ENEE633 – Statistical Pattern Recognition | Project – 1

Due: November 15th, 2018 11:59 PM (EST)
Shubham Deshkar | UID: 115484689

Introduction

This project serves as a research and evaluation for different popular techniques and methods used in modern machine learning and pattern recognition such as Bayesian classification, Support Vector Machines and different transformation such as Linear Discriminant and Principal Component Analysis and Kernel trick to enhance the performance of the base classifiers at every task.

Different data sets were used to perform the different tasks such as binary classification and multiclass subject classification.

1. Bayesian Classification for Neutral vs. Facial Expression Classification:

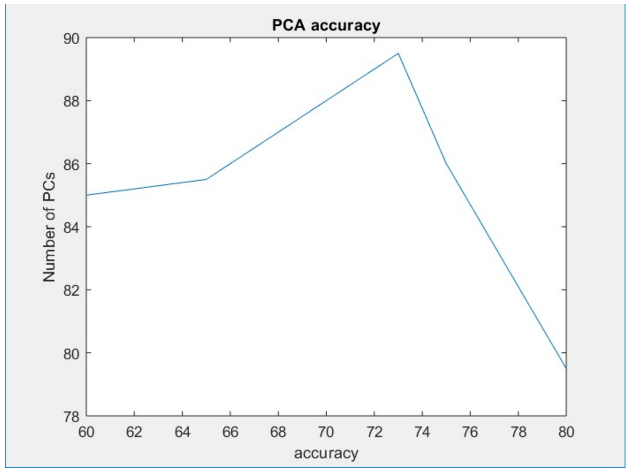
The data was partitioned in two classes namely, neutral class for neutral expression and expression class for smiling facial expression. The images were 21 by 24 pixels gray scale images and were converted into a column vector of length 504.

After careful observation, the covariance matrix formed from a single class was found to be near singular. Hence, a simple regularization technique was used to remove the singularity of the matrix by adding small amount of noise to the diagonal elements of the matrix. This made the matrix invertible and also enabled computation of a non-zero determinant.

After running several tests and partitioning the training and testing set at 50-50%, the accuracy of the classifier was 87%.

a. PCA Transformation:
As we know the PCA captures the maximum variances in the data and selects good features for classification along the components which are orthogonal to each other, it is necessary to include only the components corresponding to the maximum Eigen values, so it does not collect bad features and contaminates the data unnecessarily.

Number of components	Accuracy
60	85.00%
65	85.50%
70	88.00%
73	89.50%
75	86.00%
80	79.50%



As we established before, it is important to capture the maximum variance of the data reducing the dimensionality, we can see that for upto 73 Principal Components the components actually contribute significantly to the features and thereafter starts adding the noise to the data. Hence the accuracy goes on increasing to a certain point. At 73 components the variance captures was ~97.34%

b. Linear Discriminant Analysis:
   LDA is a very good transformation to achieve separation between the two classes on a particular dimension which minimizes the in-class variance and maximizes the between class separation.
   The accuracy achieved by this method was 90% since now the data was more separable.

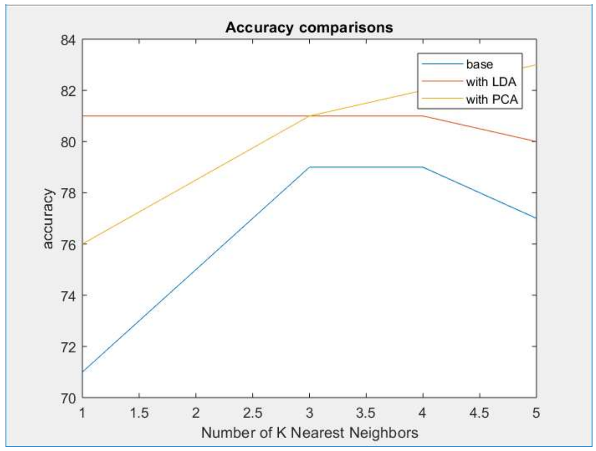
2. K – Nearest Neighbors for Neutral vs. Facial Expression:

K nearest neighbor algorithm was performed for the binary classification again. This was performed at first on the raw data and then using the two transformations. The accuracy was tested for different values of K. The results are as follows:

Number of Ks	Raw data	After PCA	After LDA
1	71.00%	76.00%	81.00%
3	79.00%	81.00%	81.00%
4	79.00%	82.00%	81.00%
5	77.00%	83.00%	80.00%

Also, this was tested for different values of principal components keeping the K = 3 as it provides the maximum accuracy and less chances of ties.

