HW2

December 13, 2020

1 Jacob Malyshev

2 Variant 20

[154]:
```
import numpy as np
import matplotlib.pyplot as plt
import math
from scipy.special import comb
import sympy as sp
np.set_printoptions(suppress=True, precision=2, linewidth=120)
```

[39]:
```
from IPython.display import display, Math
```

2.1 Useful functions

[7]:
```
A_hat
```

[7]:
```
array([[ 3,  0],
       [ 4, -4]])
```

[9]:
```
np.sum(A_hat, axis=0)
```

[9]:
```
array([ 7, -4])
```

[295]:
```
def norm_1(A):
    return np.max(np.sum(np.abs(A),axis=0))
def norm_inf(A):
    return np.max(np.sum(np.abs(A),axis=1))
def norm_1vec(b):
    return np.sum(np.abs(b))
def norm_2vec(b):
    return np.sqrt(np.sum(b**2))
def norm_2(A):
    return np.sqrt(np.max(np.linalg.eig(A.T@A)[0]))
```

3 Task 1

3.1 Find the best approximation matrix A_1 of rank 2 of the matrix A in the norm $||\cdot||_2$ and find $||A - A_1||_2$, where

$$A = \begin{bmatrix} 44 & -80 & -5 & -96 \\ 4 & 32 & -106 & 60 \\ -80 & 8 & 14 & -66 \end{bmatrix}$$

[292]:
```
A = np.array([[44,-80,-5,-96],
              [4,32,-106,60],
              [-80, 8, 14, -66]])
```

Firstly, we have to find singular value decomposition of A: $A = V\Sigma U^*$

[293]:
```
U,Sigma,V = np.linalg.svd(A)
```

Rank of A_1 must be equal to 2, so we choose first two singular values of A and build new Σ_1 in the following way: take first two rows of Σ, other rows consist of zeros.

[294]:
```
Sigma1 = np.array([[Sigma[0],0,0,0],
                   [0,Sigma[1],0,0],
                   [0, 0, 0, 0]])
```

3.1.1

$$\Sigma = \begin{pmatrix} 162 & 0 & 0 & 0 \\ 0 & 108 & 0 & 0 \\ 0 & 0 & 81 & 0 \end{pmatrix}, \Sigma_1 = \begin{pmatrix} 162 & 0 & 0 & 0 \\ 0 & 108 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Finally, we can get A_1 matrix, using the following formula: $A_1 = V\Sigma_1 U^*$

[295]:
```
A1 = U @ Sigma1 @ V
```

$$\boldsymbol{A_1 = \begin{pmatrix} 56 & -80 & 16 & -84 \\ 28 & 32 & -64 & 84 \\ -56 & 8 & 56 & -42 \end{pmatrix}}$$

Matrix A_1 is the best approximation of the matrix A in the norm $||\cdot||_2$. Let's find $||A - A_1||_2$

[297]:
```
dA = A - A1
```

To calculate $||A - A_1||_2$, we have to do several steps:

1) Calculate $A - A_1$:

$$A - A_1 = \begin{pmatrix} -12 & 0 & -21 & -12 \\ -24 & 0 & -42 & -24 \\ -24 & 0 & -42 & -24 \end{pmatrix}$$

2) Calculate $||A - A_1||_2 = \sqrt{\lambda_{max}((A - A_1)^*(A - A_1))}$

[298]:
```
dA_star = dA.T
eigenvalues,_ = np.linalg.eig(dA_star @ dA)
```

```
lambda_max = np.max(eigenvalues)
norm2 = np.sqrt(lambda_max)
```

Finally, we found, that $||\boldsymbol{A} - \boldsymbol{A_1}||_2 = \mathbf{81}$

3.2 Answer:

3.3 1)

$$A_1 = \begin{pmatrix} 56 & -80 & 16 & -84 \\ 28 & 32 & -64 & 84 \\ -56 & 8 & 56 & -42 \end{pmatrix}$$

3.4 2)

$$||A - A_1||_2 = 81$$

4 Task 2

4.1 Estimate the relative error of the approximate solution $(1, 1)$ of the system $AX = b$ in the norms $|\cdot|_1$ and $|\cdot|_2$ using the condition number of the matrix A, where

4.2

$$A = \begin{pmatrix} 2.92 & -0.02 \\ 3.96 & -4.13 \end{pmatrix}, b = \begin{pmatrix} 3.11 \\ 0.02 \end{pmatrix}$$

