БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Лабораторная работа №4

Вычисление собственных значений и собственных векторов матрицы методом Данилевского

Вариант 1

Выполнил:

Черепенников Роман

2 курс 8 группа

Преподаватель:

Радкевич Елена Владимировна

Содержание

Постановка задачи -- 3

Алгоритм решения -- 4

Листинг программы -- 7

Входные данные -- 8

Вывод программы -- 9

Вывод -- 10

Постановка задачи

1. Методом Данилевского построить характеристический многочлен матрицы
2. Найти собственные значения
3. Найти собственные вектора

Алгоритм решения

Метод опирается на известный факт, что подобные преобразования не изменяют характеристический многочлен матрицы.

$$det(B - \lambda E) = det(S^{-1}AS - \lambda E) = det(A - \lambda E)$$

Естественно попытаться преобразовать матрицу А к такому виду, где в первой строку - коэффициенты характеристического многочлена. Для этого была придумана форма Фробениуса.

$$\Phi = \begin{pmatrix} P_1 & P_2 & P_3 & \dots & P_{n-2} & P_{n-1} & P_n \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \end{pmatrix}$$

Непосредственно проверка показывает, что p_1, p_2 ... p_n - коэффициенты собственного многочлена:

$$P(\lambda) = (-1)^n(\lambda^n - p_1\lambda^{n-1} - .. - p_n)$$

Основная цель метода Данилевского - привести матрицу к форме Фробениуса с помощью подобных преобразований.

Преобразование матрицы А начнём путём последовательного приведения строк матрицы А к каноническому форме Фробениуса. Возможны 2 случая(в зависимости от коэффициентов a_{ij}): регулярный и нерегулярный. Мы будем рассматривать только регулярный.

Рассмотрим последнюю строку матрицы А - (a_{n1} … $a_{n\,n-1}$ a_{nn}). Приведем ее к последней строке формы Фробениуса, т.е. к (0 … 1 0).

Предположим, что $a_{n\,n-1} \neq 0$. Разделим все элементы (n -1) столбца матрицы А на $a_{n\,n-1}$ и полученный (n - 1) столбец умножим на a_{ni} и вычтем из столбцов под номером i. Проделав это для i = 1, 2, … , n - 2, n получим нужную строку, т.е. будет иметь вид Фробениуса.

Запишем это преобразование в виде умножения исходной матрицы справа на матрицу M_{n-1}:

$$M_{n-1} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ b_{n-1,1} & b_{n-1,2} & \dots & b_{n-1,n-1} & b_{n-1,n} \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

Получим матрицу B = AM_{n-1}.Но это преобразование не является подобным. Домножим матрицу B слева на $(M_{n-1})^{-1}$, она не изменит последнюю строку матрицы B:

$$M_{n-1}^{-1} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{n,n-1} & a_{nn} \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

На этом заканчивается первый шаг преобразования, в результате которого получим матрицу A_1:

$$\mathrm{A}_1 = \begin{pmatrix} a^1_{11} & a^1_{12} & \dots & a^1_{1n-1} & a^1_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ a^1_{n-11} & a^1_{n-12} & \dots & a^1_{n-1n-1} & a^1_{n-1n} \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}$$

Второй шаг преобразования аналогичен первому и состоит в приведении предпоследней строки матрицы A_1 к виду Фробениуса при условии неизменности последней строки матрицы A_1.

После преобразований получим матрицы M_{n-2}, M_{n-1}^{-1} и A_2:

$$\mathrm{A}_2 = \begin{pmatrix} a^2_{11} & a^2_{12} & \dots & a^2_{1n-2} & a^2_{1n-1} & a^2_{1n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 \end{pmatrix}$$

$$\mathrm{M}_{n-2} = \begin{pmatrix} 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots \\ -\frac{a^1_{n-11}}{a^1_{n-1n-2}} & \dots & \frac{1}{a^1_{n-1n-2}} & -\frac{a^1_{n-1n}}{a^1_{n-1n-2}} \\ 0 & & 1 & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

$$\mathrm{M}^{-1}_{n-2} = \begin{pmatrix} 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots \\ a^1_{n-11} & \dots & a^1_{n-1n-2} & a^1_{n-1n} \\ 0 & & 1 & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

Будем выполнять эти преобразования (n - 1) раз. Т.е. нужно чтобы $a_{n\,n-1} \neq 0, \dots, a^{n-1}_{21} \neq 0$. После выполнения этих преобразований получим:

$$A_{n-1} = M_1^{-1} M_2^{-1} \dots M_{n-1}^{-1} A\, M_{n-1} \dots M_2 M_1 = S^{-1} A\, S = \Phi$$

В первой строке полученной матрицы будут стоять коэффициенты характеристического многочлена. Составим его:

$$P(\lambda) = (-1)^n(\lambda^n - p_1\lambda^{n-1} - .. - p_n)$$

Количество операций метода Данилевского в регулярном случае оценивается как $O(n^3)$.

Метод Данилевского позволяет найти только коэффициенты характеристического многочлена. Чтобы найти собственные значения нужно решить уравнение

$$P(\lambda) = 0$$

Пусть мы решили это уравнение и нам известны собственные значения λ_i матрицы A. По определению собственный вектор является решением уравнения:

$$Ax = \lambda_i x$$

Если $B = S^{-1}AS$ и $y = (y_1 \ \dots \ y_n)^T$ есть собственный вектор B, соответствующий собственному значению $\lambda(B)$, то вектор $x = Sy$ есть собственный вектор матрицы соответствующий тому же значению λ.

