ENEE 633 Statistical Pattern Recognition

Project 1: Face Recognition

Pruthvi Sanghavi (116951005)

November 21, 2020

Introduction

The present project deals with the task of designing classifier for face recognition. In this project, various classifiers such as Bayesian, K Nearest Neighbor, Kernel SVM (Support Vector Machines) and Boosted SVM (Support Vector Machines) are implemented from scratch in MATLAB. Dimension reduction techniques such as LDA (Linear Discriminant Analysis) and PCA (Principal Component Analysis) are also applied to reduce the dimension of the dataset. The milestone of the project is to analyse these classifiers and present a comparative study of their accuracies. These classifiers are required to be applied on two classification tasks, a two class face recognition task and a subject label classification task.

Dataset

Three Data sets are provided in the .mat file format which are the data.mat file, the pose.mat and illumination.mat. Most of the processing in this project is done on the data.mat dataset which contains 600 images of 24 x 21 sized images of faces with neutral, expression and illumination effects. Data is first processed by splitting it into training and testing sets and

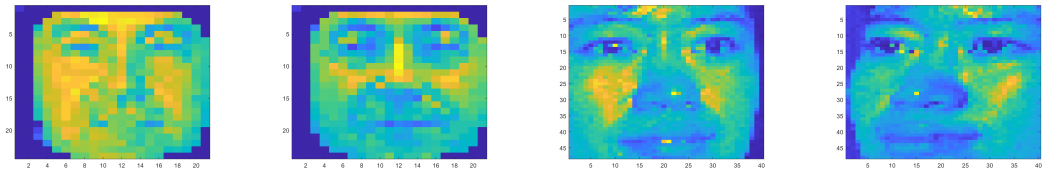

Figure 1: Neutral, Expression and Pose face image examples

Bayesian Classifier

Neutral v.s Facial Expression

For face recognition using Bayesian Classifier, the data was splitted into training and testing sets with different split ratios. The training dataset was divided into two class Neutral Face and Expression Face. It can be seen from the plots that with the increase in the split ratio; the accuracy shows an increasing trend until a split ratio of 0.6 and then it shows a decrease in the accuracy.

For the calculation of the posteriors, when the means and the co variances are calculated as per the Maximum Likelihood estimator, it is found that the co variance were singular, a noise was thus added to the co variance matrix thereby making it invertible. The optimal accuracy for the Bayes classifier was out at a split ratio of 0.6.

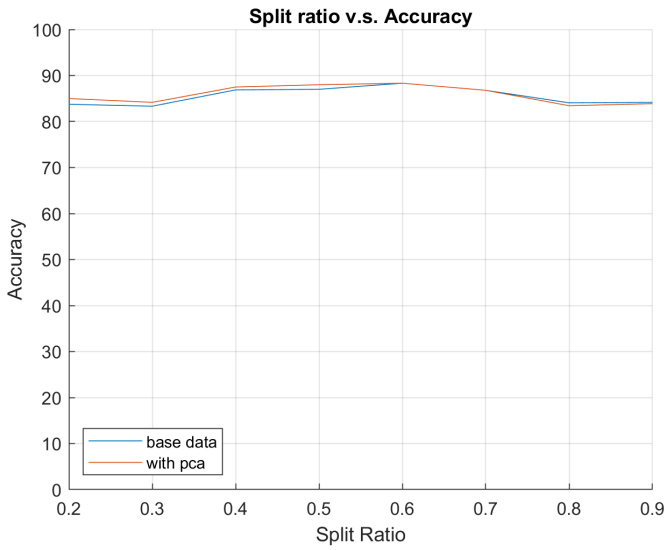

Figure 2: Bayesian: Base and PCA

1. Principal Component Analysis

Principal Component Analysis captures the linear trends in the data and thus reduces the dimensionality. Here the data was first processed using PCA transformation and then it was passed through a Bayesian Classifier. As can be seen from the plots the accuracy of the Bayesian Classifier after PCA also shows a similar trend as with the Bayesian Classifier. The accuracy measures are more for PCA+Bayesian than that for the Base Bayesian Classifier which is due to the dimension reduction after the axis is transformed.

2. Linear Discriminant Analysis

When LDA is implemented on the dataset, the dataset gets transformed to the dimension in which the between class scatter is high and within class scatter is low thereby making the classification task more easier. Thus, when LDA is used the accuracy of the classifier increases.

