EE5907R - Pattern Recognition
Sem I AY2011/2012

Part II Project

Phang Swee King
A0033585A
NGS/ECE Dept.
Faculty of Engineering
National University of Singapore
Email: king@nus.edu.sg
Website: www.PhangSweeKing.com

December 2, 2011

Abstract

In this project, two interesting pattern recognition problems is going to be investigated. First is the classification of human faces according to ethnic group, and second is the analysis on the data obtained based on the eye fixation position in an experiment. The main classification method that will be used is the Nearest Neighbor (NN) classifier. Important feature will be extracted by several feature extraction method, such as the Principal Component Analysis (PCA), the Nonnegative Matrix Factorization (NMF), and the Linear Discriminant Analysis (LDA). Also, the general Gaussian Mixture Model (GMM) will be studied to obtain the distribution properties of the database. Each of their performance will be compared with figure and discussed in detail.

Contents

1 Introduction **2**
- 1.1 Project Requirements . . . 2
- 1.2 Programming Language . . . 3

2 Principal Component Analysis **4**
- 2.1 PCA Algorithm . . . 4
- 2.2 Eigenfaces Display . . . 5
- 2.3 NN Classification Results . . . 5

3 Non-Negative Matrix Factorization **7**
- 3.1 NMF Algorithm . . . 7
- 3.2 Convergence of J . . . 8
- 3.3 Bases of NMF . . . 8

4 Linear Discriminant Analysis **13**
- 4.1 Computation of LDA . . . 13
- 4.2 Fisherfaces Display . . . 14
- 4.3 NN Classification . . . 14

5 Gaussian Mixture Model **16**
- 5.1 Expectation-Maximization algorithm . . . 16
- 5.2 Simulation Results . . . 17

6 Conclusion **19**

Chapter 1

Introduction

Pattern recognition has greatly attracted interests from many research institutes, due to its usefulness and fanciness. Recently in 2010, the famous social network service — Facebook, has incorporate an efficient face recognition algorithm in tagging photos to replace the traditional manual tagging. Also, affordable laptop such as Alienware from Dell Inc., has also embedded the face recognition feature in determining the correct user of the laptop. These examples show the importance of face recognition in the current technological world.

In this project, two database were used to investigate performance of various feature extraction methods. The first database is the ethnic database includes 240 images from four ethnic groups stored in four directories. For each group, the images indexed as 1 to 30 are used for training set and the images indexed as 31 to 60 are used as testing set. The database can be downloaded at `http://www.lv-nus.org/Ethnic.zip`.

The second database is the image saliency map. The data were collected based on the eye fixation data mainly from the students of EE5907R module. For this project, 400 images in saliency map is used and they can be downloaded at `http://www.lv-nus.org/Saliency.zip`.

1.1 Project Requirements

In this project, we will explore the feature extraction and the GMM learning problems. For both databases, the pixel gray-level values are used as features.

We are required to complete the following tasks:

1. For the ethnic database, learn the Principal Component Analysis (PCA) on the training set, a) display the first 10 Eigenfaces (PCs, rescale to 0-255 values), 2) display with figure the face recognition accuracies of the testing set over different dimension-reduced feature dimensions (based on Nearest Neighbour approach, and dimensions from 1-100).

2. For the saliency map database, learn the Nonnegative Matrix Factorization (NMF), a) display the 50 bases (rescale to 0-255 values, and set the lower feature dimension as

50), b) re-run twice with random initializations, compare the bases for all these three runs, and see whether the results are the same. If different, explain the reasons.

3. For the ethnic database, learn the Linear Discriminant Analysis (LDA) on the training set, a) display the first 10 Fisherfaces (bases, rescale to 0-255 values), 2) show with figure the face recognition accuracies of the testing set over different dimension-reduced feature dimensions (based on Nearest Neighbour approach, and dimensions from 1-100). Compare the accuracies from PCA and LDA, which is better? Please explain the reasons. Note that to avoid singular issue, you may first conduct PCA before LDA and reduce the feature dimension to 110, but for Fisherface display, you need reconstruct back to original dimension.

4. a) learn the general Gaussian Mixture Model (GMM) on the saliency database with the component number set as 8, and display the 8 centres/means of the components, b) observe these 8 centres, and describe what you may observe from the obtained centres/means (spatial distribution properties for the large values). Note that, to avoid singular issue, you may first conduct PCA before GMM to reduce the feature dimension to 40, but for centres display, you need reconstruct back to original dimension.

