EigenDecomposition

April 6, 2023

[2]:
```
import numpy as np
from numpy import linalg as la
```

[3]:
```
# 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)
```

[4]:
```
l, U
```

[4]:
```
(array([ 3., -1.]),
 array([[ 0.70710678, -0.70710678],
        [ 0.70710678,  0.70710678]]))
```

[6]:
```
# For Hermitian (symetric) matrices, Numpy has a more efficient version called␣
 ↪'eigh'
# which also returns the eigenvalues in sorted order.
l, U = la.eigh(S)
l, U
```

[6]:
```
(array([-1.,  3.]),
 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 u1 is equal to 1.
U[:,0].T @ U[:,0]
```

[ ]: `0.9999999999999999`

[ ]:
```
# Show that u1 is orthogonal to u2.
U[:,0].T @ U[:,1]
```

[ ]: `0.0`

[ ]:
```
# Compute S x U, to check that S x U = Lambda x U.
S @ U
```

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

[ ]:
```
# Compute U x Lambda, it should be equal with S x U above.
U @ 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.]])
```

[ ]:
```

```