CMSC828C: Statistical Pattern Recognition

Project-1

Umang Rastogi UID: 116773232

Date: November 18 2020

1. Problem Statement

The project is essentially divided into 2 major tasks: identification of each of the subjects in the given data sets and classification of neutral vs facial expression.
These tasks were done using 4 classifiers, namely, Bayes, k-Nearest Neighbor, Kernel SVM, and Boosted SVM classifier. Additionally, principal component analysis (PCA) and multiple discriminant analysis (MDA) had to be employed on the data before feeding it to the classifiers. Further explanation of the employment of the methods and classifiers has been described in the section below.

2. Pipeline

- First, we perform pre-processing on the required data set where it is reshaped to shift all the feature data into a single column. Note that this is not done for *Illumination.mat* data set. The data set is also split into training and testing data sets.
- Labels for training and testing data sets are generated as well during pre-processing.
- We preform the user selected analysis on the training data such as PCA and MDA. The user can select to perform no analysis as well.
- Training data is fed to the user selected classifier for learning.
- Finally, accuracy of the classifier is measured by employing the classifier on the testing data.

3. Experiments and Results

3.1 Task 1: Identification of each of the subjects in the given data sets

For task-1, the *Illumination.mat* and *Pose.mat* data sets have 2 testing samples per subject. In the *Data.mat* data set, 1 testing sample has been taken for each subject. The rest of the samples are used for training.

For this task, the change in the accuracy of the Bayes and k-NN classifiers was studied when PCA, MDA, or no analysis is applied to the training data before feeding it to the classifiers. Figure 1 shows the accuracy of all the classifiers on various data sets under varying analysis methods such as PCA and MDA. Note that I have taken 3 nearest neighbors for my analysis of the k-NN classifier.

Accuracy (%)				
Classifiers	**Dataset**	**Analysis Type**		
		Normal	PCA	MDA
Bayesian	Pose	77.94	86.76	52.94
	Illumination	100	100	100
	Data	16	16	17.5
k-Nearest Neighbor (k=3)	Pose	35.29	30.88	41.17
	Illumination	100	99.26	98.52
	Data	0.5	0.5	0.5

Figure 1: Accuracy of the Bayes and k-NN classifier under various analysis methods

3.2 Task 2: Classification of neutral vs facial expression

Here, *Data.mat* was used as the data set. A split ratio of 0.25:0.75 (test-to-train) was kept. Each run sources random samples for the training and testing data sets while maintaining the split ratio.

3.2.1 Test the impact of samples included for training

- The classifiers used for the experiment were Bayes classifier, PCA + Bayes classifier, and MDA + Bayes classifier.
- Each of the classifiers were ran 50 times on randomized training and testing data set. Refer figures 2, 3, and 4 for accuracy of the various classifiers mentioned above over the iterations.

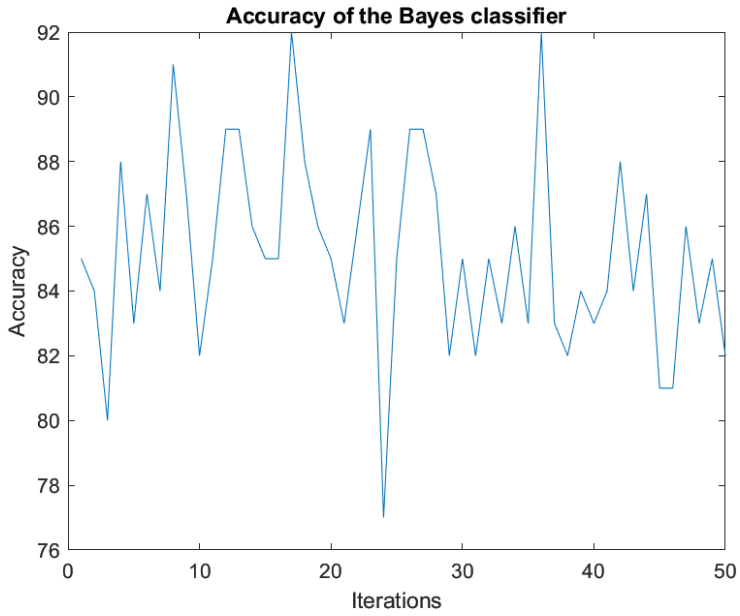


Figure 2: Accuracy of the Bayes classifier over various iterations of randomized data