1.2 Programming Language

There is no restriction on programming language. We may use any language of our preference, i.e., MATLAB, VC++, Java, etc.

In this project, the program are written in MATLAB environment. In this project report, each of the requirements mentioned above will be discussed in detail in each chapter, starting from Chapter 2 to Chapter 5. Conclusions will be given in Chapter 6.

Chapter 2

Principal Component Analysis

Principal component analysis (PCA) is a type of unsupervised feature extraction. It involves a procedure to transform a number of possibly correlated variables into a number of uncorrelated variables, which are commonly called principal components (PC). The transformation is done such that the first PC has the highest variance, followed by the second highest variance PC which is orthogonal to the first PC, and so on. It is motivated by the fact the intrinsic dimension of most of the images are relatively low compared to it's complete variables, and thus PCA can be utilized to extract only the first p number of PCs, with minimal information lost. PCA is widely used in data compression and classification and since it only captures big variability in the data and ignores small variability, sometimes it can also help remove noises.

2.1 PCA Algorithm

According to the lecture notes [1], there are two common guidelines to compute PCA. The first method is to

1. Form the covariance matrix, S, using all the training data information, $S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x}^T)$.
2. Compute its eigenvalue, $S = U^{-1}\Lambda U$.
3. The first p columns in U (first p eigen vectors of S) form the p PCs.

The second method, which is more preferred, is to

1. Form the centered data matrix using all the training data information, $X_{d,n} = [(x_1 - \bar{x}) \ ... \ (x_n - \bar{x})]$.
2. Perform singular value decomposition on X, $X_{d,n} = U_{d,d}D_{d,n}V_{n,n}^T$.
3. The first p columns in $U_{d,d}$ form the p PCs.

In practice, computing the PCs via singular value decomposition (SVD) on the centered data matrix is preferred. In this project, this preferred method is adopted. The low dimension data can be computed as

$$\hat{x}_i = \bar{x} + U_{d,p}(U_{d,p})^T(x_i - \bar{x}) \tag{2.1}$$

Note that the restoration may not be perfect, but is a good approximation.

2.2 Eigenfaces Display

Through the PCA on the set of images, a set of eigenfaces can be generated. Eigenfaces can be considered a set of standardized face ingredients, acting as a basis to form various human faces, derived from statistical analysis of many pictures of faces. In this project, the columns of $U_{d,p}$ form the eigenfaces. Every sample face x_i can be represented as a linear combination of these eigenfaces plus an average face $\bar{x}$ (shown in Eqn-2.1).

To display graphically the eigenfaces, 10 first eigenfaces, corresponding to the 10 largest eigenvalue, or simply just the first 10 columns of $U_{d,p}$ is displayed using the embedded `imshow` function in MATLAB. The resulting images are shown in Fig. 2.1.

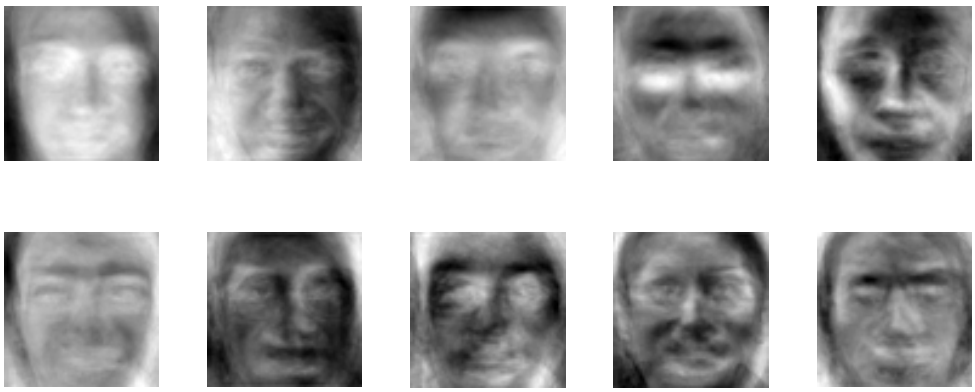

Figure 2.1: First 10 eigenfaces for ethnic group classification

2.3 NN Classification Results

PCA itself is not a classifier, it is a feature extraction method to reduce original data dimension to a lower one, while retaining as much information as possible. Next, the reduced dimension data is classified using the nearest-neighbor algorithm.