Firslty, let's write $\widehat{A}$, $\widehat{b}$, $\widehat{x}$:

$$\widehat{A} = \begin{pmatrix} 3 & 0 \\ 4 & -4 \end{pmatrix}, \widehat{b} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \widehat{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

[4]:
```
A = np.array([[2.92, -0.02],
              [3.96, -4.13]])
b = np.array([3.11,0.02]).reshape(-1,1)
A_hat = np.array([[3,0],
                  [4,-4]])
b_hat = np.array([[3],
                  [0]])
x_hat = np.array([[1],
                 [1]])
```

Then we have to calculate condition numbers $\chi_1(\widehat{A})$,$\chi_2(\widehat{A})$:

$$\chi_1(\widehat{A}) = ||\widehat{A}||_1 ||\widehat{A}^{-1}||_1,$$

where $||\widehat{A}||_1 = \max_j |\widehat{A}^j|$

$$\chi_2(\widehat{A}) = \sqrt{\frac{\lambda_{max}(\widehat{A}^* \widehat{A})}{\lambda_{min}(\widehat{A}^* \widehat{A})}}$$

[47]:
```
A_hat_inv = np.linalg.inv(A_hat)
Chi_1 = norm_1(A_hat) * norm_1(A_hat_inv)
```

To find $\chi_2(\widehat{A})$ we have to find eigenvalues of $\widehat{A}^* \widehat{A}$, using the following steps:

[48]:
```python
Dot_AhatStar_Ahat = A_hat.T @ A_hat
```

[49]:
```python
Chi_2 = np.sqrt(37.12/3.879)
```

$$\widehat{A}^*\widehat{A} = \begin{pmatrix} 25 & -16 \\ -16 & 16 \end{pmatrix}$$

$$\begin{vmatrix} 25-\lambda & -16 \\ -16 & 16-\lambda \end{vmatrix} = 0$$

$$(25-\lambda)(16-\lambda) - 256 = 0$$

We solve this quadratic equation and get two eigenvalues: $\lambda_1 = \lambda_{max}(\widehat{A}^*\widehat{A}) = 37.12$, $\lambda_2 = \lambda_{min}(\widehat{A}^*\widehat{A}) = 3.87$, and we can calculate $\chi_2(\widehat{A})$

Let's write our conditional numbers: $\chi_1(\widehat{A}) = 4.67$, $\chi_2(\widehat{A}) = 3.1$.

Finally, we can estimate the relative error of $\widehat{x}$ in the norms $|\cdot|_1, |\cdot|_2$ using the following formula:

$$\frac{\delta b + \delta A}{\chi(A)} \leq \delta x \leq \chi(A)(\delta b + \delta A),$$

To solve this task we have to find δA and δb. To do this we can use the following formulas:

$$\delta A = \frac{||\Delta A||}{||A||} = \frac{||\widehat{A} - A||}{||A||}$$

$$\delta b = \frac{|\Delta b|}{|b|} = \frac{|\widehat{b} - b|}{|b|}$$

For $|\cdot|_1$ we define two functions: 'norm_1(A)' for matrix and 'norm_1vec(b)' for vector: $\delta_1 A = 0.022$, $\delta_1 b = 0.042$. So, $\boldsymbol{\delta_1 x \in [0.0135, 0.296]}$

[59]:
```python
# First norm
deltaA_1 = norm_1(A_hat - A)/norm_1(A)
deltab_1 = norm_1vec(b_hat - b)/norm_1vec(b)
x_min = (deltaA_1 + deltab_1)/Chi_1
x_max = (deltaA_1 + deltab_1)*Chi_1
```

[72]:
```python
print('x_min in norm 1:{}'.format(x_min))
print('x_max in norm 1:{}'.format(x_max))
```

```
x_min in norm 1:0.013571972551558691
x_max in norm 1:0.29556740223394473
```

[65]:
```python
print((A_hat - A).T @ (A_hat - A))
```

```
[[0.008 0.007]
 [0.007 0.017]]
```

For $|\cdot|_2$ we have to calculate eigenvalues of $(\widehat{A}-A)^*(\widehat{A}-A)$ and A. We also have to calculate $|b|_2$ and $|\widehat{b}-b|_2$, using $|\cdot|_2$ for vectors.

