

D5 Report 線形代数-固有値

A++

さんしゃさんよー

- 36021341 末富光輝
- 36021363 道一世
- 36021364 古谷颯大

課題：ページランク

1) 上記手順を参考にして, pythonでページランクを求めよ

```
In [89]: from pprint import pprint
from numpy import array, zeros, diagflat, dot, transpose
from scipy.linalg import eig

A = array([[0, 1, 1, 1, 1, 0, 1],
           [1, 0, 0, 0, 0, 0, 0],
           [1, 1, 0, 0, 0, 0, 0],
           [0, 1, 1, 0, 1, 0, 0],
           [1, 0, 1, 1, 0, 1, 0],
           [1, 0, 0, 0, 1, 0, 0],
           [0, 0, 0, 0, 1, 0, 0]])

diag = []
for i in range(0, 7):
    tmp = 0.0
    for j in range(0, 7):
        tmp += A[i, j]
    diag.append(1.0/tmp)

D = diagflat(diag)
tA = dot(transpose(A), D)
print(tA)
l, V = eig(tA)
v0 = V[:, 0]
pprint(v0)

x = array([1/7, 1/7, 1/7, 1/7, 1/7, 1/7, 1/7])
pprint(dot(tA, x))
```

```
[[ 0.000  1.000  0.500  0.000  0.250  0.500  0.000]
 [ 0.200  0.000  0.500  0.333  0.000  0.000  0.000]
 [ 0.200  0.000  0.000  0.333  0.250  0.000  0.000]
 [ 0.200  0.000  0.000  0.000  0.250  0.000  0.000]
 [ 0.200  0.000  0.000  0.333  0.000  0.500  1.000]
 [ 0.000  0.000  0.000  0.000  0.250  0.000  0.000]
 [ 0.200  0.000  0.000  0.000  0.000  0.000  0.000]]
array([0.699+0. j, 0.383+0. j, 0.324+0. j, 0.243+0. j, 0.412+0. j, 0.103+0. j,
       0.14 +0. j])
array([ 0.321,  0.148,  0.112,  0.064,  0.29,  0.036,  0.029])
```

- 隣接行列Aを作成

- 行列Aの各行要素を合計し、それを逆数にして各行に掛けて対角化(行列D)
- 行列Aを転置した行列と行列Dを掛け、各列の合計が1になるように規格化(行列tA)
- 推移確率行列tAの固有値と固有ベクトルを計算し、最も大きい固有値をもつ1列目の固有ベクトルを表示
- 行列tAと各要素1/7のベクトルxを掛けて表示(規格化)
- 上の操作から得られた値では0.321と0.290が大きい数値となっていてID=1, ID=5はページランクが高く、優先度が高い

