BCM0505-15 - Processamento da Informação

May 12, 2020

Laboratório L04 – 12/05

Instruções

- Membros das turmas NA1 e NB{4,5} deverão submeter códigos em Python para todos os exercícios nesta página em um único arquivo `.py` via Tidia até 26/05 às 19h – sob a entrada **E8.**

Exercícios

- **(01)** *Norma de Frobënius.*
  Seja $A \in \mathbb{R}^{m \times n}$ uma matriz de pontos flutuantes. A *norma de Frobënius* de A é dada por

  $$\|A\|_F := \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left(A[i][j]\right)^2 \right)^{1/2}.$$

  Escreva uma função que recebe A e devolve $\|A\|_F$.

  Solução:

```
from math import sqrt

def frobenius_norm (A: [[float]]) -> float:
  return sqrt (sum (sum (aij*aij for aij in ai) for ai in A))
```

  □

- **(02)** *Matriz Ortogonal.*
  Uma matriz quadrada $A \in \mathbb{R}^{n \times n}$ de reais é *ortogonal* se suas colunas e linhas forem vetores ortogonais. De outra forma, A é ortogonal se

  $$A \cdot A^\top = I_n,$$

  em que I_n é a matriz identidade $n \times n$. Escreva uma função que recebe A e determina se ela é ortogonal.

  Solução:
  Uma solução consiste em usar as funções da aula A06 para transpor matriz, multiplicar matrizes, e verificar se uma matriz é identidade. A título de exemplo, fornecemos este código abaixo.

```
def is_orthogbad (A: [[float]]) -> bool:
  return matrix_is_identity (matrix_product (A, matrix_trasnpose (A)))
```

  Observe que duas matrizes novas são criadas nesta solução: a transposta de A, e a produto de A com $A^\top$. Com um pouco de reflexão, percebe-se que isto é desnecessário (e, portanto, ineficiente). O próximo código realiza a tarefa sem criar nenhuma matriz temporária / adicional.

```
def is_orthogonal (A: [[float]]) -> bool:
  n = len (A)
  for i in range (n):
    for j in range (n):
      if sum (A[i][k] * A[j][k] for k in range (n)) != (1 if i == j else 0):
        return False
  return True
```

  □

- **(03)** *Matriz Triangular Superior.*
  Uma matriz de pontos flutuantes $A \in \mathbb{R}^{n \times n}$ é *triangular superior* se

  $$A[i][j] == 0 \qquad \text{para todo} \qquad i > j.$$

  Escreva uma função que recebe A e determina se ela é triangular superior.

  Solução:

```
def is_upper_triangular (A: [[float]]) -> bool:
  for i in range (len (A)):
    for j in range (i):
      if A[i][j] != 0:
        return False
  return True
```

  □

- **(04)** *Sistema Linear Triangular Superior.*
  Dada uma matriz triangular superior $U \in \mathbb{R}^{n \times n}$ tal que $U[i][i] \neq 0$ e um vetor $b \in \mathbb{R}^n$, determine um vetor $x \in \mathbb{R}^n$ tal que

  $$Ux = b.$$

  Solução:

```
def linsys_upper_triangular (U: [[float]], b: [float]) -> [float]:
  x = [0] * (n := len (U))
  for i in reversed (range (n)):
    x[i] = (b[i] - sum (U[i][j]*x[j] for j in range (i+1, n))) / U[i][i]
  return x
```

  □

- **(05)** *Quadrado Latino.*
  Uma matriz de inteiros $A \in \mathbb{Z}^{n \times n}$, com $n \geq 0$ é um *quadrado latino* se os números do conjunto $\{1, 2, \ldots, n\}$ ocorrem em todas as linhas e em todas as colunas de A. Por exemplo, a matriz abaixo é um quadrado latino.

  $$\begin{pmatrix} 1 & 4 & 3 & 2 \\ 2 & 3 & 4 & 1 \\ 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$$

  Escreva uma função que recebe uma matriz $n \times n$ de inteiros A e determina se A é um quadrado latino.

  Solução:
  Cada linha e cada coluna devem somar $1 + 2 + \cdots + n = n(n+1)/2$. Esse teste simples é necessário, mas não é suficiente:

  $$1 + 2 + 3 + 4 = 1 + 3 + 3 + 3.$$

  É preciso verificar a ocorrência de cada valor em cada linha e cada coluna.

```
def is_latin_square (A: [[int]) -> bool:
  for i in range (n):
    lin = [False] * n
    col = [False] * n
    for j in range (n):
      lin [A[i][j]-1] = True
      col [A[j][i]-1] = True
    if not all (lin) or not all (col):
      return False
  return True
```

  □

Aritanan Gruber

Assistant Professor

"See, if y'all haven't the same feeling for *this*, I really don't give a damn. If you ain't feeling it, then dammit *this* ain't for you!"
(desconheço a autoria; agradeço a indicação)

CV

© 2021 Aritanan Gruber · Powered by the Academic theme for Hugo.