Таким образом собственные вектора матрицы A можно найти если известны собственные вектора y_i и матрица S. Вычислим собственные вектора матрицы Фробениуса:

$$\Phi y = \lambda_i y$$

Или:

$$\begin{cases} p_1 y_1 + \ldots + p_n y_n = \lambda_i y_1 \\ y_1 = \lambda_i y_2 \\ \ldots \\ y_{n-1} = \lambda_i y_n \end{cases}$$

Положим в качестве $y_n = 1$. Тогда

$$y = (\lambda_i^{n-1} \ \dots \ \lambda_i \ \ 1)^T$$

Домножив на S получим собственный вектор матрицы A.

Листинг программы

```
def danilevsky(A):
    n = A.shape[0]
    print('Матрица A: ')
    print(A)
    F = A
    S = np.identity(n)
    for i in reversed(range(1, n)):
        M_inv = np.identity(n)
        M_inv[i - 1, :] = F[i, :]
        F = np.dot(M_inv, F)
        F = np.dot(F, np.linalg.inv(M_inv))
        S = np.dot(S, np.linalg.inv(M_inv))
    print('Форма Фробениуса:')
    print(F)
    print('Матрица S: ')
    print(S)
    c = F[0, :]
    x = symbols('x')
    lambdas = solve(x ** 5 - c[0] * x ** 4 - c[1] * x ** 3 - c[2] * x ** 2 -
c[3] * x - c[4], x)
    print('Собственный многочлен: ')
    print('(-1)^',n,' * (x^5 -(', c[0], ')x^4 -(', c[1], ')x^3 -(', c[2],
')x^2 -(', c[3], ')x -(', c[4], ')', sep='')
    print('Собственные значения: ')
    print(lambdas)
    print('Собственные значения и соотв. собственные векторы: ')
    for i in range(3):
        l = lambdas[i]
        print(l, ": ", sep="", end="")
        y = [l ** (n - k - 1) for k in range(n)]
        x = np.dot(S, y)
        print(x)

A = np.array([
    [0.6444, 0.0000, -0.1683, 0.1184, 0.1973],
    [-0.0395, 0.4208, 0.0000, -0.0802, 0.0263],
    [0.0132, -0.1184, 0.7627, 0.0145, 0.0460],
    [0.0395, 0.0000, -0.0960, 0.7627, 0.0000],
    [0.0263, -0.0395, 0.1907, -0.0158, 0.5523]
])
```

Входные данные

A

0,6444	0,0000	-0,1683	0,1184	0,1973
-0,0395	0,4208	0,0000	-0,0802	0,0263
0,0132	-0,1184	0,7627	0,0145	0,0460
0,0395	0,0000	-0,0960	0,7627	0,0000
0,0263	-0,0395	0,1907	-0,0158	0,5523

Вывод программы

Матрица A:
```
[[ 0.6444  0.     -0.1683  0.1184  0.1973]
 [-0.0395  0.4208  0.     -0.0802  0.0263]
 [ 0.0132 -0.1184  0.7627  0.0145  0.046 ]
 [ 0.0395  0.     -0.096   0.7627  0.    ]
 [ 0.0263 -0.0395  0.1907 -0.0158  0.5523]]
```

Форма Фробениуса:
```
[[ 3.14290000e+00 -3.89453369e+00  2.37384138e+00 -7.10481541e-01  8.33942433e-02]
 [ 1.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00 0.00000000e+00]
 [ 0.00000000e+00  1.00000000e+00  0.00000000e+00  0.00000000e+00 0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  1.00000000e+00  0.00000000e+00 0.00000000e+00]
 [-1.35716141e-14  3.56558351e-14 -3.47886436e-14  1.00000000e+00  -2.35917963e-15]]
```

Матрица S:
```
[[ 1.09352931e+04 -2.73207557e+04  2.49998282e+04 -9.87777263e+03  1.41726654e+03]
 [-1.95587572e+03  5.13854738e+03 -5.01357191e+03  2.14771863e+03  -3.39993616e+02]
 [-2.16779427e+03  5.49115646e+03 -5.10816444e+03  2.06476982e+03  -3.05883226e+02]
 [-3.07234613e+03  7.95285059e+03 -7.50603725e+03  3.04628476e+03  -4.47833747e+02]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
```

Характеристический многочлен:
(-1)^5 * x^5 -(3.1429000000000022)x^4 -(-3.894533690000007)x^3 -(2.373841376221007)x^2 -(-0.7104815406018847)x -(0.08339424331325174)

Собственные значения:
[0.396501975572364, 0.516344259555490, 0.639749192770339, 0.795152286050903 - 0.0666540572019849*I, 0.795152286050903 + 0.0666540572019849*I]
Собственные значения и соотв. собственные векторы:
0.396501975572364: [-1.74810035289102, -4.64912133004242, -1.55707113741170, -0.219632165898020, 1.00000000000000]
0.516344259555490: [-1.59147446521229, 0.655522862127839, 0.194127703884135, 0.330820385192510, 1.00000000000000]
0.639749192770339: [8.08297985640229, -0.113300619375821, -0.956686792562834, -3.34377338122908, 1.00000000000000]

Вывод

Методом Данилевского можно найти характеристический многочлен матрицы. С помощью него можно найти её собственные значения. А запоминая порядок подобных преобразований и зная собственные значения можно найти и собственные вектора.