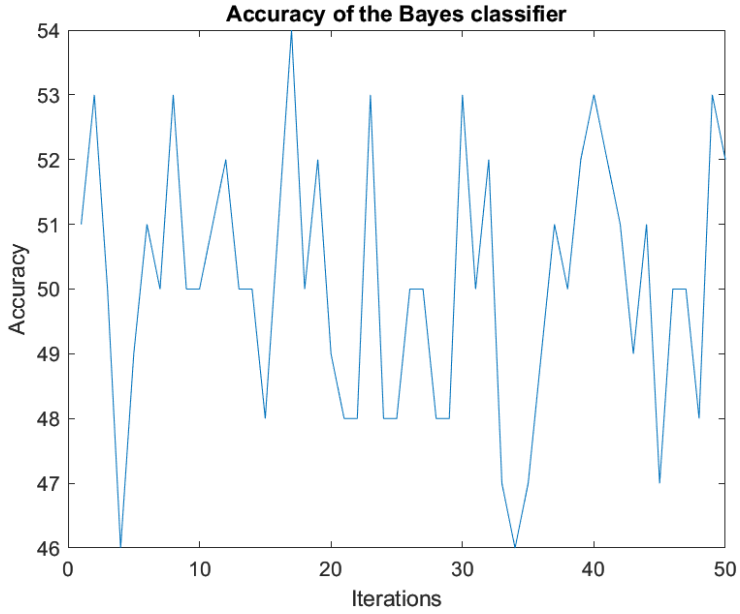


Figure 3: Accuracy of the PCA + Bayes classifier over various iterations of randomized data

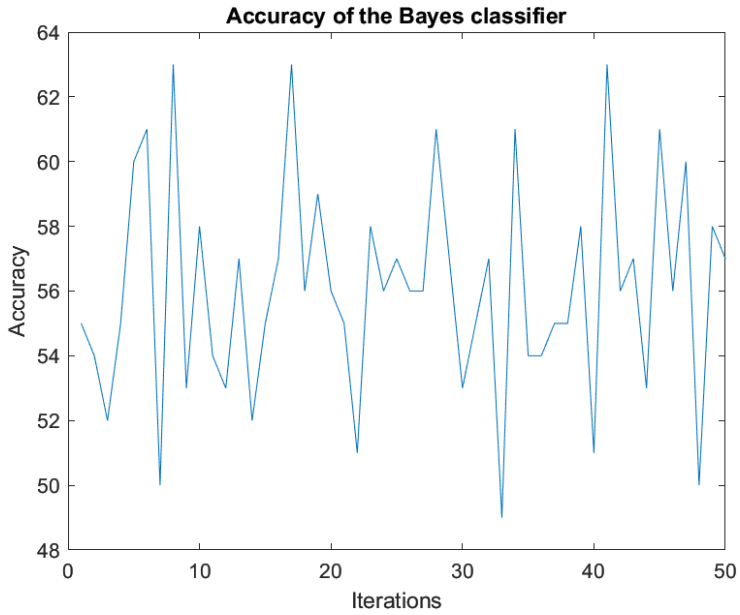


Figure 4: Accuracy of the MDA + Bayes classifier over various iterations of randomized data

3.2.2 Test the impact of number of nearest neighbors for kNN

- The classifiers used for the experiment were kNN classifier, PCA + kNN classifier, and MDA + kNN classifier.
- Each of the classifiers were ran 50 times on a set of training and testing data by varying the number of nearest neighbors from 1 to 50. Refer figures 5 and 6 for accuracy of the various classifiers mentioned above over the iterations.