$$(\widehat{A}-A)^*(\widehat{A}-A) = \begin{pmatrix} 0.008 & 0.007 \\ 0.007 & 0.017 \end{pmatrix}$$

$$\begin{vmatrix} 0.008-\lambda & 0.007 \\ 0.007 & 0.017-\lambda \end{vmatrix} = 0$$

$$(0.008-\lambda)(0.017-\lambda) - 0.000049 = 0$$

We solve this quadratic equation and get two eigenvalues: $\lambda_1 = 0.004, \lambda_2 = 0.021 = \lambda_{max}((\widehat{A}-A)^*(\widehat{A}-A))$. Then, we have to find $\lambda_{max}(A)$

$$A = \begin{pmatrix} 2.92 & -0.02 \\ 3.96 & -4.13 \end{pmatrix}$$

$$\begin{vmatrix} 2.92-\lambda & -0.02 \\ 3.96 & -4.13-\lambda \end{vmatrix} = 0$$

$$(2.92-\lambda)(-4.13-\lambda) - 0.083 = 0$$

We solve this quadratic equation and get two eigenvalues: $\lambda_1 = -4.119, \lambda_2 = 2.909 = \lambda_{max}(A)$ and we can find $\delta_2 A = \frac{\sqrt{0.021}}{\sqrt{2.909}} = 0.085$

The last step is to find $\delta_2 b$:

$$|b|_2 = \sqrt{\sum_{i=1}^{n} b_i^2} = 3.11$$

$$\Delta b = \begin{pmatrix} -0.11 \\ -0.02 \end{pmatrix} \to |\Delta b|_2 = \sqrt{\sum_{i=1}^{n} \Delta b_i^2} = 0.11$$

Finally, we can find $\delta_2 b = \frac{0.11}{3.11} = 0.035$ and can calculate $\boldsymbol{\delta_2 x \in [0.0388, 0.371]}$

4.3 Answer:

4.4 $\delta_1 x = [0.0135, 0.296]$

4.5 $\delta_2 x = [0.0388, 0.371]$

5 Task 3

5.1 Solve the system approximately and estimate the error of the solution in the norms $|\cdot|_1, |\cdot|_2, |\cdot|_\infty$:

5.2

$$\begin{cases} 3(2+\epsilon_1)x + 2(-5+\epsilon_2)y = 1+\epsilon_3 \\ 1x + (-3+\epsilon_1)y = -1+\epsilon_4 \end{cases}$$

5.3 where the unknown numbers ϵ_j satisfy the conditions $|\epsilon_j| < 0.05$ for all $j = 1, ..., 4$

Firstly, we write our matrix A, $\widehat{A}$, b, $\widehat{b}$:

$$A = \widehat{A} + \Delta A = \begin{pmatrix} 6 & -10 \\ 1 & -3 \end{pmatrix} + \begin{pmatrix} \epsilon_1 & 2\epsilon_2 \\ 0 & \epsilon_1 \end{pmatrix}$$

$$b = \widehat{b} + \Delta b = \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \begin{pmatrix} \epsilon_3 \\ \epsilon_4 \end{pmatrix}$$

We can find approximate solution: $\widehat{A}\widehat{x} = \widehat{b} \Rightarrow \widehat{x} = \widehat{A}^{-1}\widehat{b}$:

$$\widehat{x} = \begin{pmatrix} 1.625 \\ 0.875 \end{pmatrix}$$

[276]:
```
A_hat = np.array([[6,-10],
                  [1,-3]])
b_hat = np.array([1,-1]).reshape(-1,1)

A_hat_inv = np.linalg.inv(A_hat)

x_hat = A_hat_inv @ b_hat
```

To find the relatrive error of this solution in different norms we can use the following formula:

$$\frac{\delta A + \delta b}{\chi(A)} \leq \delta x \leq (\delta A + \delta b)\chi(A)$$

Finding relative error in $|.|_1$:

$$\delta_1 A = \frac{||\Delta A||_1}{||\widehat{A}||_1} = \frac{max\{\epsilon_1, 2\epsilon_2 + \epsilon_1\}}{max\{7, 13\}} < \frac{0.15}{13} = 0.011$$

$$\delta_1 b = \frac{|\Delta b|_1}{|\widehat{b}|_1} < \frac{0.1}{2} = 0.05$$

$$\chi_1(\widehat{A}) = ||\widehat{A}||_1||\widehat{A}^{-1}||_1 = 26$$

[289]:
```
print('min:{}'.format((0.011+0.05)/chi_A1 ))
print('max:{}'.format((0.011+0.05)*chi_A1 ))
```

```
min:0.0023461538461538468
max:1.5859999999999996
```

