### EigenDecomposition

November 15, 2022

```python
import numpy as np
from numpy import linalg as la

# Let's start with a symmetric matrix.
S = [[1, 2], [2, 1]]

# Let's use NumPy to do the eigenvalue decomposition of S.
l, U = la.eig(S)

l, U

(array([ 3., -1.]), array([[ 0.70710678, -0.70710678],
                       [ 0.70710678,  0.70710678]]))

# Show the vector of eigenvalues. We can see that S is not positive
  # semi-definite, as it has a negative eigenvalue.
l
array([ 3., -1.])

# Show the matrix of eigenvectors, one per column.
U

array([[ 0.70710678, -0.70710678],
        [ 0.70710678,  0.70710678]])

# Compute S x u1, in order to show that S x u1 = lambda1 x u1.
S @ U[:,0]

array([2.12132034, 2.12132034])

# Compute lambda1 x u1, it should be equal to S x u1 above.
l[0] * U[:,0]

array([2.12132034, 2.12132034])
```
# Show that the norm of $u_1$ is equal to 1.
$U[:,0].T @ U[:,0]$

0.9999999999999999

# Show that $u_1$ is orthogonal to $u_2$.
$U[:,0].T @ U[:,1]$

0.0

# Compute $S @ U$, to check that $S @ U = \Lambda @ U$.
$S @ U$

array([[ 2.12132034,  0.70710678],
        [ 2.12132034, -0.70710678]])

# Compute $U @ \Lambda$, it should be equal with $S @ U$ above.
$U @ \text{np.diag}(l)$

array([[ 2.12132034,  0.70710678],
        [ 2.12132034, -0.70710678]])

# Create the Lambda matrix containing the eigenvalues on the diagonal.
np.diag(l)

array([[ 3.,  0.],
        [ 0., -1.]])

# This shows that the eigenvectors are orthonormal.
$U.T @ U$

array([[1., 0.],
        [0., 1.]])