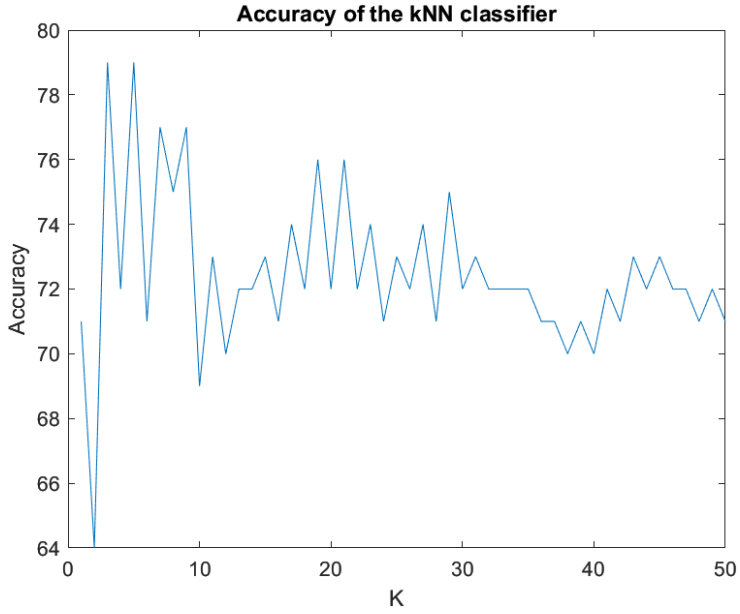


Figure 5: Accuracy of the k-NN classifier over various iterations of randomized data

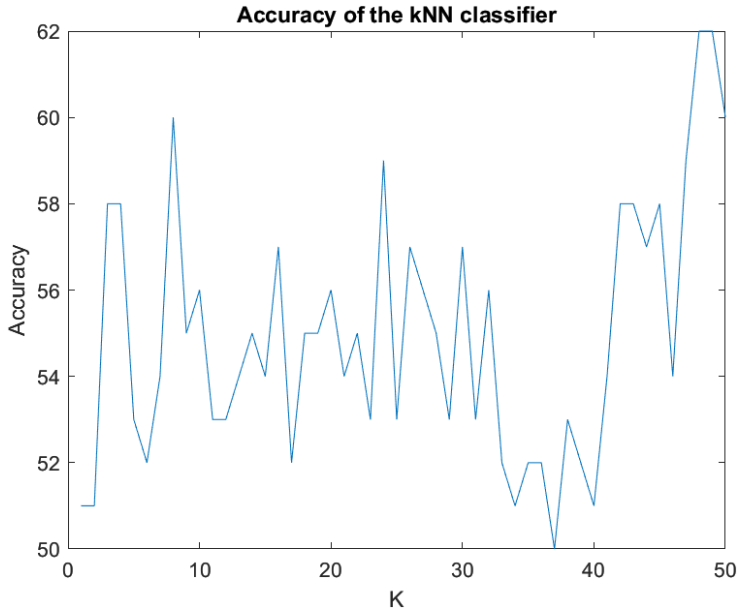


Figure 6: Accuracy of the PCA + k-NN classifier over various iterations of randomized data

