Eigendecomposition

November 2, 2020

0.0.1 EJEMPLO 1

Es necesario ocupar la función `eig()`.

[3]:
```python
# Eigendescomposición

from numpy import array
from numpy.linalg import eig
# Definiendo la matriz
A = array([
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
])
print(A)
# Factorizando
valores, vectores = eig(A)
print(valores)
print(vectores)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[ 1.61168440e+01 -1.11684397e+00 -9.75918483e-16]
[[-0.23197069 -0.78583024  0.40824829]
 [-0.52532209 -0.08675134 -0.81649658]
 [-0.8186735   0.61232756  0.40824829]]
```

0.0.2 EJEMPLO 2

[6]:
```python
# Confirmar el eigenvector

from numpy import array
from numpy.linalg import eig
# Definiendo la matriz
A = array([
    [1, 2, 3],
    [4, 5, 6],
```

```
    [7, 8, 9]
])
print(A)
# Factorizando
valores, vectores = eig(A)
# Confirmando el primer eigenvector
B = A.dot(vectores[:, 0])
print(B)
C = vectores[:, 0] * valores[0]
print(C)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[ -3.73863537  -8.46653421 -13.19443305]
[ -3.73863537  -8.46653421 -13.19443305]
```

0.0.3 EJEMPLO 3

[7]:
```
# Reconstruyendo la matriz

from numpy import diag, array
from numpy.linalg import inv, eig
# Definiendo la matriz
A = array([
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
])
print(A)
# Factorizando
valores, vectores = eig(A)
# Creando la matriz con los eigenvectores
Q = vectores
# Creando la inversa de la matriz de eigenvectores
R = inv(Q)
# Creando la matriz diagonal con los eigenvalores
L = diag(valores)
# Reconstruyendo la matriz original
B = Q.dot(L).dot(R)
print(B)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[1. 2. 3.]
```

```
 [4. 5. 6.]
 [7. 8. 9.]]
```