solucao_EP2

September 3, 2019

1 Exercício de Programação 2: Python básico

Prazo de submissão: 23:55 do dia 03/09/2019

2019.2 Álgebra Linear Computacional - DCC - UFMG

Erickson - Fabricio - Renato

Instruções: * Antes de submeter suas soluções, certifique-se de que tudo roda como esperado. Primeiro, **reinicie o kernel** no menu, selecione Kernel→Restart e então execute **todas as células** (no menu, Cell→Run All) * Apenas o arquivo .ipynb deve ser submetido. Ele não deve ser compactado. * Não deixe de preencher seu nome e número de matrícula na célula a seguir

Nome do aluno: INSIRA SEU NOME AQUI

Matricula: INSIRA SUA MATRICULA AQUI

1.1 Questão 1

Crie as matrizes A, B e C abaixo e resolva as questões: 1. [ ] Calcule $(((A^TB)+B)C^{-1})$ 2. [ ] Crie matrizes $\tilde{A}_{2x2}, \tilde{B}_{2x2}, \tilde{C}_{2x2}$, tal que sejam compostas pelo 2 primeiros elementos de cada linha das duas primeiras linhas. E repita a equação do item anterior.

Dica numpy.full, numpy.eye, numpy.ones e indexação de vetores.

$$A = \begin{bmatrix} 8 & 8 & 8 \\ 8 & 8 & 8 \\ 8 & 8 & 8 \end{bmatrix}_{3\times 3} \qquad B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}_{3\times 4} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}_{4\times 4}$$

```
In [13]: # Código para Exercício 1
         import numpy as np
         import math
         A = np.full((3,3),8)
         B = np.ones((3,4))
         C = np.eye(4)
         np.dot((np.dot(A.T,B)+B),np.linalg.inv(C))
```

```
Out[13]: array([[25., 25., 25., 25.],
                [25., 25., 25., 25.],
                [25., 25., 25., 25.]])
```

```
In [14]: A1 = A[0:2,0:2]
         B1 = B[0:2,0:2]
```

```
C1 = C[0:2,0:2]
np.dot((np.dot(A1.T,B1)+B1),np.linalg.inv(C1))
```

```
Out[14]: array([[17., 17.],
                [17., 17.]])
```

1.2 Questão 2

2A. Escreva uma função python que recebe m como entrada e executa os seguintes passos: 1. [ ] gera uma matriz aleatória $W_{m\times 4}$ (função **numpy.random.randn**), 2. [ ] divide cada uma das entradas por $\sqrt{m}$ (salva resultado em $\tilde{W}$), 3. [ ] calcula $Z = \tilde{W}^\top \times \tilde{W}$, 4. [ ] imprime Z, 5. [ ] calcula a norma Frobenius da diferença entre Z e a matriz identidade $I_{4\times 4}$.

```
In [6]: # Código para Exercício 2
        def questao_2(m):
            W = np.random.randn(m,4)
            W1 = W/math.sqrt(m)
            Z = np.dot(W1.T,W1)
            print(Z)
            return np.linalg.norm(Z-np.eye(4), 'fro')
        # np.linalg.norm(Z-np.eye(4), 'fro')
```

```
In [7]: questao_2(100)
```

```
[[ 0.87772216  0.01456443  0.01252501  0.0490636 ]
 [ 0.01456443  0.68301429 -0.10892931  0.03864704]
 [ 0.01252501 -0.10892931  0.89486186 -0.04713815]
 [ 0.0490636   0.03864704 -0.04713815  0.81878836]]
```

```
Out[7]: 0.4427621843851814
```

```
In [10]: questao_2(10000)
```

```
[[ 9.80620639e-01  1.67171601e-03 -1.75512761e-02  6.61939806e-04]
 [ 1.67171601e-03  9.97818528e-01  7.57331463e-03 -7.97286048e-03]
 [-1.75512761e-02  7.57331463e-03  1.01012416e+00 -3.32963973e-03]
 [ 6.61939806e-04 -7.97286048e-03 -3.32963973e-03  9.92877563e-01]]
```

```
Out[10]: 0.03768451345502954
```

2B. Qual a norma da diferença obtida para $m = 100$? E para $m = 10000$?

Resposta: Para $m = 100$ temos a norma igual a 0.4427621843851814. Para $m = 10000$ temos a norma igual a 0.03768451345502954

2C. O que podemos dizer sobre a matriz $\tilde{W}$?

Reposta: A medida que aumentamos m, a matriz Z se aproxima da matriz identidade. Logo, a matriz $\tilde{W}$ fica mais "ortonormal".

1.3 Questão 3

Seja $x = (1, 2, 3)^\top$. Calcule a projeção x_u de x em $u = (1, 1, 0)^\top$ e a projeção x_v de x em $v = (1, -1, 1)^\top$.

Resposta:

$x_u = \|x\| \frac{\langle x,y \rangle}{\|x\|\|u\|} \frac{u}{\|u\|} = \frac{3}{2}u = (\frac{3}{2}, \frac{3}{2}, 0)^\top \setminus x_v = \|x\| \frac{\langle x,y \rangle}{\|x\|\|v\|} \frac{v}{\|v\|} = \frac{2}{3}v = (\frac{2}{3}, -\frac{2}{3}, \frac{2}{3})^\top$

1.4 Questão 4

Encontre os autovalores e autovetores para ambas matrizes Markovianas A e A^∞. Explique a partir dessas respotas por que $A^{100} \approx A^\infty$:

$$A = \begin{bmatrix} .6 & .2 \\ .4 & .8 \end{bmatrix}_{2\times 2} \qquad A^\infty = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{2}{3} \end{bmatrix}_{2\times 2}$$

```
In [27]: A = np.array([[.6,.2],[.4,.8]])
         A_inf = np.array([[1/3,1/3],[2/3,2/3]])
         autovalores_A,autovetores_A = np.linalg.eig(A)
         autovalores_A_inf,autovetores_A_inf = np.linalg.eig(A_inf)
         print("autovalores de A: ",autovalores_A," autovalores de A_inf: ",autoval
         print("autovetores de A: \n",autovetores_A," \nautovetores de A_inf: \n",a

         A_100 = A
         for i in range(100):
             A_100 = np.dot(A,A_100)



         autovalores_A_100,autovetores_A_100 = np.linalg.eig(A_100)
         print("\n\n\nautovalores de A_100: ",autovalores_A_100," autovalores de A_
         print("autovetores de A_100: \n",autovetores_A_100," \nautovetores de A_in
```

```
autovalores de A:  [0.4 1. ]  autovalores de A_inf:  [0. 1.]
autovetores de A:
 [[-0.70710678 -0.4472136 ]
 [ 0.70710678 -0.89442719]]
autovetores de A_inf:
 [[-0.70710678 -0.4472136 ]
 [ 0.70710678 -0.89442719]]



autovalores de A_100:  [0. 1.]  autovalores de A_inf:  [0. 1.]
autovetores de A_100:
 [[-0.70710678 -0.4472136 ]
 [ 0.70710678 -0.89442719]]
autovetores de A_inf:
 [[-0.70710678 -0.4472136 ]
```

```
 [ 0.70710678 -0.89442719]]
```

Resposta: Como os autovetores de A e A^{∞} são os mesmos e os autovalores são respectivamente $0.4, 1$ e $0, 1$, os autovalores de A^{100} tendem a um valor muito próximo de $0, 1$

```
In [ ]:
```