UNIVERSITY OF MARYLAND, COLLEGE PARK

Classic Classification Algorithms

Pranav Inani

pranavinani94@gmail.com

November 2018

CONTENTS

1 DATASET

All the algorithms were applied on the "Data" dataset. The data set consisted of 600 images of 200 subjects (3 images per person) each of size 24x21. Each subject had 3 types of images: a neutral face, an image with a facial expression and third image had illumination variations.

The data set was used to perform two types of classification tasks:
1) Train multi-class classifiers for person classification
2) train 2-class classifier for expression (neutral vs facial expression) classifier.

2 PERSON CLASSIFICATION

In this section we will discuss the Pipeline, results and effect of various variations on the performance of the Bayes Classifier and the K- Nearest Neighbour classifier applied to the problem of person classification on the "DATA" dataset.

2.1 DIMENSIONALITY REDUCTION

First, the images were unrolled into vectors and the relatively high dimensional data (1 x 504) was reduced using either Principal Component Analysis or Fisher's Linear Discriminant Analysis.

2.1.1 PRINCIPAL COMPONENT ANALYSIS

To apply apply PCA the data was first centered. However, in addition to making the data zero mean, experiments were also done by standardizing the data (making standard deviation = 1). The different results obtained by taking these different approaches will be summarized later but the on the whole it can be surmised that standardizing the data improved the performance slightly.

The PCA algorithm was set up such that the ratio of the eigen values corresponding to the selected eigen vectors divided by the sum of all the eigen values is greater than or equal to 0.95. This resulted in choosing the first 144 eigen vectors corresponding to the first 144 largest eigen values of the centered data multiplied with its transpose. The matrix containing these vectors were multiplied to the data to obtain the lower dimensional (1 x 144) dataset.

There are some articles [1] that suggest that the data centering and the PCA algorithm is to be done only on the training data and then the same transformations need to be applied on the test data. However, there is another school of thought that says since PCA is done on completely unlabled data, it is OK to apply PCA on entire dataset. This is even more justified when the data available is not large as is the case especially in the person detection problem.

2.1.2 FISHER'S LINEAR DISCRIMINANT ANALYSIS

Since, the person classification problem is a 200 class problem we can only find multiple discriminants (because Fisher's LDA can produce m linear discriminant functions, where m < #classes-1).

For the person detection problem 125 discriminant directions were chosen.

NOTE: It is worth noting that performing LDA on top of PCA did not have any significant change in the performance of the classifiers.

2.2 Test-Train Split

First, the dataset was split into training and testing sets and use the training data was used to train the classifiers and testing set was used to evaluate how the algorithm generalizes to unseen data..

For the person classification problem, the best performance was achieved by taking all the expression images (100) as testing data and the model was trained on the neutral and illumination images. The effect of various ways of splitting the data set will be explored in the results and conclusions section.

2.3 Bayes Classifier

First the Maximum Likelihood estimates of means and covariances were found for all 200 classes. To ensure the covariance matrix is invertible, an identity matrix was added to each of the ML estimates of it. Then the posterior probabilities were calculated to evaluate how likely the data sample belongs to one of the 200 classes. The class assignment is done to the class with maximum posterior. This is done iteratively over all the testing and training data and the accuracy was evaluated by comparing it with the true labels.

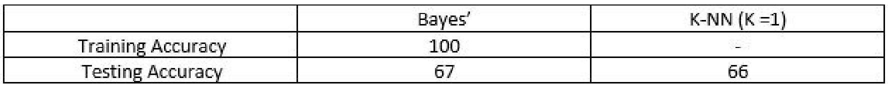
2.4 K- Nearest Neighbour Algorithm

The standard K-NN algorithm was applied. The tie breaking rule was handled as follows:
1) The k nearest neighbours were found and replaced with their class labels.
2) The mode of this vector was found.
3) If the frequency of the mode 1 (all k neighbours belong to different classes), then the data point is assigned to the nearest neighbour.
4) If the frequency of the mode is greater than one, then the data point is assigned to the mode i.e, the most repeated neighbour.