Here, our aim is to test the performance of the PCA and not the NN classifier. Thus, accuracies of the testing set are recorded over different dimension-reduced feature dimensions, from $p = 1$ to $p = 100$. Fig. 2.2 shows the accuracies result, with each circle point represent

the percentage of hit (classified as correct class) with respect to the number of dimension, p.

We observe that the results are floating around 30% to 35% accuracy, which indicate that about 30 to 35 correct classifications can be obtained from 100 different pictures. The low classification accuracies obtained are due to the insufficient training set, where only 30 picture from each races are trained. Also, it might be caused by the conversion of RGB to grayscale of the images, which greatly distorted one important feature for ethnic classification, which is the skin color. In general, however, the classification accuracies show an increasing trend towards higher p value, which physically means more feature information are retained when p value is higher.

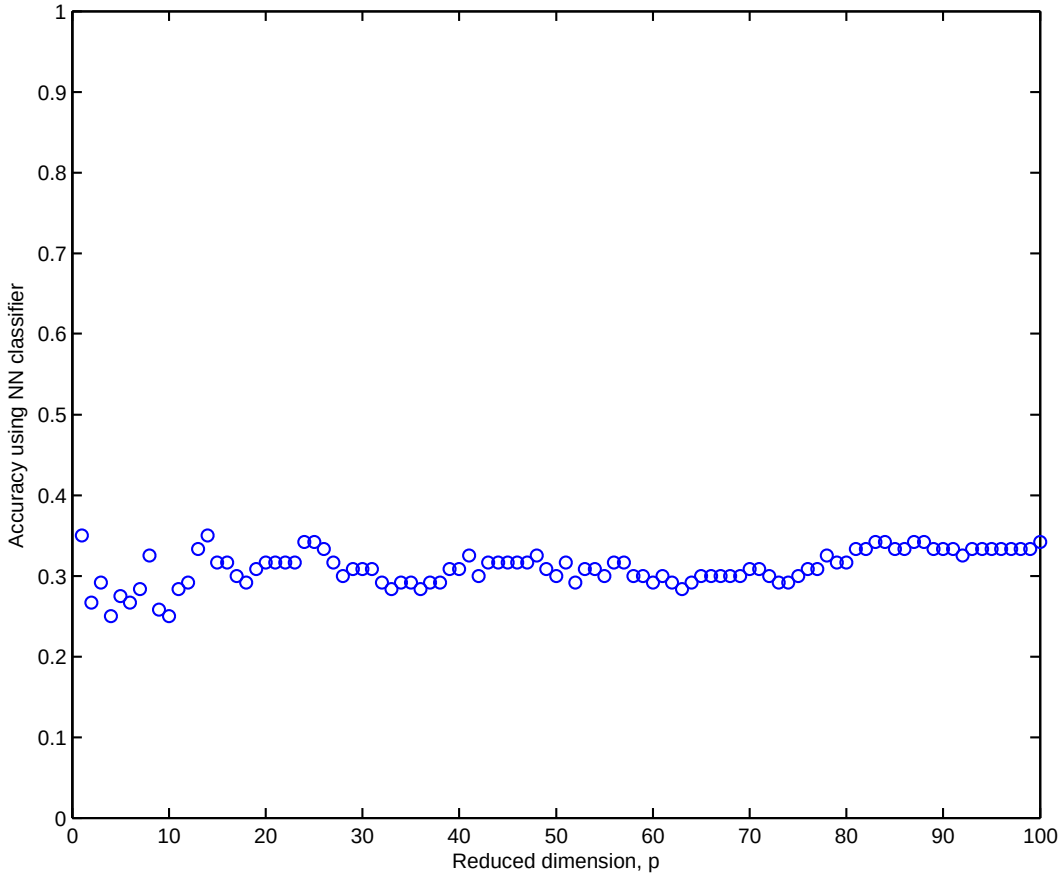

Figure 2.2: NN classification result upon PCA algorithm

Chapter 3

Non-Negative Matrix Factorization

A big shortcoming of using PCA for pattern recognition problem is that their basis vectors are not physically intuitive for many applications, such as face recognition. In PCA, subtracting the basis vector does not make sense in context of some applications. Here, Non-negative Matrix Factorization (NMF) is invented to overcome the issue. NMF is similar to the PCA except that the coefficients in the linear combination must not be a negative number. Due to this property, it leads to forming nice basis vectors that are usually representing some physical parts of the subjects. Here in this project, NMF algorithm is applied to the saliency map database, to obtained the basis for the saliency map.

