İrem Şahin

ENGR421

Mehmet Gönen

HW3-Report

For this project, I have written my code in a Jupyter Notebook .ipynb file. I have started my code by importing necessary libraries and writing our parameters down which were given in the pdf file. After this I have imported our data set and class labels from the given .csv files. I have continued by taking K(# of classes) and N(# of samples) values for future use. Then, as requested in the pdf file, I have divided the first 25 samples of each class for the training set, and the remaining 14 samples for the test set. In the end, I had a training set which is composed of 125 points(125x320) and a test set composed of 70 points(70x320). I have used normal list appending in these as using numpy is still new for me and I couldn't comprehend how to implement appending in numpy arrays. After using normal list appending, I just turned them into numpy arrays to use their transpose,shape etc. built-ins.

Next, I have estimated the class priors and prior class densities(pcd) to use in score function. After this, I have printed the estimated images for each class using seaborn.heatmap. Images looked like this:

Next, we needed to calculate each data point's score for each class and assign the data point to the class of their highest score. For this, I have defined my score function for my data points according to the function in the section 5.7. The method took the data set that will be

evaluated and the pcd as parameters, and returned a matrix with dimensions data_set_row x class_number, meaning 125x5 for the training set.

After this, I have printed the confusion matrix. First I took the train_predicted using argmax, then crosstabbed it with train_truth that was calculated at the beginning. The confusion matrix for the training set was like this:

train_truth	1	2	3	4	5
train_pred					
1	25	0	0	0	0
2	0	24	1	0	1
3	0	0	24	0	0
4	0	1	0	25	0
5	0	0	0	0	24

After the training set, I have done the same steps for the test set. By calling the score function with the test set and calculating the test_predicted, I have printed the following confusion matrix(which was the same with the one in the pdf).

test_truth	1	2	3	4	5
test_pred					
1	7	0	0	0	0
2	0	11	3	2	4
3	0	0	7	0	0
4	7	3	3	12	0
5	0	0	1	0	10