Number of PCs	Accuracy
25	83.00%
35	84.00%
45	84.00%
50	81.00%
55	81.00%



As discussed previously, there is a growing trend in accuracy and then it minimizes the effect of adding more principal components. This captures approximately 97% of the variance from the data.

3. Support Vector Machines for Neutral vs. Facial Expression Classification:

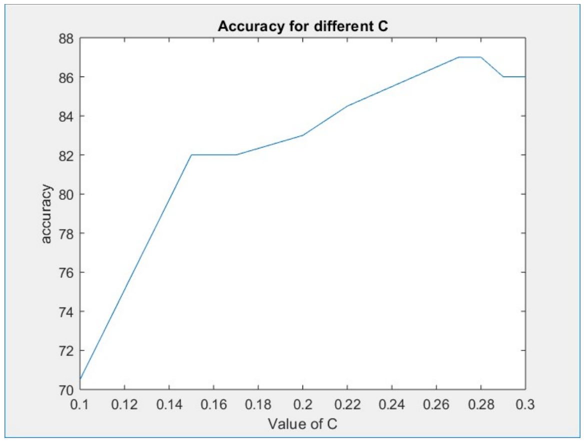
The primal problem was SVM was formulated in Dual Lagrangian form and was solved using the *quadprog(..)* function in MATLAB. The function very conveniently supports all the KKT conditions and they can be modelled very easily and were used to solve the dual problem for the SVM.

Although the data was linearly separable, the slack parameter was chosen to generalize the linear inseparability of the data. Hence, we now know that if the value of the slack (C parameter) is too low, it acts as a hard margin and hence the training error will be high, whereas if the slack is high the training error will be reduced but the testing error might go high.

Following shows the comparison of accuracies for different values of slack parameters.

Slack parameter	Accuracy
0.10	70.50%
0.15	82.00%
0.17	82.00%
0.20	83.00%
0.22	84.50%
0.27	87.00%
0.28	87.00%
0.29	86.00%
0.30	86.00%

Hence, we can see an approximate value for the classification task could be chosen as 0.27.

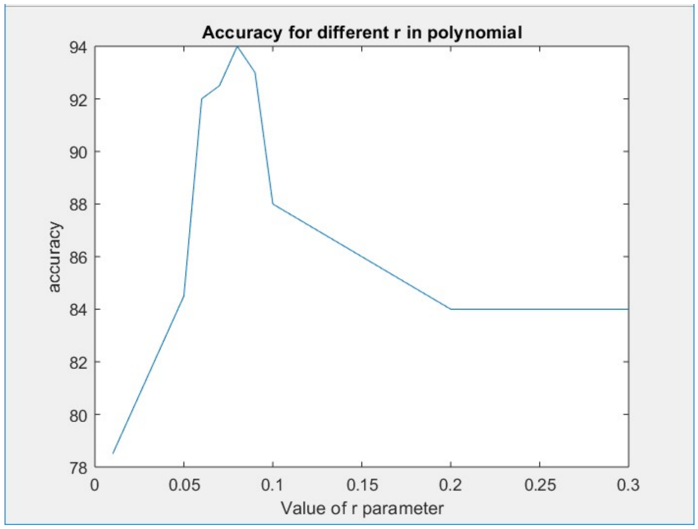


a. Kernel Trick:
   Kernel trick can help raise the dimensionality of the data and can make the data linearly separable.
   Radial **B**asis **F**unction Kernel was used to model the transformed data to improve the classification task. Since the data was already linearly separable, the kernel trick was expected to elevate the dimensionality to a more separable one and hence it was found that RBF function improved the classification task to 100% accuracy for many values of sigma^2.