3.1 NMF Algorithm

Most of the NMF algorithm targets to minimize

$$J = ||V - WH||^2, \tag{3.1}$$

where V is the sample data matrix and W, H are constrained in such a way that all their elements are non-negative.

Among all the NMF calculation methods, multiplicative update rule is a good compromise between speed and ease of implementation [2]. It is already proven that $||V - WH||^2$ is non-increasing under the multiplicative update rule as shown below:

$$\begin{aligned} H_{a\mu} &\leftarrow H_{a\mu}\frac{(W^TV)_{a\mu}}{(W^TWH)_{a\mu}} \\ W_{ia} &\leftarrow W_{ia}\frac{(VH^T)_{ia}}{(WHH^T)_{ia}} \end{aligned} \tag{3.2}$$

By this update rule, if W and H are initialized as matrices with non-negative entries, then it is guaranteed that W and H will remain non-negative because of the multiplicative nature.

3.2 Convergence of J

One important issue while implementing the NMF algorithm is the initialization of each matrix W and H. Here one must taken note that the dimension of W is given by $d \times p$, and dimension of H is given by $p \times n$, where d is the feature dimension, p is the number of desired bases, and n is the number of sample. In this project, they are

$$\begin{aligned} d &= 6912, \\ p &= 50, \\ n &= 400. \end{aligned}$$

By randomly assigned the initial condition for W and H with value between 0 and 1, the NMF algorithm is run with 500 iteration. Fig. 3.1 shows the optimization error J against the number of iteration run. It can be seen that the error converged to a small value after about 100 iterations. In other words, upon 100 and above iterations, the basis of NMF will not have significant changes.

One must take note that, in the saliency map images, most of the pixel are in black color, corresponding to 0 value in grayscale. This will cause a 'divide by zero' problem when updating the matrix W and H as W will be approximated zero. To overcome this problem, one possible way is to add a matrix with all element 1 to the feature matrix, such that W constructed later will not be near zero. It can be implemented easily in MATLAB.

3.3 Bases of NMF

NMF through multiplicative updating rule only guaranteed to converge to local optima, and since the initialization of W and H are random, different resulting bases will be expected for different initializations. In this section, 3 different runs are initiated, and their corresponding 50 bases are shown in Fig. 3.2, Fig. 3.3 and Fig. 3.4. This result has verified that NMF only converges to local optima, as the 50 bases of these 3 cases show different result. The outcome is very sensitive towards initializations.

