ENTS 669D-Introduction to Machine Learning

Nihal A. Paul

114836427

**Project 1: Face Recognition**

**Aim**: My objective of this project is to primarily achieve face recognition using Baye’s Classifier and K-Nearest Neighbours Classifiers. Input images are usually large sized. Therefore, a re-implementation of these classifiers by first reducing the dimension by standard dimension reduction techniques has achieved. I have tested on two of the provided datasets, Data.mat and Pose.mat, to see how the classifier performs on both these datasets. Also, I have experimented with a number of parameters to notice the effect of small changes in the model to the accuracy of the classifiers.

The Report has been organized as follows:

1. Bayes Classification Results
2. K-Nearest Neighbours Classification Results
3. PCA Result
4. LDA Results

Important Notes:

1. Feature Selection: To accurately represent features of the image, I have chosen LBP features. The computational complexity involved is considerable, but the features are represented fairly accurately.
2. Effectiveness Measure: Accuracy is chosen as the effectiveness measure, where

Accuracy= number of correctly classified test points  
 total number of test points

1. **Bayes Classifier Results**

*Brief summary about the classifier:* The classifier models the training data as Gaussians.

The main aim is to find the conditional probability p(wi/x). This is done according to Baye’s Theorem, p(wi/x) =p(x/wi)×p(wi)/p(x).

We need to only compute the p(x/wi) term for each i=1..c, as the priors p(wi) and p(x) are constants for all terms.

Assuming Gaussian distribution, we fit a Guassian on the data. Therefore, we get,

p(x/wi)= (1 /((2π)^d/2)\*|Σ|1/2) \* exp[−1/ 2((x−µ)tΣ^(−1)(x−µ))]

x here, are the features.

That value of I for which p(x/wi) attains maximum value is assumed as the class of the input test vector.

Baye’s Classification was tested on two datasets, Data.mat and Pose.mat.

The results obtained for Data.mat were as follows:

1. Data.mat: The dataset contains 200 subjects in total and has a total of three face variations per subject. So, varying training and testing ratio is meaningless. We need to have a minimum of at least two training points per class for Bayes.

As such, number of training points were chosen as the first two images of each class, ie, normal face and expression face. Therefore, the total size of training data taken is: 200 ×2 = 400 training images. Illumination was chosen as test data.

The table below summarizes the variation of accuracy vs number of test points fed into the classifier

|  |  |
| --- | --- |
| **Number of test points** | **Accuracy(%)** |
| 4 | 58.33 |
| 12 | 58.33 |
| 33 | 58.75 |
| 75 | 55 |
| 120 | 50 |
| 150 | 50.75 |
| 175 | 53.67 |
| 200 (all test points) | 54 |

Table 1: variation of number of test points and accuracy for Data.mat

1. Pose.mat: This dataset has 13 images per subject, each with a different pose, for a total of 68 subjects. I have chosen the first 6 poses per subject to be reserved as training data to be fed into the classifier. Therefore, my total number of training samples fed into the classifier for configuration were: 68 × 6 = 408 images. Variable size of test data were fed into the classifier. The variation of accuracy vs number of testing data was as follows:

|  |  |
| --- | --- |
| Number of test points | Accuracy |
| 4 | 22.23 |
| 12 | 31.34 |
| 33 | 43.75 |
| 75 | 49.67 |
| 120 | 42.34 |
| 150 | 38.69 |
| 175 | 33.67 |
|  |  |
| 200 (all test points) | 41.75 |

Table 2: variation of number of test points and accuracy for Pose.mat

**Training Vs Testing:**

For the Pose Dataset, training and testing data points were varied per class. In the above example, 6 of the first 13 poses per subject were chosen as training data. Experiments were carried out to see the variation when 2, 4 and 9 of the first 13 poses per subject were chosen as training data as well. For each case, test samples of 9 and 16 were fed into the classifier. The accuracy was observed to increase as the number of training samples were increased, for a fixed number of testing samples. The graph below shows the variation of accuracy obtained vs number of training samples, for a fixed 12 test samples.



Few Points of Observation:

1. The Covariance matrix turns out to be singular to working precision, while working on matlab. To tackle this problem, an identity matrix was added to the covariance matrix.
2. In the Pose dataset, the term blows up, because the value of d is 1920. Therefore, this term was removed from calculations as this is a constant term and won’t affect the decision of the classifier.
3. Input images were adjusted in the range [0,1].
4. **K-NN Classifier Results**

*Brief summary about the classifier:* This is a very straightforward classifier, which computes a distance metric of the test feature to all training features. Then, K of the least distant training features are chosen and a poll is taken among these K training features. The test feature is then assigned the label of the class which has maximum poll among the K nearest neighbours.

Modification: While working on Data dataset, there were only two samples to train per subject. Therefore, it was common to find two classes contributing two samples each to the K nearest neighbours. Eg: KNN=[ 121, 54, 1, 54, 149, 121]. In such a scenario, a the two maximum contributing classes are evaluated separately on the basis of the minimum distance. The one closer to the test point is chosen as the final class. In case of no majority poll among the K nearest neighbours, a random class is chosen and assigned.