Subject Label Classification

For this task, the given dataset contains 3 types of images for 200 subjects. We have to classify the subject of the image based on the test data. The dataset data.mat is divided such that two out of the three images for each subject are selected for training and the remaining one out of three images is selected for testing.

Case	Training Dataset	Testing Dataset	Bayesian	PCA-Bayesian
1	Neutral + Expression	Illumination	64.00%	63.50%
2	Neutral + Illumination	Expression	67.00%	67.00%
3	Expression + Illumination	Neutral	71.50%	70.50%

As can be seen in the table, experimentation was done by taking the Neutral and Expression, Neutral and Illumination and Expression and Illumination as the training sets and the Illumination, Expression and the Neutral Images respectively as the testing set. Accuracy for the first case is 64.00% which is less because all the three images look similar and thus the classification becomes difficult. This task is also very time complex, since the co variance matrix and mean vector for all the 200 subjects is required to be calculated.

K Nearest Neighbor Classifier

Neutral v.s. Expression Face Classification

In the K nearest neighbor classifier, the accuracy depends upon the number of K. First the classifier is implemented on the basic data set and then the transformed data set which is obtained after LDA and PCA. The maximum accuracy is obtained when K is chosen to be 4. The accuracy when the value of K = 1 is 100.00% for the training data as the values are already seen by the model and a rough decision boundary is already formed. For testing data, the accuracy for K = 1 is found to be 71.00% which is the least because when you calculate the accuracy for the unseen data it performs really bad and so the error would be high. In case of KNN, when LDA and PCA is used on the dataset, we see a similar effect on the accuracy of the classifiers.

Number of Nearest Neighbors (K)	Raw Data	PCA	LDA
1	77.00%	77.50%	69.50%
2	79.00%	89.50%	68.50%
3	78.00%	81.00%	54.00%
4	83.00%	91.50%	54.00%
5	82.00%	84.00%	50.00%

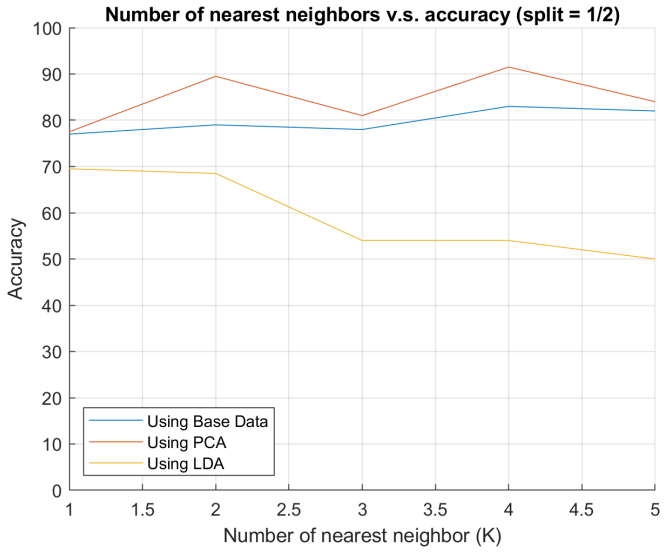

Figure 3: Number of nearest neighbor v.s. Accuracy

Subject Label Classification

In this case, the data was splitted in the case same as the Bayesian Subject Classification and experiments were carried out on the data.mat file.

Case	Training Dataset	Testing Dataset	Base	PCA+KNN
1	Neutral + Expression	Illumination	87.00%	52.50%
2	Neutral + Illumination	Expression	99.50%	99.00%
3	Expression + Illumination	Neutral	97.50%	98.50%

Kernel Support Vector Machines

In this project, the SVM primal problem was formulated in a Dual Lagrangian form and quadratic programming was used to solve the optimization problem. quadprog() function in MATLAB was used for the solution.

In this experiment, the value of C - the slack parameter was varied to find the optimum accuracy of the classifier.

