Análise Estatística de dados

Inteligência Artificial

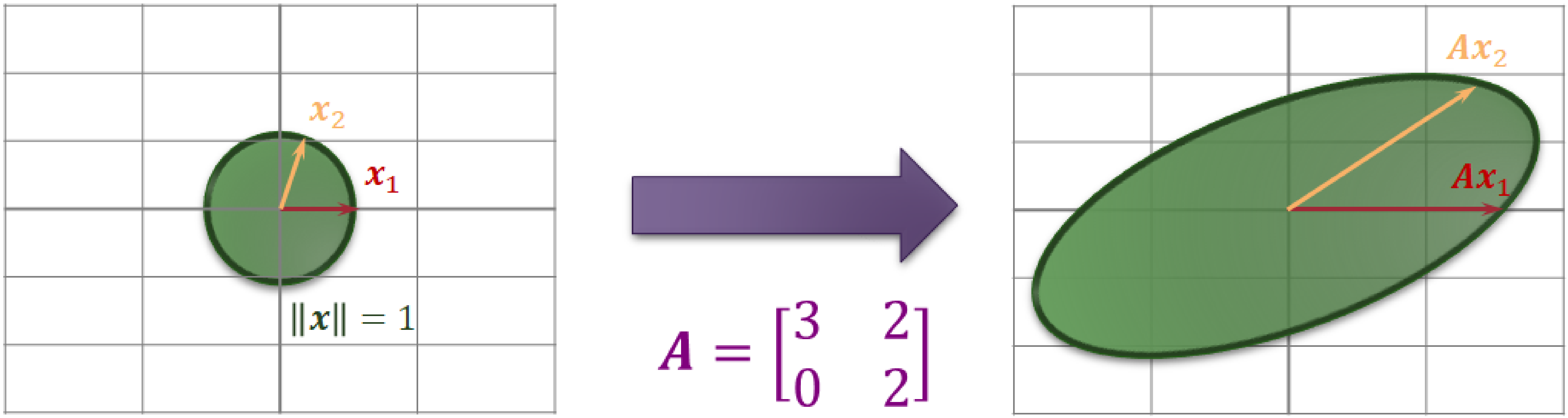
AULA 04 — DECOMPOSIÇÃO DE VALOR SINGULAR

Arturo Forner-Cordero
Larissa Driemeier

PROGRAMA DO CURSO

Aula	Data	Conteúdo da Aula
01	27/02	Aula Inaugural
02	05/03	Introdução ao Curso. Noções de Álgebra Linear , Geometria Analítica Parte I
03	12/03	Noções de Álgebra Linear , Geometria Analítica Parte II
04	19/03	Decomposição de valor singular (SVD)
05	26/03	Otimização: derivadas, derivadas parciais (operadores gradiente, Jacobiano, Hessiano e Laplaciano), algoritmos de gradiente
06	02/04	Variáveis independentes e não independentes. Estatística Descritiva e Indutiva. Definições de medidas de dispersão e tendência central.
07	09/04	Probabilidade e Teorema de Bayes
08	16/04	Modelos de probabilidade discretos
09	23/04	Modelos de probabilidade contínuos
10	30/04	Modelo de Markov. Modelo de Markov oculto.

TRANSFORMAÇÃO



Suponha que a transformação $T(x) = Ax$ seja aplicada a um círculo de raio unitário centrado na origem.

PECE Programa de Educação Continuada
Escola Politécnica da USP

AUTOVETORES E AUTOVALORES

$$\boldsymbol{A} = \begin{pmatrix} 3 & 2 \\ 0 & 2 \end{pmatrix}$$

$$\det \boldsymbol{A} = 6, tr\ \boldsymbol{A} = 5$$

$$\lambda^2 - 5\lambda + 6 = 0$$

$$\lambda_1 = 3$$
$$\lambda_2 = 2$$

autovalores

$$\begin{pmatrix} 3-3 & 2 \\ 0 & 2-3 \end{pmatrix}\begin{pmatrix} u_{1,1} \\ u_{2,1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} u_{1,1} \\ u_{2,1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$2u_{2,1} = 0 \rightarrow u_{2,1} = 0, \qquad u_{1,1} = t$$

$$\boldsymbol{u}_1 = \begin{pmatrix} t \\ 0 \end{pmatrix}$$

$$\|\boldsymbol{u}_1\|^2 = t^2 + 0^2 = 1$$
$$t = \pm 1$$

$$\boldsymbol{u}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 3-2 & 2 \\ 0 & 2-2 \end{pmatrix}\begin{pmatrix} u_{1,2} \\ u_{2,2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} u_{1,2} \\ u_{2,2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$u_{1,2} + 2u_{2,2} = 0, u_{2,2} = t$$
$$u_{1,2} = -2u_{2,2} = -2t$$

$$\boldsymbol{u}_2 = \begin{pmatrix} -2t \\ t \end{pmatrix}$$

$$\|\boldsymbol{u}_2\|^2 = (-2t)^2 + t^2 = 1$$
$$5t^2 = 1 \rightarrow t = \pm\frac{1}{\sqrt{5}}$$

$$\boldsymbol{u}_2 = \begin{pmatrix} -0{,}8944 \\ 0{,}4472 \end{pmatrix}$$

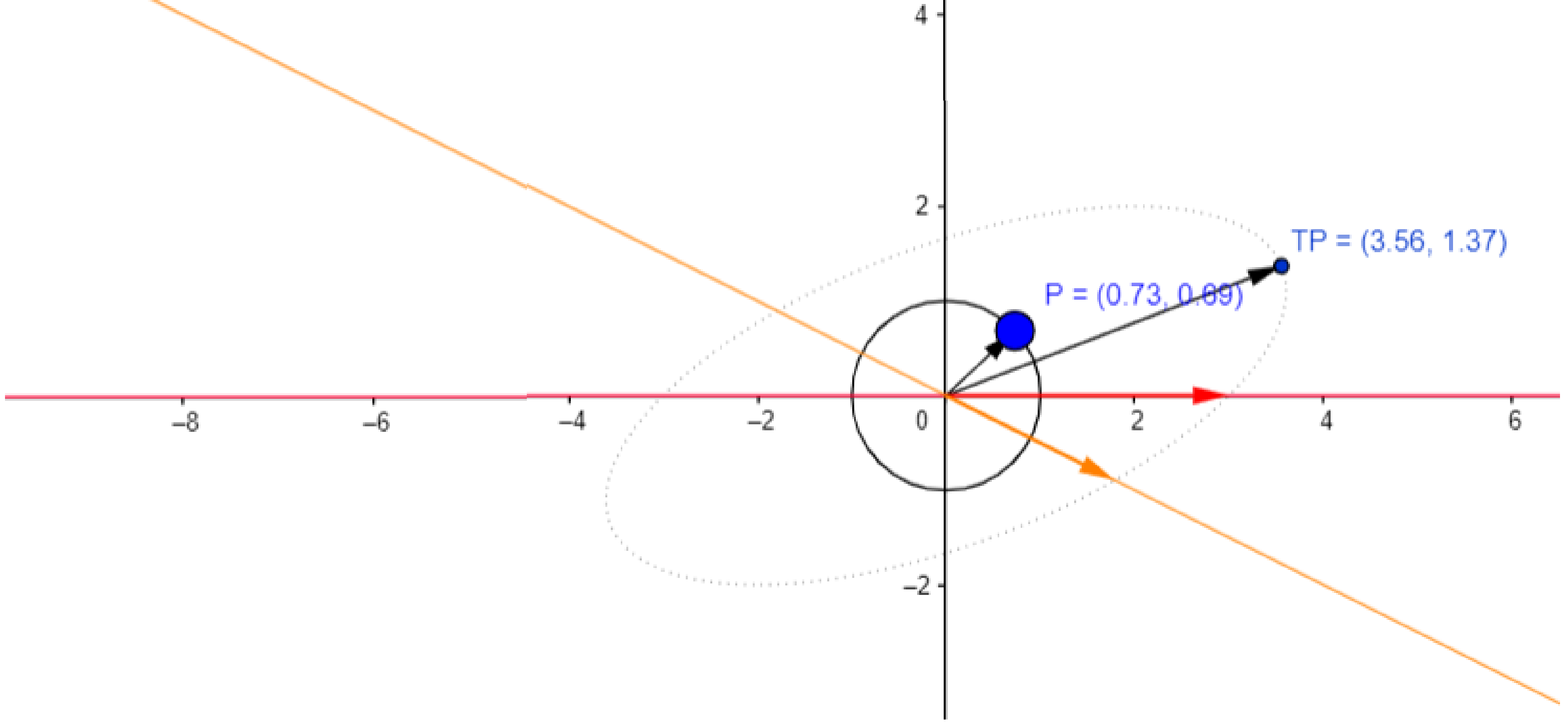
$$T = \begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix}$$

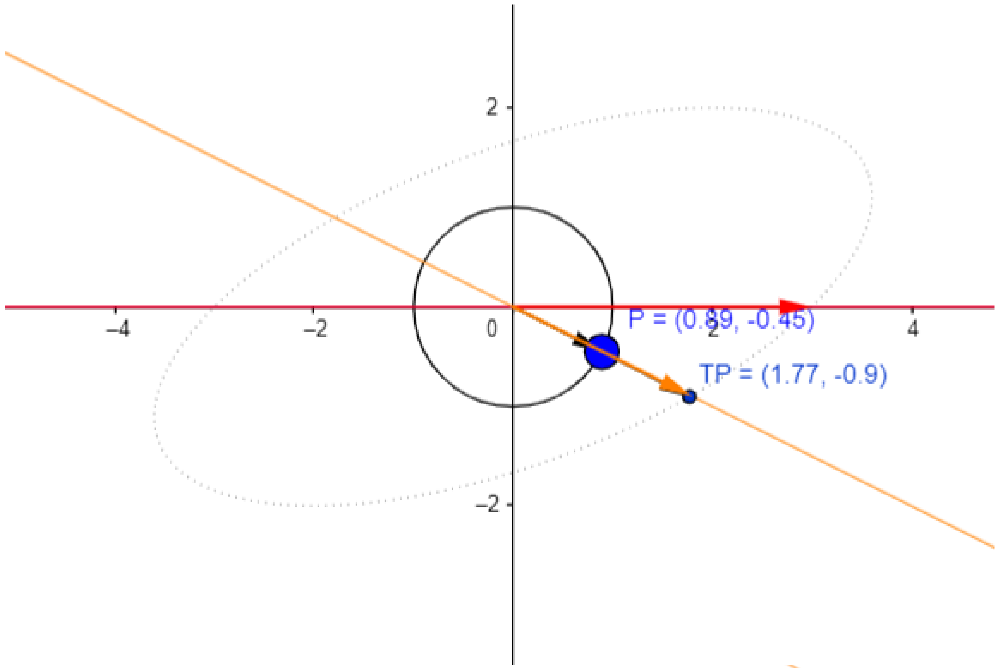
Brinque no link: https://www.geogebra.org/classic/mdvN0HTt

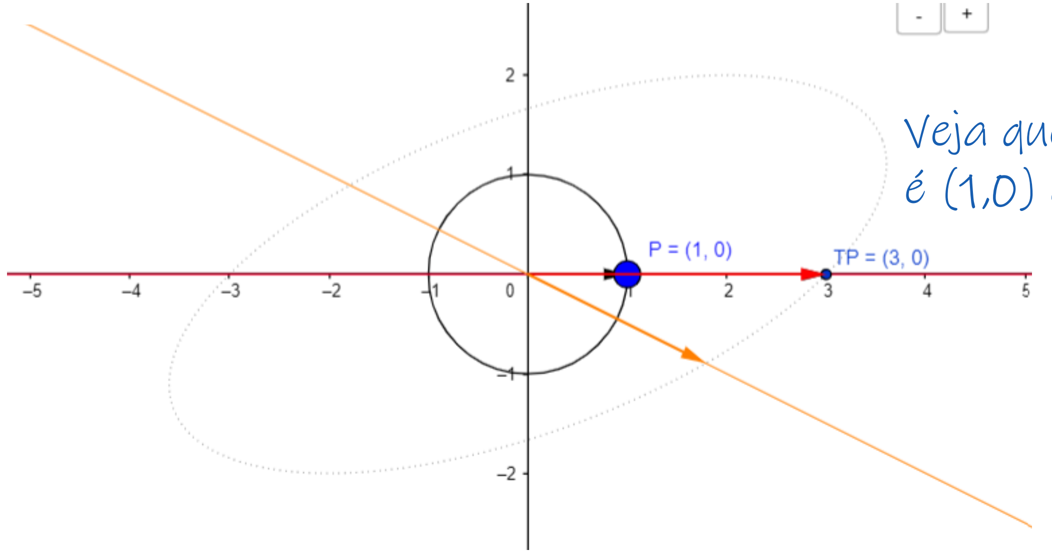
$\lambda_1 = 3$ $\lambda_2 = 2$

$u_1 = (1, 0)$ $u_2 = (0.89, -0.45)$

det T = 6

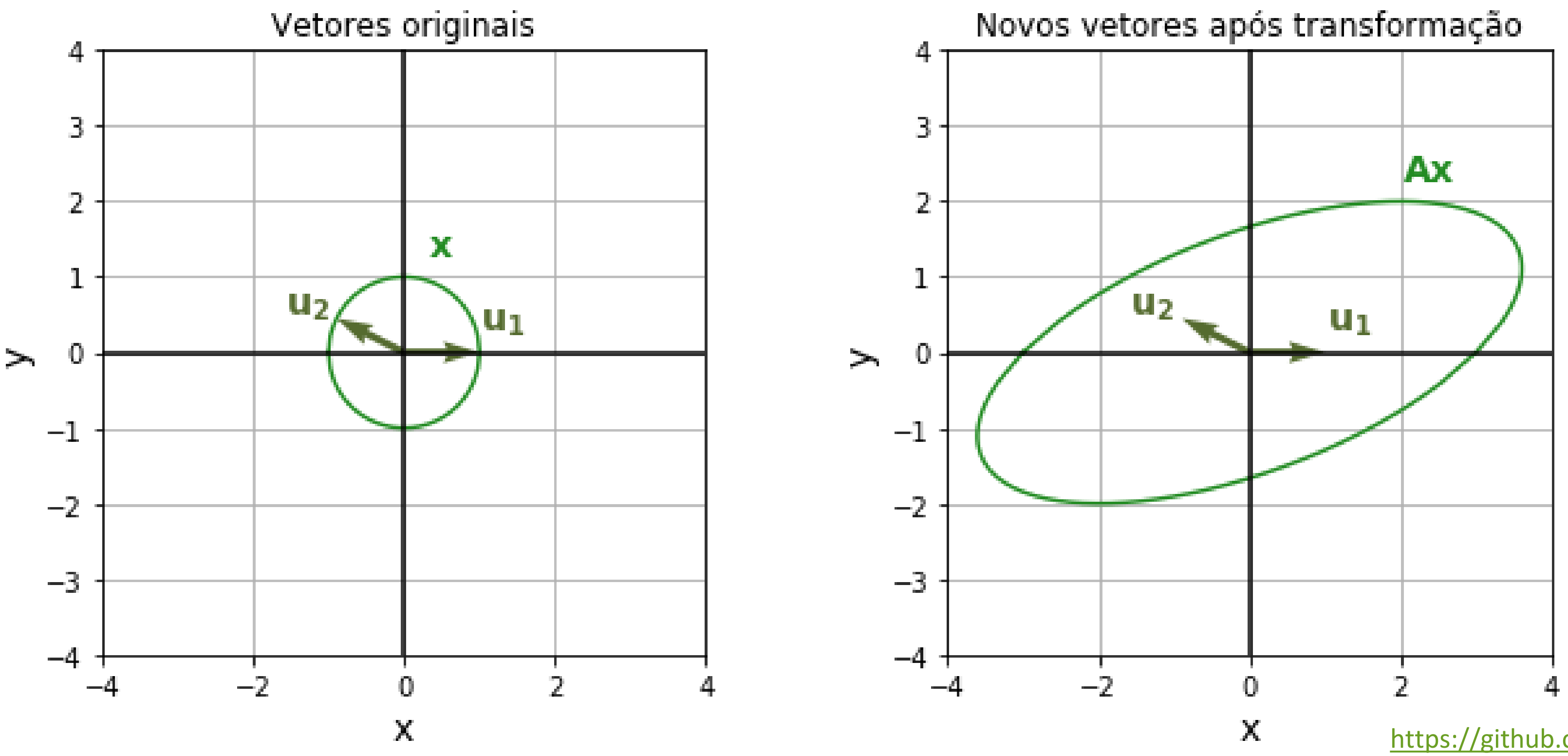






Veja que um dos autovetores é (1,0) com autovalor 3.

AUTOVETORES



$$A = \begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix}$$

$$\boldsymbol{u}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \boldsymbol{u}_2 = \begin{bmatrix} -0{,}8944 \\ 0{,}4472 \end{bmatrix}$$

$$\lambda_1 = 3, \qquad \lambda_2 = 2$$

https://github.com/reza-bagheri/SVD_article

Os vetores $\boldsymbol{u}_i$ são os autovetores de $\boldsymbol{A}$; os escalares λ_i são chamados autovalores.

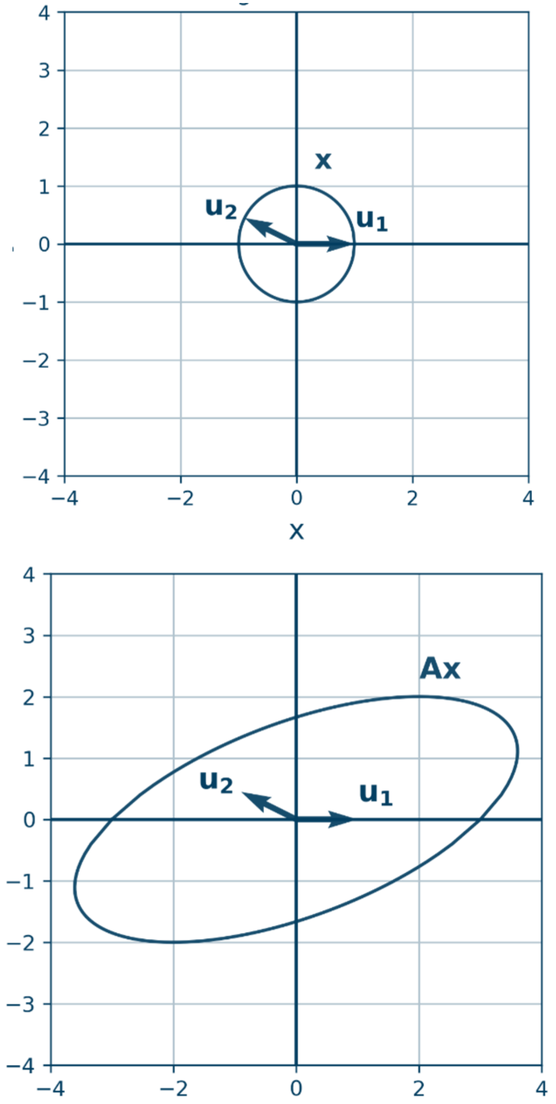
EM PYTHON $A = \begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix}$