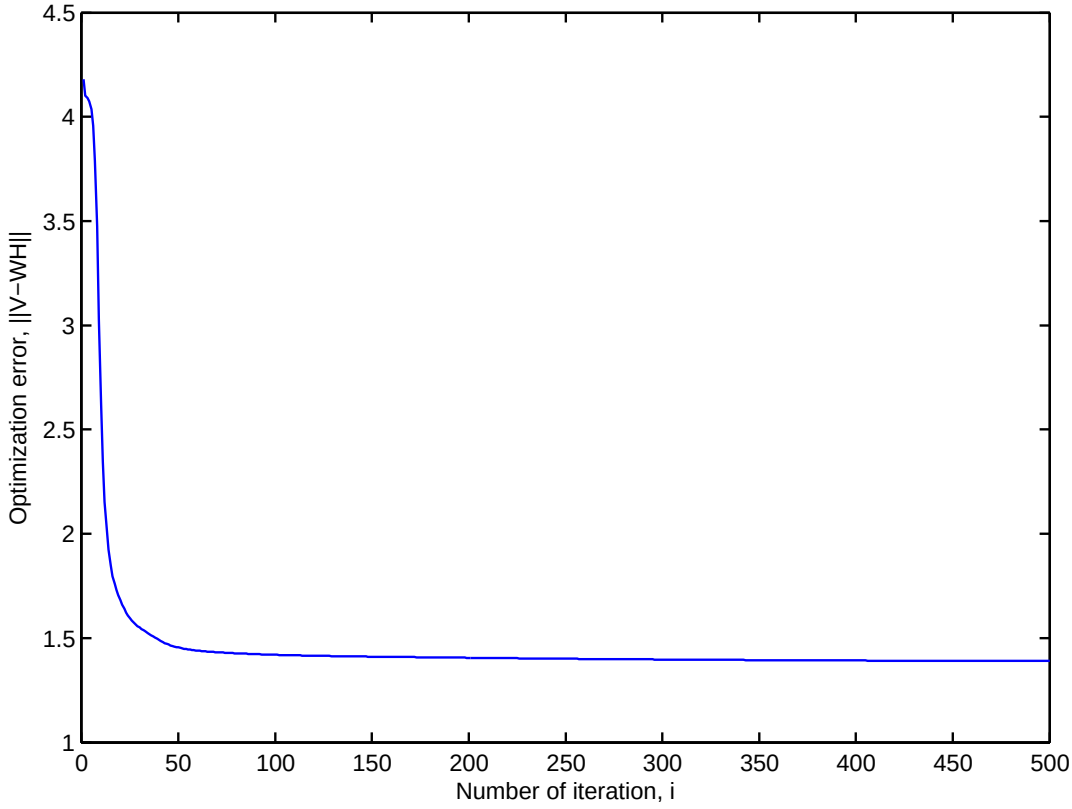

Figure 3.1: Optimization error against iteration number

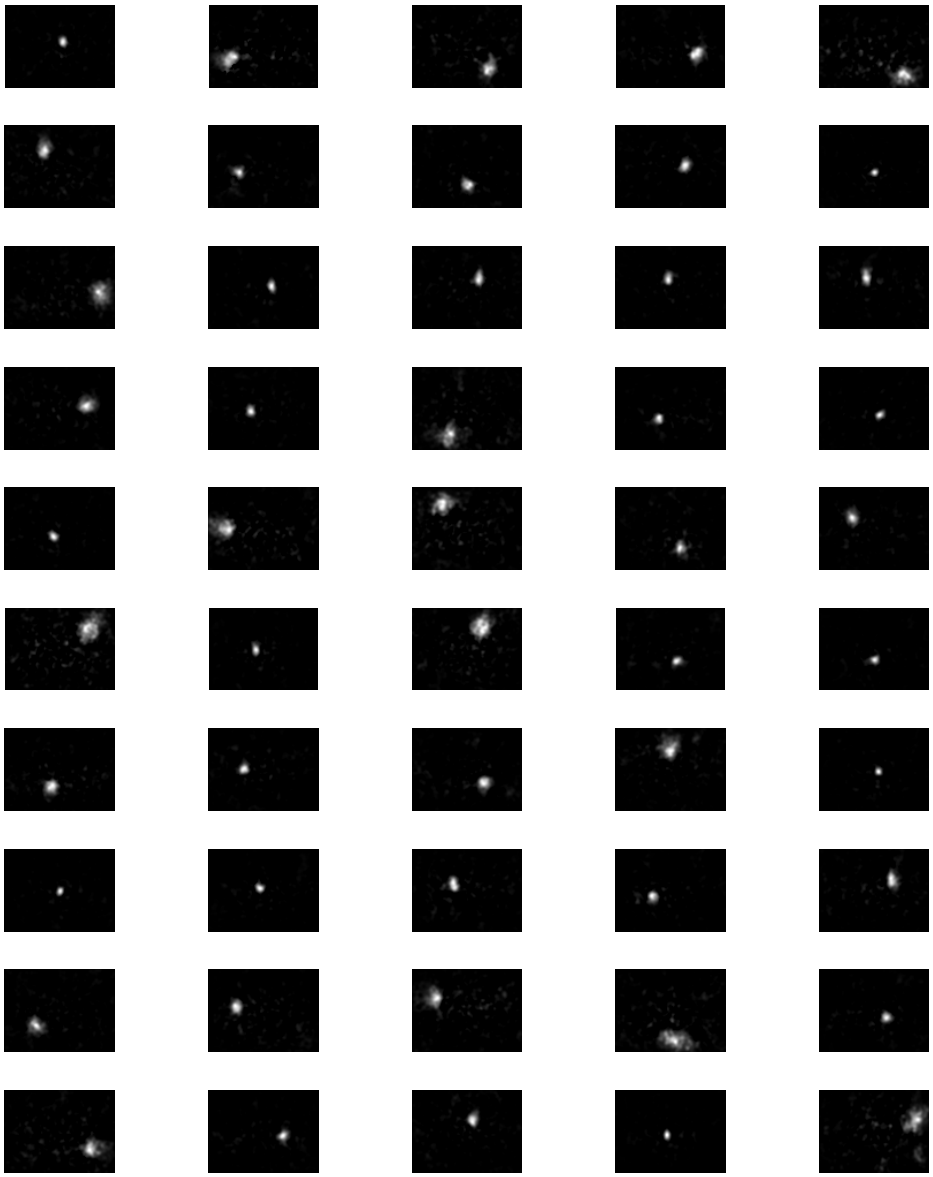