To find $|| \cdot ||_1$ and $| \cdot |_1$ we used predefined functions 'norm_1(A)' and 'norm_1vec(b)'. Finally, we can estimate relative error of approximate solution: $\delta_1 x = [0, 1.586)$

[286]:
```
chi_A1 = norm_1(A_hat) * norm_1(np.linalg.inv(A_hat))
print(chi_A1)
```

```
25.999999999999993
```

Finding relative error in $|\cdot|_\infty$:

$$\delta_\infty A = \frac{||\Delta A||_\infty}{||\widehat{A}||_\infty} = \frac{max\{\epsilon_1, 2\epsilon_2 + \epsilon_1\}}{max\{16, 4\}} < \frac{0.15}{16} = 0.0093$$

$$\delta_\infty b = \frac{|\Delta b|_\infty}{|\widehat{b}|_\infty} < \frac{0.05}{1} = 0.05$$

$$\chi_\infty(\widehat{A}) = ||\widehat{A}||_\infty ||\widehat{A}^{-1}||_\infty = 26$$

[293]:
```
chi_Ainf = norm_inf(A_hat) * norm_inf(np.linalg.inv(A_hat))
print(chi_Ainf)
```

```
25.999999999999996
```

[294]:
```
print('min:{}'.format((0.0093+0.05)/chi_A1 ))
print('max:{}'.format((0.0093+0.05)*chi_A1 ))
```

```
min:0.0022807692307692316
max:1.5417999999999996
```

Finally, we can estimate relative error of approximate solution: $\delta_\infty x = [0, 1.542)$
Finding relative error in $|\cdot|_\infty$:

$$\delta_2 A = \frac{||\Delta A||_2}{||\widehat{A}||_2} = \frac{\sqrt{\lambda_{max}(\Delta A^* \Delta A)}}{\sqrt{\lambda_{max}(\widehat{A}^* \widehat{A})}} < \frac{0.366}{12.064} = 0.03$$

$$\delta_2 b = \frac{|\Delta b|_2}{|\widehat{b}|_2} < \frac{0.005}{1.414} = 0.0035$$

$$\chi_2(\widehat{A}) = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}} = \sqrt{\frac{145.56}{0.439}} = 18.19$$

[312]:
```
print('min:{}'.format((0.03+0.0035)/18.19 ))
print('max:{}'.format((0.03+0.0035)*18.19 ))
```

```
min:0.0018416712479384276
max:0.609365
```

Finally, we can estimate relative error of approximate solution: $\delta_2 x = [0, 0.0609)$

5.4 Answer

5.5

$$\delta_1 x = [0, 1.586)$$

5.6

$$\delta_\infty x = [0, 1.542)$$

5.7

$$\delta_2 x = [0, 0.609)$$

6 Task 4

6.1 Find the approximate inverse matrix to the matrix A and evaluate the approximation error with respect to the uniform norm $||\cdot||_1$ if the elements of the matrix A are known with an absolute error of 0.01:

$$A \approx \begin{pmatrix} 8 & -8 \\ -8 & -8 \end{pmatrix}$$

[81]:
```
A_hat = np.array([[8, -8],
                  [-8, -8]])
```

[83]:
```
np.linalg.inv(A_hat)
```

[83]:
```
array([[ 0.062, -0.062],
       [-0.062, -0.062]])
```

Firstly, let's rewrite matrix A in the following form:

$$A = \widehat{A} + \Delta A = \begin{pmatrix} 8 & -8 \\ -8 & -8 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} & \epsilon_{12} \\ \epsilon_{21} & \epsilon_{22} \end{pmatrix}$$

Let's find $\widehat{A}^{-1}$:

$$\widehat{A}^{-1} = \frac{1}{|\widehat{A}|}\begin{pmatrix} -8 & 8 \\ 8 & 8 \end{pmatrix} = \mathbf{\frac{1}{-128}\begin{pmatrix} -8 & 8 \\ 8 & 8 \end{pmatrix}}$$

We can find the approximation error with respect to the uniform norm $||\cdot||_1$ using the following formula:

$$\delta_1 A^{-1} \leq \frac{\chi_1(\widehat{A})\delta_1\epsilon}{1 - \chi_1(\widehat{A})\delta_1\epsilon}, \delta_1\epsilon = \frac{||\epsilon||_1}{||\widehat{A}||_1}$$