Slack Parameters (C)	Accuracy
0.1	76.00%
0.2	84.00%
0.3	82.00%
0.4	82.00%
0.5	82.00%

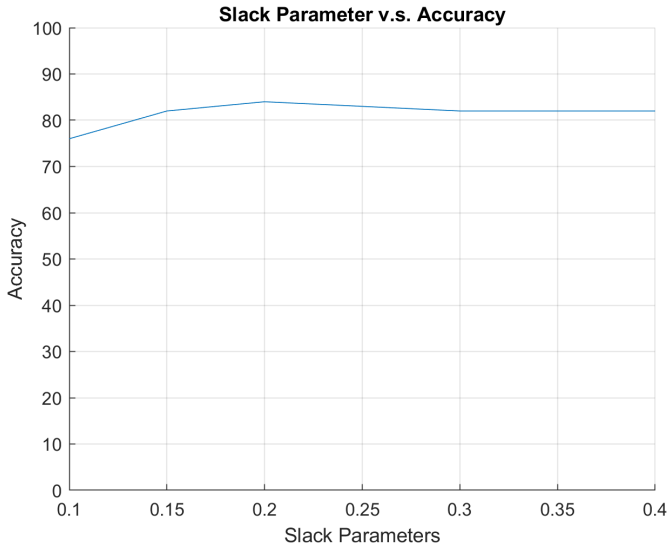

Figure 4: Slack Parameters v.s. Accuracy

Any linear model can be turned into a non-linear model by applying the kernel trick to the model: replacing its features (predictors) by a kernel function. Usually when the data is not linearly separable, kernel trick helps in raising the dimensionality and makes it separable.

Radial Basis Function Kernel

The RBF kernel function is $K = \exp \frac{-\|x-y\|^2}{\sigma^2}$ is used to model the data to improve the classification task. The data provided in the case of data.mat file was already linearly separable. Thus on using the RBF kernel the data got transformed to a more separable transform. Thus, the accuracy improved as compared to the linear SVM task.

Sigma	Accuracy
1	91.00%
2	91.00%
3	79.00%
4	75.00%
5	74.00%

Polynomial Kernel

The equation of the polynomial kernel is $K = (x^T y + 1)^r$. In the implementation of the polynomial kernel SVM on the neutral v.s. facial task, the change in the value of r changes the accuracy. The polynomial kernel gives the maximum accuracy of 92.00%

r	Accuracy
0	0%
5	0%
10	0%
15	0%
20	100%
25	100%
30	100%

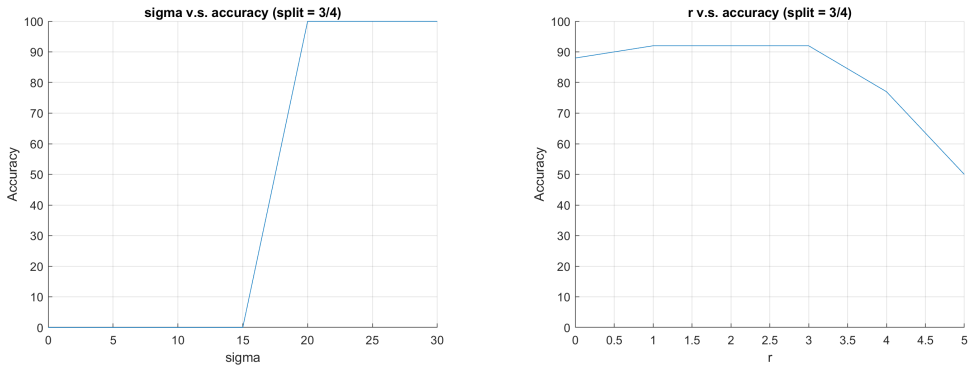

Figure 5: rbf and poly kernel svm

Boosting AdaBoost Algorithm

Adaboost boosts a weak learner to increase the accuracy of the classifier. Unlike many ML models which focus on high quality prediction done by a single model, boosting algorithms seek to improve the prediction power by training a sequence of weak models, each compensating the weaknesses of its predecessors. Boosting needs you to specify a weak model (e.g. regression, shallow decision trees, etc) and then improves it.

Conclusions

The choice of classification techniques highly depends upon the classification task. Over-fitting can be avoided by carefully choosing the training and the testing data. Transformation such as LDA and PCA and the kernel trick improve the performance since they try to exploit the linear separability in a higher and lower dimensions. Kernel SVM and Boosted SVM employ a lot of tricks to perform classification by taking the advantage of the dimensionalities of the data.