Data Science - Assignment 3

Dora Dimitrova Martin Vestergaard

Spring 2021

The following tasks should be solved, individually or in groups of 2. Write the answers to the questions in a readme. When reviewing, check that you got the same answers.

Matrix fun

Solve the following using Python with numpy.

In numpy, there are some handy ways of working with matrixes (they are all explained later in this document):

```python
import numpy as np
from numpy.linalg import inv

# Creating matrices
A = np.array([[ 1, 2 ],[ 3, 4]])
B = np.array([[ 9, 8 ],[ 7, 6]])

# Transposing:
A.T # A transposed (danish: A transponeret)
B.T # B transposed

# Matrix multiplication:
A @ B

# Inverse:
inv(A)
```

Figure 1: Basic matrix functionality with numpy.

In the following, when talking about multiplication, we implicitly mean *matrix multiplication* (same as *dot product*).

Task 1

Given the two following matrices

$$A = \begin{pmatrix} 3 & 1 \\ 2 & 6 \end{pmatrix} \qquad B = \begin{pmatrix} -1 & 4 \\ 3 & 8 \end{pmatrix}$$

(a) Find A^T

(b) Find B^T

(c) Find AB (matrix multiplication). Compare with simple multiplication (using `*` instead of `@` in Python). Can you see what is the difference?

(d) Find AB^T

(e) Compare AB^T and $B^T A^T$

(f) Find $(A^T)^T$

(g) Find AA^T

Task 2

Given

$$A = \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

(a) Find AB

(b) Find BA

Confirm that they are different! Clearly, when doing matrix multiplication, order matters! $AB \neq BA$, so matrix multiplication is *not* commutative.

Task 3

The inverse of a matrix $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is found by

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \tag{1}$$

As seen in listing 1, the inverse of a matrix can be found easily with numpy (after having imported numpy.linalg.inv) by: `inv(A)`.

Using the same matrices from Task 2:

(a) Find A^{-1}

(b) Find B^{-1}

(c) Find AA^{-1}. Look closely at the result.

(d) Find $A^{-1}A$. Look closely at the result.

(e) Find BB^{-1}. Look closely at the result.

(f) Find $B^{-1}B$. Do you start to see a pattern?

It appears that a matrix multiplied with its inverse always gives $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Incidentally, a matrix with only ones in the diagonal is called an *identity matrix*, often denoted I.

Task 4

Given

$$A = \begin{pmatrix} 2 & 4 \\ 1 & 2 \end{pmatrix}$$

(a) Find A^{-1}

Oops. We see that not all matrices have an inverse! Looking at equation 1 (the equation for finding the inverse), can you figure out why? (hint: look at the denominator!)

Task 5

Plotting (lines, graphs, coordinates, etc) can be done using `matplotlib`. Try the following:

```
1  import numpy as np
2  from matplotlib import pyplot as plt
3  from numpy.linalg import inv
4
5  xs = np.array([0,0,3,3,0,1.5,3]) # List of x coordinates
6  ys = np.array([1,0,0,1,1,2 ,1]) # List of y coordinates
7
8  fig = plt.figure()
9  fig, ax = plt.subplots()
10
11 xs_ys = np.array([xs,ys])
12 ax.axis('equal')
13
14 # Plot the points
15 ax.plot( *xs_ys, '-', color='b')
16
17 # Create a rotation matrix
18 rot = np.array([[1, 0],[0, 1]]) # <-- CHANGE THIS
19
20 # turn the two lists (xs, ys) into a list of (x,y) tuples
21 points = np.array([[x,y] for x,y in zip(xs,ys)])
22
23 # Make the transformation:
24 points_rot = (points @ rot)
25
26 # Turn it into a row of xs and a row of ys:
27 xs_ys_rot = np.array([points_rot[:,0], points_rot[:,1]])
28
29 # Finally, plot it
30 ax.plot( *xs_ys_rot, '-', color='r')
31 fig
```

(a) Set the rotation matrix (line 18) to rotate the shape 45 degrees ($\frac{\pi}{4}$ radians).