$||\epsilon||_1 = ||\Delta A||_1 = \max\limits_j |\Delta A^j| = \max\limits_j\{|0.01| + |0.01|, |0.01| + |0.01|\} = 0.02$

$||\widehat{A}||_1 = \max j|\widehat{A}^j| = \max\limits_j\{16, 16\} = 16 \Rightarrow \delta_1\epsilon = \frac{0.02}{16} = 0.00125$

We have to calculate $\chi_1(\widehat{A}) = ||\widehat{A}||_1||\widehat{A}^{-1}||_1$, we have already calculated first multiplier, so: $||\widehat{A}^{-1}||_1 = \max\limits_j\{0.128, 0.128\} = 0.128$. So, $\chi_1(\widehat{A}) = 16 \cdot 0.128 = 2.048$

Finally, we can find $\delta_1 A^{-1}$:

$$\boldsymbol{\delta_1 A^{-1} \leq \frac{2.048 \cdot 0.00125}{1 - 2.048 \cdot 0.00125} = 0.0026}$$

6.2 Answer

6.3 1)

$$\widehat{A}^{-1} = \frac{1}{-128}\begin{pmatrix} -8 & 8 \\ 8 & 8 \end{pmatrix}$$

6.4 2)

$$\delta_1 A^{-1} \leq 0.0026$$

7 Task 5

7.1 Use simple iteration method for finding the solution of the given linear system

7.2

$$\begin{cases} 24x + 1y + 2z = 2 \\ 1x + 26y + 6z = 6 \\ 5x + 5y + 25z = 9 \end{cases}$$

7.3 Determine the iteration number after which the approximation error for each coordinate does not exceed 0.01 and find the corresponding approximate solution. Start with $x_0 = [0,0,0]^T$.

[198]:
```
x_0 = np.array([0,0,0]).reshape(-1,1)
```

[199]:
```
A = np.array([[26, 4, 5],
              [7, 24, 6],
              [5, 3, 27]])
B = np.array([5,6,2]).reshape(-1,1)
```

[200]:
```
c = max(A[0][0],A[1][1],A[2][2])
```

[201]:
```
C = np.array([[1/c,0,0],
              [0,1/c,0],
              [0,0,1/c]])
```

[202]:
```
C
```

[202]:
```
array([[0.037, 0.   , 0.   ],
       [0.   , 0.037, 0.   ],
       [0.   , 0.   , 0.037]])
```

We have the following matrices:

$$A = \begin{pmatrix} 24 & 1 & 2 \\ 1 & 26 & 6 \\ 5 & 5 & 25 \end{pmatrix}, B = \begin{pmatrix} 2 \\ 6 \\ 9 \end{pmatrix}, x^0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

So, matrix C is:

$$C = \begin{pmatrix} 0.038 & 0 & 0 \\ 0 & 0.038 & 0 \\ 0 & 0 & 0.038 \end{pmatrix}$$

Then we have to find matrix P:

$$P = I - CA = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 0.038 & 0 & 0 \\ 0 & 0.038 & 0 \\ 0 & 0 & 0.038 \end{pmatrix} \begin{pmatrix} 24 & 1 & 2 \\ 1 & 26 & 6 \\ 5 & 5 & 25 \end{pmatrix} = \begin{pmatrix} 0.0769 & -0.0385 & -0.0769 \\ -0.0385 & 0 & -0.2308 \\ -0.1923 & -0.1923 & 0.0385 \end{pmatrix}$$

[203]:
```
P = np.identity(3) - (C @ A)
```

Then we have to find b:

$$b = CB = \begin{pmatrix} 0.038 & 0 & 0 \\ 0 & 0.038 & 0 \\ 0 & 0 & 0.038 \end{pmatrix} \begin{pmatrix} 2 \\ 6 \\ 9 \end{pmatrix} = \begin{pmatrix} 0.0769 \\ 0.2308 \\ 0.3462 \end{pmatrix}$$

[204]:
```
b = C @ B
```

[205]:
```
P_norm1 = norm_1(P)
print(P_norm1)
```

```
0.4814814814814815
```

Calculate norm of matrix P using our function 'norm_1(A)': $||P||_1 = 0.346 < 1$, so iteration process is convergent.

