Fundamentos Matemáticos e Computacionais de Machine Learning

Especialização em Machine Learning e Big Data

Profa. Dra. Juliana Felix

jufelix16@uel.br

UEL

Fundamentos Matemáticos

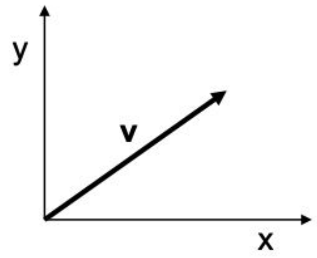
Vetor

Segmento de reta, direcionado, de dimensão N

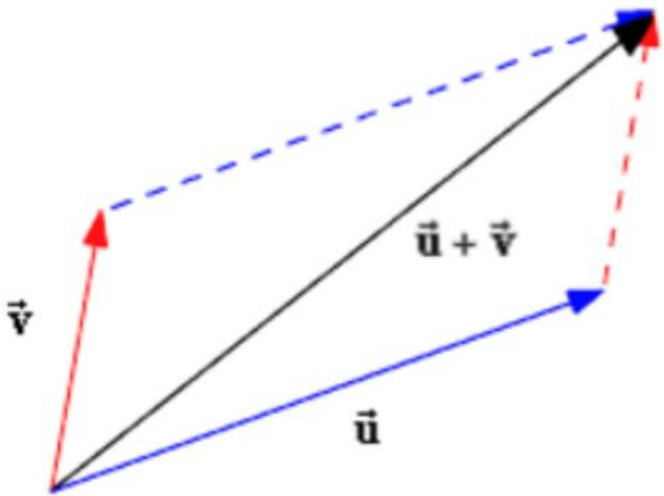
Um vetor tem tamanho e direção

$$\vec{v} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$\mathbf{v} = [\mathbf{a}\ \mathbf{b}\ \mathbf{c}]^T$



Adição de Vetores



$u = (u_1, u_2)$

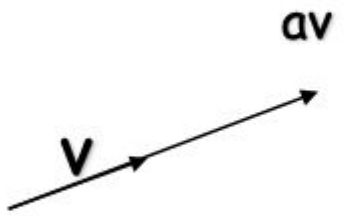
$v = (v_1, v_2)$

$u+v = (u_1+v_1, u_2+v_2)$

Multiplicação de vetor por escalar: av

O **produto av,** de um vetor **v** por um **escalar a,** muda apenas o tamanho do vetor, mantendo-se a mesma direção.

$$av = a(x_1, x_2) = (ax_1, ax_2)$$



Matriz

Uma **matriz** é um conjunto de elementos, organizado em **linhas** (rows) e **colunas** (columns)

$$\begin{array}{cc} & \text{linhas} \\ \text{colunas} & \begin{bmatrix} a & b \\ c & d \end{bmatrix} \end{array}$$

Matriz 2x2

Matrizes

```
In [30]: a = np.matrix('1 2; 3 4')
         print(a)

         print("\nOutra forma de criar matrizes:")
         a = np.arange(1,21)
         print(a)
         print("Dimensões:", a.shape)

         print("\n")
         print("Readequando em forma de matriz 5x4")
         a = a.reshape(5,4)
         print(a)
         print("Dimensões:", a.shape)
```

```
[[1 2]
 [3 4]]

Outra forma de criar matrizes:
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20]
Dimensões: (20,)


Readequando em forma de matriz 5x4
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]
 [17 18 19 20]]
Dimensões: (5, 4)
```

Operações básicas

Adição de elementos correspondentes

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$$

Subtração de elementos correspondentes

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} - \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} a-e & b-f \\ c-g & d-h \end{bmatrix}$$

Multiplica-se cada linha por cada coluna

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$

Operações básicas

```
In [8]: a = np.array(np.arange(4)).reshape(2,2)
        b = np.array(np.arange(4,8)).reshape(2,2)

        print("Matriz A:\n", a)
        print("\n")

        print("Matriz B:\n", b)
        print("\n")

        print("Matriz A+B:\n", a+b)
        print("\n")

        print("Matriz A-B:\n", a-b)
        print("\n")
```

```
Matriz A:
 [[0 1]
 [2 3]]


Matriz B:
 [[4 5]
 [6 7]]


Matriz A+B:
 [[ 4  6]
 [ 8 10]]


Matriz A-B:
 [[-4 -4]
 [-4 -4]]
```