It was observed that as K increased the accuracy decreased. This is obvious since there are only 2 lables for each class. So the performance is bound to worsen if you go beyond K =2.

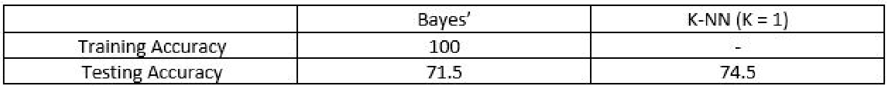
2.5 Results

It was found that in addition to centering the data to have zero mean, it was also beneficial to standardize it, i.e., to have standard deviation = 1. This was achieved using the zscore() function in MATLAB. This tended to improve the results by a few percentage points. However, something unnatural happens in one of the experiments. Specifically, for K-NN performed on top of PCA for zscored data gives a performance accuracy that is higher than the Bayes' accuracy which goes against the bound derived in the class. It is curious why this happens, my guess is some of the tie breaking is working in the favor of Bayes. The results of the various experiments are summarized below.



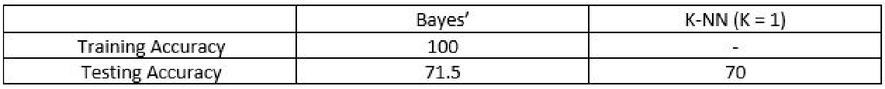
	Bayes'	K-NN (K =1)
Training Accuracy	100	-
Testing Accuracy	67	66

Figure 2.1: Person Classification: PCA with mean centering



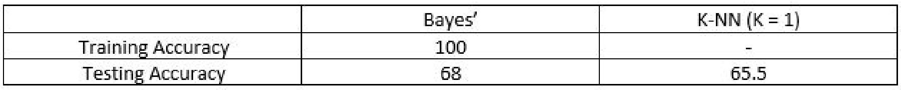
	Bayes'	K-NN (K = 1)
Training Accuracy	100	-
Testing Accuracy	71.5	74.5

Figure 2.2: Person Classification: PCA with zscore



	Bayes'	K-NN (K = 1)
Training Accuracy	100	-
Testing Accuracy	71.5	70

Figure 2.3: Person Classification: LDA with zsore



	Bayes'	K-NN (K = 1)
Training Accuracy	100	-
Testing Accuracy	68	65.5

Figure 2.4: Person Classification: LDA without zscore

3 Expression Classification

In this section we will discuss the Pipeline, results and effect of various variations on the performance of the Bayes Classifier and the K- Nearest Neighbour, RBF Kernel SVM, Polynomial Kernel SVM and Boosted SVM classifiers applied to the problem of expression classification on the "DATA" dataset.

3.1 Dimensionality Reduction

Before Dimensionality reduction, the illumination images were discarded as they were superfluous to the classification problem. Then, either Principal Component Analysis or Fisher's Linear Discriminant Analysis was applied.

3.1.1 Principal Component Analysis

PCA was applied in exactly the same manner as the person classification problem

3.1.2 Fisher's Linear Discriminant Analysis

Since, the expression classification problem is a 2 class problem we can only find 1 discriminant (because Fisher's LDA can produce m linear discriminant functions, where m < #classes-1). This results in the entire data being reduced to 1 dimensional data.

3.2 Test-Train Split

For the expression classification problem, the experiments were performance by taking the first 100 images as testing data and the model was trained on the rest of the 300 images.

3.3 Bayes Classifier

First the Maximum Likelihood estimates of means and covariances were found for all 200 classes. To ensure the covariance matrix is invertible, an identity matrix was added to each of the ML estimates of it. Then the posterior probabilities were calculated to evaluate how likely the data sample belongs to one of the 2 classes. The class assignment is done to the class with maximum posterior. This is done iteratively over all the testing and training data and the accuracy was evaluated by comparing it with the true labels.