The pseudo code for selection of the test point’s class after obtaining K nearest neighbors is:

if only one class is repeated twice

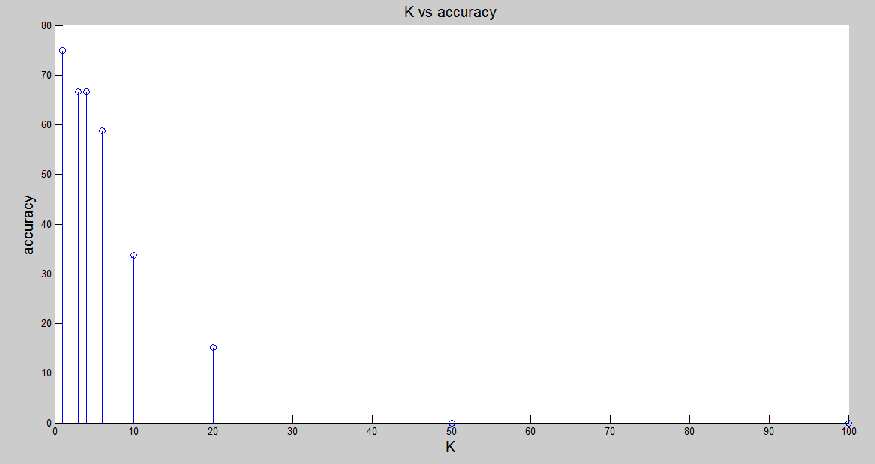
assignedClass=class that repeats twice

else if more that one class appears twice  
 assignedClass=minimum of the two repeating classes

else if no majority poll among K nearest neighbours

assignedClass= random value sampled from the K nearest neighbours.

Variation of accuracy with K: For the case of 400 training samples, and choosing a fixed 12 samples to test, it was observed that as the number of neighbours is increased, the accuracy decreases monotonically. The graph is shown below:



1. **PCA Results**

*Brief summary about PCA:* A component of along which the variance among the data is maximized is known as the first principle component. Q principle components are chosen, thereby reducing the data dimension from d ( d-component x) to q. This significantly reduces computational complexity.

PCA was applied to the Data.mat Dataset, for 400 train images (first two from each class). 6 nearest neighbors for Knn were chosen. For 33 test samples, the variation of accuracy with the number of features (q) is shown in the table below, for both, Bayes and Knn.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| q | 504 | 400 | 300 | 200 | 100 | 50 | 10 |
| accuracy | 54 | 48.48 | 42.42 | 39.39 | 18.18 | 15.15 | 0 |

Table 3: variation of accuracy with number of principle components for Bayes clasifier

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| q | 504 | 400 | 300 | 200 | 100 | 50 | 10 |
| accuracy | 52.56 | 47.8 | 33.3 | 26.67 | 15.34 | 8.33 | 0 |

Table 4: variation of accuracy with number of principle components for Knn classifier

*Take aways*: As the number of features is decreases continuously, the accuracy decreases as well. This follows the logical inference, that the more features that are selected to represent the data, the less information loss incurred and consequently, the more precise will be our classification results. Notice also, that for all 504 features, the accuracy is the same as using the classifier (Bayes or Knn), without dimensionality reduction.  
The time complexity however, decreases markedly. Eg. For 50 features in Bayes, the time to run was 5.725 seconds only while for 400 features, this number jumped to 92.48 seconds.

1. **LDA Results**

*Brief summary about LDA:* This is primarily a classification technique (known as Fisher’s discriminant analysis). But this can also be used to reduce dimensions of the image to c-1 dimesions. We not only find the axis that maximizes the scatter between classes (between class scatter matrix) but also find the axis that corresponds to minimum scatter within a class (within class scatter matrix). Unlike PCA, LDA is supervised as we need to provide class labels.

LDA was tested on Data.mat dataset, for 33 sample test images. A total of 200 training images were chosen. 6 nearest neighbors for Knn were chosen. The dimensions were reduced by finding the eigan vectors that correspond to inv(Sw)\*Sb. It was observed that LDA gave poor results compared to PCA for similar dimensions. This is summarized in the two tables below:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| q | 504 | 400 | 300 | 200 | 100 | 50 | 10 |
| accuracy | 41.67 | 30.3 | 27.17 | 21.21 | 18.18 | 3.03 | 0 |

Table 3: variation of accuracy with number of principle components for Bayes clasifier

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| q | 504 | 400 | 300 | 200 | 100 | 50 | 10 |
| accuracy | 52.56 | 30.3 | 26.67 | 15.15 | 9.09 | 0 | 0 |

Table 4: variation of accuracy with number of principle components for Knn classifier

While selecting all features in LDA, the accuracy was still lesser than running the two classifiers without any dimensionality reduction.