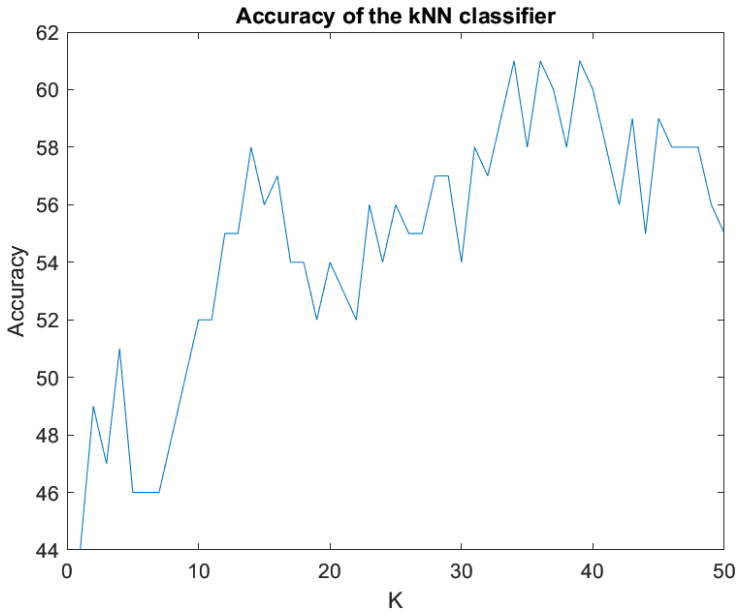
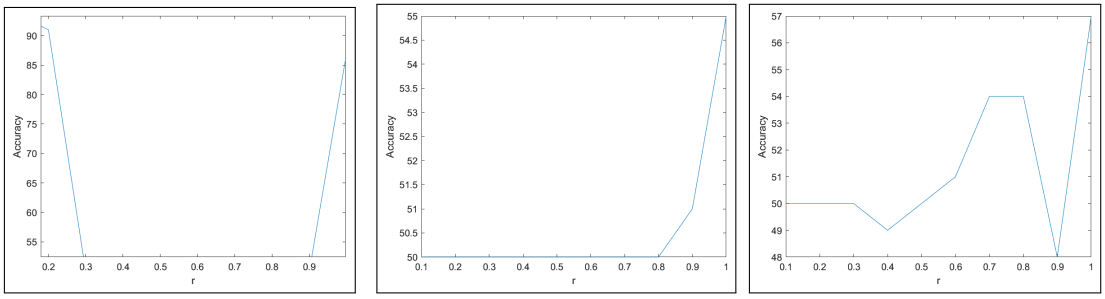


Figure 7: Accuracy of the MDA + k-NN classifier over various iterations of randomized data

In general, it is easy to notice that classifiers run better without PCA or MDA. I believe this is happening because the analysis methods are causing important features to disappear for proper classification.

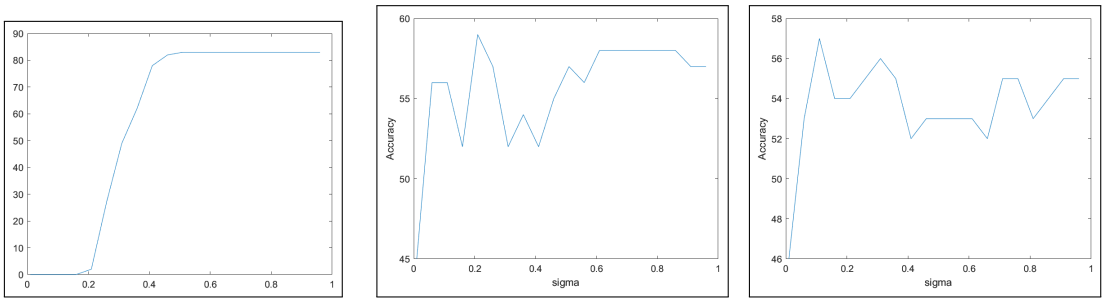
3.2.3 Test the impact of r and σ in Kernel SVM

- An SVM classifier with 2 kernels: RBF and Polynomial kernels, was used for the experiment.
- Below we can see the impact of varying r and σ on the accuracy of the Kernel SVM classifier in figures 8 and 9 respectively. Note that in these figures sub-labels a, b, and c represent normal, PCA, and MDA analysis on the kernel SVM models.
- While getting the plots in figure 8, r was varied from 0.1 to 1 and σ was kept constant at 0.01.
- While getting the plots in figure 9, σ was varied from 0.01 to 1 and r was kept constant at 0.1.
- We can see the best accuracy is at $r = \sigma = 1$ when we compare all of the classifiers.



(a) (b) (c)

Figure 8: Accuracy of SVM classifier using polynomial Kernel



(a) (b) (c)

Figure 9: Accuracy of SVM classifier using RBF Kernel

3.2.4 Performance of the Boosted SVM classifier

- In this experiment, a linear SVM classifier's performance was tried to be boosted by applying the Adaboost algorithm on it.
- Figure 10 shows the accuracy of the classifier under normal and after using PCA and MDA on the training data set.
- The adaboost algorithm was ran for 50 iterations. However, no improvement was seen during the iterations which was shocking.

Accuracy (%)			
Classifier	**Analysis Type**		
	Normal	PCA	MDA
Boosted SVM	90	50	50

Figure 10: Accuracy of the Boosted SVM classifier over various analysis methods