3.4 K- Nearest Neighbour Algorithm

The KNN algorithm is applied exactly as in the previous section. In general it was observed that increasing the trend did not give a clear patter and the error oscillated without displaying a general trend.

3.5 Kernel SVM

The kernel SVM was implemented by solving the dual SVM problem using the **quadprog** function in MATLAB. Once the λ_i's were found the biases were found slightly differently than what is described in the text book. Instead the following formula was used [2]:

$$b^* = -\frac{\max_{i:y^{(i)}=-1} w^{*T}x^{(i)} + \min_{i:y^{(i)}=1} w^{*T}x^{(i)}}{2}.$$

Figure 3.1: Alternate formula to compute bias

The reason being this formula generalizes better rather than evaluating the bias at one of the support vectors.

3.5.1 RBF Kernel

For RBF kernel the optimal sigma was found to be 0.0039 using cross-validation and this value was used to compute the Kernel Matrix.

3.5.2 Polynomial Kernel

The data initially suggested a hint of slight linear separability but the test error was slightly poor. The best results were optained for d = 3.

3.5.3 Boosted SVM

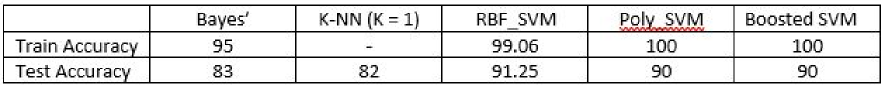
The boosted SVM for a class of linear SVMs was solved using CVX. The primal problem was solved in this case. The weights of the adaboost algorithm were incorporated by multiplying them to the respective hinge loss term in the objective function.

Since the data was close to linearly separated the algorithm ran only for 4 iterations. The errors go down monotonically within the algorithm. however the training and testing accuracies oscillate and finally training data reaches 100 percent accuracy and the algorithm converges.

3.6 Results

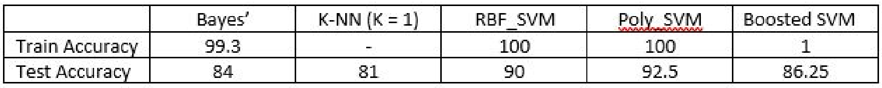
It was observed that standardizing the data was not as beneficial in this task. It sometimes improved and sometimes even worsened the reuslts as is shown in the tables below.

LDA reduced that data to a single dimension and the results obtained for it were generally worse than the PCA results. Further it can clearly be seen that the SVM methods performed better than Bayes and K-NN classifiers.



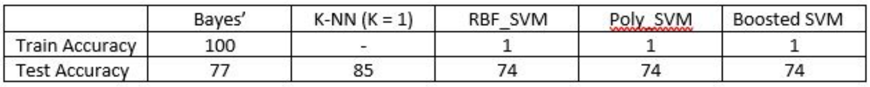
	Bayes'	K-NN (K = 1)	RBF_SVM	Poly_SVM	Boosted SVM
Train Accuracy	95	-	99.06	100	100
Test Accuracy	83	82	91.25	90	90

Figure 3.2: Expression Classification: PCA with mean centering



	Bayes'	K-NN (K = 1)	RBF_SVM	Poly_SVM	Boosted SVM
Train Accuracy	99.3	-	100	100	1
Test Accuracy	84	81	90	92.5	86.25

Figure 3.3: Expression Classification: PCA with zscore



	Bayes'	K-NN (K = 1)	RBF_SVM	Poly_SVM	Boosted SVM
Train Accuracy	100	-	1	1	1
Test Accuracy	77	85	74	74	74

Figure 3.4: Expression Classification with LDA

REFERENCES

[1] PatEugene, "Zero-centering the testing set after pca on the training set," 2018. [Online; accessed 15-November-2018].

[2] A. Ng, "Cs229 lecture notes: Support vector machines," 2018. [Online; accessed 15-November-2018].