Sigma^2	Accuracy
1	100.00%
2	100.00%
5	100.00%
10	100.00%
100	89.00%

Also, the polynomial kernel was used to transform the data using the parameter *r* and the SVM was performed on the data to improve the accuracy. Just like the RBF function, the polynomial kernel also gave a high accuracy.

R parameter	Accuracy
0.01	78.50%
0.05	84.50%
0.06	92.00%
0.07	92.50%
0.08	94.00%
0.09	93.00%
0.10	88.00%
0.15	86.50%
0.20	84.00%
0.30	84.00%



4. Boosted SVM for Binary Classification:

Since the data was highly separable, it was difficult to construct a classifier whose accuracy was low. The worst-case scenario would be to train the classifier using only 1 point in each class and test the classifier on the remaining data set of 199 points in each class. This gave an accuracy of 65.8291%, which can be considered as better than chance classifier.

Hence the boosting algorithm of Adaboost was implemented starting with a classifier with ~ 66% accuracy.

But there was a problem of diminishing the values of multiplying factor *a* and *theta_0,* hence the iterations could not sustain itself more than 3 times.

Number of iterations	Accuracy
1	65.8291%
2	68.3572%
3	74.6231%

5. Multi-class Classification

a. Bayesian Classifier:
The task of subject classification using ***data.mat*** was very difficult, since it provided merely 3 images in each of the 200 classes (subjects). The data was now partitioned in such a way that the training consisted only 2 points and testing was performed on 1 point for each subject. Hence, the anticipated accuracy was already very low.

This way one point was evaluated with 200 Bayesian posterior probabilities and this was performed for 200 testing points. Hence the evaluation was done 200x200 times which consumed a lot of computing power and also a lot of time.

The following evaluation was done keeping the neutral face and expression face in training set and illumination variation was used for testing.
Accuracy = 63.50% and Time = 2001.724 sec

b. K-NN:
The same classification task was carried out using K-NN algorithm, and as previous, L2 norm (Euclidean distance) was used to calculate the distance between to points. This algorithm was fast to give out results and hence accuracy was recorded for different combinations of the data.

- Neutral and Expression as training and Illumination variation as testing:
  Accuracy = 59.00%

- Neutral and Illumination variation as training and Expression as testing:
  Accuracy = 65.00%

- Expression and Illumination variation as training and Neutral as testing:
  Accuracy = 58.50%

6. Multi-class SVM for subject classification: One vs. all

We know that SVMs only classify for two classes and forms a linear classifier between them. In order to create a classifier between many classes we can perform SVM with one vs. all classification task.

In this case, the best option for the data set was to use ***illumination.mat***. It contained 68 subjects with 21 different illumination variations in them. For each class, first 20 images were selected as the training data and only one image was selected as the testing image to make the experiment computationally fast.

The data would be considered only as classified if it is classified in some class (one) rather than any other class (all). Then next task would be to check if *'some class'* would actually be the correct class. Only then it is correctly classified otherwise it its misclassified or simply ignored.

Since the classification task was modelled in a way that one class data is placed on one side and all other variations of the data which is either vague or very broadly generalized is kept in other side, it is highly likely that any random point to get classified correctly. Moreover, the number of points in one class is 20, which are perfect points and all other vague points and testing is done using only one point at a time, again goes to show that accuracy would be extremely high.

Hence, finally after evaluation the accuracy was 100%.

7. Change of data sets: Illumination Vs. Pose

The ***illumination.mat*** data set is really a robust data set with good features for each subject which makes it easy for the classifiers to classifies test-points correctly. As evident from above the illumination data set performed accurately for SVM which suggests that there is good separability between the classes. This will be further evident from the results evaluated below.

On the other hand, the ***pose.mat*** data set consistently performed poor for Bayesian and KNN classification models.

This evaluation was performed for both data sets using the KNN classification techniques and this time for fair evaluation and increase the randomness the data set was not split into training and testing sets but rather every point was evaluated against every other point in the data set keeping the labels safe. This was easily possible in KNN.

