Shiraz University

Statistical Pattern Recognition (Fall 2020)

Homework#4: Non-Parametric Density Estimation, and PCA

Due date: 30th January 2021

In order to do this homework, you have to go through density estimations and principal component analysis theories and concepts.

Part A: Non-Parametric Density Estimation

For this part of homework, you have to generate a dataset for three classes each with 500 samples from three Gaussian distribution described below:

$$\text{Class1:} \quad \mu = \begin{pmatrix} 2 \\ 5 \end{pmatrix} \quad \sum = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\text{Class2:} \quad \mu = \begin{pmatrix} 8 \\ 1 \end{pmatrix} \quad \sum = \begin{pmatrix} 3 & 1 \\ 1 & 3 \end{pmatrix}$$

$$\text{Class3:} \quad \mu = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \quad \sum = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

Use your generated data and estimate the density without pre-assuming a model for the distribution which is done by a non-parametric estimation.

- Implement the below PDF estimation methods using h=0.09,0.3,0.6 and answer the next questions for all of them:
    a) Histogram
    b) Parzen Window
    c) Gaussian kernel (Standard Deviations of 0.2,0.6,0.9)
    d) KNN Estimator (Fork=1,9,99)
- Estimate P(X) and Plot the true and estimated PDF and compare them for each of the separate model.
- Find the best value for h in the Gaussian kernel model with the standard deviation of 0.6 using 5-Fold cross-validation.
- Plot the bias and variance functions of estimated PDFs of class2, using Gaussian kernel with standard deviation of 0.2 and 0.6 and KNN estimator with k=1 and k=99.
- Employ the estimated Gaussian kernel for each class and do the followings with standard deviation 0.6:
    a) Divide the samples into a 90% train and 10% test data randomly.
    b) Use Bayesian estimation and predict the class labels while reporting train, test and total accuracies.

Part B: PCA

Dataset: jaffe

In this part, you will compute a PCA from a set of images of faces, each of size 128*128 in RGB format. The database provides facial images of 30 different people. For simplicity, convert the images into gray scale and resize them into 64*64. Attention that, eigenfaces are sets of eigenvectors which can be used to work with face recognition applications. Each eigenface, as we'll see in a bit, appears as an array vector of pixel intensities. We can use PCA to determine which eigenfaces contribute the largest amount of variance in our data, and to eliminate those that don't contribute much. This process lets us determine how many dimensions are necessary to recognize a face as 'familiar'

- Visualize the dataset.
- Preprocess and normalize the dataset.
- Implement the PCA function, then apply it on the dataset.
- Visualize the reduced dataset using 2D and 3D plots.
- Reconstruct the original data by using K principle components (Show reconstructed images of each individual for K=1,40,120).
- Plot the MSE between the original and reconstructed images in terms of number of eigenvectors.
- Visualize some of the first principal components.
- How many principle components are enough so that you have acceptable reconstruction? How do you select them?

Notes:

- **Pay extra attention to the due date. It will not extend.**
- **Be advised that submissions after the deadline would not grade.**
- **Your implementation should be functional.**
- **Prepare your full report in PDF format and include the figures and results.**
- **Do not use sklearn or any similar library and write your own code.**
- **The allowed programming languages are any language and feel free.**
- **Feel free for using sklearn in python for split train and test dataset.**
- **Feel free for using numpy, pandas, or any regular library in python.**
- **Submit your assignment using a zipped file with the name of "StdNum_FirstName_LastName.zip" to csehws.shirazu.ac.ir or for someone has problem with it to compuscien@gmail.com with SPR-Fall 2020-HW#4 subject.**