Operações básicas

```
print("Matriz AxB:\n", np.matmul(a,b)) #multiplicacao de matrizes
print("\n")

print("Matriz A.B:\n", a*b) # produto escalar entre 2 matrizes
print("\n")
```

```
Matriz A:
 [[0 1]
 [2 3]]


Matriz B:
 [[4 5]
 [6 7]]
```

```
Matriz AxB:
 [[ 6  7]
 [26 31]]


Matriz A.B:
 [[ 0  5]
 [12 21]]
```

Operações básicas

```
print("Matriz 3*A:\n", 3*a)
print("\n")

print("Matriz A/3:\n", a/3)
print("\n")
```

```
Matriz A:
 [[0 1]
 [2 3]]


Matriz B:
 [[4 5]
 [6 7]]
```

```
Matriz 3*A:
 [[0 3]
 [6 9]]


Matriz A/3:
 [[0.         0.33333333]
 [0.66666667 1.        ]]
```

Multiplicação de matrizes

Na multiplicação $\mathbf{L} = \mathbf{M}\mathbf{N}$, o elemento da linha 1 e coluna 2 da matriz resultante $\mathbf{L}$ será obtido pela multiplicação dos elementos da linha 1 da matriz $\mathbf{M}$ pelos elementos da coluna 2 da matriz $\mathbf{N}$

$$\mathbf{L} = \mathbf{M} \cdot \mathbf{N}$$

$$\begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \cdot \begin{bmatrix} n_{11} & n_{12} & n_{13} \\ n_{21} & n_{22} & n_{23} \\ n_{31} & n_{32} & n_{33} \end{bmatrix}$$

$$l_{12} = m_{11}n_{12} + m_{12}n_{22} + m_{13}n_{32}$$

Multiplicação de matrizes

A multiplicação AB de uma matriz A por B só pode acontecer caso o número de colunas da matriz A seja igual ao número de linhas da matriz B.

A: B: AB:

$$\begin{bmatrix} 1 & 3 & 2 \\ 4 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 3 \\ 0 & 1 \\ 5 & 2 \end{bmatrix} = \begin{bmatrix} \cdots & \cdots \\ \cdots & \cdots \end{bmatrix}$$

$\mathbb{R}^{2x3}$ $\mathbb{R}^{3x2}$ $\mathbb{R}^{2x2}$

$$\mathbb{R}^{2\times\boxed{3}} \qquad \mathbb{R}^{\boxed{3}\times 2}$$

Multiplicação de matrizes

```
In [9]: a = np.array([[1,3,2],[4,0,1]])
        b = np.array([[1,3], [0,1], [5,2]])
        ab = np.matmul(a,b)

        print("Matriz A:\n", a)
        print("\n")
        print("Matriz B:\n", b)
        print("\n")
        print("Matriz AxB:\n", ab)
```

```
Matriz A:
 [[1 3 2]
 [4 0 1]]


Matriz B:
 [[1 3]
 [0 1]
 [5 2]]


Matriz AxB:
 [[11 10]
 [ 9 14]]
```

Multiplicação de matrizes

A **multiplicação** de matrizes **não é comutativa**, isto é, AB != BA.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & \ldots \\ \ldots & \ldots \end{bmatrix}$$

AB

$$\begin{bmatrix} e & f \\ g & h \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ea+fc & \ldots \\ \ldots & \ldots \end{bmatrix}$$

BA

Multiplicação de matrizes

A multiplicação de matrizes é **Associativa**.

$$A \times (B \times C) = D$$

$$A \times B \times C = D$$

$$(A \times B) \times C = D$$

Operações com escalares

Na multiplicação de matrizes por escalar, multiplica-se cada elemento da matriz. A dimensão da matriz permanece a mesma.

$$3 \times \begin{bmatrix} 4 & 0.5 \\ 2 & 5 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 12 & 1.5 \\ 6 & 15 \\ 0 & 3 \end{bmatrix}$$

Operações com escalares

```
In [10]: a = np.mat([[4, 0.5], [2, 5], [0,1]])

         print("Matriz A:\n", a)
         print("\n")
         print("Matriz resultante de 3*A:\n", 3*a)
         print("\n")

Matriz A:
 [[4.  0.5]
 [2.  5. ]
 [0.  1. ]]


Matriz resultante de 3*A:
 [[12.   1.5]
 [ 6.  15. ]
 [ 0.   3. ]]
```