PECE Programa de Educação Continuada
Escola Politécnica da USP

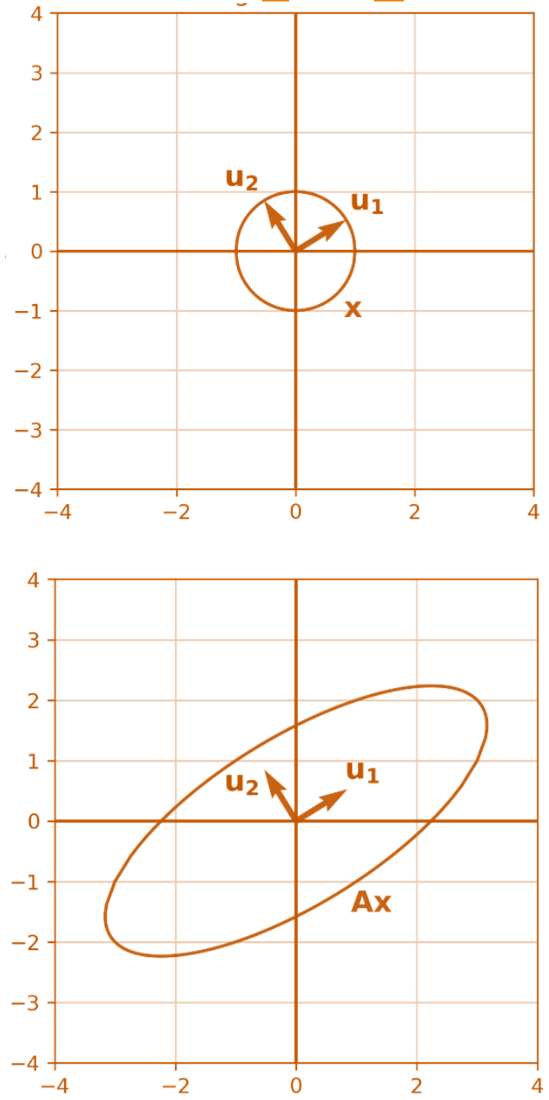
```
A = np.array([[3,2],[0,2]])
# Cálculo de autovalores e autovetores
lam, u = linalg.eig(A)
print("autovalores=\n", np.round(lam, 4))
print("autovetores=\n", np.round(u, 4))
```

```
autovalores=
 [3. 2.]
autovetores=
 [[ 1.        -0.8944]
 [ 0.         0.4472]]
```

$$A = \begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix}$$



$$A = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

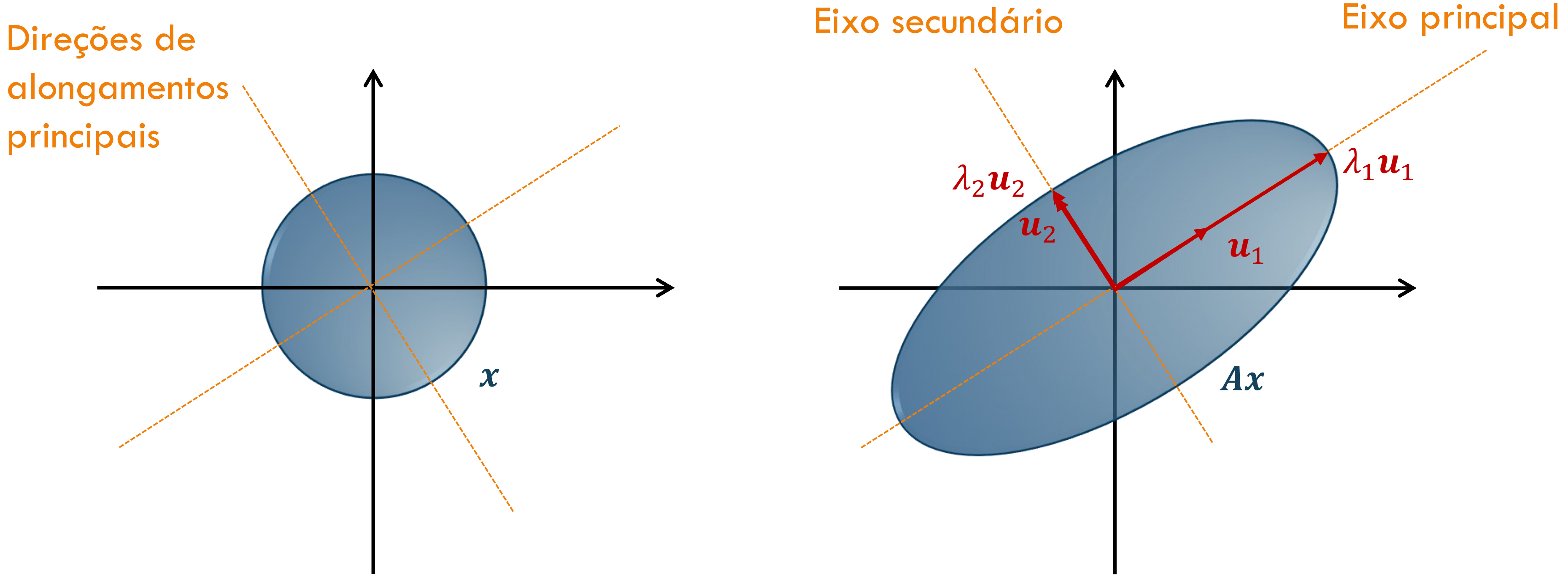


PROBLEMA DE AUTOVALORES E MATRIZES SIMÉTRICAS

Duas propriedades importantes surgem quando olhamos para os autovalores e autovetores de uma matriz simétrica $\boldsymbol{A}_{n\times n}$,

- todos n os autovalores de $\boldsymbol{A}$ são reais;
- os n autovetores de $\boldsymbol{A}$ são ortonormais, ou seja, a matriz composta dos autovetores é uma matriz ortogonal.

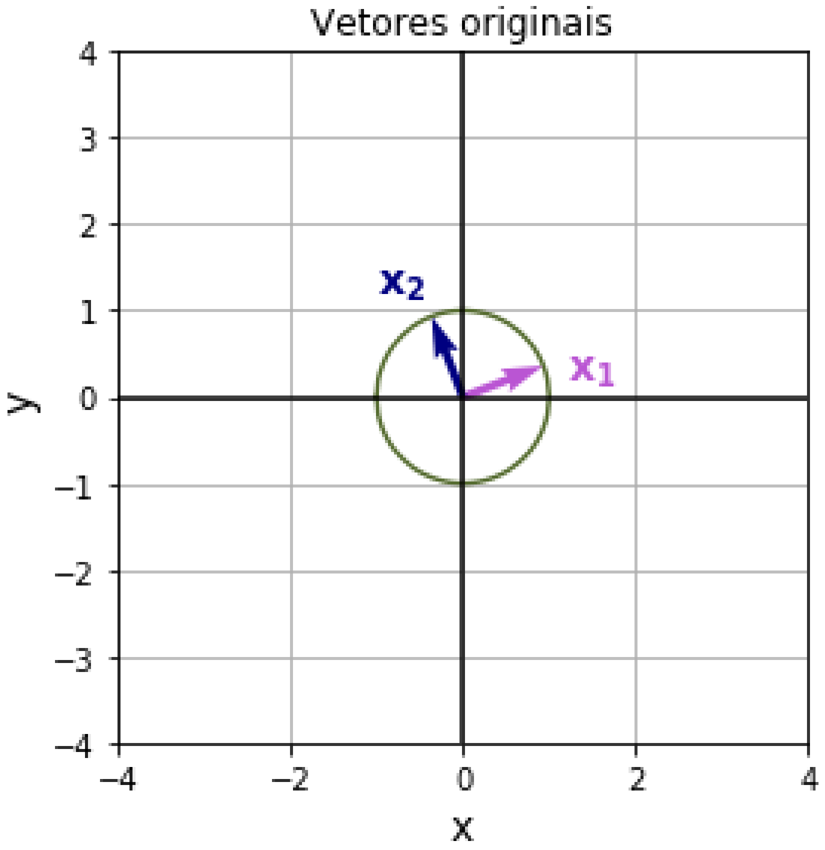
INTUIÇÃO GEOMÉTRICA PARA MATRIZ 2X2 SIMÉTRICA QUADRADA

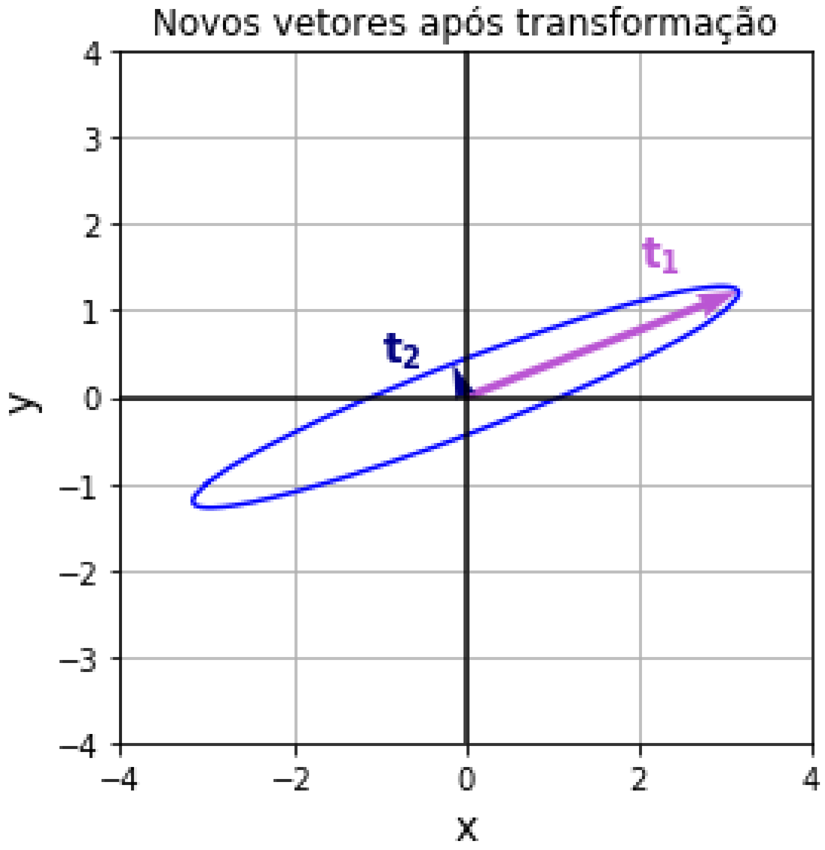


```python
A = np.array([[3, 1],
              [1, 0.8]])
# Cálculo de autovalores e autovetores
lam, u = linalg.eig(A)
print("autovalores=\n", np.round(lam, 4))
print("autovetores=\n", np.round(u, 4))
```

```
autovalores=
[3.3866 0.4134]
autovetores=
[[ 0.9327 -0.3606]
[ 0.3606 0.9327]]
```

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 0{,}8 \end{bmatrix}$$





DECOMPOSIÇÃO EM AUTOVALORES

Toda matriz simétrica quadrada pode ser decomposta em,

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T$$

$$\boldsymbol{A} = \begin{pmatrix} | & | & & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_n \\ | & | & & | \end{pmatrix} \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{pmatrix} \begin{pmatrix} - & \boldsymbol{u}_1^T & - \\ - & \boldsymbol{u}_2^T & - \\ & \vdots & \\ - & \boldsymbol{u}_n^T & - \end{pmatrix}$$

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$$

Veja que $\lambda_i \geq 0$ significa matriz definida positiva (ou semidefinida, se pelo possuir qualquer $\lambda_i = 0$)

onde $\boldsymbol{u}_i$ são n LI autovetores de $\boldsymbol{A}$ e λ_i são os autovalores de $\boldsymbol{A}$

POR EXEMPLO

$$\boldsymbol{A} = \begin{pmatrix} 7 & 3 \\ 3 & -1 \end{pmatrix}$$

$$\det \boldsymbol{A} = -16, tr\ \boldsymbol{A} = 6$$

$$\lambda^2 - 6\lambda - 16 = 0$$

```
A = np.array([[7,3],[3,-1]])
# Cálculo de autovalores e autovetores
lam, u = linalg.eig(A)
print("autovalores=\n", np.round(lam, 4))
print("autovetores=\n", np.round(u, 4))
```

```
autovalores=
[ 8. -2.]
autovetores=
[[ 0.9487 -0.3162]
[ 0.3162 0.9487]]
```

$$\lambda_1 = 8 \qquad u_1 = \begin{pmatrix} 0{,}9487 \\ 0{,}3162 \end{pmatrix}$$

$$\lambda_2 = -2 \qquad u_2 = \begin{pmatrix} -0{,}3162 \\ 0{,}9487 \end{pmatrix}$$

POR EXEMPLO...

$$\boxed{\boldsymbol{A} = \begin{pmatrix} 7 & 3 \\ 3 & -1 \end{pmatrix}}$$

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T$$

$$\boldsymbol{A} = \begin{pmatrix} | & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 \\ | & | \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} - & \boldsymbol{u}_1^T & - \\ - & \boldsymbol{u}_2^T & - \end{pmatrix}$$

$$\lambda_1 = 8 \quad \rightarrow \quad u_1 = \begin{pmatrix} 0{,}9487 \\ 0{,}3162 \end{pmatrix}$$

$$\lambda_2 = -2 \quad \rightarrow \quad u_2 = \begin{pmatrix} -0{,}3162 \\ 0{,}9487 \end{pmatrix}$$

$$\boldsymbol{U} = \begin{pmatrix} 0{,}9487 & -0{,}3162 \\ 0{,}3162 & 0{,}9487 \end{pmatrix} \qquad \boldsymbol{S} = \begin{pmatrix} 8 & 0 \\ 0 & -2 \end{pmatrix}$$

$$\boldsymbol{A} = \begin{pmatrix} 0{,}9487 & -0{,}3162 \\ 0{,}3162 & 0{,}9487 \end{pmatrix} \begin{pmatrix} 8 & 0 \\ 0 & -2 \end{pmatrix} \begin{pmatrix} 0{,}9487 & 0{,}3162 \\ -0{,}3162 & 0{,}9487 \end{pmatrix}$$

$$\boldsymbol{A} = \begin{pmatrix} 0{,}9487 \times 8 + (-0{,}3162) \times 0 & 0{,}9487 \times 0 + (-0{,}3162) \times (-2) \\ 0{,}3162 \times 8 + 0{,}9487 \times 0 & 0{,}3162 \times 0 + 0{,}9487 \times (-2) \end{pmatrix} \begin{pmatrix} 0{,}9487 & 0{,}3162 \\ -0{,}3162 & 0{,}9487 \end{pmatrix}$$

$$\boldsymbol{A} = \begin{pmatrix} 7{,}5896 & 0{,}6324 \\ 2{,}5296 & -1{,}8974 \end{pmatrix} \begin{pmatrix} 0{,}9487 & 0{,}3162 \\ -0{,}3162 & 0{,}9487 \end{pmatrix} = \boxed{\begin{pmatrix} 7{,}0002 & 2{,}9998 \\ 2{,}9998 & -1{,}0002 \end{pmatrix}}$$

OUTRO EXEMPLO

$$\boldsymbol{A} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

```
A = np.array([[3, 1],
              [1, 2]])
lam, u = linalg.eig(A)
print("autovalores=\n", np.round(lam, 4))
print("autovetores=\n", np.round(u, 4))
S=np.array([[lam[0], 0],
              [0, lam[1]]])
u1 = np.array(u[:,0])
u2 =np.array(u[:,1])
P=np.array([u1,u2]).T

t = P @ S @ P.T
print("Recupera-se a matriz A\n",t)
```

```
autovalores=
[3.618 1.382]
autovetores=
[[ 0.8507 -0.5257]
[ 0.5257 0.8507]]
Recupera-se a matriz
A [[3. 1.]
[1. 2.]]
```

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T$$

$$\boldsymbol{A} = \begin{pmatrix} | & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 \\ | & | \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} - & \boldsymbol{u}_1^T & - \\ - & \boldsymbol{u}_2^T & - \end{pmatrix}$$

$$\boldsymbol{u_1} = \begin{bmatrix} 0.8507 \\ 0.5257 \end{bmatrix}, \lambda_1 = 3.618$$

$$\boldsymbol{u_2} = \begin{bmatrix} -0.5257 \\ 0.8507 \end{bmatrix}, \lambda_2 = 1.382$$

$$\boldsymbol{S} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} 3.618 & 0 \\ 0 & 1.382 \end{bmatrix}$$

$$\boldsymbol{U} = \begin{bmatrix} 0.8507 & -0.5257 \\ 0.5257 & 0.8507 \end{bmatrix}$$

$$\boldsymbol{U}^T = [\boldsymbol{u_1} \quad \boldsymbol{u_2}]^T = \begin{bmatrix} \boldsymbol{u_1^T} \\ \boldsymbol{u_2^T} \end{bmatrix} = \begin{bmatrix} 0.8507 & 0.5257 \\ -0.5257 & 0.8507 \end{bmatrix}$$

$$A = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 0.8507 & -0.5257 \\ 0.5257 & 0.8507 \end{bmatrix} \begin{bmatrix} 3.618 & 0 \\ 0 & 1.382 \end{bmatrix} \begin{bmatrix} 0.8507 & 0.5257 \\ -0.5257 & 0.8507 \end{bmatrix}$$

REPARE QUE...

$$A = \boxed{US}U^T$$

$$S = \begin{bmatrix} 3.618 & 0 \\ 0 & 1.382 \end{bmatrix}$$

$$A = \boxed{\begin{pmatrix} | & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 \\ | & | \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}} \begin{pmatrix} - & \boldsymbol{u}_1^T & - \\ - & \boldsymbol{u}_2^T & - \end{pmatrix}$$

$$U = \begin{bmatrix} 0.8507 & -0.5257 \\ 0.5257 & 0.8507 \end{bmatrix}$$

$$U = \begin{bmatrix} \boxed{0.8507} \times \boxed{3.618} + \cancel{(-0.5257) \times 0} & \cancel{0.8507 \times 0} + \boxed{(-0.5257)} \times \boxed{1.382} \\ \boxed{0.5257} \times \boxed{3.618} + \cancel{0.8507 \times 0} & \cancel{0.5257 \times 0} + \boxed{0.8507} \times \boxed{1.382} \end{bmatrix}$$

$$\boldsymbol{US} = \begin{pmatrix} | & | \\ \lambda_1 \boldsymbol{u}_1 & \lambda_2 \boldsymbol{u}_2 \\ | & | \end{pmatrix}$$

REPARE QUE...

$$\boldsymbol{A} = \boldsymbol{U}\boxed{\boldsymbol{S}\boldsymbol{U}^T}$$

$$\boldsymbol{S} = \begin{bmatrix} 3.618 & 0 \\ 0 & 1.382 \end{bmatrix}$$

$$\boldsymbol{U} = \begin{bmatrix} 0.8507 & -0.5257 \\ 0.5257 & 0.8507 \end{bmatrix}$$

$$\boldsymbol{A} = \begin{pmatrix} | & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 \\ | & | \end{pmatrix} \boxed{\begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} - & \boldsymbol{u}_1^T & - \\ - & \boldsymbol{u}_2^T & - \end{pmatrix}}$$

Similarmente,

$$\boldsymbol{US} = \begin{pmatrix} | & | \\ \lambda_1 \boldsymbol{u}_1 & \lambda_2 \boldsymbol{u}_2 \\ | & | \end{pmatrix}$$

$$\boldsymbol{US}^T = \begin{pmatrix} - & \lambda_1 \boldsymbol{u}_1^T & - \\ - & \lambda_2 \boldsymbol{u}_2^T & - \end{pmatrix}$$

GENERALIZANDO...

$$\boldsymbol{S}\boldsymbol{U}^T \rightarrow \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{u_1}^T \\ \mathbf{u_2}^T \\ \vdots \\ \mathbf{u_n}^T \end{bmatrix} =$$

$$\begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ u_{n1} & u_{n2} & \dots & u_{nn} \end{bmatrix} =$$

$$\begin{bmatrix} \lambda_1 u_{11} & \lambda_1 u_{12} & \dots & \lambda_1 u_{1n} \\ \lambda_2 u_{21} & \lambda_2 u_{22} & \dots & \lambda_2 u_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_n u_{n1} & \lambda_n u_{n2} & \dots & \lambda_n u_{nn} \end{bmatrix} = \begin{bmatrix} \lambda_1 \mathbf{u_1}^T \\ \lambda_2 \mathbf{u_2}^T \\ \vdots \\ \lambda_n \mathbf{u_n}^T \end{bmatrix}$$

PORTANTO...

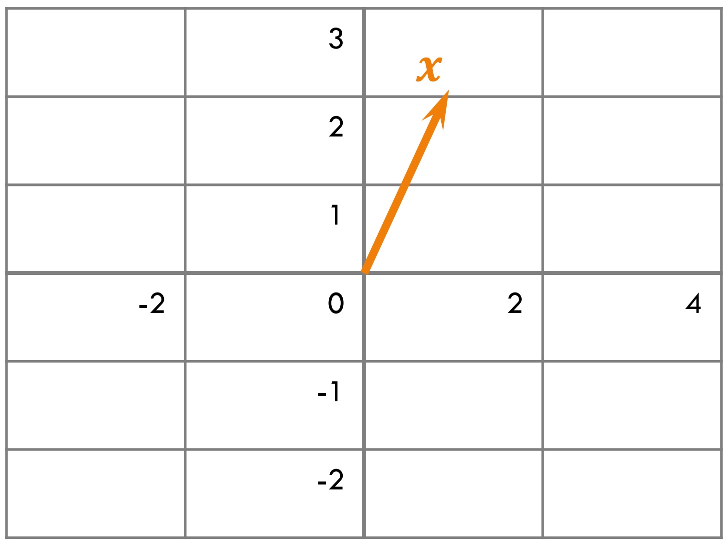
$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{u_1} & \mathbf{u_2} & \dots & \mathbf{u_n} \end{bmatrix} \begin{bmatrix} \lambda_1 \mathbf{u_1}^T \\ \lambda_2 \mathbf{u_2}^T \\ \vdots \\ \lambda_n \mathbf{u_n}^T \end{bmatrix} =$$

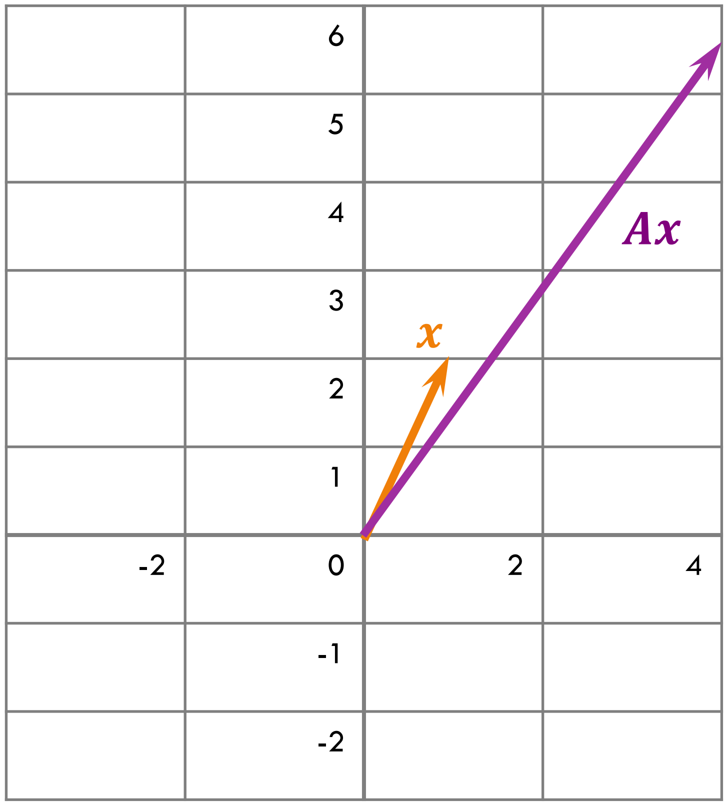
$$\boldsymbol{A} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^T + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^T + \cdots + \lambda_n \boldsymbol{u}_n \boldsymbol{u}_n^T$$

Portanto, a matriz $\boldsymbol{A}$ pode ser decomposta em n matrizes $\boldsymbol{u}_i \boldsymbol{u}_i^T$ $n \times n$ ponderadas de λ_i

VAMOS DEVAGAR PARA ENTENDER...