Finally, we can find x using the following process:

$$x^{k+1} = Px^k + b$$

[210]:
```
x_k1 = x_0
cond = True
k = 0
while cond:
    k+=1
    x_k2 = P @ x_k1 + b
    if (np.absolute(x_k2 - x_k1) < 0.01).all():
        cond = False
    else:
        x_k1 = x_k2
print('Solution:{}'.format(x_k1.T))
print('Iteration:{}'.format(k))
```

```
Solution:[[0.1607 0.2014 0.0269]]
Iteration:4
```

7.4 Answer

7.5 $x^4 = \begin{pmatrix} 0.1607 \\ 0.2014 \\ 0.0269 \end{pmatrix}$

7.6 We need 4 iterations

8 Task 6

8.1 Find the most influential vertex in the graph using the Page Rank algorithm with $\beta = 0.15$, where the graph adjacency matrix is defined as follows

8.2

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

[214]:
```
A = np.array([[1,1,1,1,1],
              [0,0,1,0,1],
              [0,0,0,1,1],
              [0,0,1,0,1],
              [0,1,0,0,1]])
P = np.array([el/sum(row) for row in A for el in row]).reshape(5,5)
beta = 0.15
```

So, our matrix P

$$P = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0.5 & 0 & 0 & 0.5 \end{bmatrix}$$

However, we have $\beta = 0.15$, so we have to find $\tilde{P} = P(1-\beta) + \beta Q$, where

$$Q = \begin{bmatrix} 0.04 & 0.04 & 0.04 & 0.04 & 0.04 \\ 0.04 & 0.04 & 0.04 & 0.04 & 0.04 \\ 0.04 & 0.04 & 0.04 & 0.04 & 0.04 \\ 0.04 & 0.04 & 0.04 & 0.04 & 0.04 \\ 0.04 & 0.04 & 0.04 & 0.04 & 0.04 \end{bmatrix}$$

$$\tilde{P} = \begin{bmatrix} 0.176 & 0.176 & 0.176 & 0.176 & 0.176 \\ 0.006 & 0.006 & 0.431 & 0.0.06 & 0.431 \\ 0.006 & 0.006 & 0.006 & 0.431 & 0.431 \\ 0.006 & 0.006 & 0.431 & 0.006 & 0.431 \\ 0.006 & 0.431 & 0.006 & 0.006 & 0.431 \end{bmatrix}$$

[220]:
```
Q = np.ones((5,5)) * 1/A.shape[0]**2
```

```
[221]: P_hat = P*(1-beta) + Q*beta
```

In the initial state $x_0 = \begin{pmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{pmatrix}^T$. Let's calculate next steps using iterative process.

```
[231]: x_0 = np.array([0.2,0.2,0.2,0.2,0.2]).reshape(-1,1)
for i in range(4):
    x_i = P_hat.T @ x_0
    x_0 = x_i
    print('x_{}:{}'.format(i+1,x_0.T))
```

```
x_1:[[0.04   0.125 0.21   0.125 0.38 ]]
x_2:[[0.0121 0.1736 0.1183 0.1013 0.3691]]
x_3:[[0.0067 0.1636 0.1235 0.057  0.3307]]
x_4:[[0.0052 0.1458 0.099  0.0577 0.292 ]]
```

We get:

$$x_1 = \begin{pmatrix} 0.04 & 0.125 & 0.21 & 0.125 & 0.38 \end{pmatrix}^T$$

$$x_2 = \begin{pmatrix} 0.012 & 0.173 & 0.118 & 0.101 & 0.369 \end{pmatrix}^T$$

$$x_3 = \begin{pmatrix} 0.007 & 0.163 & 0.123 & 0.057 & 0.33 \end{pmatrix}^T$$

$$x_4 = \begin{pmatrix} 0.0052 & 0.146 & 0.1 & 0.058 & 0.29 \end{pmatrix}^T$$

We see, that the most influential vertex is vertex number 5

8.3 Answer

8.4 Vertex 5

9 Task 7

9.1 Find the value $f(A)$ of the function $f(l) = e^{l+1}$, where

9.2

$$A = \begin{pmatrix} -6 & 26 & 9 \\ -6 & 22 & 7 \\ 8 & -28 & -8 \end{pmatrix}$$

Firstly, we have to find Jordan form of A. To do this we find eigenvalues, and then eigenvectors.