Without randomness for pose data set:

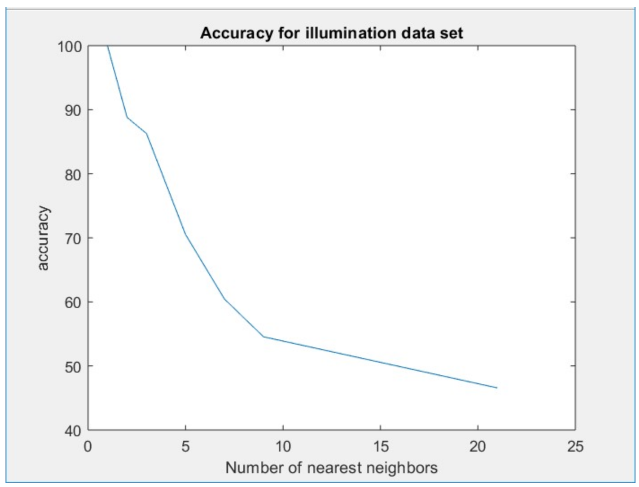
K	Accuracy
1	61.76%
2	36.27%
3	34.31%

With randomness for pose data set:

K	Accuracy
1	56.22%
2	40.95%
3	37.20%
4	36.20%

Illumination data set:

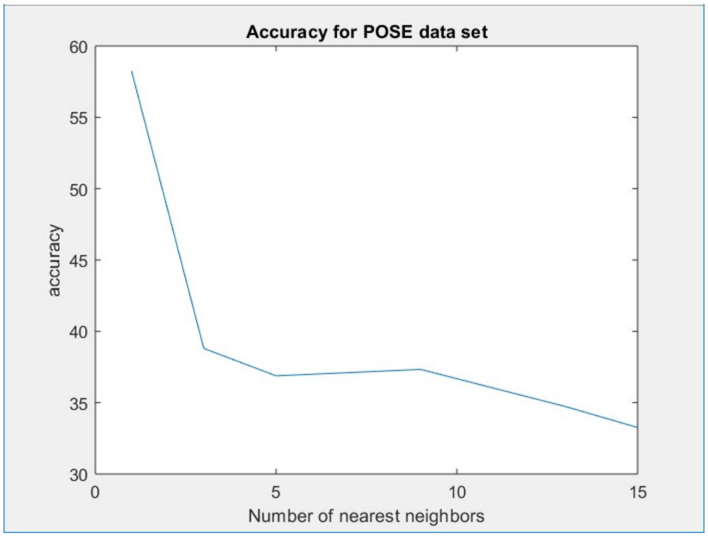
K	Accuracy
1	99.93%
2	88.80%
3	86.27%
5	70.52%
7	60.43%
9	54.55%
21	46.57%



As the number of nearest neighbors increased the classification performed poor because there are a lot of cases where the multiple classes get the same number of votes and tie breaking rules does not provide much help to improve the performance.

Pose data set:

K	Accuracy
1	58.26%
3	38.80%
5	36.88%
9	37.33%
13	34.73%
15	33.26%



The same trend can be observed for the pose data set. It already started with a poor accuracy and it went on getting worse as the number of K increased.

8. Conclusions:

Various techniques were finally evaluated and experimented with different transformations for different data sets with varying parameters and results were duly noted. It could be said:

- The choice of classification techniques highly depends upon the problem that one is trying to solve. Example Illumination set performed best for most classifiers.
- Transformations such as LDA, PCA and kernel trick are more likely to improve the performance since they try to exploit linear separability in higher and lower dimensions respectively but not always.
- Training accuracy and error should be carefully chosen so as to avoid overfitting of the data.
- The Kernel SVM and Boosted SVM were two of the very powerful classification techniques since they employ a lot of tricks to perform classification in a better way and takes full advantages of the dimensionalities of the data.
- Further, different and varying results could be obtained by making different combinations of kernelizations and transformation techniques along with Bayesian, KNN and SVM classification.