

数值分析实验报告

张钰晖

2015011372, yuhui-zh15@mails.tsinghua.edu.cn, 185-3888-2881

2017 年 6 月 15 日

1 题目描述

5-1 用幂法求下列矩阵按模最大的特征值 λ_1 及其对应的特征向量 \mathbf{x}_1 , 使 $|(\lambda_1)_{k+1} - (\lambda_1)_k| < 10^{-5}$.

$$(1) \mathbf{A} = \begin{bmatrix} 5 & -4 & 1 \\ -4 & 6 & -1 \\ 1 & -4 & 7 \end{bmatrix}$$

$$(2) \mathbf{B} = \begin{bmatrix} 25 & -41 & 10 & -6 \\ -41 & 68 & -17 & 10 \\ 10 & -17 & 5 & -3 \\ -6 & 10 & -3 & 2 \end{bmatrix}$$

2 解题思路

本题要求是求解矩阵的模最大特征值和特征向量, 可以用幂法来实现问题的求解。但常规的幂法由于目前计算机浮点数系统的有限性, 会出现上溢和下溢的问题, 所以需要对常规幂法进行改进。

具体来说, 常规幂法中 $v_k \approx \lambda_1^k x_1$, 而 x_1 可以用 n 个特征向量线性表出。当 k 趋于无穷的时候, 一方面其余特征值被主特征值“掩盖”, 但另一方面, 由于 λ_1 是最大的特征值, 这个系数会趋于无穷大, 会出现溢出的问题。可以使用归一化解决这个问题, 即在幂法的每一步中, 对向量除以其绝对值最大的分量。

通过幂法, 可以通过数次的迭代求解出问题的主特征值和其对应的特征向量。

3 实验结果

算法运行的结果如下：

(1) $\lambda_1 = 12.254322$

$x_1 = [-0.6740, 1.0000, -0.8896]$

经过 15 步迭代收敛，结果误差 $\|Ax_1 - \lambda_1 x_1\| = 1.17 * 10^{-5}$

(2) $\lambda_1 = 98.521698$

$x_1 = [-0.6040, 1.0000, -0.2511, 0.1490]$

经过 5 步迭代收敛，结果误差 $\|Ax_1 - \lambda_1 x_1\| = 1.35 * 10^{-8}$

通过亲自验证，该求解得到的特征值和特征向量十分准确。

4 实验心得

本次试验，我深入理解了幂法收敛的原理，编写了幂法的相关程序。可以看出，幂法求解准确度比较高，收敛也比较快，而且编程实现简单。

通过编写幂法的程序，我亲身体会到了幂法中归一化思想的巨大好处。正是因为归一化操作的存在，使得幂法可以简洁清晰的在计算机编程实现。当然，除了幂法之外还有很多方法可以求解特征值、特征向量，但都没有幂法这么高效、简洁。

同时可以通过一些改进技术，使得幂法可以求最小特征值、以及最接近某个值的特征值等问题，但核心都是幂法。可见，幂法是解决矩阵特征值和特征向量求解问题的有力工具。

5 源代码

```
% 幂法求矩阵最大特征值和对应特征向量，输入初始向量v，矩阵A，迭代误差限epsilon，输出最大特征值lambda1和对应特征向量x1
function [x1, lambda1] = power_method(v, A, epsilon)
u = v;
lambda1 = 0;
last_lambda1 = inf;

% 当不满足终止条件，进行迭代
while abs(lambda1 - last_lambda1) > epsilon
    last_lambda1 = lambda1;
    v = A * u;
    % 选取v中绝对值最大元素
    if abs(max(v)) > abs(min(v))
        lambda1 = max(v);
```

```

else
    lambda1 = min(v);
end
u = v ./ lambda1;
end
x1 = u;

% 初始矩阵和误差限
A = [5,-4,1;-4,6,-4;1,-4,7];
B = [25,-41,10,-6;-41,68,-17,10;10,-17,5,-3;-6,10,-3,2];
epsilon = 1e-5;

% 幂法进行计算
[x1, lambda1] = power_method(rand(3, 1), A, epsilon);
fprintf('lambda1 = %.6f\n', lambda1);
fprintf('x1 =\n');
disp(x1');

% 幂法进行计算
[x1, lambda1] = power_method(rand(4, 1), B, epsilon);
fprintf('lambda1 = %.6f\n', lambda1);
fprintf('x1 =\n');
disp(x1');

```

参考文献

[1] 数值分析与算法 (第二版), 喻文健, 2015.