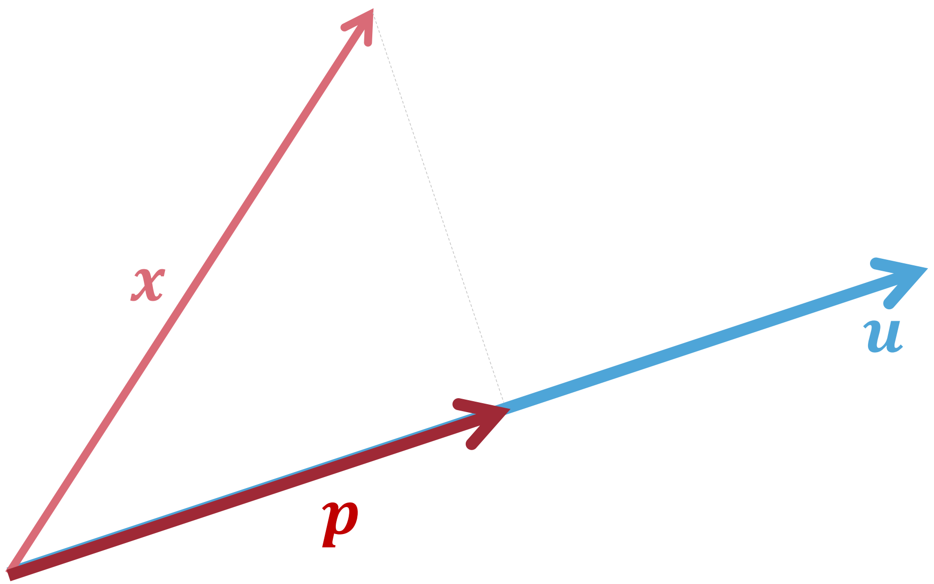
Suponha um vetor x qualquer, e aplicamos uma matriz $\boldsymbol{A}$ de transformação, simétrica $n \times n$.



https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d

$$\boldsymbol{Ax} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^T \boldsymbol{x} + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^T \boldsymbol{x}$$

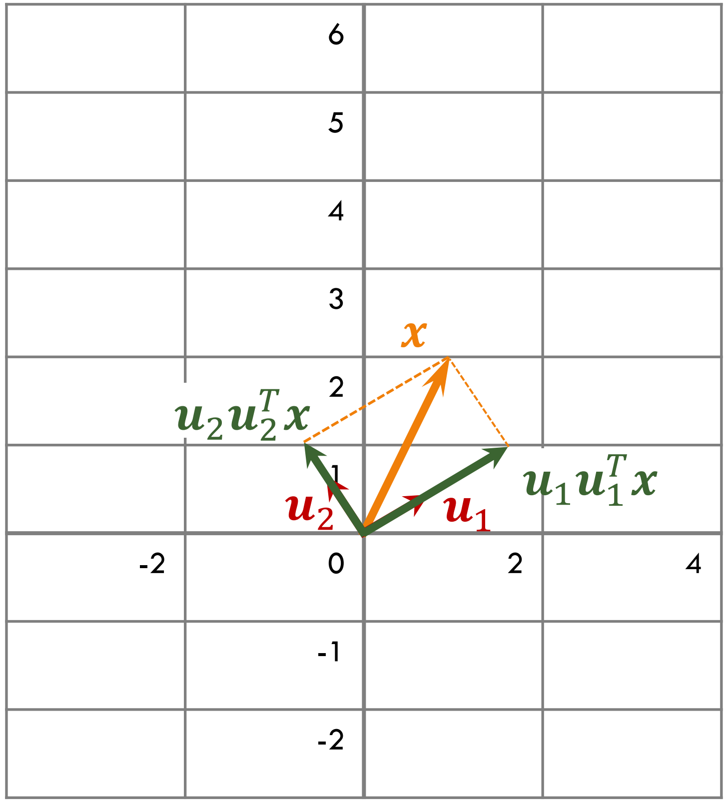
PROJEÇÃO



$$p = \frac{u}{u^T u} u^T x$$

Dimensão da componente de x ao longo de u

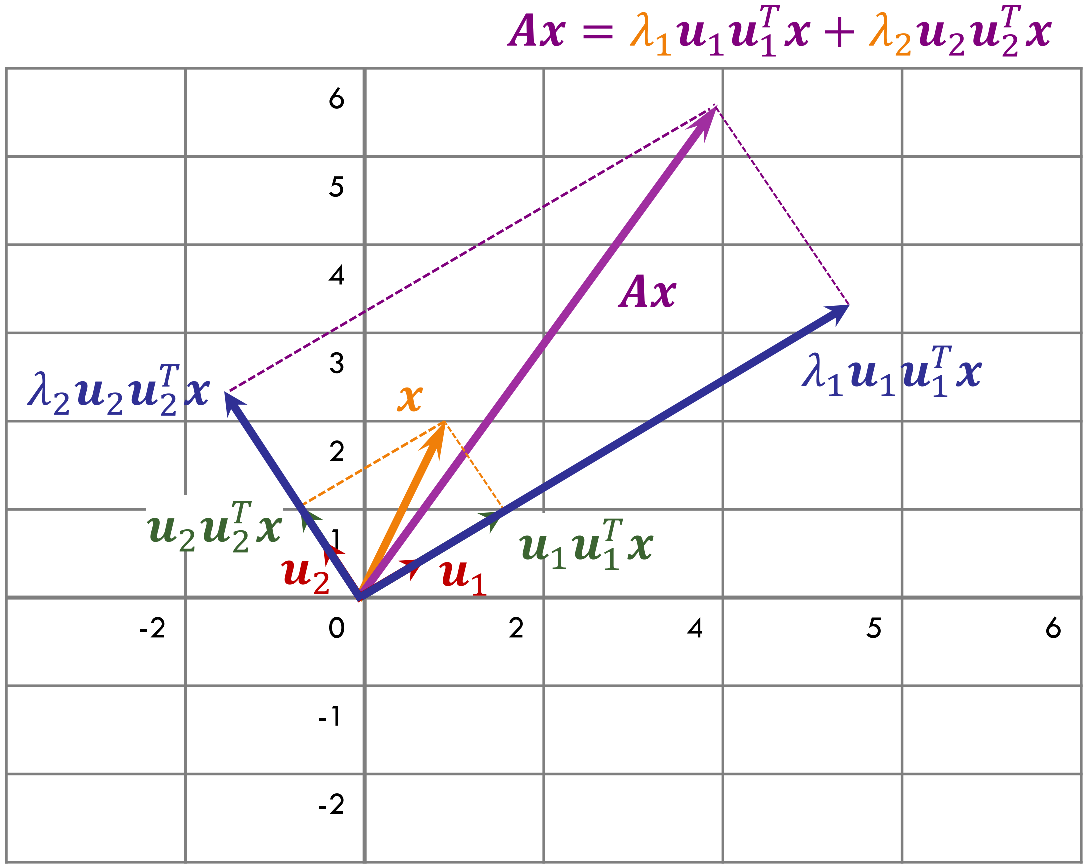
Vetor u normalizado dá a direção de p



$$Ax = \lambda_1 u_1 u_1^T x + \lambda_2 u_2 u_2^T x + \cdots + \lambda_n u_n u_n^T x$$

$u_i u_i^T x \rightarrow$ Projeta o vetor x na i−ésima direção principal!!!

$$Ax = \lambda_1 u_1 u_1^T x + \lambda_2 u_2 u_2^T x$$



Uma matriz simétrica transforma um vetor alongando-o ou encolhendo-o ao ***longo de seus autovetores***.

Além disso, sabemos que as matrizes transformam um autovetor ***multiplicando seu comprimento (ou magnitude) pelo autovalor correspondente***.

https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d

INSIGHT PARA MATRIZ SIMÉTRICA DEFINIDA POSITIVA...

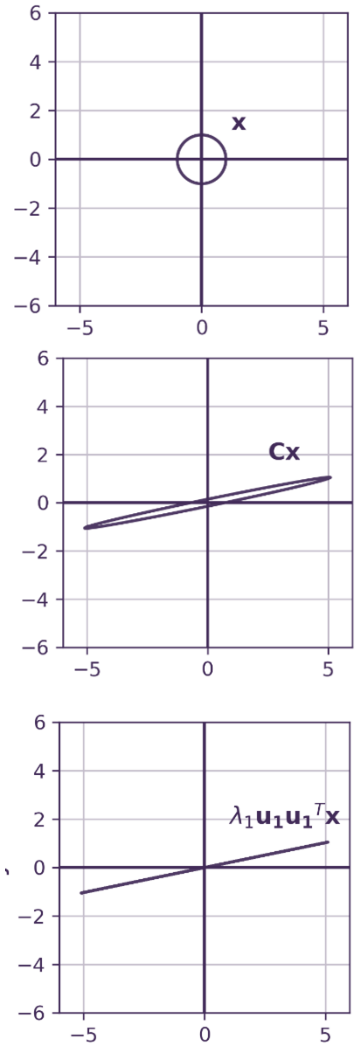
$$\boldsymbol{A} = \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^T + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^T + \cdots + \lambda_k \boldsymbol{u}_k \boldsymbol{u}_k^T + \cdots + \lambda_n \boldsymbol{u}_n \boldsymbol{u}_n^T$$
$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k \cdots \geq \lambda_n$$

Quanto maior o autovalor, maior o comprimento do vetor resultante $\lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^T$ e mais peso é dado à sua matriz correspondente $\boldsymbol{u}_i \boldsymbol{u}_i^T$.

Assim, podemos aproximar nossa matriz simétrica original $\boldsymbol{A}$ somando os termos que têm os maiores autovalores,

$$\boldsymbol{A} \approx \lambda_1 \boldsymbol{u}_1 \boldsymbol{u}_1^T + \lambda_2 \boldsymbol{u}_2 \boldsymbol{u}_2^T + \cdots + \lambda_k \boldsymbol{u}_k \boldsymbol{u}_k^T$$
$$\boldsymbol{A} \approx \boldsymbol{A}_1 + \boldsymbol{A}_2 + \cdots + \boldsymbol{A}_k$$

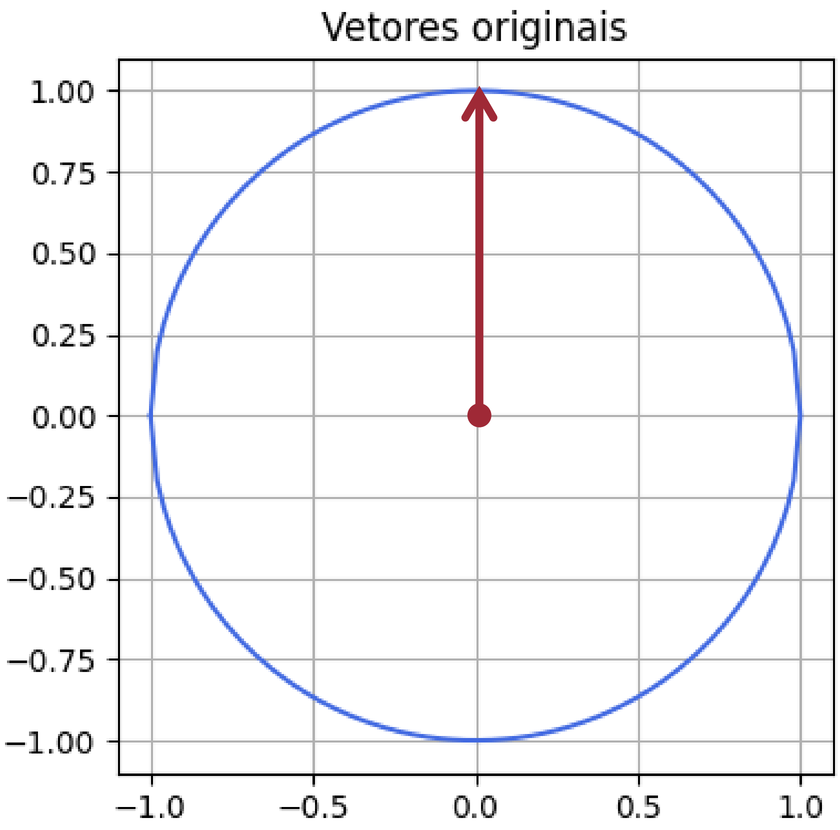
onde $\boldsymbol{A}_i$ é uma matriz de projeção e deve projetar tudo em $\boldsymbol{u}_i$.



https://github.com/reza-bagheri/SVD_article

ACHE A PROJEÇÃO A_1

$$\boldsymbol{A} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$



ACHE AS PROJEÇÕES A_1 E A_2

Já usamos essa matriz...

$$\boldsymbol{A} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\boldsymbol{u_1} = \begin{bmatrix} 0.8507 \\ 0.5257 \end{bmatrix}, \lambda_1 = 3.618$$

$$\boldsymbol{u_2} = \begin{bmatrix} -0.5257 \\ 0.8507 \end{bmatrix}, \lambda_2 = 1.382$$

```
# Cálculo de autovalores e autovetores
#A = np.array([[3, 2],
#              [0, 2]])
A = np.array([[3, 1],
              [1, 2]])
lam, u = linalg.eig(A)
print("autovalores=", np.round(lam, 4))
print("autovetores=", np.round(u, 4))

```

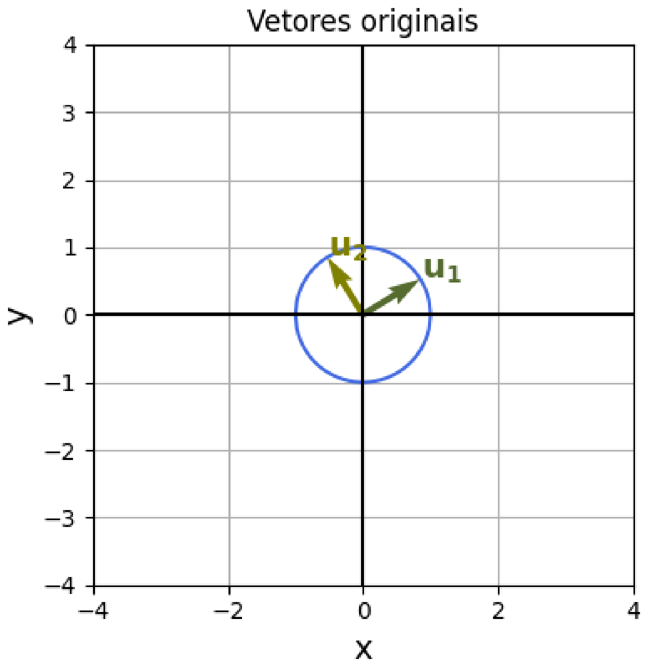
```
autovalores= [3.618 1.382]
autovetores= [[ 0.8507 -0.5257]
 [ 0.5257  0.8507]]
```

ACHE A PROJEÇÃO A_1

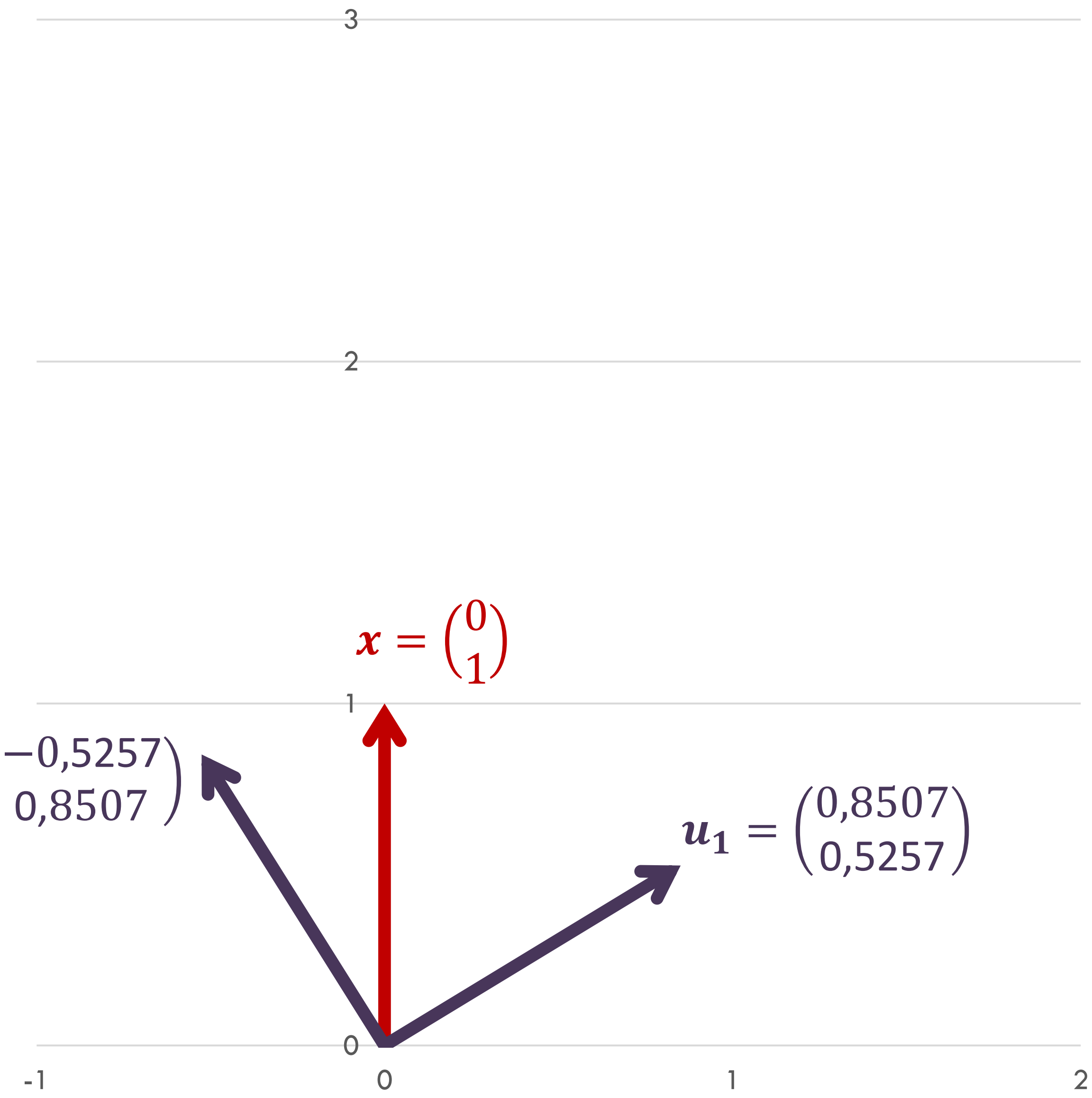
$$\boldsymbol{A} = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\boldsymbol{u_1} = \begin{bmatrix} 0.8507 \\ 0.5257 \end{bmatrix}, \lambda_1 = 3.618$$

$$\boldsymbol{u_2} = \begin{bmatrix} -0.5257 \\ 0.8507 \end{bmatrix}, \lambda_2 = 1.382$$



