Finding Eigenvalues of a 2×2 Matrix

1. Definition of Eigenvalues and Eigenvectors

Consider the linear transformation of n-dimensional vectors v defined by an $n \times n$ matrix A,

$$Av = w$$

or

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$$

λ is the **eigenvalues**, and v is the **eigenvectors** of the matrix A, if

$$Av = w = \lambda v$$

E.g. for a 2×2 matrix A,

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \tag{1}$$

The two sets of **eigenvalues** and **eigenvectors** of the matrix are:

$$\lambda_1 = -1, \lambda_2 = -2, v_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, v_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}, \tag{2}$$

2. Solution for finding the eigenvalues

Rewritten the function above,

$$Av - \lambda v = (A - \lambda I)v = 0$$
$$|A - \lambda I| = 0$$

E.g. for the given matrix A

$$|A - \lambda I| = |\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}| = |\begin{bmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{bmatrix}| = \lambda^2 + 3\lambda + 2 = 0 \tag{3}$$

According to the function above, we then get $\lambda_1 = -1$ and $\lambda_2 = -2$.

3. Solution for finding the eigenvectors

$$(A - \lambda I)v = \begin{bmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0 \tag{4}$$

For the first eigenvalue $\lambda_1 = -1$, we have

$$\begin{bmatrix} 1 & 1 \\ -2 & -2 \end{bmatrix} \begin{bmatrix} v_{1,1} \\ v_{1,2} \end{bmatrix} = \begin{bmatrix} v_{1,1} + v_{1,2} \\ -2v_{1,1} - 2v_{1,2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5}$$

Hence, the first eigenvector v_1 can be $[1 - 1]^T$.

Likewise, we can calculate the second eigenvector v_2 = $[1 - 2]^T$ for the second eigenvalue $\lambda_2 = -2$.

Exercise: Calculate the eigenvalues and eigenvectors using Numpy

Task:

1. Define the matrix A and convert it to a numpy array
2. Find the eigenvalues and eigenvectors via the function, (values, vectors) = np.linalg.eig(matrix)
3. Print the eigenvalues and eigenvectors one by one. (Format)
4. Plot the Arrows for the two eigenvectors using Matploblib.pyplot lib

*Tips: np.linalg.eig() can automatically calculate the results for you.

Print format:

'Eigenvalue: -1, EigenVector: [1.00, -1.00]'

'Eigenvalue: -2, EigenVector: [1.00, -2.00]'

Keep 2 decimal space

Plotting Arrow: import matplotlib.pyplot as plt plt.arrow(start_x, start_y, offset_x, offset_y, head_width, head_length, color)

Show Solution

In [1]:

```
import numpy as np

A = np.array([[0, 1],[-2, -3]])
e_val, e_vec = np.linalg.eig(A)

for i in range(len(A)):
    print('Eigenvalue:{:.2f}, EigenVector:[{:.2f}, {:.2f}]'.format(e_val[i], e_vec[0, i], e_vec[1, i]))
```

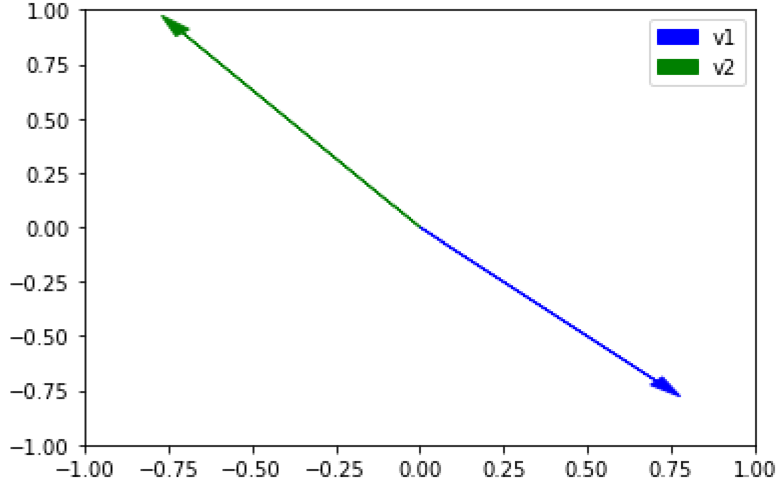
```
Eigenvalue:-1.00, EigenVector:[0.71, -0.71]
Eigenvalue:-2.00, EigenVector:[-0.45, 0.89]
```

In [2]:
```
import matplotlib.pyplot as plt

plt.axes()
plt.xlim([-1, 1])
plt.ylim([-1, 1])
arrow1 = plt.arrow(0, 0, e_vec[0, 0], e_vec[1, 0], head_width=0.05, head_length=0.1, color='blue')
arrow2 = plt.arrow(0, 0, e_vec[1, 0], e_vec[1, 1], head_width=0.05, head_length=0.1, color='green')

plt.legend([arrow1,arrow2], ['v1', 'v2'])
```

Out[2]:
```
<matplotlib.legend.Legend at 0x12123bb38>
```



In [ ]: