Universidade Federal da Bahia
Programa de Pós-Graduação em Ciência de Dados e Big Data
Carlos Magno Santos Ribeiro de Brito

Prof Dra. Majela Penton Machado

Respostas da quarta lista de exercício

Questão 01) Considere o espaço vetorial $M_2(\mathbb{R})$ com o produto interno dado por $\langle U, V\rangle = tr(U^T V)$. Determine se as seguintes funções são transformações lineares ou não. Justifique!

a) $T : M_2(\mathbb{R}) \to \mathbb{R}$ com $T(U) = \|U\|$:

Devemos considerar a soma e a multiplicação por uma constante e $\|U\| = \sqrt{tr(U^T U)}$. Assim sendo, tendo a matriz U:

$$U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}, \quad U^T = \begin{bmatrix} u_{11} & u_{21} \\ u_{12} & u_{22} \end{bmatrix}$$

Assim:

$$U^T U = \begin{bmatrix} u_{11}u_{11} & u_{21}u_{21} & u_{11}u_{12} & u_{21}u_{22} \\ u_{12}u_{11} & u_{22}u_{21} & u_{12}u_{12} & u_{22}u_{22} \end{bmatrix}$$

$$tr(U^T U) = u_{11}^2 + u_{21}^2 + u_{12}^2 + u_{22}^2$$
$$T(U) = \|U\| = \sqrt{u_{11}^2 + u_{21}^2 + u_{12}^2 + u_{22}^2}$$

1) Para a multiplicação por uma contante α:

$$T(\alpha U) = \sqrt{\alpha^2(u_{11}^2 + u_{21}^2 + u_{12}^2 + u_{22}^2)}$$
$$T(\alpha U) = \alpha\sqrt{u_{11}^2 + u_{21}^2 + u_{12}^2 + u_{22}^2}$$
$$T(\alpha U) = \alpha T(U)$$

1) Para a soma com uma matriz V:

$$
\begin{aligned}
&T(U+V) = \sqrt{(u_{11}+v_{11})^2 + (u_{21}+v_{21})^2 + (u_{12}+v_{12})^2 + (u_{22}+v_{22})^2} \\
&T(U)+T(V) = \sqrt{u_{11}^2 + u_{21}^2 + u_{12}^2 + u_{22}^2} + \sqrt{v_{11}^2 + v_{21}^2 + v_{12}^2 + v_{22}^2} \\
&T(U+V) \neq T(U)+T(V)
\end{aligned}
$$

Sendo assim, não é uma transformação linear, já que $T(U+v) = 0$ e $T(U)+T(V) = 2T(U)$, sendo $V = -U$.

b) Se B é uma matriz fixa de tamanho 2×3, $T : M_2(\mathbb{R}) \to M_{2\times 3}(\mathbb{R})$ com $T(U) = UB$:

Temos as três matrizes:

$$
B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix}, \quad U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}, \quad V = \begin{bmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{bmatrix}
$$

Assim sendo, temos:

$$
T(U) = \begin{bmatrix} u_{11}b_{11} + u_{12}b_{21} & u_{11}b_{12} + u_{12}b_{22} & u_{11}b_{13} + u_{12}b_{23} \\ u_{21}b_{11} + u_{22}b_{21} & u_{21}b_{12} + u_{22}b_{22} & u_{21}b_{13} + u_{22}b_{23} \end{bmatrix}
$$

1) Para a multiplicação por uma contante α:

$$
\begin{aligned}
&T(\alpha U) = \begin{bmatrix} \alpha(u_{11}b_{11} + u_{12}b_{21}) & \alpha(u_{11}b_{12} + u_{12}b_{22}) & \alpha(u_{11}b_{13} + u_{12}b_{23}) \\ \alpha(u_{21}b_{11} + u_{22}b_{21}) & \alpha(u_{21}b_{12} + u_{22}b_{22}) & \alpha(u_{21}b_{13} + u_{22}b_{23}) \end{bmatrix} \\
&T(\alpha U) = \alpha \begin{bmatrix} u_{11}b_{11} + u_{12}b_{21} & u_{11}b_{12} + u_{12}b_{22} & u_{11}b_{13} + u_{12}b_{23} \\ u_{21}b_{11} + u_{22}b_{21} & u_{21}b_{12} + u_{22}b_{22} & u_{21}b_{13} + u_{22}b_{23} \end{bmatrix} \\
&T(\alpha U) = \alpha T(U)
\end{aligned}
$$