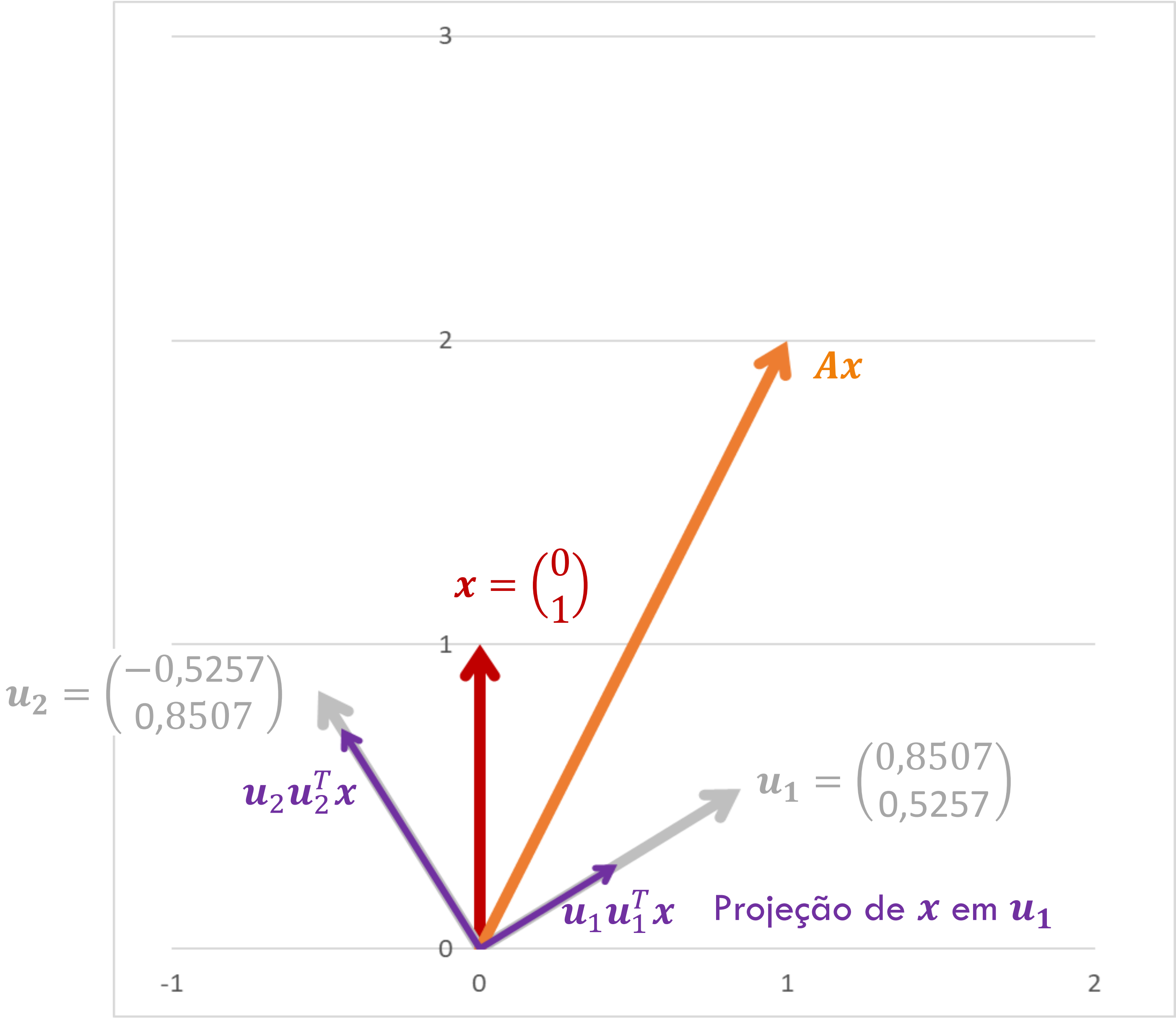
$$A = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$



$$\boldsymbol{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\boldsymbol{u_2} = \begin{pmatrix} -0{,}5257 \\ 0{,}8507 \end{pmatrix}$$

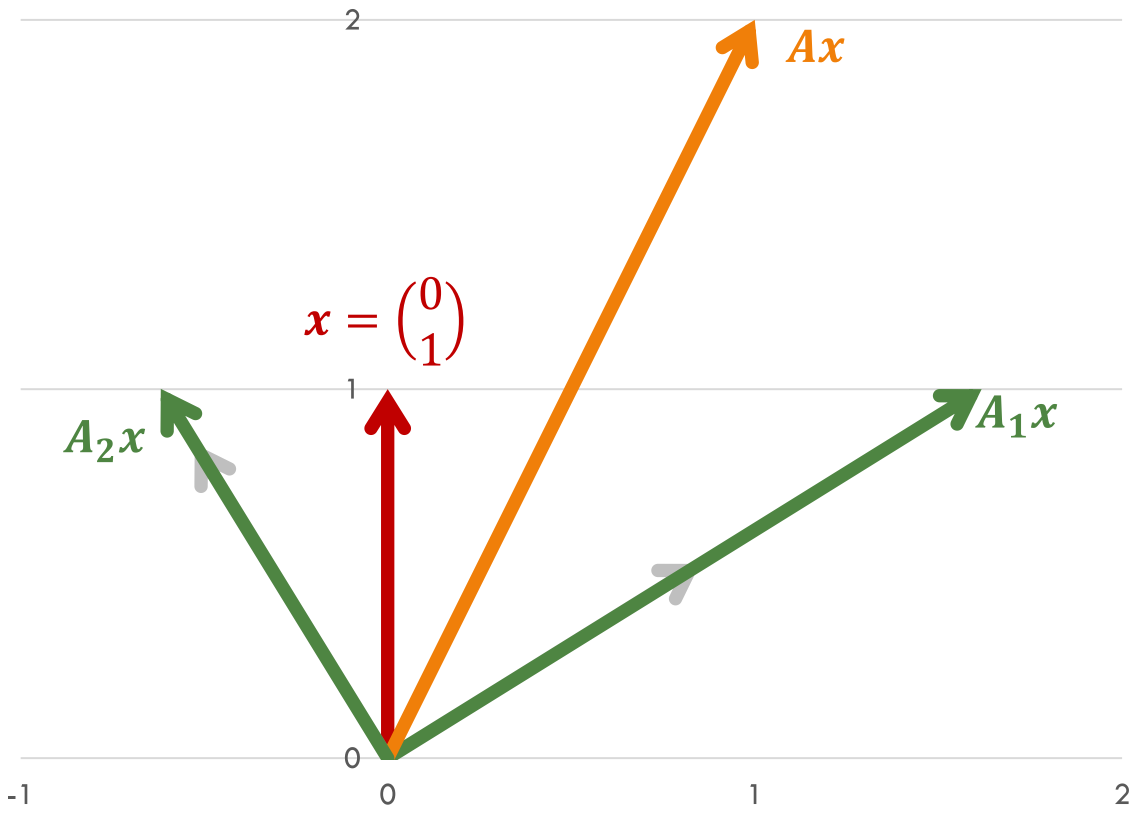
$$\boldsymbol{u_1} = \begin{pmatrix} 0{,}8507 \\ 0{,}5257 \end{pmatrix}$$



$$A = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\boldsymbol{A_1} = \lambda_1 \boldsymbol{u_1} \boldsymbol{u_1^T} = 3.61 \begin{bmatrix} 0.8507 \\ 0.5257 \end{bmatrix} \begin{bmatrix} 0.8507 & 0.5257 \end{bmatrix} = \begin{bmatrix} 2.618 & 1.618 \\ 1.618 & 1 \end{bmatrix}$$

$$\boldsymbol{A_2} = \lambda_2 \boldsymbol{u_2} \boldsymbol{u_2^T} = 1.382 \begin{bmatrix} -0.5257 \\ 0.8507 \end{bmatrix} \begin{bmatrix} -0.5257 & 0.8507 \end{bmatrix} = \begin{bmatrix} 0.382 & -0.618 \\ -0.618 & 1 \end{bmatrix}$$



PECE Programa de Educação Continuada
Escola Politécnica da USP

ACHE A PROJEÇÃO A_1

$$\boldsymbol{A} = \begin{pmatrix} 5 & 2 \\ 2 & 1 \end{pmatrix}$$

$$\boldsymbol{u_1} = \begin{bmatrix} 0.9239 \\ 0.3287 \end{bmatrix}, \lambda_1 = 5.8284$$

$$\boldsymbol{u_2} = \begin{bmatrix} -0.3827 \\ 0.9239 \end{bmatrix}, \lambda_2 = 0.1716$$

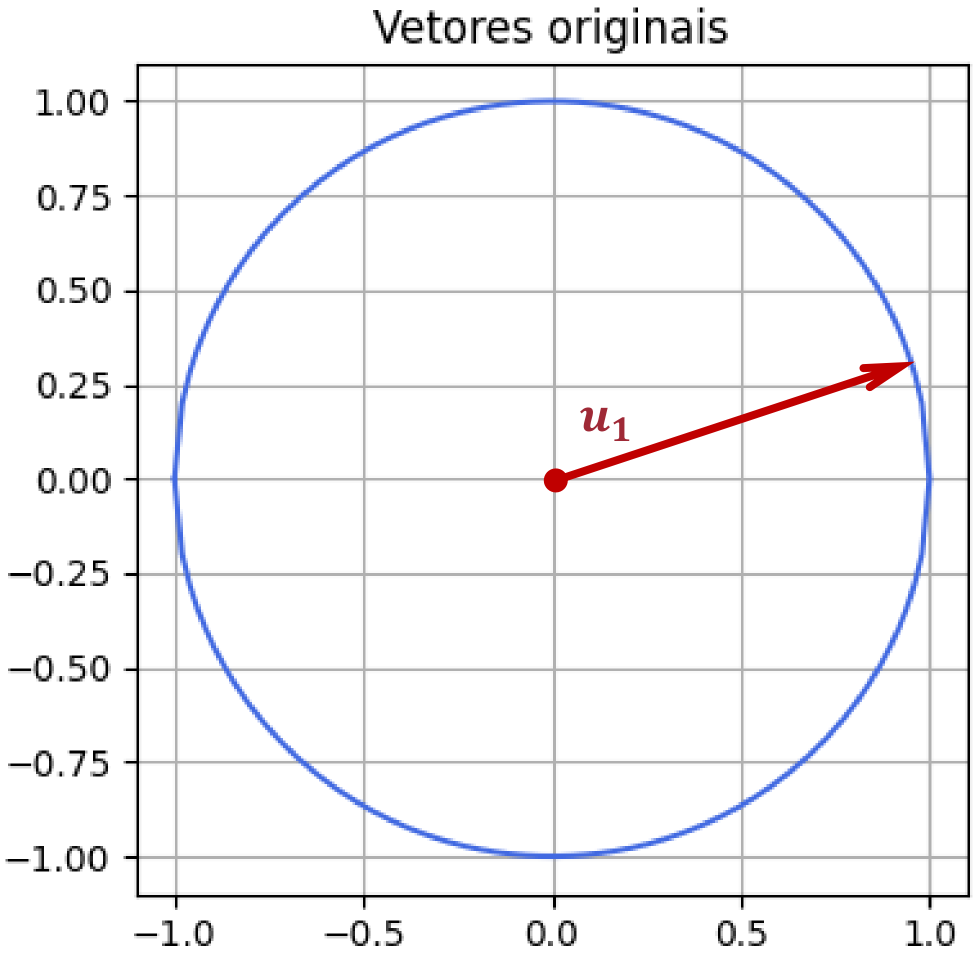
$$\boldsymbol{A_1} = \lambda_1 \boldsymbol{u_1} \boldsymbol{u_1^T} = \begin{bmatrix} 4.9749 & 2.0607 \\ 2.0606 & 0.8535 \end{bmatrix}$$

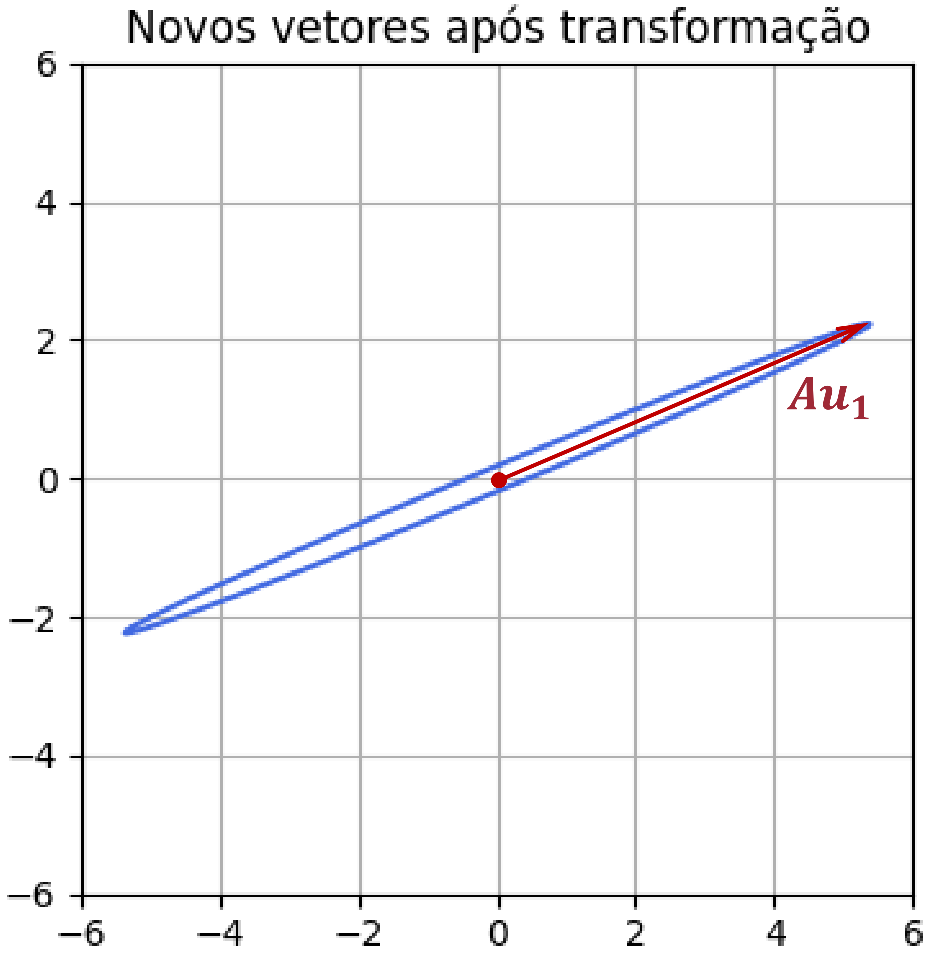
```python
A = np.array([[5, 2],
              [2,1]])
lam, u = linalg.eig(A)
print("autovalores=\n", np.round(lam, 4))
print("autovetores=\n", np.round(u, 4))
S=np.array([[lam[0], 0],
              [0, lam[1]]])
u1 = np.array(u[:,0])
u2 =np.array(u[:,1])
P=np.array([u1,u2]).T


t = P @ S @ P.T
print("Recupera-se a matriz A\n",t)
```

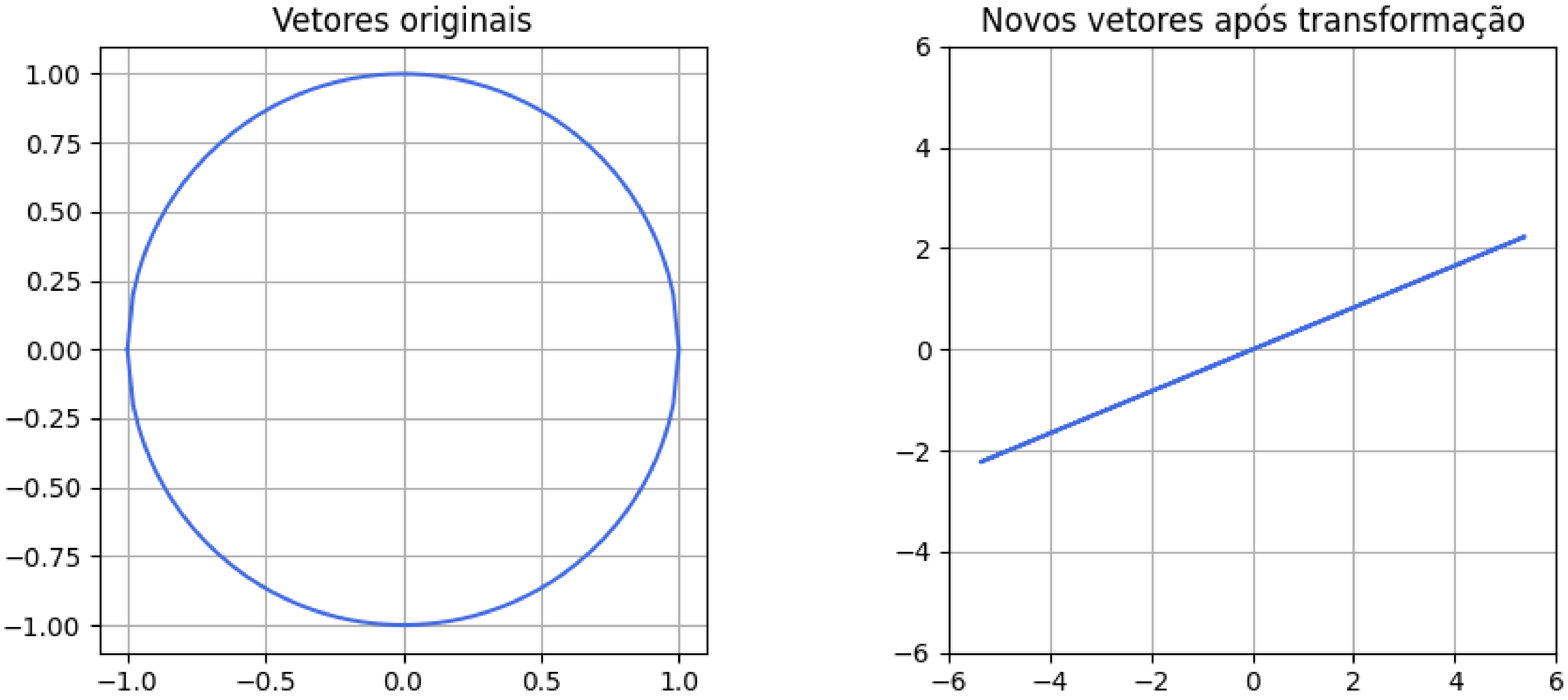
```
autovalores=
 [5.8284 0.1716]
autovetores=
 [[ 0.9239 -0.3827]
 [ 0.3827  0.9239]]
Recupera-se a matriz A
 [[5. 2.]
 [2. 1.]]
```

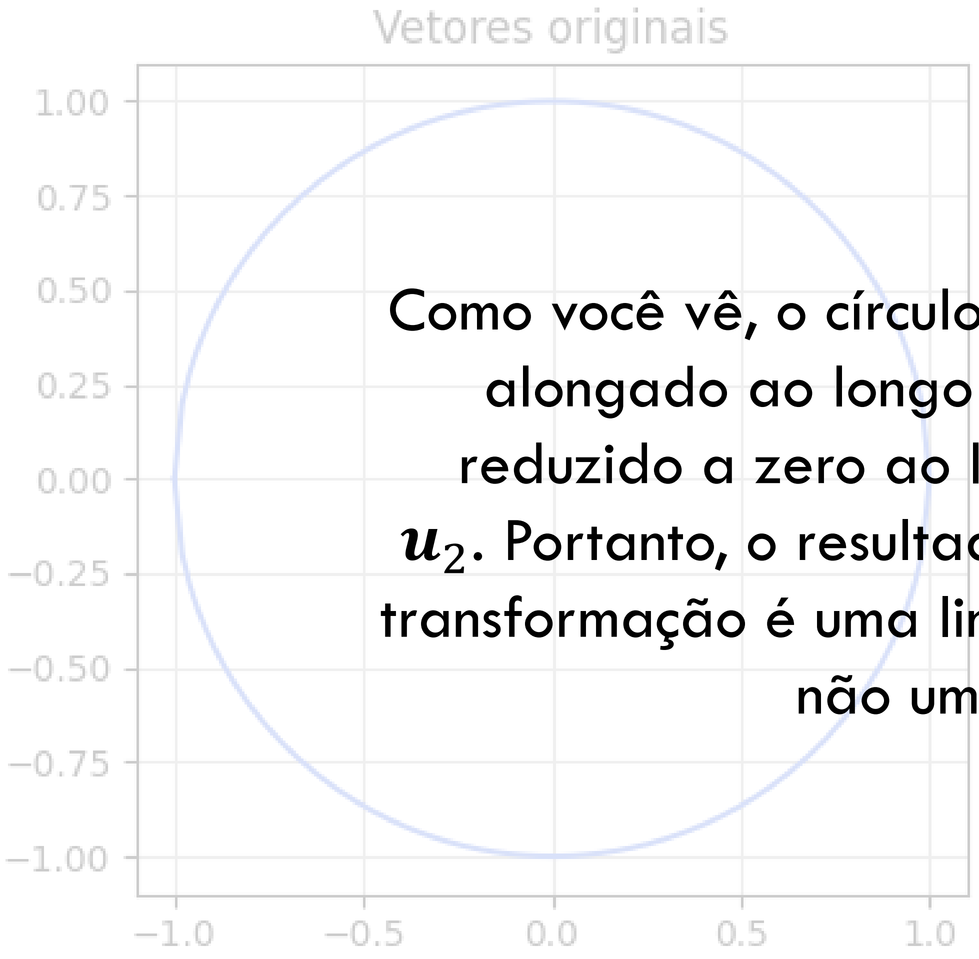
USANDO $A_1 = \lambda_1 u_1 u_1^T + \lambda_2 u_2 u_2^T$



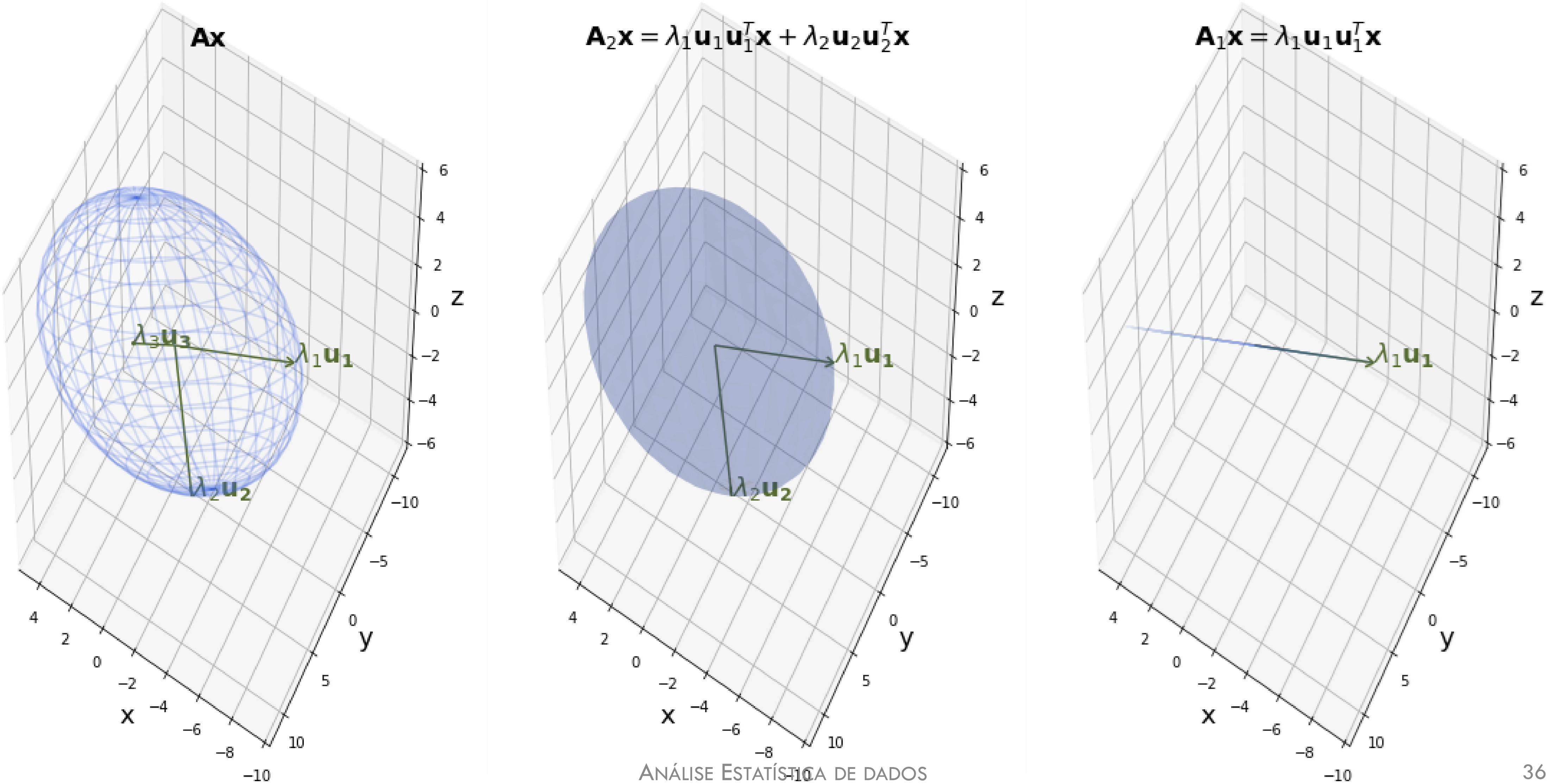


USANDO $A_1 = \lambda_1 u_1 u_1^T$





Como você vê, o círculo inicial é alongado ao longo de $\boldsymbol{u}_1$ e reduzido a zero ao longo de $\boldsymbol{u}_2$. Portanto, o resultado dessa transformação é uma linha reta, não uma elipse.


