HW7

CISC648010 - Spring 2022

Due Date: April 15th at 11 PM

1 PCA 5 pts each part

Assume that the covariance matrix of data points $x_1, x_2 \dots x_n \in \mathbb{R}^3$ is

$$S = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using PCA, we reduce the dimension to 2. Let $\theta_i \in \mathbb{R}^2$ be the vector corresponding to x_i after dimensionality reduction.

a) Calculate

$$\frac{1}{n} \sum_{i=1}^{n} \theta_i$$

The final answer should be a real-valued 2-dimensional vector.

b) In the lecture note, we found A such that $\theta_i = A^T(x_i - \bar{x})$. What is A in this example?

c) Assume $x_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$ and $\theta_1 = \begin{bmatrix} 2\sqrt{2} \\ -1 \end{bmatrix}$.

Find the inner product between $\bar{x}$ and vector $\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$.

($\bar{x}$ is the mean of $x_1, x_2 \dots x_n$).

2 PCA (coding problem) 10 pts each part

In this exercise you will apply PCA to a modified version of the Extended Yale Face Database B. The modified database is available in the file yalefaces.mat on Canvas. For a tour of the data, issue the following commands:

```
import scipy.io
import numpy as np
from matplotlib import pyplot as plt
import numpy as np
data = scipy.io.loadmat('yalefaces.mat')
data = data['yalefaces']
index = 10
plt.imshow(data[:,:,index])
```

Change the index to see other images. The dataset includes several different subjects (38 total) under a variety of lighting conditions.

a) By viewing each image as a vector in a high dimensional space, perform PCA on the full dataset. **Hand in** a plot the sorted eigenvalues of the sample covariance matrix. How many principal components are needed to represent 95% of the total variation? How about 99%? What is the percentage reduction in dimension in each case? Note that each image is a 48 by 42 matrix and you need to vectorize them (you should turn each image to a 2016 by 1 vector). The commands that may be helpful: np.reshape, np.linalg.eig, np.mean, np.diag.

b) **Hand in** a 4×5 array of subplots showing principal eigenvectors ('eigenfaces') 0 through 19 as images, treating the sample mean as the zeroth order principal eigenvector. Comment on what facial or lighting variations some of the different principal components are capturing. PLEASE NOTE: For uniformity, please do not standardize the features as described in the "preprocessing" section of the PCA notes. Also, to receive full credit, please submit your code in a file named hw6_prob2_UDID.

3 K-means (10pts each part)

a) download 'data1.csv' from the canvas. Implement k-means algorithm to divide the data points into **two clusters**. Please **hand in** a plot of the data points. Use two colours to show different clusters. Include the center of each cluster in you plot. Please use black and green colours for the centroids. In your report, please include the value of $W(\hat{c})$, where $\hat{c}$ is the clustering map that you have found using k-means.

In order to load the data, you can use the following code:

```
import csv
import numpy as np
import matplotlib.pyplot as plt
data1 = [ ]
with open('data1.csv') as csv_file:
  csv_reader = csv.reader(csv_file, delimiter=',', quoting=csv.QUOTE_NONNUMERIC)
  for row in csv_reader:
      data1.append(row)
data1 = np.array(data1)
plt.plot(data1[:,0],data1[:,1],'o')
```

b) Download the 'data2.csv' from the canvas and repeat part a for this dataset. Does k-means work well for this dataset? If not, how should we modify the datapoints to cluster them successfully using k-means.