2) このような問題ではすべての固有値・固有ベクトルを求める必要はなく、最大の固有値を示す固有ベクトルを求めるだけでよい。初期ベクトルを適当に決めて、何度も推移確率行列を掛ける反復法でページランクを求めよ。

```
In [95]: import numpy as np
from pprint import pprint

np.set_printoptions(formatter={'float': '{: 0.3f}'.format})
#pprint(A)
#pprint(D)
#pprint(transpose(A))
tA = dot(transpose(A), D)
pprint(tA)

l, V = eig(tA)
v0 = V[:, 0]
pprint(l)
pprint(v0)

x = np.ones(A.shape[0]) / A.shape[0]

for _ in range(20):
    x = np.dot(tA, x)
    pprint(x)
#pprint(x)
```

```

array([[ 0.000,  1.000,  0.500,  0.000,  0.250,  0.500,  0.000],
       [ 0.200,  0.000,  0.500,  0.333,  0.000,  0.000,  0.000],
       [ 0.200,  0.000,  0.000,  0.333,  0.250,  0.000,  0.000],
       [ 0.200,  0.000,  0.000,  0.000,  0.250,  0.000,  0.000],
       [ 0.200,  0.000,  0.000,  0.333,  0.000,  0.500,  1.000],
       [ 0.000,  0.000,  0.000,  0.000,  0.250,  0.000,  0.000],
       [ 0.200,  0.000,  0.000,  0.000,  0.000,  0.000,  0.000]])
array([ 1. +0.j , -0.444+0.234j, -0.444-0.234j,  0.027+0.314j,
        0.027-0.314j, -0.166+0.j ,  0. +0.j  ])
array([0.699+0.j, 0.383+0.j, 0.324+0.j, 0.243+0.j, 0.412+0.j, 0.103+0.j,
        0.14 +0.j])
array([ 0.321,  0.148,  0.112,  0.064,  0.290,  0.036,  0.029])
array([ 0.294,  0.142,  0.158,  0.137,  0.132,  0.073,  0.064])
array([ 0.290,  0.184,  0.137,  0.092,  0.205,  0.033,  0.059])
array([ 0.320,  0.157,  0.140,  0.109,  0.164,  0.051,  0.058])
array([ 0.294,  0.170,  0.141,  0.105,  0.184,  0.041,  0.064])
array([ 0.308,  0.165,  0.140,  0.105,  0.178,  0.046,  0.059])
array([ 0.302,  0.166,  0.141,  0.106,  0.178,  0.045,  0.062])
array([ 0.304,  0.166,  0.140,  0.105,  0.180,  0.045,  0.060])
array([ 0.304,  0.166,  0.141,  0.106,  0.178,  0.045,  0.061])
array([ 0.303,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.304,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.303,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.304,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.304,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.304,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.304,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.304,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.304,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.304,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.304,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])
array([ 0.304,  0.166,  0.141,  0.105,  0.179,  0.045,  0.061])

```

OK

- 行列Aを転置し規格化(行列tA)
- 行列tAの固有値と固有ベクトルを求め、最大固有値である1列目の固有ベクトルを表示
- 上の操作で求めた行列Aの1列目を用いて適当に規格化されたベクトルxを作成
- 20回程度、推移確率行列とベクトルxを掛ける操作を繰り返した
- その結果、13回目から安定した値になってきた
- よって、ID=1が一番ページランクが大きくて優先度が高い。ID=5, 2, 3辺りも比較的大きい値になっている。

考察

2.の1回目のページランクの値は1.で求めたページランクと等しくなっているが2.の2回目以降の値は13回目収束していることから、精度が良くなっていると考えた。

課題：対角化

1. 次の行列 A の固有値をpythonで求めよ。また、対角化行列 P を求めて、対角化せよ。

$$A = \begin{bmatrix} 4 & 5 & 5 \\ -4 & -5 & -7 \\ 4 & 4 & 6 \end{bmatrix}$$

```
In [86]: import numpy as np
np.set_printoptions(formatter={'float': '{: 0.3f}'.format})

A = np.array([[4, 5, 5], [-4, -5, -7], [4, 4, 6]])
l, V = np.linalg.eig(A)
print(l)
#print(V)

from pprint import pprint
P = np.linalg.inv(V)
dP = np.dot(P, np.dot(A, V))
np.set_printoptions(precision=3, suppress=True)
print(format(P))
print(format(dP))
```

```
[ 4.000  2.000 -1.000]
[[-3.   -3.   -3.  ]
 [ 2.828  2.828  1.414]
 [-0.    1.414  1.414]]
[[ 4.  0.  0.]
 [ 0.  2. -0.]
 [ 0.  0. -1.]]
```

OK.

- 浮動小数点数の表示を小数点以下3桁までに設定
- 行列Aの固有値lと固有ベクトルVを計算
- 行列Aの固有値を求められた
- 行列Aの固有ベクトルを対角化(行列P)
- 対角化行列Pを求められた
- 対角化行列Pと、行列Aと固有ベクトルVの積を掛けて対角化(行列dP)
- 対角化行列Pの対角化することができた