$\mathbf{Ax}$
$\mathbf{A}_2\mathbf{x} = \lambda_1\mathbf{u}_1\mathbf{u}_1^T\mathbf{x} + \lambda_2\mathbf{u}_2\mathbf{u}_2^T\mathbf{x}$
$\mathbf{A}_1\mathbf{x} = \lambda_1\mathbf{u}_1\mathbf{u}_1^T\mathbf{x}$
$\lambda_1\mathbf{u_1}$
$\lambda_2\mathbf{u_2}$
$\lambda_3\mathbf{u_3}$
x
y
z

Se você tiver não uma matriz quadrada simétrica, não poderá usar o método de composição por autovalores para aproximar matrizes...

E a maioria das matrizes que você irá trabalhar não são nem quadradas e nem simétricas...

VALORES SINGULARES

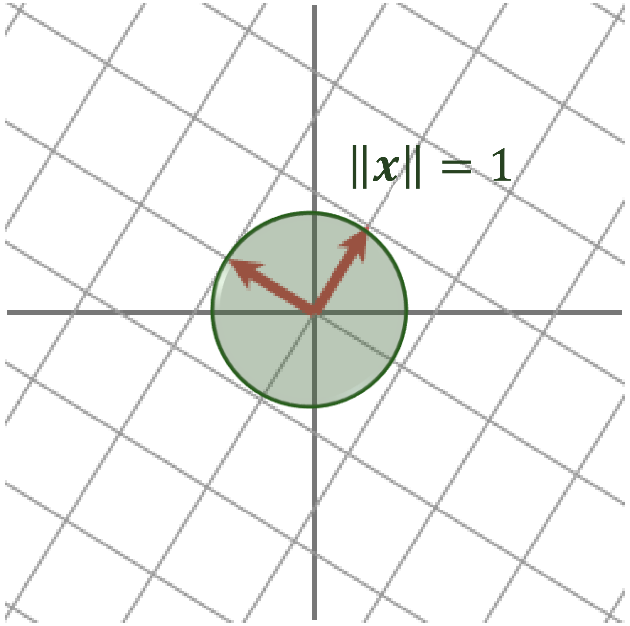
Antes de falar em SVD devemos encontrar uma maneira de calcular as direções de alongamento para uma matriz não simétrica. Para isso, definimos “valores singulares”.

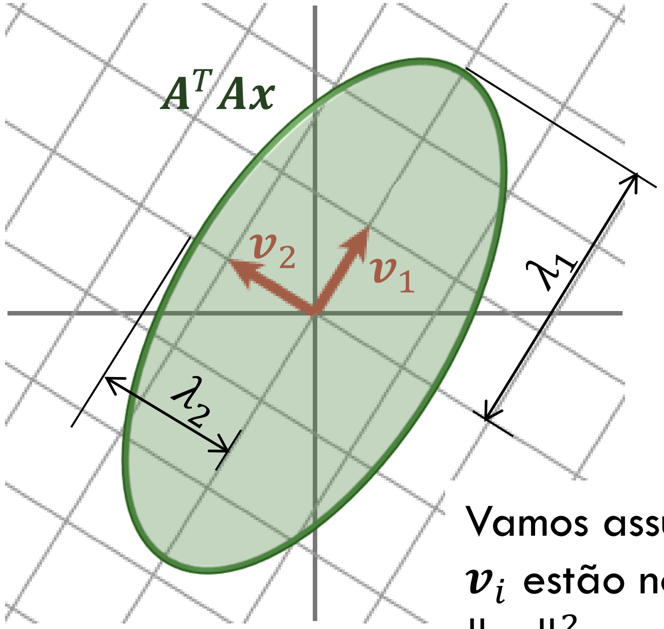
$$(A^T A)^T = A^T (A^T)^T = A^T A$$

MATRIZ $A^T A$

$\boldsymbol{A}$ é uma matriz $m \times n$ e não necessariamente simétrica.

A matriz $\boldsymbol{A}^T\boldsymbol{A}$ é positiva semi definida, quadrada $(n \times n)$ e simétrica. Portanto, possui autovetores $\boldsymbol{v}_i$ ortogonais e autovalores λ_i não negativos.



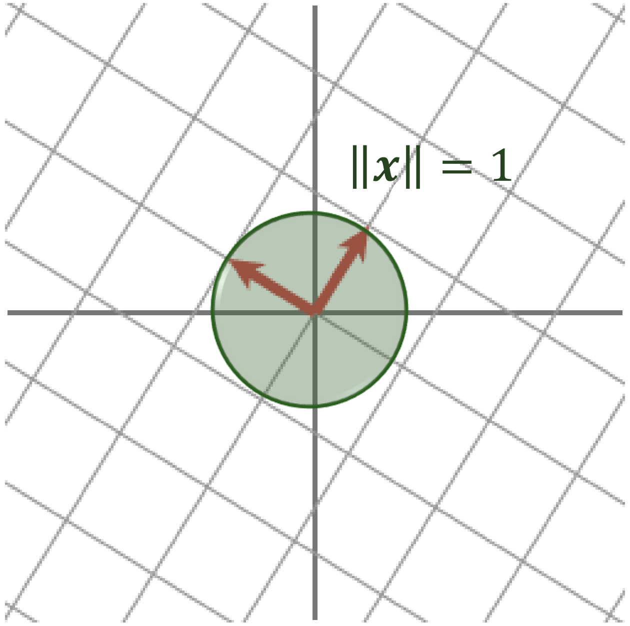


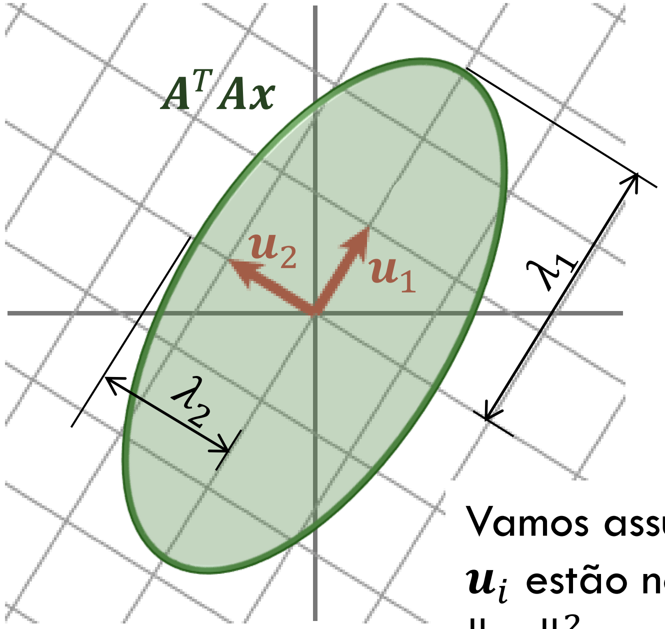
Vamos assumir que os vetores $\boldsymbol{v}_i$ estão normalizados, ie, $\|\boldsymbol{v}_i\|^2 = \boldsymbol{v}_i^T\boldsymbol{v}_i = 1$

MATRIZ AA^T

$\boldsymbol{A}$ é uma matriz $m \times n$ e não necessariamente simétrica.

A matriz $\boldsymbol{AA}^T$ é positiva semi definida, quadrada $(n \times n)$ e simétrica. Portanto, possui autovetores $\boldsymbol{u}_i$ ortogonais e autovalores λ_i não negativos.





Vamos assumir que os vetores $\boldsymbol{u}_i$ estão normalizados, ie, $\|\boldsymbol{u}_i\|^2 = \boldsymbol{u}_i^T\boldsymbol{u}_i = 1$

EXEMPLO

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix}$$

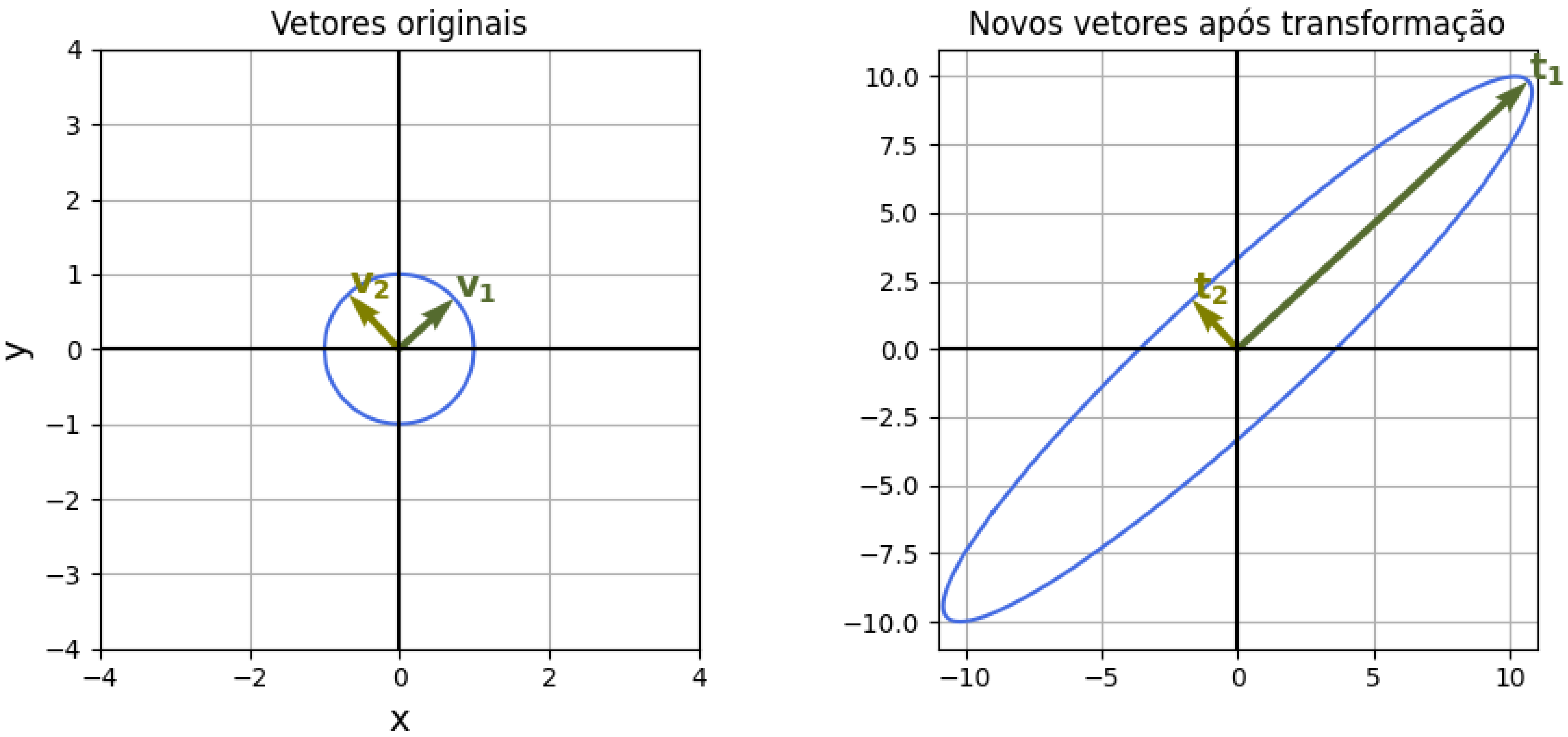
$$\boldsymbol{A^T A} = \begin{bmatrix} 9 & 6 \\ 6 & 8 \end{bmatrix}$$

$$\boldsymbol{v_1} = \begin{bmatrix} 0.7359 \\ 0.6771 \end{bmatrix}, \lambda_1 = 14.5208$$

$$\boldsymbol{v_2} = \begin{bmatrix} -0.6771 \\ 0.7359 \end{bmatrix}, \lambda_2 = 2.4792$$

```python
ATA = A.T @ A
# Cálculo de autovalores e autovetores
lamv, v = linalg.eig(ATA)
print("A^TA=\n", np.round(ATA, 4))
print("autovalores=\n", np.round(lamv, 4))
print("autovetores=\n", np.round(v, 4))
```

```
A^TA=
 [[9 6]
 [6 8]]
autovalores=
 [14.5208  2.4792]
autovetores=
 [[ 0.7359 -0.6771]
 [ 0.6771  0.7359]]
```


Vetores originais
Novos vetores após transformação
x
y
v1
v2
t1
t2

EXEMPLO

$$\boldsymbol{A} = \begin{bmatrix} 3 & 2 \\ 0 & 2 \end{bmatrix}$$

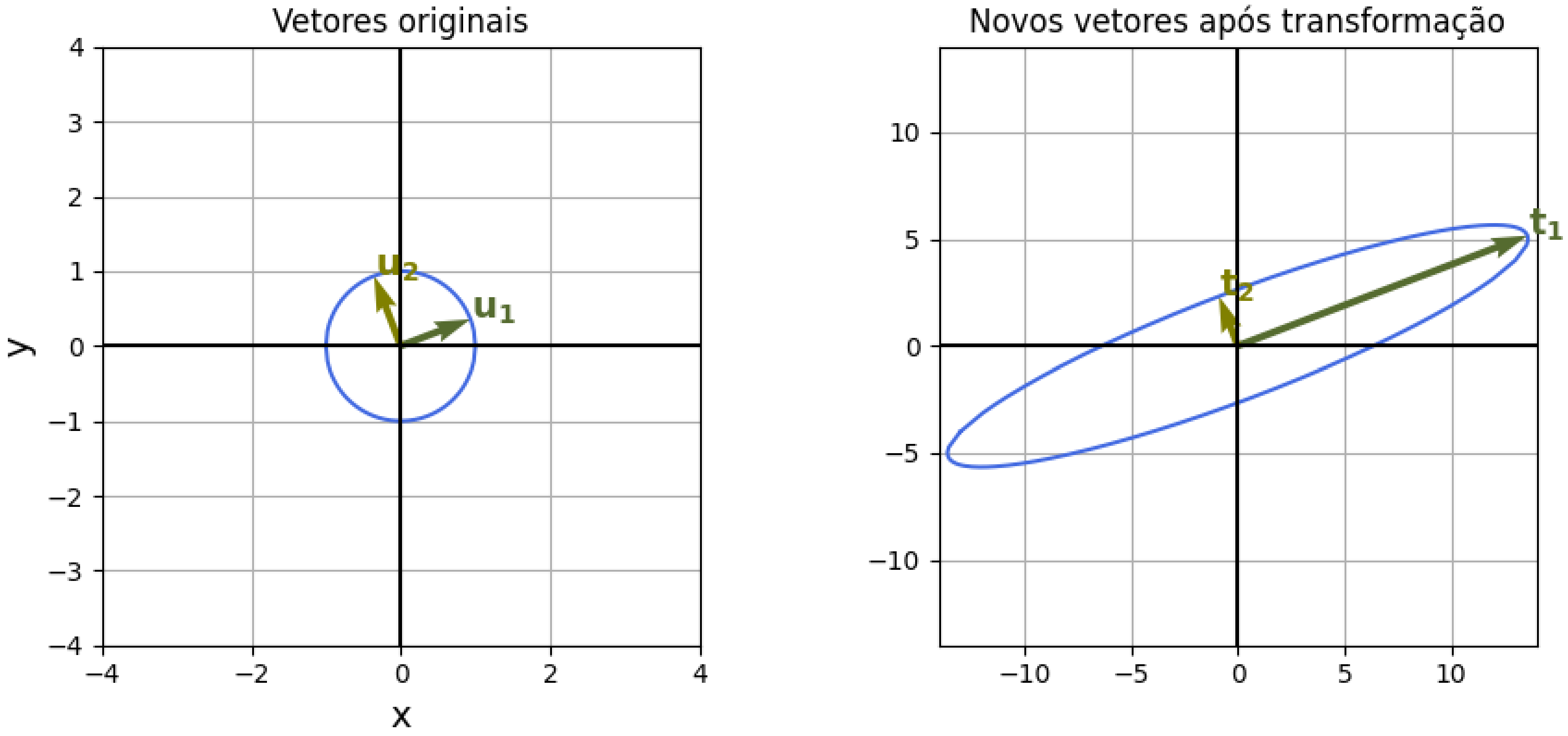
$$\boldsymbol{A}\boldsymbol{A}^T = \begin{bmatrix} 13 & 4 \\ 4 & 4 \end{bmatrix}$$

$$\boldsymbol{u_1} = \begin{bmatrix} 0.9347 \\ 0.3554 \end{bmatrix}, \lambda_1 = 14.5208$$

$$\boldsymbol{u_2} = \begin{bmatrix} -0.3554 \\ 0.9347 \end{bmatrix}, \lambda_2 = 2.4792$$

```
AAT = A @ A.T
# Cálculo de autovalores e autovetores
lamu, u = linalg.eig(AAT)
print("AA^T=\n", np.round(AAT, 4))
print("autovalores=\n", np.round(lamu, 4))
print("autovetores=\n", np.round(u, 4))
```

```
AA^T=
 [[13  4]
 [ 4  4]]
autovalores=
 [14.5208  2.4792]
autovetores=
 [[ 0.9347 -0.3554]
 [ 0.3554  0.9347]]
```


Vetores originais
4
3
2
1
0
−1
−2
−3
−4
y
−4
−2
0
2
4
x
u_2
u_1
Novos vetores após transformação
10
5
0
−5
−10
−10
−5
0
5
10
t_1
t_2

VETORES E VALORES SINGULARES

Definição final

VETORES E VALORES SINGULARES

Dada uma matriz $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, as matrizes à esquerda e à direita de $\boldsymbol{A}$ são, respectivamente,

$$\underset{m \times m}{\boldsymbol{A}\boldsymbol{A}^T} \text{ e } \underset{n \times n}{\boldsymbol{A}^T\boldsymbol{A}}.$$

As matrizes $\boldsymbol{A}\boldsymbol{A}^T$ e $\boldsymbol{A}^T\boldsymbol{A}$ têm os mesmos autovalores não nulos.

$\boldsymbol{A}\boldsymbol{A}^T$ e $\boldsymbol{A}^T\boldsymbol{A}$ são matrizes semi definidas positivas, simétricas, reais e quadradas. Portanto, autovalores λ_i de $\boldsymbol{A}\boldsymbol{A}^T$ ou $\boldsymbol{A}^T\boldsymbol{A}$ são reais não negativos, iguais ou distintos.

