ENEE 439M: Machine Learning

PROJECT-1

November 15, 2019

Shreya Gummadi
UID: 116195799
University of Maryland College Park
M.Engg in Robotics

CONTENTS

1 Pre Classification

1.1 Data Extraction

Each of the three datasets: data.mat, pose.mat and illumination.mat is unrolled into a 2D matrix of MxN containing N samples over M features. Each image is reshaped into a dx1 vector where d is the number of dimensions.

1. Data.mat
   The dataset data.mat has 3 images per 200 subjects. After unrolling, we have a 540x600 matrix with 600 data samples to build our classifier. For the two class; neutral vs facial expression problem, we take the neutral and expression images of all the subjects. Out of these we take a certain percent as train samples and the rest as test samples. We discard the illumination samples for this task. For the 200 class: identifying subject label problem, we take the neutral and expression images for each class as training and use the illumination samples as the test samples.
2. Pose.mat
   The dataset pose.mat has 13 images per 68 subjects. After unrolling we have a 1920x884 matrix with 884 data samples to build our classifier. We take a certain percent of data as test samples and rest as test samples.
3. The illumination.mat has 21 images per 68 images. After unrolling we have 1920x1428 samples with 1428 data samples. We take a certain percent of data as test samples and rest as test samples.

1.2 Dimensionality Reduction

1.2.1 Principal Component Analysis (PCA)

To reduce the dimensionality of the feature space we perform Principal Component Analysis (PCA). The goal of dimensionality reduction is to find a linear subspace of the feature space that has a reduced dimensionality.
This is done by procuring the first M' different principal components or eigenvectors ordered by their decreasing eigenvalues λi where, i = 1, ..., M .
M' is chosen based on the tolerance of error, e calculated from these eigenvalues.

$$\frac{\sum_{i=1}^{M'} \lambda i}{\sum_{i=1}^{M} \lambda i} > e$$

In this project e is chosen to be 0.95.

Projecting the dataset on the first two significant eigenvectors allows us to get some clarity of the classification task ahead. Figure 1, visualizes the dataset for neutral vs facial expression classification.

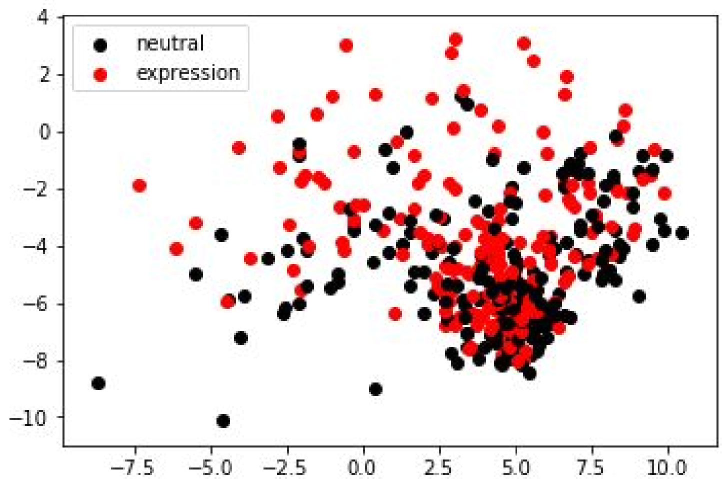

Fig 1. Projecting the data along 2 significant Principal Components for 2 class problem.

1.2.2 Fischer Linear Discriminant (LDA/MDA)

LDA can also be used as a tool for data reduction. LDA also aids classification as projecting the dataset increases class separation and reduces the within class variance.. It is observed that running Bayes classification with LDA improves the performance of classification in the course of the project. The LDA projection reduces the original dimension to c-1 dimensions as shown in table 1.

Dataset	Original dimension	Reduced Dimension
Data.mat (2 class)	504	1
Data.mat (200 class)	504	199
Pose.mat	1920	67
Illumination.mat	1920	67

Table 1. Dimensionality Reduction

2 Classification

2.1 Bayes Classifier

In Bayes classifier sample mean and covariance matrix for each class is calculated. Due to small data size and highly correlated images the covariance matrix becomes singular. To overcome this, noise is added to the covariance matrix in the form of a small constant diagonal matrix. This makes the covariance matrix non-singular The images are classified by assigning its label as the class with which the discriminant function has the highest value.

2.2 K- Nearest Neighbours

kNN classifies the test sample based on the voting evaluated by k of its nearest neighbors.The closeness is defined by a similarity measure and in this project the Euclidean distance is used.It is observed that as the value of k is increased the accuracy of performance but decreases after an optimal value of k.

3 Results

3.1 Two class Problem

Train-test split	Bayes	Bayes +PCA	Bayes+LDA	KNN (k=9)	KNN+PCA	KNN+LDA
60-40	0.8625	0.85625	0.9	0.7875	0.78125	0.90625
75-25	0.83	0.82	0.86	0.8	0.81	0.88
80-20	0.825	0.825	0.775	0.8	0.8	0.8875

Table 2. Effect of Train-test split percentage on accuracy

K	KNN	KNN+PCA	KNN+LDA
1	0.78	0.78	0.84
5	0.82	0.82	0.86

9	0.8	0.81	0.88
15	0.8	0.78	0.9

Table 3. Effect of K on accuracy (constant split 75-25)

3.2 N-class Problem

Data. mat:

Bayes	Bayes +PCA	Bayes+LDA	KNN (k=1)	KNN+PCA	KNN+LDA
0.64	0.61	0..55	0.595	0.58	0.575

Table 4. Accuracy with neutral and expression as train and illumination as test.

Other datasets:

Dataset	Train-test split	Bayes	Bayes +PCA	Bayes+LDA	KNN (k=1)	KNN+PCA	KNN+LDA
pose.mat	60-40	0.712	0.818	0.835	0.7323	0.7470	0.7794
pose.mat	70-30	0.74	0.772	0.809	0.6985	0.7058	0.7573
pose.mat	80-20	0.6875	0.71	0.77	0.617	0.6323	0.66
illuminatio.mat	60-40	0.902	0.9908	1.0	0.922	0.77	0.974
illumination.mat	70-30	1.0	1.0	1.0	1.0	0.995	1.0
illumination.mat	80-20	1.0	1.0	1.0	1.0	0.9926	1.0

Table 5. Effect of train-test split on pose and illumination data.

For the above above three datasets and experiment, k=1 which is nearest neighbour classifier gives the best accuracy. It is observed that as k is increased the accuracy decreases.

Discussion

- For all the datasets, Bayes classifier out performs the KNN classifier. The error is however bounded by the Bayes error.
- An optimal value of k gives the best accuracy.
- We can also see that dimensionality reduction with LDA out performs dimensionality reduction with PCA in most cases.
- As can be seen for the illumination.mat dataset the accuracy achieves 100 percent as the training data is increased. This could be due to some overfitting due to increased train samples.
- As training data size increases the accuracy generally increases.