# ENEE439M Project 1
# Classifier Design for Face Recognition

Wenyan Li
wenyanli@umd.edu

April 10, 2017

**Abstract**

This project mainly focuses on the implementation of the classifiers for face recognition. Data of faces with different poses, expressions and illuminations are preprocessed and divided into trainning sets and testing sets for experiments. Principal Component Analysis(PCA) and Fisher Linear Discrimenant Analysis(LDA) are implemented for dimension reduction. Bayes rule based on Maximum Likelihood Estimation(MLE) and k-Nearest Neighbor(K-NN) rule are applied in classifier design. Classification results are analyzed and discussed.

## 1 Introduction

In this project, following classifiers are implemented for face recognition

1. MLE with Gaussian assumption followed by Bayes rule
2. K-Nearest Neighbor(KNN) rule
3. PCA followed by KNN and Bayes rule
4. LDA followed by KNN and Bayes rule

With classifiers succesfully implemented, classification results of different classifiers are analysed and further measures taken for imporving the accuracy of classification are discussed.

## 2 Data preprocessing

All three provided datasets: DATA, POSE, and ILLUMINATION are used in this project, and samples in each datastet are divided into training data and test data for further experiment as shown in Table.1. Labels for each sample are also extracted and stored for accuracy computation.

### 2.1 DATA

In this dataset, 600 images with size of $21 \times 24$ are provided for 200 subjects, and 3 images each(i.e. this is a problem with 200 classes and 3 samples in each class). Each image is reshaped into a $504 \times 1$ vector, and two images from each class are extracted and used as trainning samples, and the one left is used for testing,i.e. the dataset is preprocessed into a trainning set of size$400 \times 504$ and a testing set of $200 * 504$. All three ways that use different type of images in the class for trainning and testing are all experimented.

Table 1: Data preprocessing

| No. | Dataset | Size | Trainning set | Testing set |
|---|---|---|---|---|
| 1 | DATA | $200 \times 3 \times 20 \times 24$ | $400 \times 504$ | $200 \times 504$ |
| 2 | POSE | $68 \times 13 \times 48 \times 40$ | $680 \times 1920$ | $204 \times 1920$ |
| 3 | ILLUMINATION | $68 \times 21 \times 1920$ | $952 \times 1920$ | $476 \times 1920$ |

### 2.2 POSE

$68 \times 13$ images are provided with each image of size $48 \times 40$ in this dataset. Similarly, each image is first reshaped into a $1920 \times 1$vector, and then the dataset is divided in to a $680 \times 1920$ trainning set and a $204 \times 1920$ testing set.(i.e.10 images from each class(68 classses in total) are used for trainning while 3 images are for testing) The propotion of images used for trainning and testing are adjusted for higher accuracy in later experiment.

### 2.3 ILLUMINATION

In this dataset, the image matrices are already reshaped into vectors. So the data are simply divided into a trainning set of size $952 \times 1920$ and a testing set of size $476 \times 1920$, i.e. the propotion of the data used for trainning and testing in each class is 2:1.

## 3 Classifier implementations

### 3.1 KNN rule

Nearest neighbor is first implemented, i.e Euclidean distances from each test point to all the points in the trainnning set are computed, and the label of the test point is set as the same as its nearest neighbor. Then for KNN rule, the label of the test point is determined based on the vote of the $k$ neighbor points. Here, it is observed that with the number of neighbors increase, there's no significant increment of the classification accuracy, on the contrary, when $k > 4$ the accuracy keeps decreasing as $k$ increases.
In order to obtain a higher accuracy, improvements are made on KNN s.t. if the vote for label $i$, where, take dataset 1 for example, $i = 1, 2, ...200$, does not have majority vote, the decision will be made following the NN rule rather than KNN rule.

### 3.2 Bayes rule

First, sample mean and covariance matrix for each class is calculated, and if the covariance matrix is singular, a small constant diagnoal matrix is added to make it non-singular. Bayes decision rule for multivariate condition is then used for classifying the faces, i.e. for each test point, its label is set as the class with which the determinant function: $g_i(x) = x^t X_i x + {w_i}^t x + w_i 0$ has maximum value.

### 3.3 Use PCA for dimension reduction

Singular Vector Decomposition(SVD) of the coviarnace matrix is used for finding the direction descried by the principal components, and then the original data is projected into $k$ dimensions, where $k$ is set as a parameter in the function applyPCA(). Here for data obtained in Table.2 and Table.3, we project the original data into 100 dimension. In order to compare the performance

Table 2: Classification results on dataset 1

| No. | Faces for training | Faces for testing | KNN | PCA + KNN | LDA + KNN | Bayes | PCA + Bayes | LDA + Bayes |
|---|---|---|---|---|---|---|---|---|
| 1 | Neutral & Expression | Illumination | 0.615 | 0.600 | 0.620 | 0.640 | 0.635 | 0.640 |
| 2 | Neutral & llumination | Expression | 0.650 | 0.645 | 0.660 | 0.665 | 0.670 | 0.665 |
| 3 | Illumination & Expression | Neutral | 0.545 | 0.545 | 0.565 | 0.72 | 0.69 | 0.71 |