2) Para a adição de uma matriz V:

$$T(U+V) = \begin{bmatrix} (u_{11}+v_{11})b_{11} + (u_{12}+v_{12})b_{21} & \cdots & (u_{11}+v_{11})b_{13} + (u_{12}+v_{12})b_{23} \\ (u_{21}+v_{21})b_{11} + (u_{22}+v_{22})b_{21} & \cdots & (u_{21}+v_{21})b_{13} + (u_{22}+v_{22})b_{23} \end{bmatrix}$$
$$T(U+V) = \begin{bmatrix} (u_{11}b_{11} + u_{12}b_{21}) + (v_{11}b_{11} + v_{12}b_{21}) & \cdots & (u_{11}b_{13} + u_{12}b_{23}) + (v_{11}b_{13} + v_{12}b_{23}) \\ (u_{21}b_{11} + u_{22}b_{21}) + (v_{21}b_{11} + v_{22}b_{21}) & \cdots & (u_{21}b_{13} + u_{22}b_{23}) + (v_{21}b_{13} + v_{22}b_{23}) \end{bmatrix}$$
$$T(U+V) = T(U) + T(V)$$

Sendo assim, é uma transformação linear, já que ambas as condições são satisfeitas.

Questão 02) Considere a transformação linear $T : \mathbb{R}^2 \to \mathbb{R}^2$ tal que $T(1,0) = (1,-1)$ e $T(0,1) = (1,1)$. Determine:

a) $T(2,3)$:

$$T(2,3) = T(2,0) + T(0,3)$$
$$T(2,3) = 2T(1,0) + 3T(0,1)$$
$$T(2,3) = 2(1,-1) + 3(1,1)$$
$$T(2,3) = (5,1)$$

b) O vetor $(x,y) \in \mathbb{R}^2$ tal que $T(x,y) = (4,-2)$:

$$T(x,y) = (4,-2)$$
$$T(x,y) = 3(1,-1) + (1,1)$$
$$T(x,y) = 3T(1,0) + T(0,1)$$
$$T(x,y) = T(3,1)$$
$$(x,y) = (3,1)$$

c) O núcleo e a imagem de T:

Sabemos que (x,y) é igual a:

$$(x,y) = vT(0,1) + uT(1,0)$$
$$(x,y) = v(1,1) + u(1,-1)$$

com $x = u + v$ e $y = v - u$.

$$u + v = v - u$$
$$u = -u$$
$$2u = 0$$
$$u = 0$$

Assim sendo, $(0,0) = v(1,1)$, $v = 0$ e o núcleo desse vetor $(0,0)$ será dado como:

$$T(x,y) = yT(0,1) + xT(1,0)$$
$$T(x,y) = y(1,1) + x(1,-1)$$
$$T(x,y) = (y,y) + (x,-x)$$
$$T(x,y) = (y + x, y - x)$$

Questão 03) Considere a matriz:

$$A = \begin{bmatrix} 7 & -2 & 0 \\ -2 & 6 & -2 \\ 0 & -2 & 5 \end{bmatrix}$$

a) Determine os autovalores de A uma base ortonormal do autoespaço associado ao autovalor dominante de A.

Sabemos que o valor de $A - \lambda I$ será:

$$A - \lambda I = \begin{bmatrix} 7-\lambda & -2 & 0 \\ -2 & 6-\lambda & -2 \\ 0 & -2 & 5-\lambda \end{bmatrix}$$

Segunda etapa, encontra-se o determinante dessa matriz para saber os λ_n:

$$det(A - \lambda I) = (\lambda - 3)(\lambda - 6)(\lambda - 9)$$

Assim, os valores de λ são $(\lambda_1, \lambda_2, \lambda_3) = (3{,}6{,}9)$ que são substituídos na matriz A e resolvida como um sistema homogêneo:

Para $\lambda_2 = 9$, temos então:

$$\begin{bmatrix} 7-9 & -2 & 0 \\ -2 & 6-9 & -2 \\ 0 & -2 & 5-9 \end{bmatrix} = \begin{bmatrix} -2 & -2 & 0 \\ -2 & -3 & -2 \\ 0 & -2 & -4 \end{bmatrix}$$

$$\begin{bmatrix} -2 & -2 & 0 \\ -2 & -3 & -2 \\ 0 & -2 & -4 \end{bmatrix} \cdot \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

Resolvendo o sistema de equações lineares (a matriz aumentada) com eliminação de gauss, temos:

$$\left[\begin{array}{ccc|c} 1 & 0 & -2 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

O que significa que:

$$\begin{aligned} x_1 &= 2x_3 \\ x_2 &= -2x_3 \\ x_3 &= x_3 \end{aligned}$$

A solução é dada como $\{x_3(2, -2, 1)\}$ o que ao tomar $x_3 = 1$, ter-se-á $v_3 = (2, -2, 1)$. A norma de $v_3 \rightarrow \|v_3\| = 3$. Assim $({}^2\!/\!_3, -{}^2\!/\!_3, {}^1\!/\!_3)$.

b)Com ajuda do Python realize 5 iterações do método das potências aplicado à matriz A, começando com x_0 = (1,0,0) Compare a aproximação resultante com o valor exato do autovalor dominante e o autovetor unitário correspondente. Observação: Especifique os comandos utilizados em Python.

```
import numpy as np
from math import sqrt

#Calcula a norma do vetor
def norm(vet):
    sum = 0
    for v in vet:
        sum += v[0]**2
    return sqrt(sum)

A = np.array([[7, -2, 0],
        [-2, 6, -2],
        [0, -2, 5]])

xi = np.array([[1], [0], [0]])
print(f"x0:\n{xi}\n")

for i in range(5):
    Ax = A @ xi #multiplicacao da matriz por um vetor i
    print(f"Ax{i+1}:\n{Ax}") #imprime o resultado da multiplicacao
```

```
    xi = Ax/norm(Ax) #divide o vetor por sua norma
    print(f"x{i+1}:\n{xi}") #imprime o resultado da divisao
    print(f"Autovalor na iteracao {i+1} e: {norm(Ax)}\n") #imprime
    o autovalor

# x0:
# [[1]
#  [0]
#  [0]]

# Ax1:
# [[ 7]
#  [-2]
#  [ 0]]
# x1:
# [[ 0.96152395]
#  [-0.27472113]
#  [ 0.        ]]
# Autovalor na iteracao 1 e: 7.280109889280518

# Ax2:
# [[ 7.28010989]
#  [-3.57137466]
#  [ 0.54944226]]
# x2:
# [[ 0.89573556]
#  [-0.43941744]
#  [ 0.06760268]]
# Autovalor na iteracao 2 e: 8.12752137946034

# Ax3:
# [[ 7.14898378]
#  [-4.56318114]
#  [ 1.2168483 ]]
# x3:
# [[ 0.83437796]
#  [-0.53258167]
#  [ 0.14202178]]
# Autovalor na iteracao 3 e: 8.568040094092964

# Ax4:
# [[ 6.90580904]
#  [-5.14828952]
#  [ 1.77527225]]
# x4:
```

```
# [[ 0.78522432]
#  [-0.58538574]
#  [ 0.20185715]]
# Autovalor na iteracao 4 e: 8.794695848686912

# Ax5:
# [[ 6.6673417 ]
#  [-5.48647736]
#  [ 2.18005723]]
# x5:
# [[ 0.74867875]
#  [-0.61607897]
#  [ 0.24479959]]
# Autovalor na iteracao 5 e: 8.905477454667345
```

O autovalor dominante encontrado deu muito próximo do valor calculado manualmente. Ja o autovetor, seria necessaria mais iterações para melhorar sua precisão.

Questão 04) Se A é uma matriz de ordem n diagonalizável, mostre que:

a) $det(A)$ é igual ao produto dos autovalores de A:

Com um matriz qualquer A, tem-se:

$$A = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}$$

O valor do determinante de A é igual ao produto dos autovalores de A. Ja que $det(A) = a \times b \times c \rightarrow A : (a - \lambda_1)(b - \lambda_2)(c - \lambda_3) = 0.$ o que nos traz $\lambda_1 = a$,$\lambda_2 = b\lambda_3 = c$.

b) $tr(A)$ é igual à soma dos autovalores de A:

Com um matriz qualquer A, tem-se:

$$A = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix}$$

O valor do da soma dos autovalores de A é igual ao traço de A. Ja que $tr(A) = a+b+c \rightarrow A : (a - \lambda_1)(b - \lambda_2)(c - \lambda_3) = 0$. o que nos traz $\lambda_1 = a, \lambda_2 = b\lambda_3 = c$.

c) Se A é invertível, então A^{-1} é diagonalizável.

Nesse caso, para diagonalizar A é necessario existir uma matriz B (invertível) para que $B^{-1}AB$ seja diagonal. Considerando A invertível, pode-se encontrar uma matriz diagonal com $BA^{-1}B$ que será inversa de $B^{-1}AB$.

Questão 05) Realize o seguinte experimento para compressão de imagens usando decomposição SVD. No Python, digite as seguintes linhas de comando:

```
# Importar a biblioteca de imagens
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np

img = mpimg.imread('pirata.png') # Lendo o arquivo da imagem
original
plt.figure() # Criando uma figura
imgplot = plt.imshow(img, cmap='gray') # Plotando a imagem
original
plt.title('Original') # Titulo da imagem
plt.axis('off') # Desabilitando os eixos
U, S, V = np.linalg.svd(img) # Realizando a decomposicao SVD
postos = [10, 30, 50, 100, 200] # Lista de postos para a
compressao

for posto in postos: # Para cada posto
    aproxSigmas = S[0:posto] # Aproximando os valores de S
    aproxS = np.diag(aproxSigmas, 0) # Criando a matriz diagonal
    aproxU = U[:, 0:posto] # Aproximando os valores de U
    aproxV = V[0:posto] # Aproximando os valores de V
    aproxImg = aproxU @ aproxS @ aproxV # Criando a imagem
    aproximada
    plt.figure() # Criando uma figura
    aproxImgplot = plt.imshow(aproxImg, cmap='gray') # Plotando
     a imagem aproximada
    plt.title('Aproximacao de posto ' + str(posto)) # Titulo da
     imagem com posto
    plt.axis('off') # Desabilitando os eixos
```

- Você consegue explicar o que cada linha acima está mandando o Python fazer?
- Execute as linhas acima e observe os resultados.


Original
Aproximacao de posto 10
Aproximacao de posto 30
Aproximacao de posto 50
Aproximacao de posto 100
Aproximacao de posto 200

- Para cada posto considerado determine a quantidade de números que devem ser armazenados na aproximação da imagem e compare com a quantidade de números necessários para armazenar a imagem original. Analise as imagens obtidas para postos diferentes.

Para a análise de imagens, utilza-se do código em python a seguir:

```
# Importar a biblioteca de imagens
import matplotlib.image as mpimg
import numpy as np

img = mpimg.imread('pirata.png')
U, S, V = np.linalg.svd(img)
postos = [10, 30, 50, 100, 200]

print("Imagem original tem dimensao: {}x{}\n".format(
        img.shape[0], img.shape[1]))
print("Imagem original tem armazenamento: {}".format(
        img.shape[0]*img.shape[1]))
for posto in postos:
    print("Imagem do posto {} tem armazenamento: {}".format(
        posto, posto * (img.shape[0]+img.shape[1]+1)))

#Imagem original tem dimensao: 512x51

#Imagem original tem armazenamento: 262144
#Imagem do posto 10 tem armazenamento: 10250
#Imagem do posto 30 tem armazenamento: 30750
#Imagem do posto 50 tem armazenamento: 51250
#Imagem do posto 100 tem armazenamento: 102500
#Imagem do posto 200 tem armazenamento: 205000
```