MAD Assignment 3

Ask Jensen

12. december 2021

Indhold

1 Problem 1 **1**
1.1 (a) 1
1.2 (b) 2

2 Problem 2 **3**

3 Problem 3 **4**
3.1 (a) 4

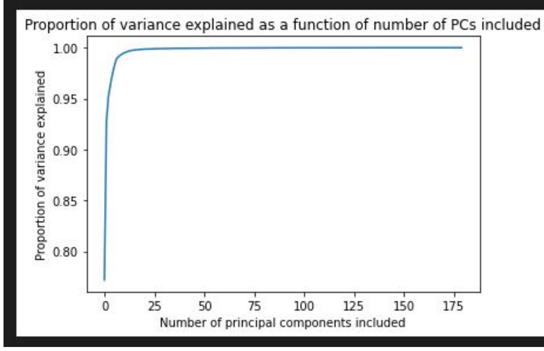
4 Problem 4 **4**
4.1 (a) 4

1 Problem 1

1.1 (a)

```python
import numpy.matlib

def pca(data):
    # Creating "clone" of matrix
    data_cent = np.full_like(data,0)

    # Iterate the matrix subtracting the mean diatiom
    #  from each row
    for i in range(780):
        data_cent[:,i] = diatoms[:,i] - mean_diatom

    # Create the covariance matrix
    cov_matrix = np.cov(data_cent)

    # Calculate the eigenvecotrs and eigenvalues
    PCevals, PCevecs = np.linalg.eigh(cov_matrix)


    # linalg.eigh returns the vectors and values
    # in the wrong order.
    # Np.flip will reverse the order so it is correct and
    # corrosponding to the exercise requirements
    PCevals = np.flip(PCevals)
    PCevecs = np.flip(PCevecs, axis=1)
    return PCevals, PCevecs, data_cent

PCevals, PCevecs, data_cent = pca(diatoms)
```

```
Proportion of variance explained by the first 1 principal components: 0.7718721493017527
Proportion of variance explained by the first 2 principal components: 0.9276996293043024
Proportion of variance explained by the first 3 principal components: 0.9521198453942007
Proportion of variance explained by the first 4 principal components: 0.9637878603999529
Proportion of variance explained by the first 5 principal components: 0.9739084497954094
Proportion of variance explained by the first 6 principal components: 0.98236065164916
Proportion of variance explained by the first 7 principal components: 0.9889975933245944
Proportion of variance explained by the first 8 principal components: 0.9910287023941854
Proportion of variance explained by the first 9 principal components: 0.9926692113360289
Proportion of variance explained by the first 10 principal components: 0.9939926229665051
```

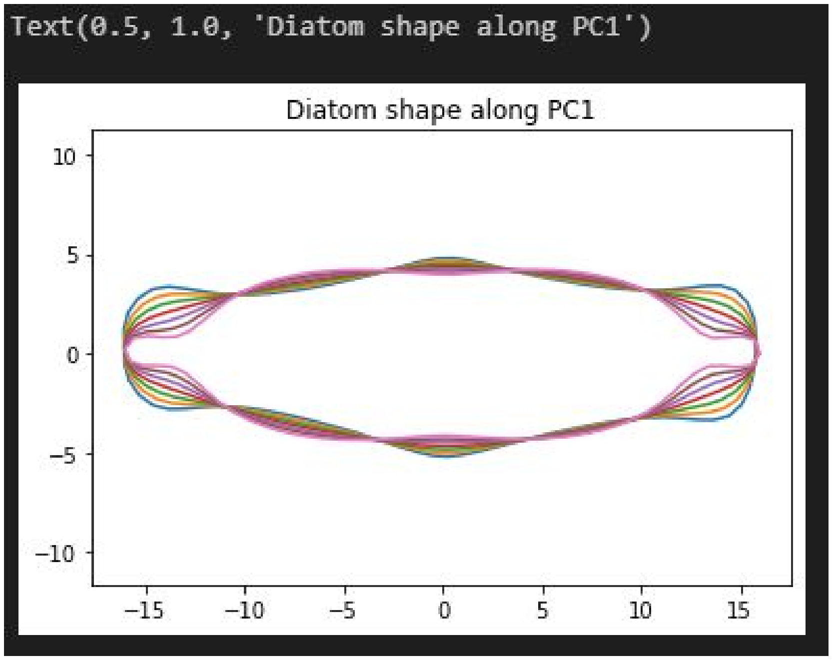


Figur 1: Values for the first 10 proportions of variance, and the corrosponding graph

1.2 (b)

```
    # gets the fourth eigenvector
    e4 = PCevecs[:, 3]
    # gets the fourth eigenvalue
    lambda4 = PCevals[3]
    # In case the naming std is confusing --
    # the eigenvalues have a statistical interpretation
    # print(std4)
    std4 = np.sqrt(lambda4)

    # Makes matrix filled with zeros
    diatoms_along_pc = np.zeros((7, 180))

    # Iterates the length of the matrix
    # For each row, add the mean diatom with added
    # values
    for i in range(7):
        diatoms_along_pc[i] = mean_diatom + ( e4 * std4 * (i-3))

    # Plotting each diatom
    for i in range(7):
```

```
        plot_diatom(diatoms_along_pc[i])

    plt.title('Diatom shape along PC1')
```



Figur 2: Plotted diatom

2 Problem 2

assesses the given claim $E[(X-\mu)^4] \geq \sigma^4$

X has the mean μ and the variance σ^4 which can be rewritten as $\sigma^4 = (Var(X))^2$
$E[(X-\mu)^4]$ kan be rewritten as $E(g(x))$ where $g(x) = (x-\mu)^4$
It is possible to Jensen's inequality if $g''(x) \geq 0$
$g(x) = (x-\mu)^4$
$g(x) = 4(x-\mu)^3$
$g(x) = 12(x-\mu)^2$
g(x) is convex, since the second derivative of the function is quadratic. Hence it will always be greater than zero. Which means that it its possible to make use of Jensen's inequality.
$E(cX-\mu)^4 \geq (E(X-\mu))^4$
$(E(X-\mu))^4$
$((E(X-\mu))^2)^2 = (Var(X))^2$

thus the claim is true, and shown by Jensen's inequality

$$E[(X-\mu)^4] \geq \sigma^4$$

3 Problem 3

3.1 (a)

4 Problem 4

4.1 (a)

In this exercise I'm asked to choose the null hypothesis.
The dataset is $X_i - Y_i$, which gives me

1	0.5	-0.5	1.5	0.5

Calculating the observed mean = $\frac{1+0.5-0.5+1.5+0.5}{5} = 0.6$
$\mu_0 = 0$ since the assumption is, that there is no difference between the two types of flowers
Thus my null hypothesis is: $H_0 : \mu_0 = 0$