VALORES SINGULARES DE UMA MATRIZ

Chamamos de **valores singulares** da matriz $\boldsymbol{A}$, $\lambda_1, \lambda_2, \ldots, \lambda_r$, as r raízes quadradas dos autovalores não nulos $\boldsymbol{A}\boldsymbol{A}^T$ ou $\boldsymbol{A}^T\boldsymbol{A}$;

$$\sigma_i = \sqrt{\lambda_i}$$

Os ***vetores singulares*** $\boldsymbol{u}_i$ à esquerda da matriz $\boldsymbol{A}$ são os autovetores de $\boldsymbol{A}\boldsymbol{A}^T, m \times m$.

Os ***vetores singulares*** $\boldsymbol{v}_i$ à direita da matriz $\boldsymbol{A}$ são os autovetores de $\boldsymbol{A}^T\boldsymbol{A}, n \times n$.

MATRIZES $\boldsymbol{U}$ E $\boldsymbol{V}$

Os autovetores de $\boldsymbol{AA}^T$ formam as colunas da matriz $\boldsymbol{U}$ e os autovetores de $\boldsymbol{A}^T\boldsymbol{A}$ formam as colunas da matriz $\boldsymbol{V}$. Como são simétricos, os autovetores podem ser definidos ortonormais, ié, perpendiculares entre si e de comprimento unitário, de forma que:

$$\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I} \qquad \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$$

$$\boldsymbol{U} = \begin{pmatrix} | & | & & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_m \\ | & | & & | \end{pmatrix}_{m \times m} \qquad \boldsymbol{V} = \begin{pmatrix} | & | & & | \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_n \\ | & | & & | \end{pmatrix}_{n \times n}$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m$$

POR EXEMPLO...

$$\boldsymbol{A} = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$$

```
A = np.array([[3, 2, 2],
              [2, 3, -2]])
ATA = A.T @ A
AAT = A @ A.T
```

$$\boldsymbol{A}\boldsymbol{A}^T = \begin{pmatrix} 17 & 8 \\ 8 & 17 \end{pmatrix} \quad \boldsymbol{A}^T\boldsymbol{A} = \begin{pmatrix} 13 & 12 & 2 \\ 12 & 13 & -2 \\ 2 & -2 & 8 \end{pmatrix}$$

$$AA^T = \begin{pmatrix} 17 & 8 \\ 8 & 17 \end{pmatrix}$$

PECE Programa de Educação Continuada
Escola Politécnica da USP

```
# Cálculo de autovalores e autovetores
lamu, u = linalg.eig(AAT)
print("AA^T=\n", np.round(AAT, 4))
print("autovalores=\n", np.round(lamu, 4))
print("autovetores=\n", np.round(u, 4))
```

$$\lambda_1 = 25 \rightarrow \sigma_1 = 5$$
$$\lambda_2 = 9 \rightarrow \sigma_2 = 3$$

$$\boldsymbol{U} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

```
AA^T=
 [[17  8]
 [ 8 17]]
autovalores=
 [25.  9.]
autovetores=
 [[ 0.7071 -0.7071]
 [ 0.7071  0.7071]]
```

$$\boldsymbol{A}^T\boldsymbol{A} = \begin{pmatrix} 13 & 12 & 2 \\ 12 & 13 & -2 \\ 2 & -2 & 8 \end{pmatrix}$$

```
# Cálculo de autovalores e autovetores
lamv, v = linalg.eig(ATA)
print("A^TA=\n", np.round(ATA, 4))
print("autovalores=\n", np.round(lamv, 4))
print("autovetores=\n", np.round(v, 4))
```

$$\lambda_1 = 25 \rightarrow \sigma_1 = 5$$
$$\lambda_2 = 9 \rightarrow \sigma_2 = 3$$
$$\lambda_3 = 0 \rightarrow \sigma_3 = 0$$

$$\boldsymbol{V} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{18} & 2/3 \\ 1/\sqrt{2} & -1/\sqrt{18} & -2/3 \\ 0 & 4/\sqrt{18} & -1/3 \end{bmatrix}$$

```
A^TA=
 [[13 12  2]
 [12 13 -2]
 [ 2 -2  8]]
autovalores=
 [25.  0.  9.]
autovetores=
 [[-0.7071 -0.6667  0.2357]
 [-0.7071  0.6667 -0.2357]
 [-0.      0.3333  0.9428]]
```

PECE Programa de Educação Continuada
Escola Politécnica da USP

SVD

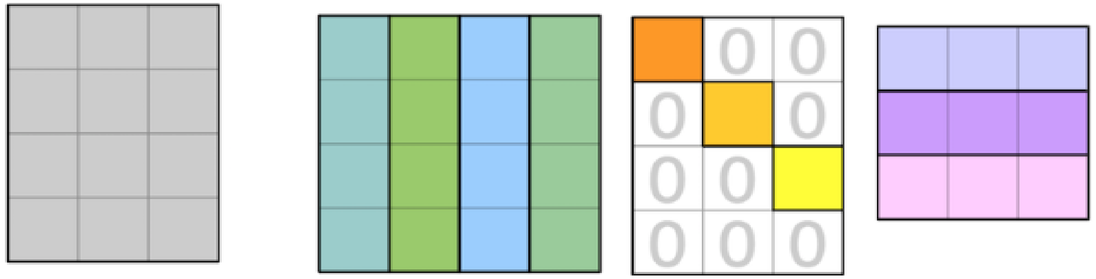
Definição

SVD

$$A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^T$$

$$\boldsymbol{A}_{m\times n} = \boldsymbol{U}_{m\times m}\boldsymbol{S}_{m\times n}\boldsymbol{V}_{n\times n}^T$$

$\boldsymbol{A}$ $\boldsymbol{U}$ $\boldsymbol{S}$ $\boldsymbol{V}^T$



$\boldsymbol{A}_{m\times n}$é uma matriz de valores reais

$\boldsymbol{U}_{m\times m}$ é uma matriz ortogonal

$\boldsymbol{S}_{m\times n}$ é uma matriz diagonal

$\boldsymbol{V}_{n\times n}^T$ é uma matriz ortogonal

DECOMPOSIÇÃO EM VALORES SINGULARES

O **SVD (Singular Value Decomposition)** é um método para fatorar matrizes:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \boldsymbol{U}\begin{bmatrix} \boldsymbol{D}_{r\times r} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}_{m\times n} \boldsymbol{V}^T$$

onde $\boldsymbol{U}$ e $\boldsymbol{V}$ são as matrizes ortogonais que definimos e $\boldsymbol{S}$ é diagonal.

Os valores da matriz diagonal $\boldsymbol{S}$ são os valores singulares σ_i que acabamos de aprender, e por isso a decomposição recebe este nome.

O número de valores singulares diferentes de zero é igual ao rank da matriz $\boldsymbol{A}$.

MATRIZ DIAGONAL $\boldsymbol{S}$

Se $\boldsymbol{A}$ tem r valores singulares não nulos, $\mathbf{S}$ é uma matriz da forma:

$\boldsymbol{D}_{r\times r}$

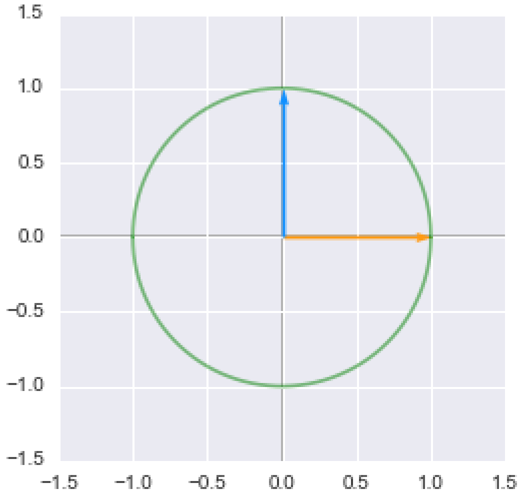
$$\boldsymbol{S} = \begin{pmatrix} \sigma_1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \sigma_r & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Em SVD , os elementos diagonais da matriz $\boldsymbol{S}$, ié, os valores singulares σ_i, são computados em ordem decrescente $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r$

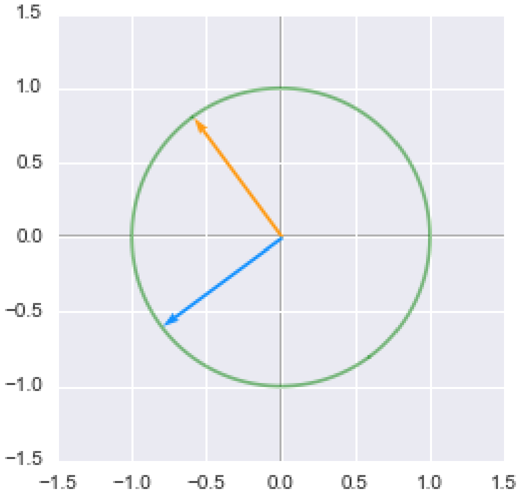
OU SEJA...

$$\boldsymbol{A} = \begin{pmatrix} 3 & 7 \\ 5 & 2 \end{pmatrix} = \boldsymbol{USV}^T = \begin{bmatrix} -0{,}8506 & -0{,}52573 \\ -0{,}52573 & 0{,}8506 \end{bmatrix} \begin{bmatrix} 8{,}7134 & 0 \\ 0 & 3{,}3282 \end{bmatrix} \begin{bmatrix} -0{,}5945 & -0{,}8040 \\ 0{,}8040 & -0{,}5945 \end{bmatrix}$$

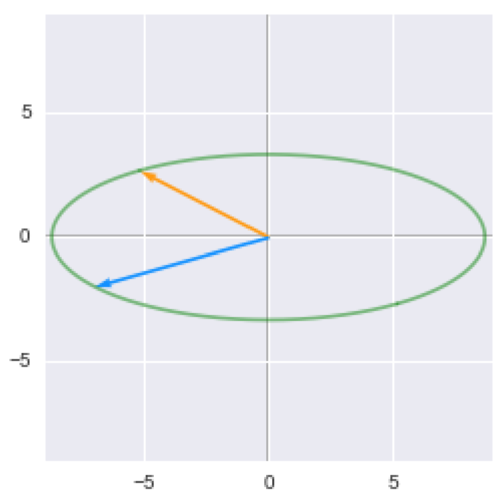
$$\boldsymbol{A} = \begin{pmatrix} 3 & 7 \\ 5 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



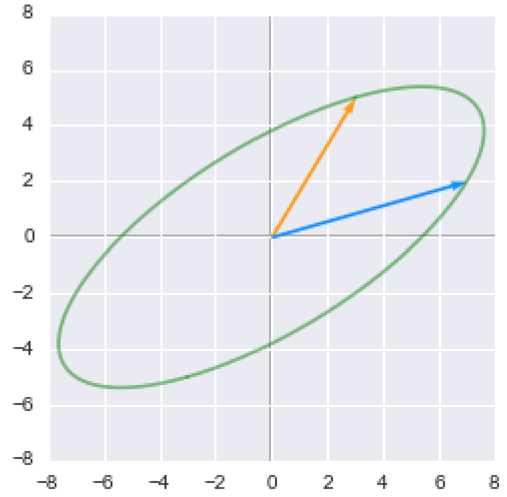
Círculo inicial



Rodado pela matriz $\boldsymbol{V}$



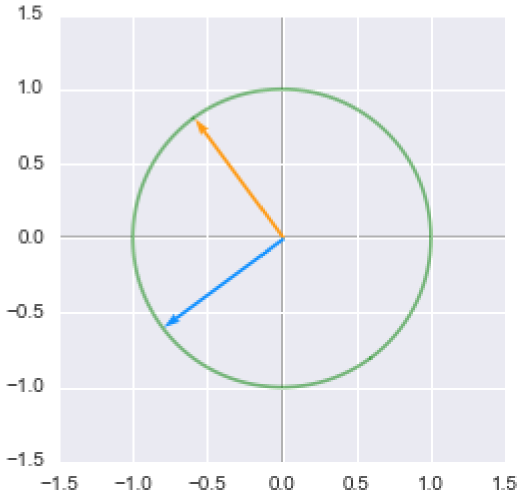
Escalonado pela matriz $\boldsymbol{S}$



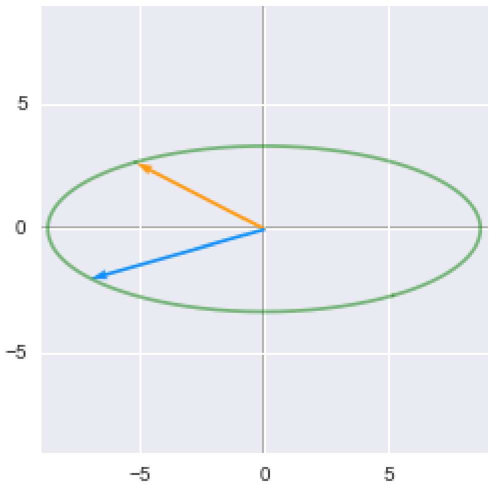
Rodado pela matriz $\boldsymbol{U}$

PECE Programa de Educação Continuada
Escola Politécnica da USP

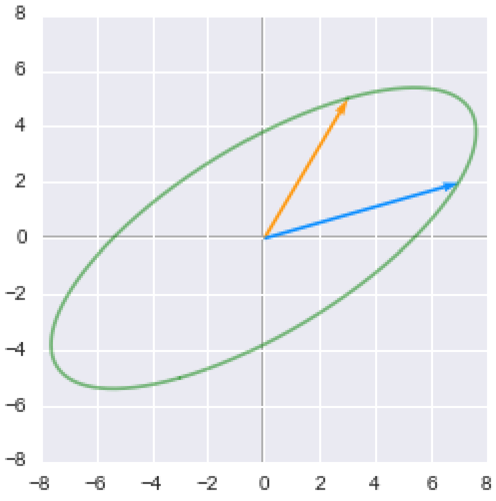
PORTANTO...



Rodado pela matriz $\boldsymbol{V}$



Escalonado pela matriz $\boldsymbol{S}$



Rodado pela matriz $\boldsymbol{U}$

$\boldsymbol{A}$ é uma matriz que pode ser vista como uma transformação linear. Essa transformação pode ser decomposta em três sub-transformações: rotação, escalonamento, rotação. Esses três passos correspondem às três matrizes: $\boldsymbol{U}$, $\boldsymbol{S}$ e $\boldsymbol{V}$.

RECUPERANDO A IDEIA DE DECOMPOSIÇÃO

$$\boldsymbol{A}_{m\times n} \qquad \boldsymbol{U}_{m\times m} \qquad \boldsymbol{S}_{m\times n} \qquad \boldsymbol{V}^T_{n\times n}$$

$$\begin{pmatrix} \boldsymbol{a}_{11} & \boldsymbol{a}_{12} & \cdots & \boldsymbol{a}_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{a}_{m1} & \boldsymbol{a}_{m2} & \cdots & \boldsymbol{a}_{mn} \end{pmatrix} = \begin{pmatrix} | & | & & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots & \boldsymbol{u}_m \\ | & | & & | \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \sigma_r & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_1^T & - \\ - & \boldsymbol{v}_2^T & - \\ - & \vdots & - \\ - & \boldsymbol{v}_n^T & - \end{pmatrix}$$

$$\boldsymbol{S}\boldsymbol{V}^T = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \sigma_r & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{v}_1^T \\ \vdots \\ \boldsymbol{v}_n^T \end{pmatrix} = \begin{pmatrix} \sigma_1 \boldsymbol{v}_1^T \\ \vdots \\ \sigma_r \boldsymbol{v}_r^T \\ \vdots \\ \boldsymbol{0} \end{pmatrix}$$

$$\boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = (\boldsymbol{u}_1 \quad \cdots \quad \boldsymbol{u}_m) \begin{pmatrix} \sigma_1 \boldsymbol{v}_1^T \\ \vdots \\ \sigma_r \boldsymbol{v}_r^T \\ \vdots \\ \boldsymbol{0} \end{pmatrix} = (\boldsymbol{u}_1)(\sigma_1 \boldsymbol{v}_1^T) + \cdots + (\boldsymbol{u}_r)(\sigma_r \boldsymbol{v}_r^T)$$

$$A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^T$$

INTERPRETAÇÃO

$$\boldsymbol{A} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^T$$

Os valores singulares são ordenados por ordem decrescente. Eles correspondem a um novo conjunto de características que são uma combinação linear dos características originais. A primeira característica explica a maior parte da variância dos dados.

VOLTEMOS AO NOSSO EXEMPLO...

$$\boldsymbol{A} = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$$

A matriz $\boldsymbol{A}$ é 2×3 e possui 2 autovalores não nulos $(r = 2)$.

$\boldsymbol{U}$ tem dimensão 2×2, $\boldsymbol{S}$ tem dimensão 2×3 e $\boldsymbol{V}^T$ tem dimensão 3×3,

$$\boldsymbol{U} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \quad \boldsymbol{S} = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \quad \boldsymbol{V}^T = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{bmatrix}$$

$$\boldsymbol{A} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T$$

$$\boldsymbol{A} = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$$

$$\boldsymbol{U} = \begin{array}{cc} \boldsymbol{u}_1 & \boldsymbol{u}_2 \\ \left[\begin{matrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{matrix}\right. & \left.\begin{matrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{matrix}\right] \end{array} \qquad \boldsymbol{S} = \begin{pmatrix} \overset{\sigma_1}{5} & 0 & 0 \\ 0 & \underset{}{\overset{\sigma_2}{3}} & 0 \end{pmatrix} \qquad \boldsymbol{V}^T = \begin{array}{c} \\ \boldsymbol{v}_1^T \\ \boldsymbol{v}_2^T \\ \\ \end{array} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{bmatrix}$$

$$\boldsymbol{A} = \underset{\sigma_1}{5} \underset{\boldsymbol{u}_1}{\begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}} \underset{\boldsymbol{v}_1^T}{\begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \end{pmatrix}} + \underset{\sigma_2}{3} \underset{\boldsymbol{u}_2}{\begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}} \underset{\boldsymbol{v}_2^T}{\begin{pmatrix} 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \end{pmatrix}}$$

VEJA O EXEMPLO DO NOTEBOOK

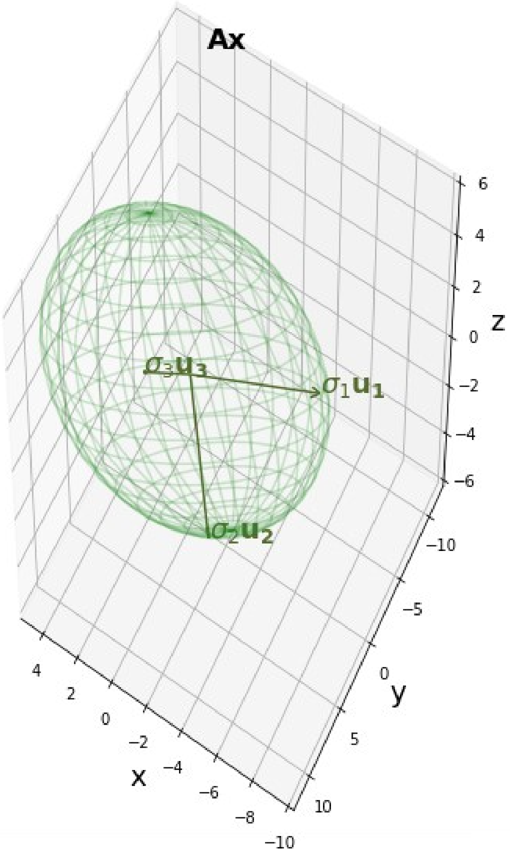
$$\boldsymbol{A} = \begin{bmatrix} 4 & 1 & 3 \\ 8 & 7 & -2 \\ 1 & 5 & 4 \end{bmatrix}$$

Os valores singulares são

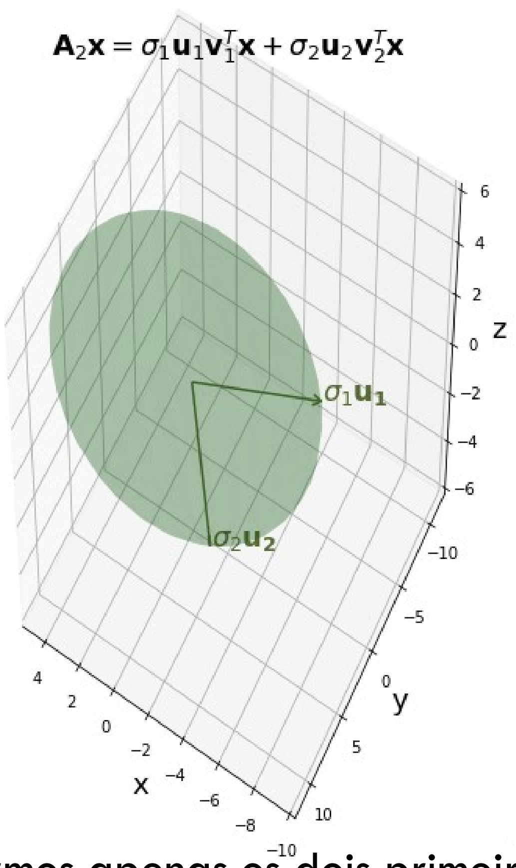
$\sigma_1 = 11{,}97$

$\sigma_2 = 5{,}57$

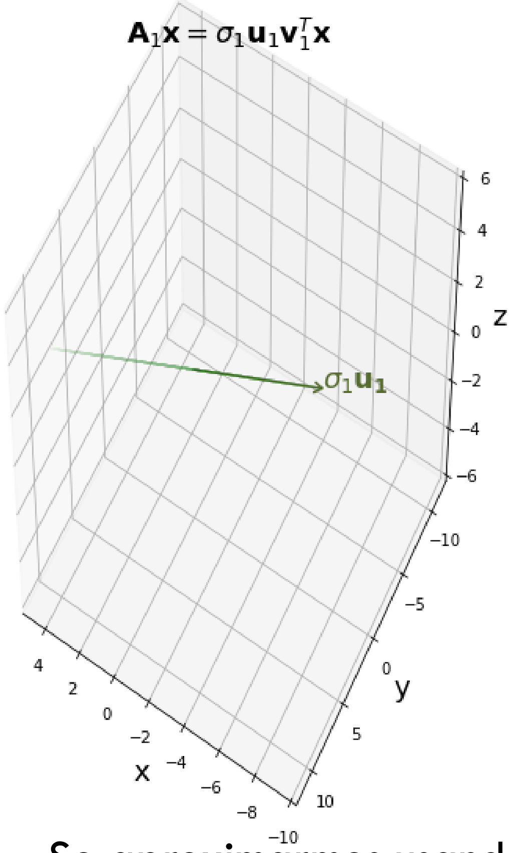
$\sigma_3 = 3{,}25$



O rank de $\boldsymbol{A_k}$ é 3. Então $\boldsymbol{Ax}$ é um elipsóide no espaço 3-d.



Se usarmos apenas os dois primeiros valores singulares, o rank de $\boldsymbol{A_k}$ será 2 e $\boldsymbol{A_k}$ multiplicado por $\boldsymbol{x}$ será um plano.



Se aproximarmos usando o primeiro valor singular, o rank de $\boldsymbol{A_k}$ será 1 e $\boldsymbol{A_k}$ multiplicado por $\boldsymbol{x}$ será uma linha

DIFERENÇAS ENTRE SVD E DECOMPOSIÇÃO EM AUTOVALORES

The singular value decomposition (SVD) provides another way to factorize a matrix, into singular vectors and singular values. The SVD allows us to discover some of the same kind of information as the eigendecomposition. However, the SVD is more generally applicable.
— Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning (Adaptive Computation and Machine Learning series)*, pags. 44-45, 2016.

- Nem todas as matrizes possuem uma decomposição em autovalores, porém toda matriz tem uma decomposição em valores singulares.

SVD: O TEOREMA FUNDAMENTAL DA ANÁLISE DE DADOS COM MÚLTIPLAS VARIÁVEIS

A decomposição em valor singular é uma técnica fundamental para análise de dados multivariados. Por exemplo, SVD é usado para cálculo de **Pseudo inversa.**

Um objetivo comum da análise de dados multivariada é reduzir a dimensão do problema, escolhendo um pequeno subespaço que captura propriedades importantes dos dados.

Dado que os elementos diagonais da matriz (chamados de valores singulares) no SVD são calculados em ordem decrescente, usa-se SVD para uma **aproximação por matriz de baixo rank,** i.é, redução de dimensão:

- **Análise de componentes principais**

INVERSA E DADOS REAIS

$$\boldsymbol{Ax = b}$$
$$\boldsymbol{x = A^{-1}b}$$

Mas nem todas as matrizes são inversíveis.

PSEUDOINVERSA MOORE-PENROSE

Pseudo inversa de $\boldsymbol{A}$:

$$\boldsymbol{A}^{+}$$

de forma que

$$\boldsymbol{x} \approx \boldsymbol{A}^{+}\boldsymbol{b}$$

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$$(\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}) \leftarrow svd(\boldsymbol{A})$$

$$\boldsymbol{A} = \boldsymbol{USV}^{\boldsymbol{T}}$$

$$\boldsymbol{A}^{-1} = \left(\boldsymbol{USV}^{\boldsymbol{T}}\right)^{-1}$$

$$\boldsymbol{A}^{-1} = \left(\boldsymbol{V}^{\boldsymbol{T}}\right)^{-1}\boldsymbol{S}^{-1}\boldsymbol{U}^{-1}$$

$$\boldsymbol{A}^{+} = \mathbf{V}\boldsymbol{S}^{+}\boldsymbol{U}^{T}$$

onde $\boldsymbol{S}^{+}$ é o recíproco $\frac{1}{x_i}$ dos elementos não nulos de $\boldsymbol{S}$

$$\begin{bmatrix} -3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}^{+} = \begin{bmatrix} -1/3 & 0 & 0 \\ 0 & -1/4 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

EXEMPLO

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$ (singular, inversa não existe)

$$A = USV^T = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$A^+ = VS^+U^T = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1/2 & 0 \\ 0 & 0 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

PECE Programa de Educação Continuada
Escola Politécnica da USP

APROXIMAÇÃO POR MATRIZ DE BAIXO RANK

Eliminação de ruídos,PCA, compactação de arquivos, etc...

PECE Programa de Educação Continuada

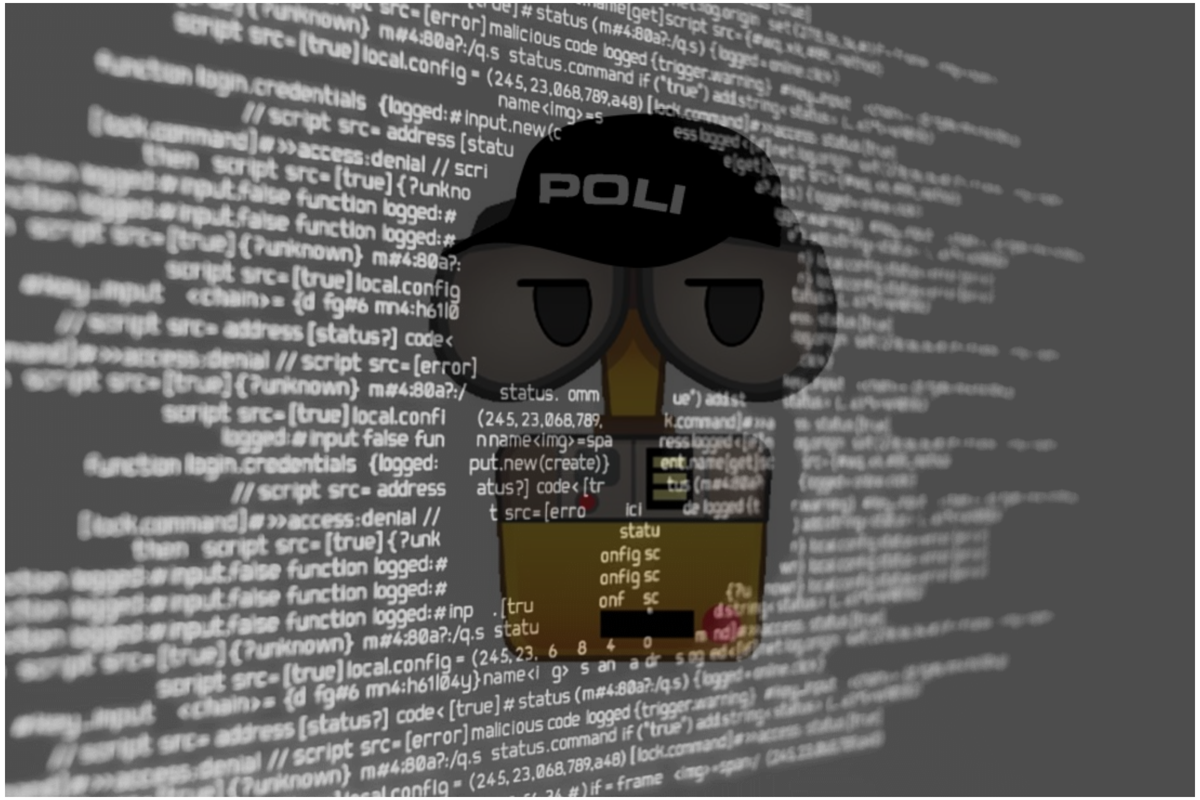
Escola Politécnica da USP

PCA

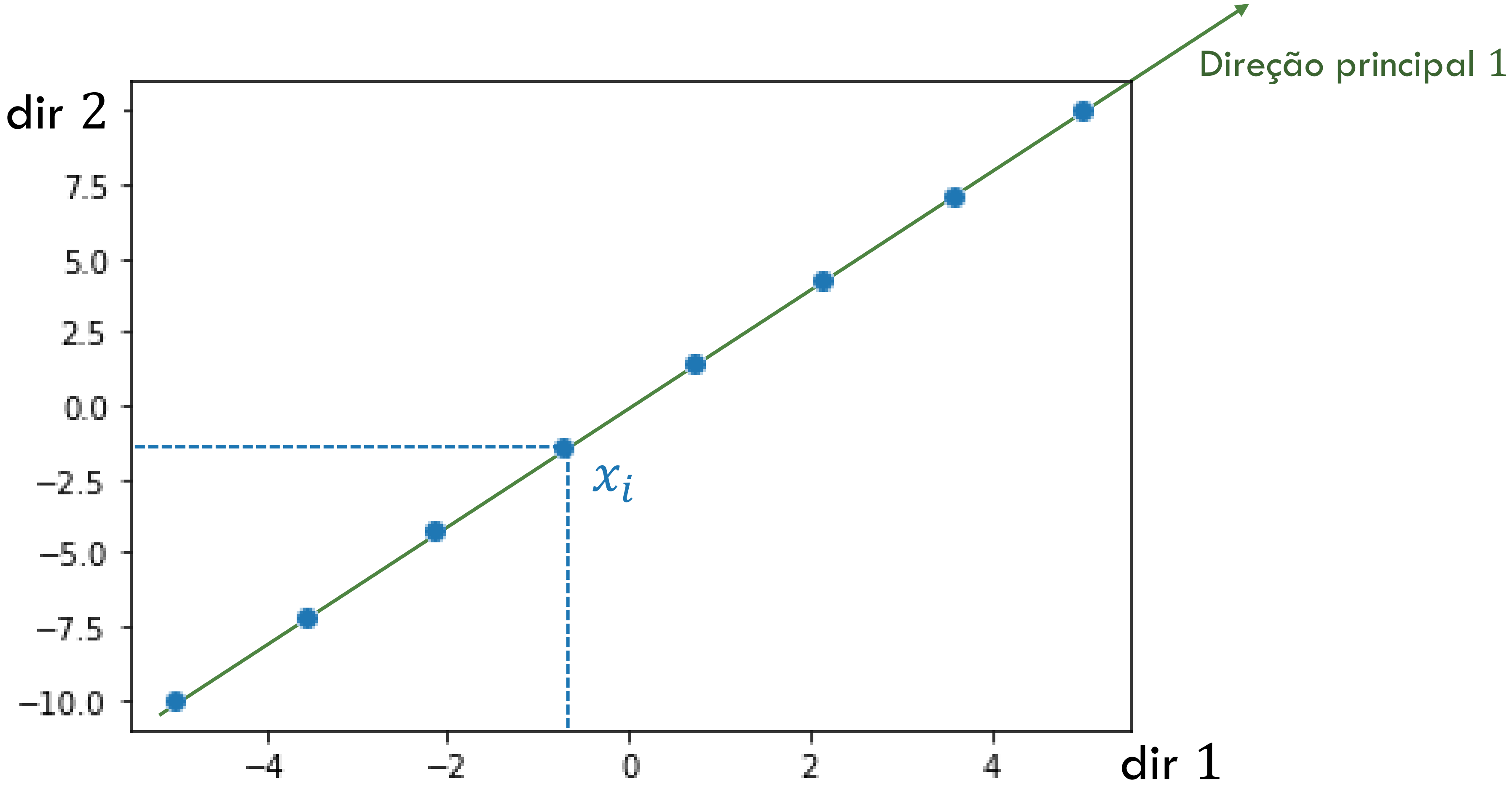
Qual a ideia?

ANÁLISE DE COMPONENTES PRINCIPAIS

A Análise de Componentes Principais ou PCA (Principal Component Analysis) tem como objetivo **encontrar um meio de condensar a informação contida em várias variáveis originais em um conjunto menor de variáveis estatísticas (componentes) com uma perda mínima de informação.**



Uma questão-chave na análise de dados é descobrir **quais variáveis são responsáveis pela maior parte da variação nos dados –** essas podem não ser as variáveis que você mediu, mas podem ser uma combinação linear das variáveis medidas!

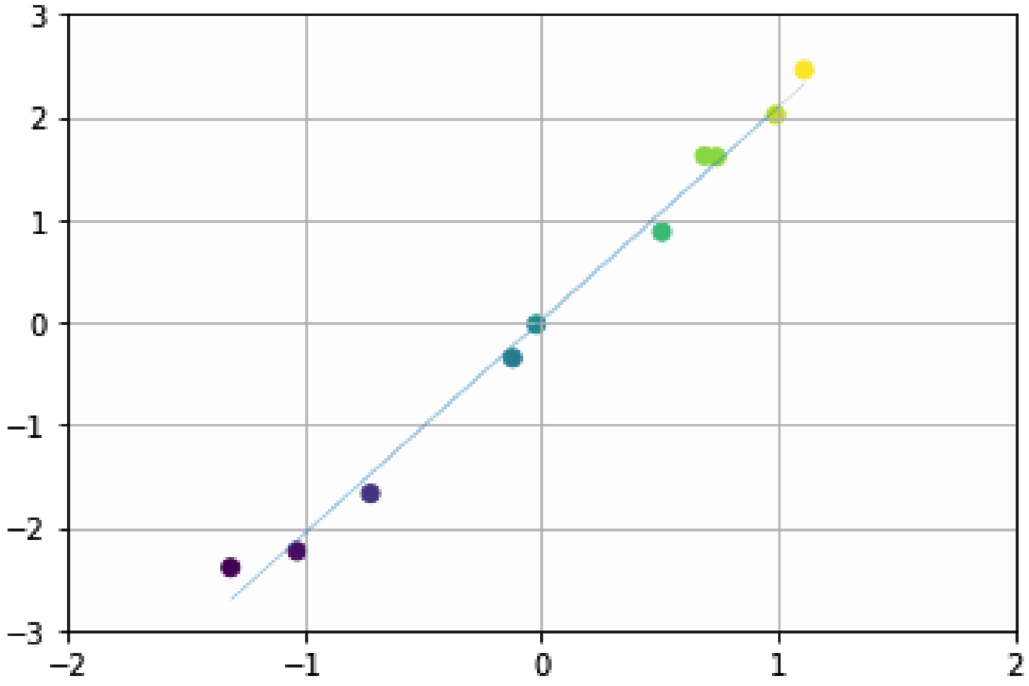


EXEMPLO

$$\boldsymbol{A} = \begin{bmatrix} -1{,}03 & 0{,}74 & -0{,}02 & 0{,}51 & -1{,}31 & 0{,}99 & 0{,}69 & -0{,}12 & -0{,}72 & 1{,}11 \\ -2{,}23 & 1{,}61 & -0{,}02 & 0{,}88 & -2{,}39 & 2{,}02 & 1{,}62 & -0{,}35 & -1{,}67 & 2{,}46 \end{bmatrix}$$

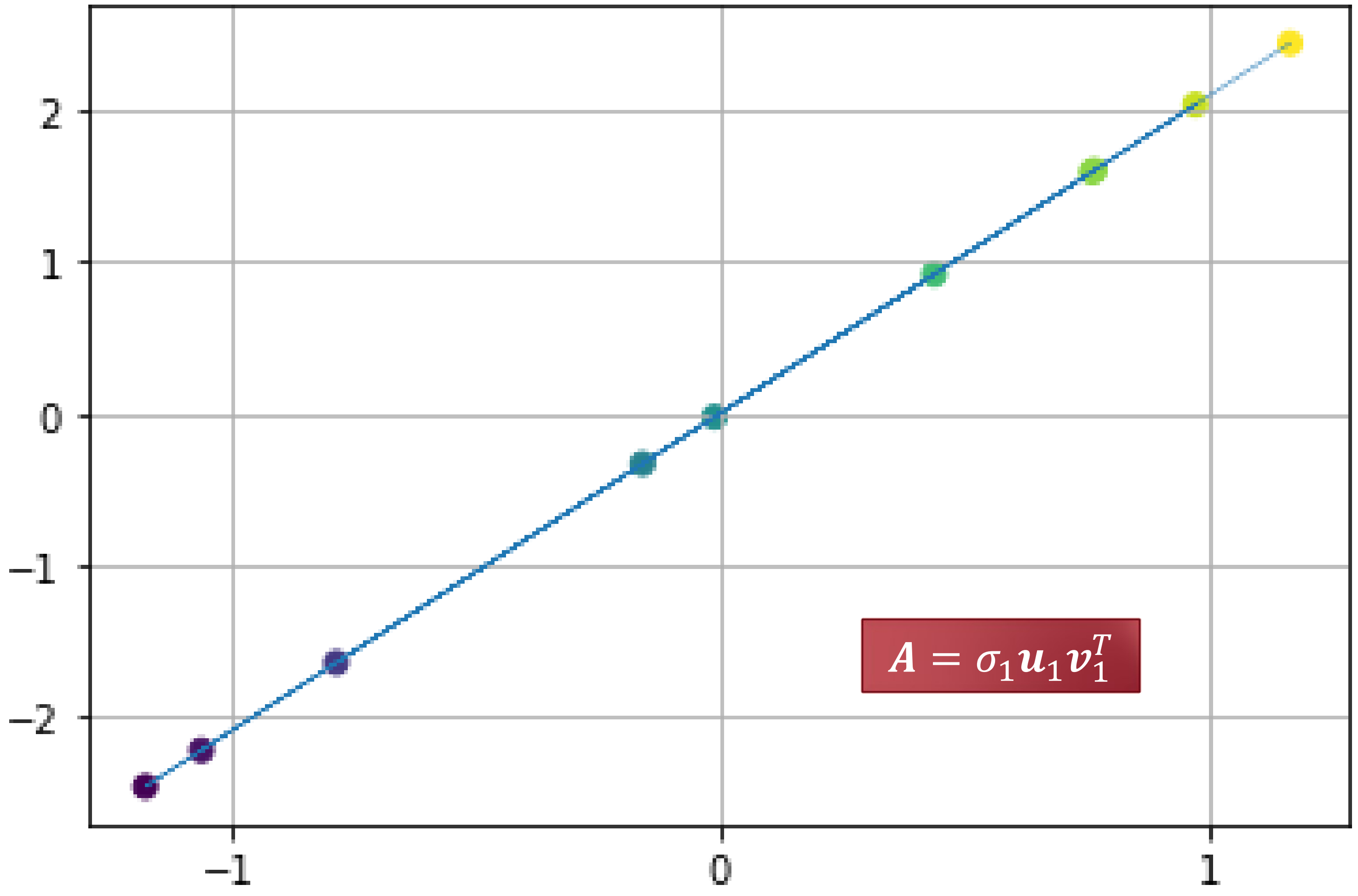
SVD:

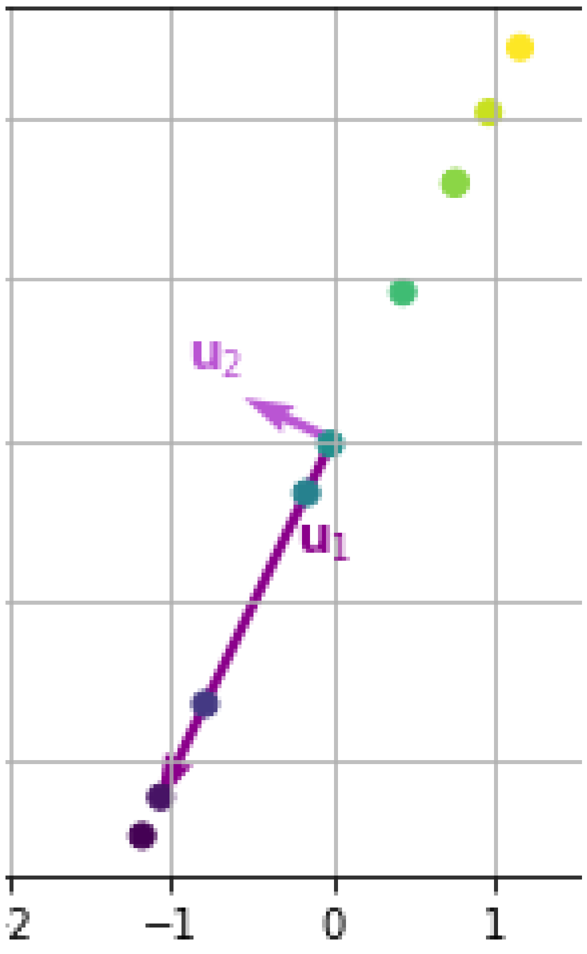
$$\boldsymbol{S} = \begin{bmatrix} 6{,}040 & 0 \\ 0 & 0{,}218 \end{bmatrix}$$



Com um valor singular muito maior que o outro, pode-se assumir que o pequeno valor de σ_2 é devido ao ruído nos dados e que esse valor singular seria idealmente zero.

Nesse caso, a matriz teria uma classificação, o que significa que todos os dados estão na linha definida por $\boldsymbol{u}_1$.





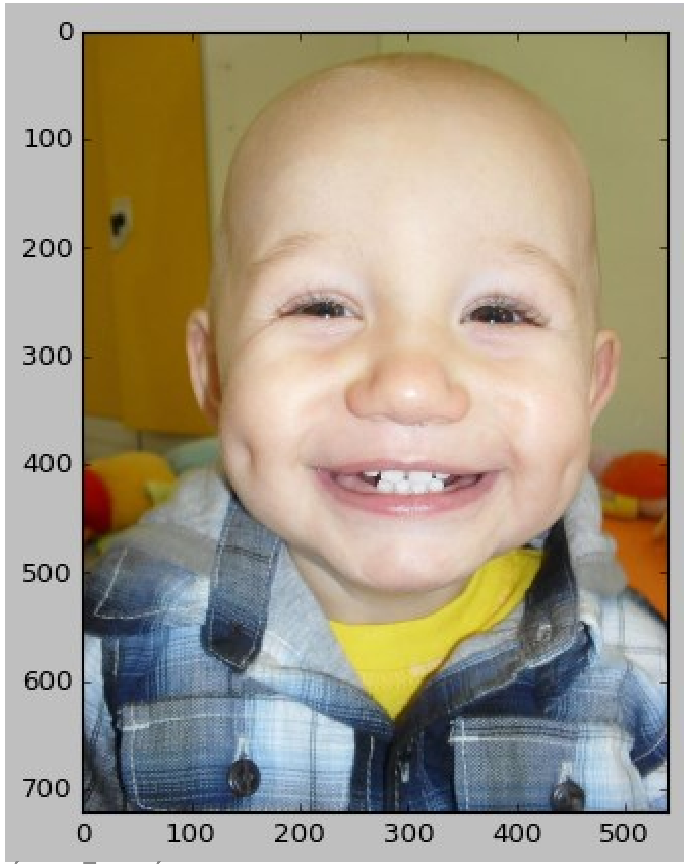
FAÇA VOCÊ

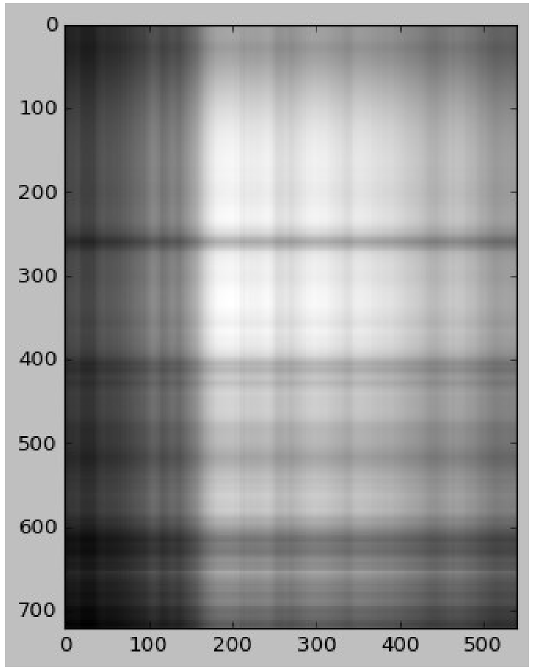
$$A = \begin{bmatrix} 2.9 & -1.5 & 0.1 & -1.0 & 2.1 & -4.0 & -2.0 & 2.2 & 0.2 & 2.0 & 1.5 & -2.5 \\ 4.0 & -0.9 & 0.0 & -1.0 & 3.0 & -5.0 & -3.5 & 2.6 & 1.0 & 3.5 & 1.0 & -4.7 \end{bmatrix}$$

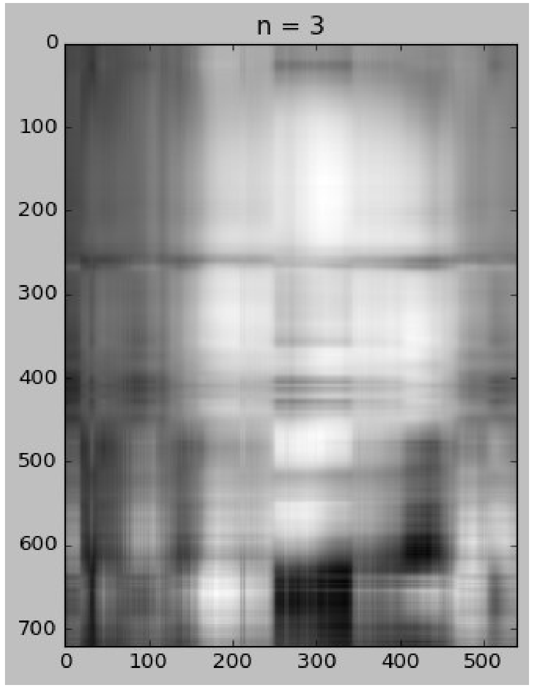
COMPACTAÇÃO DE IMAGEM (720, 540)

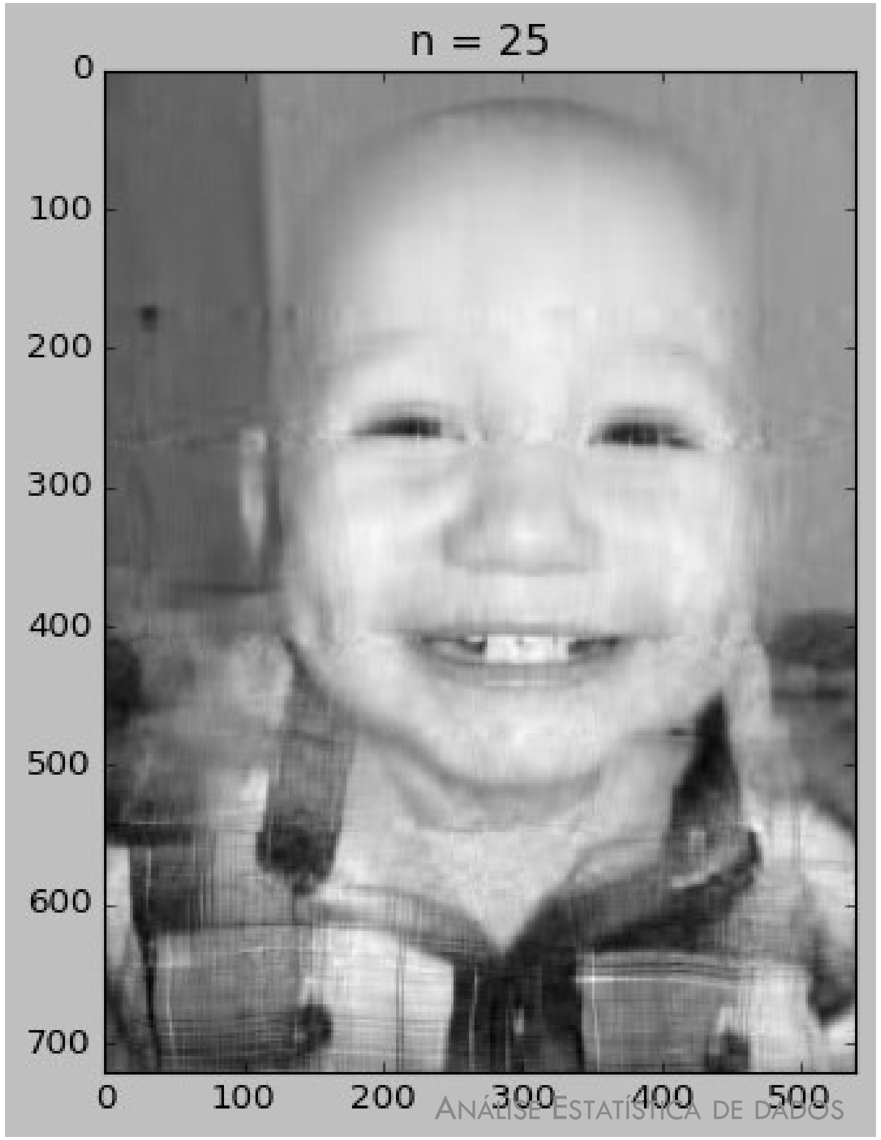
No Notebook, tente acompanhar a compactação da imagem feita utilizando SVD.

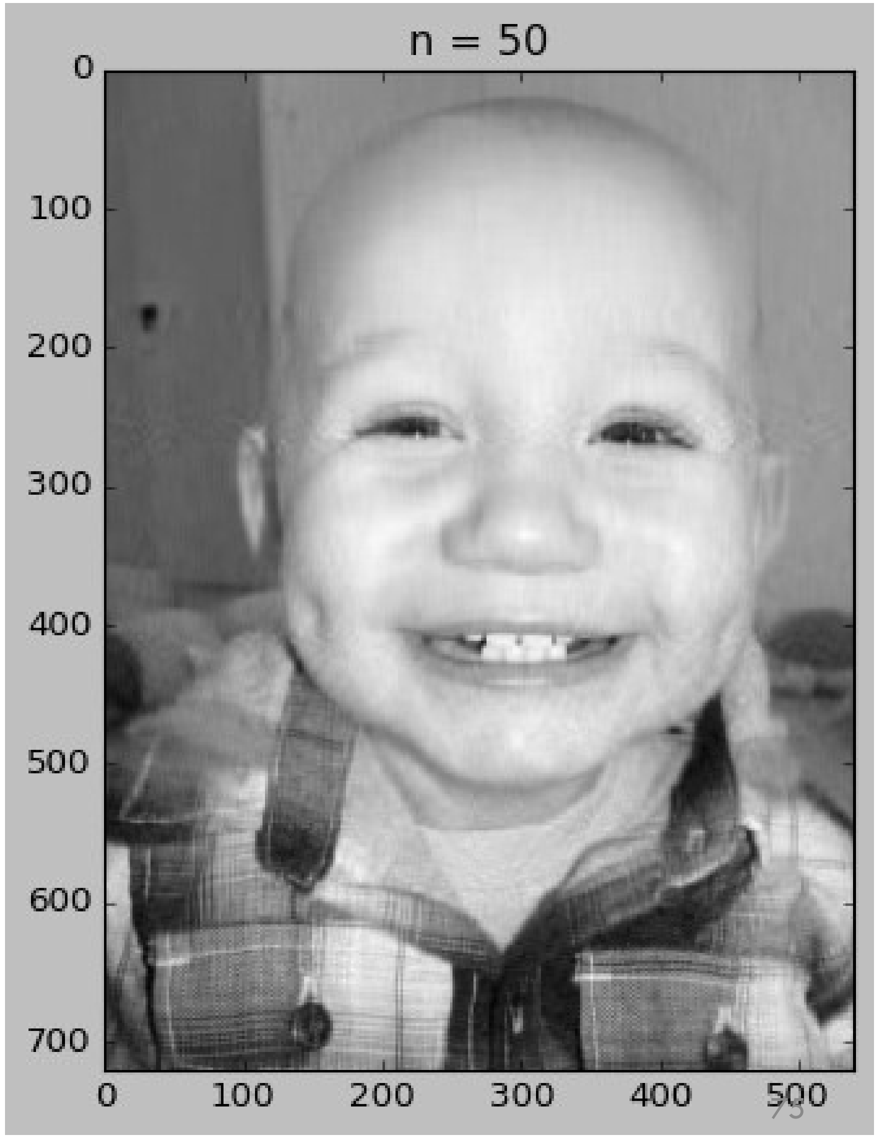
$$A = \sigma_1 u_1 v_1^T + \cdots + \sigma_r u_r v_r^T$$




0
100
200
300
400
500
600
700
0 100 200 300 400 500


n = 3
0
100
200
300
400
500
600
700
0 100 200 300 400 500

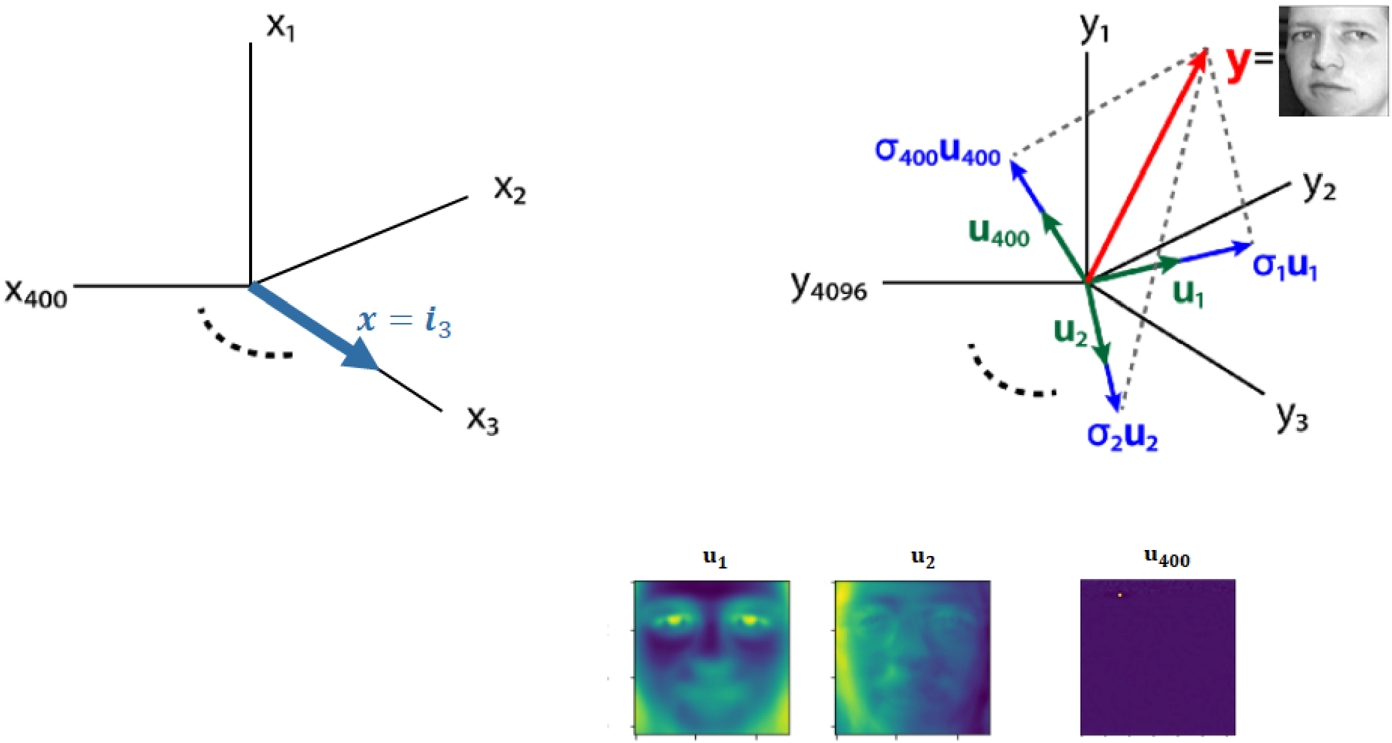

n = 25
0
100
200
300
400
500
600
700
0 100 200 300 400 500


n = 50
0
100
200
300
400
500
600
700
0 100 200 300 400 500

EIGENFACES

O termo *Eigenfaces* refere-se, basicamente, a um conjunto de características que definem a variação global entre as imagens faciais. O objetivo é **representar uma imagem que retrata o rosto de uma pessoa como uma combinação linear de um conjunto de imagens básicas chamadas de eigenfaces.**

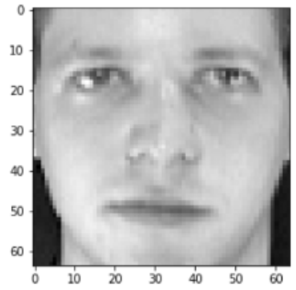
$$i_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad y = M i_3 \qquad y = \begin{bmatrix} 0.19835 \\ 0.19421 \\ \vdots \\ 0.75207 \\ 0.73967 \end{bmatrix}$$



EXEMPLO

Para ilustrar usaremos um dataset de imagens de rosto da biblioteca sklearn.

```
from sklearn.datasets import
fetch_olivetti_faces
data = fetch_olivetti_faces()
imgs = data.images
```



$$= \begin{bmatrix} 0.310 & 0.368 & \dots & 0.306 \\ 0.343 & 0.405 & \dots & 0.314 \\ \vdots & \vdots & \dots & \vdots \\ 0.202 & 0.207 & \dots & 0.157 \end{bmatrix} \rightarrow \mathbf{f}_1 = \begin{bmatrix} 0.310 \\ 0.368 \\ \vdots \\ 0.306 \\ 0.343 \\ 0.405 \\ \vdots \\ 0.314 \\ \vdots \\ 0.202 \\ 0.207 \\ \vdots \\ 0.157 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{f_1} & \mathbf{f_2} & \mathbf{f_3} & \dots & \mathbf{f_{400}} \end{bmatrix}$$

TEREMOS LIÇÃO DE CASA...

Antes de fazer a lição, estude a aula. Se precisar, veja os textos dos links:

1) https://towardsdatascience.com/understanding-singular-value-decomposition-and-its-application-in-data-science-388a54be95d

2) https://blogs.sas.com/content/iml/2017/08/30/svd-low-rank-approximation.html

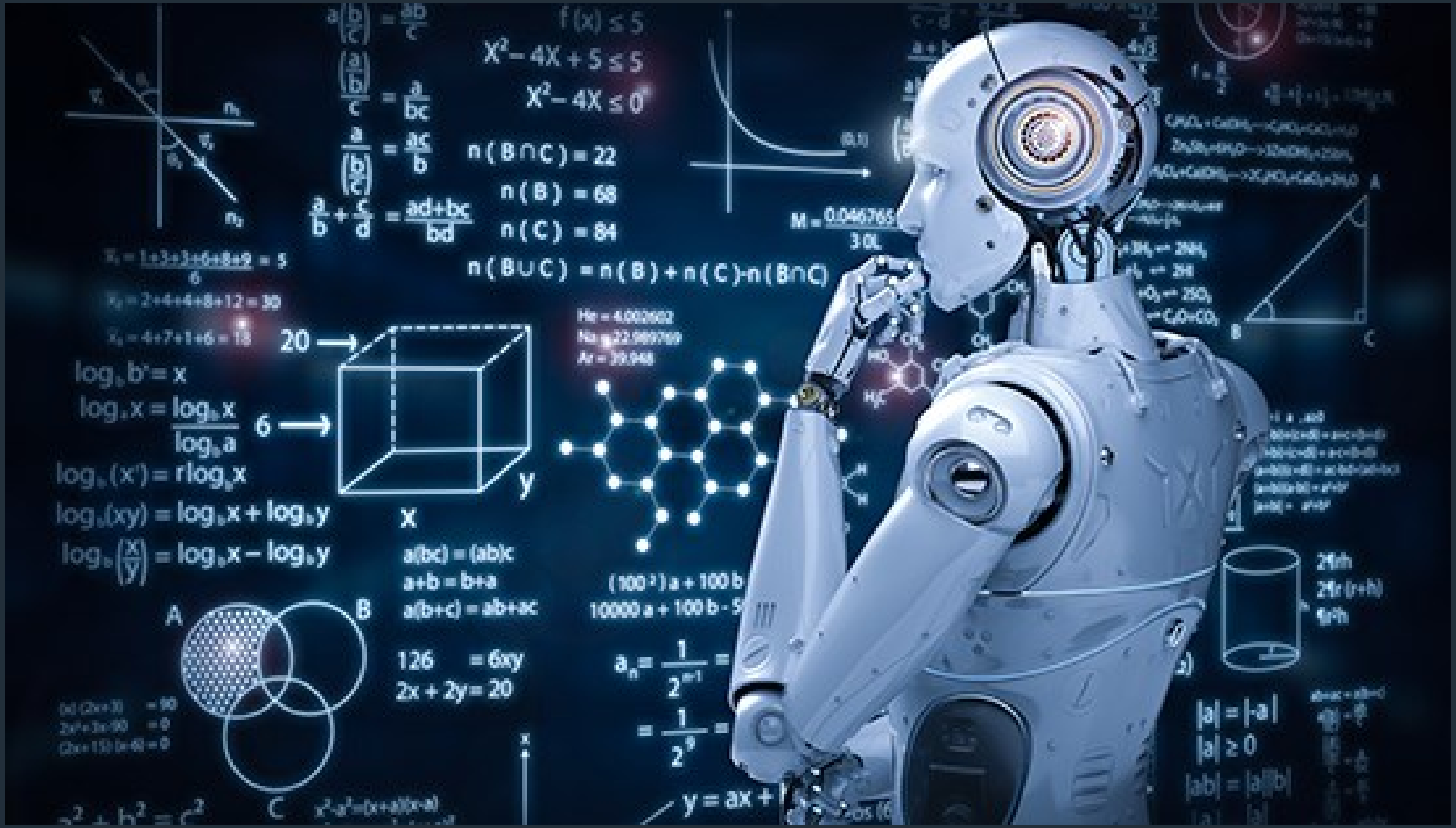
Se quiser, rode o programa sugerido em Python no segundo link (não é necessário, voltaremos a este tema futuramente).

PECE Programa de Educação Continuada
Escola Politécnica da USP

ENTREGA NO MOODLE ATÉ 24/03, 23:59

1. Com os dados de altura e peso de homens e mulheres, que você pode retirar do link* ,
   - Ache a matriz $\boldsymbol{A}$;
   - Ache os valores singulares;
   - Ache $\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}^T$;
   - Escreva a matriz na forma aproximada, $\boldsymbol{A}_k = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \cdots + \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^T$;
   - Discuta suas conclusões, analisando a matriz $\boldsymbol{A}_k$.
2. Use uma foto da sua face de dimensão 64 x64, e tente recuperá-la utilizando os valores singulares da matriz `M`. Discuta o resultado.

*https://www.kaggle.com/datasets/mustafaali96/weight-height



E AGORA...????

Revejam o material disponibilizado em aula.
NÃO PERCAM A PRÓXIMA AULA.
Até a próxima semana!!!