Table 3: Classification results on dataset 2 and 3

| Dataset | Training set | Testing set | KNN | PCA+KNN | LDA+KNN | Bayes | PCA+Bayes | LDA+Bayes |
|---|---|---|---|---|---|---|---|---|
| **POSE** | $680 \times 1920$ | $204 \times 1920$ | 0.5882 | 0.5833 | 0.5784 | 0.6814 | 0.7304 | 0.7794 |
| **ILLUMINATION** | $952 \times 1920$ | $476 \times 1920$ | 0.9538 | 0.9013 | 0.8214 | 1.0000 | 1.0000 | 1.0000 |

of LDA and PCA, we project the original data into $(c-1) = 67$ dimension for data acquired in Table.4.

### 3.4 Use LDA for dimension reduction

Using Fisher LDA, within-class scatter and between-class scatter are computed . Again SVD is used for solving eigenvectors, and pseudoinverse pinv() function is used if the with-class scatter has a determinant of zero. The original data is projected into $(c-1)$ dimension.

## 4 Classification testing and result analysis

From Table.2 and Table.3, we can observed that using Bayes rule for classification has highest accuracy as expected. However, classifying with Bayes will takes a longer time. Using PCA and LDA, the classifying process can be accelerated. Here we reduce the dimension into 100 for PCA, and $(c-1)$ for LDA.

### 4.1 Selection of data

By observing the differences among the three datasets, we can generalize that the selection of data used for training and testing also influence the accuracy. Especially, as shown in Table.2, when training with faces under illuminations and faces of expressions, the classifier has a higher accuracy than being trained with neutral faces. Moreover, when using the dataset ILLUMINATION for training and testing, the accuracy for bayes classification becomes 100%, which is much higher than using the first two dataset.

Table 4: Classification results using different size of training data

| Dataset | Training set | Testing set | KNN | PCA+LNN | LDA+KNN | Bayes | PCA+Bayes | LDA+Bayes |
|---|---|---|---|---|---|---|---|---|
| **POSE** | $276 \times 1920$(30%) | $612 \times 1920$ | **0.4984** | 0.4967 | 0.5065 | **0.5621** | 0.5686 | 0.5686 |
| **POSE** | $340 \times 1920$(40%) | $544 \times 1920$ | **0.6085** | 0.5956 | 0.5882 | **0.6673** | 0.6654 | 0.7004 |
| **POSE** | $476 \times 1920$(50%) | $408 \times 1920$ | **0.7304** | 0.7255 | 0.7206 | **0.8309** | 0.8382 | 0.8652 |
| **POSE** | $544 \times 1920$(60%) | $340 \times 1920$ | **0.7059** | 0.7059 | 0.7294 | **0.8029** | 0.8353 | 0.8529 |
| **POSE** | $612 \times 1920$(70%) | $276 \times 1920$ | **0.6765** | 0.6691 | 0.6801 | **0.7500** | 0.8015 | 0.8346 |
| **POSE** | $680 \times 1920$(80%) | $204 \times 1920$ | **0.5822** | 0.5784 | 0.5784 | **0.6814** | 0.7304 | 0.7794 |
| **POSE** | $816 \times 1920$(90%) | $68 \times 1920$ | **0.5882** | 0.5735 | 0.5441 | **0.7794** | 0.8676 | 0.9118 |

### 4.2 A good training data size

Taking dataset 2 for experiment to see how the size of the training dataset influence the accuracy. From Table.4, it can be observed that when the size of training data varies from 30% to 90%, the accuracy of KNN and Bayes classification first increases and then drops(regardless of impact of PCA and LDA, here reduce to $c-1$ dimension for both). The reason might be that when training data size is too small, the model may be under-fitting, but when training data size is too large, the model may be overfitting.

### 4.3 PCA and LDA followed by Bayes rule

If we project the original data to $c-1$ dimension both for PCA and LDA, with data obtained in Table.4, it can be seen that LDA followed by Bayes outperforms than PCA followed by Bayes rule, which might because of LDA explicitly attempts to model the difference between the classes of data and PCA on the other hand does not take into account any difference in class.

## 5 Conclusion

In this project, classifiers with good accuracy are successfully designed and compared for face recognition. Classification results using different methods are analyzed and discussed. High accuracy for classification can be obtained if the classifier is trained with a proper amount of data which better have more variances(such as data of faces with both expression and illumination). Dimension reduction techniques such as PCA and LDA has great performance for speeding up the classification while outputing good classification results.

## References


[1] Richard O. Duda, Peter E. Hart, David G. Stork. *Pattern Classification.* 1973

[2] In Jae Myung. *Tutorial on Maximum Likelihood Estimation.* Journal of Mathematical Psychology,47,(2003) 90100

[3] Jon Shlens. *A Tutorial on Principal Component Analysis.* Retrieved from https://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition