DESCOMPONER UNA MATRIZ EN 3 MATRICES

%matplotlib inline

import numpy as np

import matplotlib.pyplot as plt

A = np.array([[-1,3],[2,-2]])

vector = np.array([[2],[1]])

def graficarVectores(vecs, cols, alpha=1):

    #coordenadas

    plt.axvline(x=0, color="grey", zorder=0)

    plt.axhline(y=0, color="grey", zorder=0)

    for i in range(len(vecs)):

        x=np.concatenate([[0,0], vecs[i]])

        plt.quiver([x[0]],

                   [x[1]],

                   [x[2]],

                   [x[3]],

        angles = 'xy', scale\_units='xy',

        color = cols[i],

        scale=1, alpha = alpha )

def graficarMatriz(matriz, vectorCols=['red','blue']):

    #circulo unitario

    x = np.linspace(-1,1,100000)

    y = np.sqrt(1-(x\*\*2))

    #circulo unitario transformado

    x1 = matriz[0,0]\*x + matriz[0,1]\*y

    y1 = matriz[1,0]\*x + matriz[1,1]\*y

    x1\_neg = matriz[0,0]\*x - matriz[0,1]\*y

    y1\_neg = matriz[1,0]\*x - matriz[1,1]\*y

    #vectores

    u1 = [matriz[0,0], matriz[1,0]]

    v1 = [matriz[0,1], matriz[1,1]]

    graficarVectores([u1,v1], cols=[vectorCols[0], vectorCols[1]])

    plt.plot(x1,y1, 'green', alpha=0.7)

    plt.plot(x1\_neg, y1\_neg, 'green', alpha=0.7)

|  |  |
| --- | --- |
| A = np.array([[-1,3],[2,-2]])  vector = np.array([[2],[1]])  vector\_transformado = A.dot(vector)  graficarVectores([vector.flatten(),                    vector\_transformado.flatten()],                   cols = ['blue','orange'])  #transformado (2, 1) -> (1,2) - rotacion  #flatten pasa de 2,1 a 2, 1  plt.xlim(-0.5, 2.5)  plt.ylim(-0.5, 2.5) |  |

|  |  |
| --- | --- |
| # vector x autovalor = le da mas amplitud,  # misma direccin pero amplitud cambia  X = np.array([[3,2],[4,1]])  v = np.array([[1], [1]])  u = X.dot(v)  graficarVectores([u.flatten(), v.flatten()],                   cols=['orange','blue'])  plt.xlim(-0.5, 6)  plt.ylim(-0.5, 6) |  |

|  |  |
| --- | --- |
| #misma direccion, pero cambio de sentido  s= np.array([[-1],[2]])  t= X.dot(s)  graficarVectores([t.flatten(), s.flatten()],                   cols=['orange','blue'])  plt.xlim(-3,3)  plt.ylim(-3,3) |  |

|  |  |
| --- | --- |
| #circulo unitario  graficarMatriz(np.array([[1,0],[0,1]]))  plt.xlim(-1.5,1.5)  plt.ylim(-1.5,1.5) |  |

|  |  |
| --- | --- |
| A = np.array([[3,7], [5,2]])  graficarMatriz(A)  plt.xlim(-8,8)  plt.ylim(-8,8)  plt.show()  Una forma simple de visualizar el efecto que la aplicación de una matriz A de 2x2 tiene es:  Graficar el círculo unitario y el circulo unitario transformado. |  |

La transformación del espacio de una matriz A es igual a la transformación de las matrices SVD (Valores singulares)

**Efecto de la descomposición de una matriz en vectores singulares:**

V → rota el espacio  
D → Escala el espacio (agranda o encoje los vectores, también puede cambiar su sentido)  
U → rota nuevamente los vectores

**Nota:**  
La descomposición por valores singulares tiene efectos similares:

* autovectores → rota el espacio
* diag(autovalores) → escala el espacio
* inv(autovectores) → rota el espacio

Usar np.linalg.svd para descomponer una matriz por el método SVD nos devuelve 3 objetos U, D, V ¿Qué es D?

Un vector que contiene los valores singulares en orden descendente.

Al descomponer una matriz no cuadrada A por el método SVD obtenemos 3 matrices, U, D, V. Donde podemos interpretar a cada matriz como una transformación que debe ser aplicada en el siguiente orden: (2 de rotacion)

V -> Primera rotación D -> Escala U -> Segunda rotación

Cuando aplicamos la descomposición SVD a una imagen podemos:

Visualizar la imagen con menor calidad usando U[:, :1] , D[:1] y V[:1,:]

Un autovector de una matriz de 2x2 es aquel que cuando le aplico la matriz al autovector:

Su dirección no cambia.

Una forma simple de visualizar el efecto que la aplicación de una matriz A de 2x2 tiene es:

Graficar el círculo unitario y el circulo unitario transformado.

Una matriz A no cuadrada, ¿Cuándo se puede descomponer?

Siempre, podemos usar la descomposición en valores singulares (SVD).

A = np.array([[3,7],[5,2]])

U,V,D = np.linalg.svd(A)

print(U)

|  |  |
| --- | --- |
| graficarMatriz(A)  plt.xlim(-2,2)  plt.ylim(-2,2)  plt.show() |  |
| #PRIMERA ROTACION ROTACION V  graficarMatriz(V)  plt.xlim(-2,2)  plt.ylim(-2,2)  plt.show() |  |
| #ESCALA D  graficarMatriz(np.diag(D).dot(V)) |  |
| #SEGUNDA ROTACION U  graficarMatriz(U.dot(np.diag(D).dot(V)))  plt.xlim(-9,9)  plt.ylim(-9,9)  plt.show() |  |