$$\begin{vmatrix} -6-\lambda & 26 & 9 \\ -6 & 22-\lambda & 7 \\ 8 & -28 & -8-\lambda \end{vmatrix} = 0$$

$$(-6-x)(22-x)(-8-x) + 6 \cdot 28 \cdot 9 + 26 \cdot 7 \cdot 8 - (72(22-x) + 7(-6-x)(-28) + (-6) \cdot 26(-8-x)) = 0$$

From this equation we got $\lambda_{1,2} = 2$, $\lambda_3 = 4$ and we have to find their eigenvectors:

first step:$\lambda = 2$

$$(A-2I) = \begin{pmatrix} -8 & 26 & 9 \\ -6 & 20 & 7 \\ 8 & -28 & -10 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0.5 \\ 0 & 0 & 0 \end{pmatrix} \Rightarrow v_1 = \begin{pmatrix} -1 \\ -1 \\ 2 \end{pmatrix}$$

second step:

$$(A-2I)^2 = \begin{pmatrix} -20 & 60 & 20 \\ -16 & 48 & 16 \\ 24 & -72 & -24 \end{pmatrix} \sim \begin{pmatrix} 1 & -3 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \Rightarrow v_2 = \begin{pmatrix} 3c_1 + c_2 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

third step:

$$v_2' = (A-2I)v_2 = \begin{pmatrix} 1 \\ 1 \\ -2 \end{pmatrix}$$

forth step:$\lambda = 4$

$$(A-4I) = \begin{pmatrix} -10 & 26 & 9 \\ -6 & 18 & 7 \\ 8 & -28 & -12 \end{pmatrix} \sim \begin{pmatrix} 8 & -28 & -12 \\ 0 & -24 & -16 \\ 0 & 0 & 0 \end{pmatrix} \Rightarrow v_4 = \begin{pmatrix} -\frac{23}{6}c \\ -\frac{2}{3}c \\ c \end{pmatrix} = \begin{pmatrix} -23 \\ -4 \\ 6 \end{pmatrix}$$

Then, we have to build matrix J, T:

$$J = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

$$T = \begin{pmatrix} 1 & 1 & -23 \\ 1 & 0 & -4 \\ -2 & 1 & 6 \end{pmatrix}$$

Finally, we can find $f(A) = e^A e$ in the following form: $f(A) = Tf(J)T^{-1}$

$$e^J = \begin{pmatrix} e^2 & e & 0 \\ 0 & e^2 & 0 \\ 0 & 0 & e^4 \end{pmatrix} \Rightarrow f(J) = ee^J = \begin{pmatrix} e^3 & e & 0 \\ 0 & e^3 & 0 \\ 0 & 0 & e^5 \end{pmatrix}$$

$$f(A) = \begin{pmatrix} 1 & 1 & -23 \\ 1 & 0 & -4 \\ -2 & 1 & 6 \end{pmatrix} \begin{pmatrix} e^3 & e & 0 \\ 0 & e^3 & 0 \\ 0 & 0 & e^5 \end{pmatrix} \begin{pmatrix} -0.235 & 1.706 & 0.235 \\ -0.118 & 2.353 & 1.118 \\ -0.059 & 0.176 & 0.059 \end{pmatrix} = \begin{pmatrix} 70.94 & -185.21 & -60.83 \\ 10.788 & -19.54 & -8.07 \\ -16.02 & 37.194 & 17.975 \end{pmatrix}$$

[268]:
```
A = np.array([[-6,26,9],
              [-6,22,7],
              [8,-28,-8]])

# A = np.array([[0,0,-1],
#               [-14,-10,-13],
#               [20,16,20]])
```

```
# A = np.array([[2,0],
#                  [-1,1]])
```

[272]:
```
T = np.array([[1,1,-23],
              [1,0,-4],
              [-2,1,6]])
Tinv = np.linalg.inv(T)
```

[274]:
```
T @ np.array([[np.exp(2),np.exp(1),0],
              [0, np.exp(2),0],
              [0, 0, np.exp(4)]]) @ Tinv
```

[274]:
```
array([[  70.9404, -185.2174,  -60.833 ],
       [  10.7882,  -19.5391,   -8.0699],
       [ -16.0224,   37.1942,   17.9749]])
```

9.3 Answe

9.4

$$f(A) = \begin{pmatrix} 70.94 & -185.21 & -60.83 \\ 10.788 & -19.54 & -8.07 \\ -16.02 & 37.194 & 17.975 \end{pmatrix}$$

[ ]: