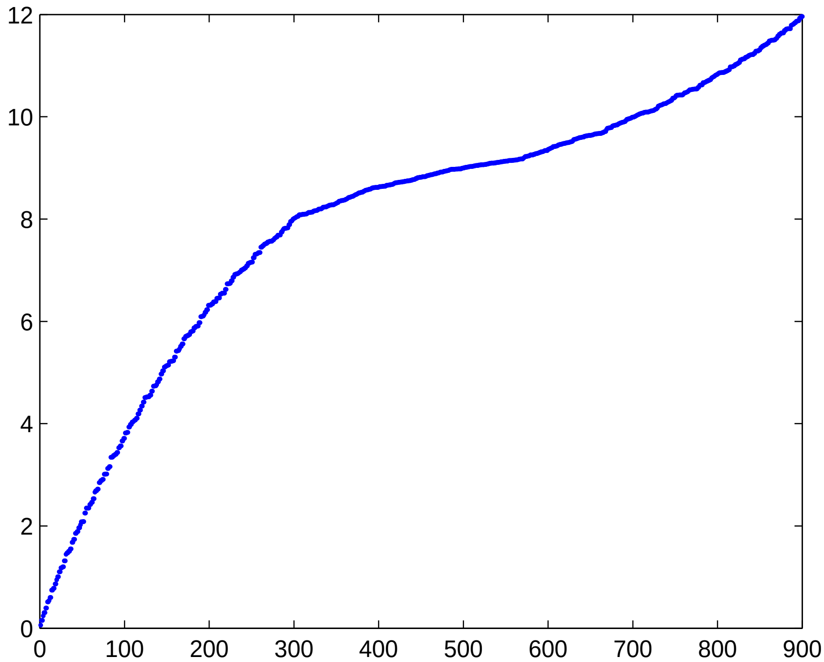

图 4: A的900个特征值从小到大排列