Matriz Identidade

A matriz **identidade** $\mathbf{I}$ é uma matriz **quadrada** cujos elementos da **diagonal principal** são 1 e todos os demais elementos são zeros.

$$\begin{bmatrix} 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$1 \times 1 \qquad 2 \times 2 \qquad 3 \times 3 \qquad 4 \times 4$$

Matriz Identidade

```
In [11]: identidade = np.eye(5)

         print(identidade)

[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

Matriz Transposta

A **transposta** A^{T} de uma matriz A é a matriz obtida pela troca ordenada das linhas pelas colunas da matriz original.

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 5 & 9 \end{bmatrix} \qquad A^T = \begin{bmatrix} 1 & 3 \\ 2 & 5 \\ 0 & 9 \end{bmatrix}$$

Matriz Transposta

A **transposta** A^{T} de uma matriz A é a matriz obtida pela troca ordenada das linhas pelas colunas da matriz original.

$$A = \begin{bmatrix} \boxed{1 \ 2 \ 0} \\ 3 \ 5 \ 9 \end{bmatrix} \qquad A^T = \begin{bmatrix} 1 & 3 \\ 2 & 5 \\ 0 & 9 \end{bmatrix}$$

Matriz Transposta

A **transposta** A^{T} de uma matriz A é a matriz obtida pela troca ordenada das linhas pelas colunas da matriz original.

$$A = \begin{bmatrix} 1 & 2 & 0 \\ \boxed{3 \quad 5 \quad 9} \end{bmatrix} \qquad A^T = \begin{bmatrix} 1 & 3 \\ 2 & 5 \\ 0 & 9 \end{bmatrix}$$

Matriz Transposta

```
In [12]: a = np.array([[1, 2, 0], [3, 5, 9]])
         #transposta = np.swapaxes(a, 0, 1)
         #transposta = a.transpose(1,0)
         transposta = a.T

         print("Matriz A:\n", a)
         print("\n")
         print("Matriz transposta:\n", transposta)
         print("\n")
```

```
Matriz A:
 [[1 2 0]
 [3 5 9]]


Matriz transposta:
 [[1 3]
 [2 5]
 [0 9]]
```

Matriz Inversa

A **inversa** A^{-1} de uma matriz A é obtida utilizando-se a seguinte propriedade:

$$A \cdot A^{-1} = A^{-1} \cdot A = I$$

Exemplo:

$$\underset{A}{\begin{bmatrix} 3 & 4 \\ 2 & 16 \end{bmatrix}} \times \underset{A^{-1}}{\begin{bmatrix} a & c \\ b & d \end{bmatrix}} = \underset{I}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}$$

Matriz Inversa

$$\begin{bmatrix} 3 & 4 \\ 2 & 16 \end{bmatrix} \times \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

A $\quad$ A^{-1} $\quad$ I

Sistema de equações resultantes:

$$\begin{cases} 3 \times a + 4 \times b = 1 \\ 2 \times a + 16 \times b = 0 \end{cases}$$

$$\begin{cases} 3 \times c + 4 \times d = 0 \\ 2 \times c + 16 \times d = 1 \end{cases}$$

Resultado: $\begin{bmatrix} 3 & 4 \\ 2 & 16 \end{bmatrix} \times \begin{bmatrix} 0.4 & -0.1 \\ -0.05 & 0.075 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Matriz Inversa

```
In [13]: a = np.array([[3,4],[2,16]])
         print("Matriz A:\n", a)
         print("\n")

         inversa = np.linalg.inv(a)
         print("Matriz inversa:\n", inversa)
         print("\n")

Matriz A:
 [[ 3  4]
 [ 2 16]]


Matriz inversa:
 [[ 0.4   -0.1  ]
 [-0.05   0.075]]
```

Matriz e Sistemas lineares

Matrizes podem ser utilizadas para representar funções e resolver sistemas de equações lineares.

- $x' = ax + by$
- $y' = cx + dy$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Links úteis

Calculadoras gráficas:

Desmos | Calculadora Gráfica

Calculadora 3D - GeoGebra