Figure 3.2: 50 NMF basis for 1st run

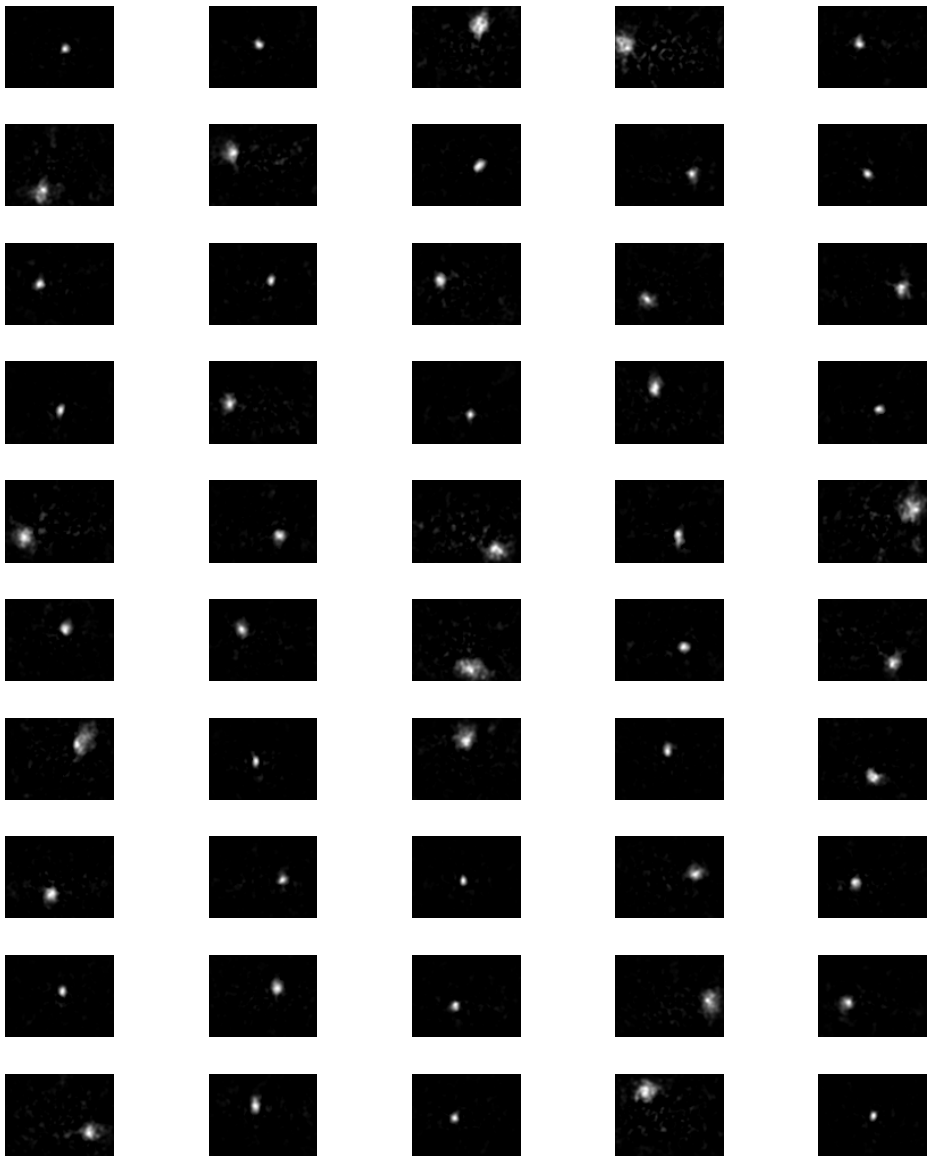

Figure 3.3: 50 NMF basis for 2nd run

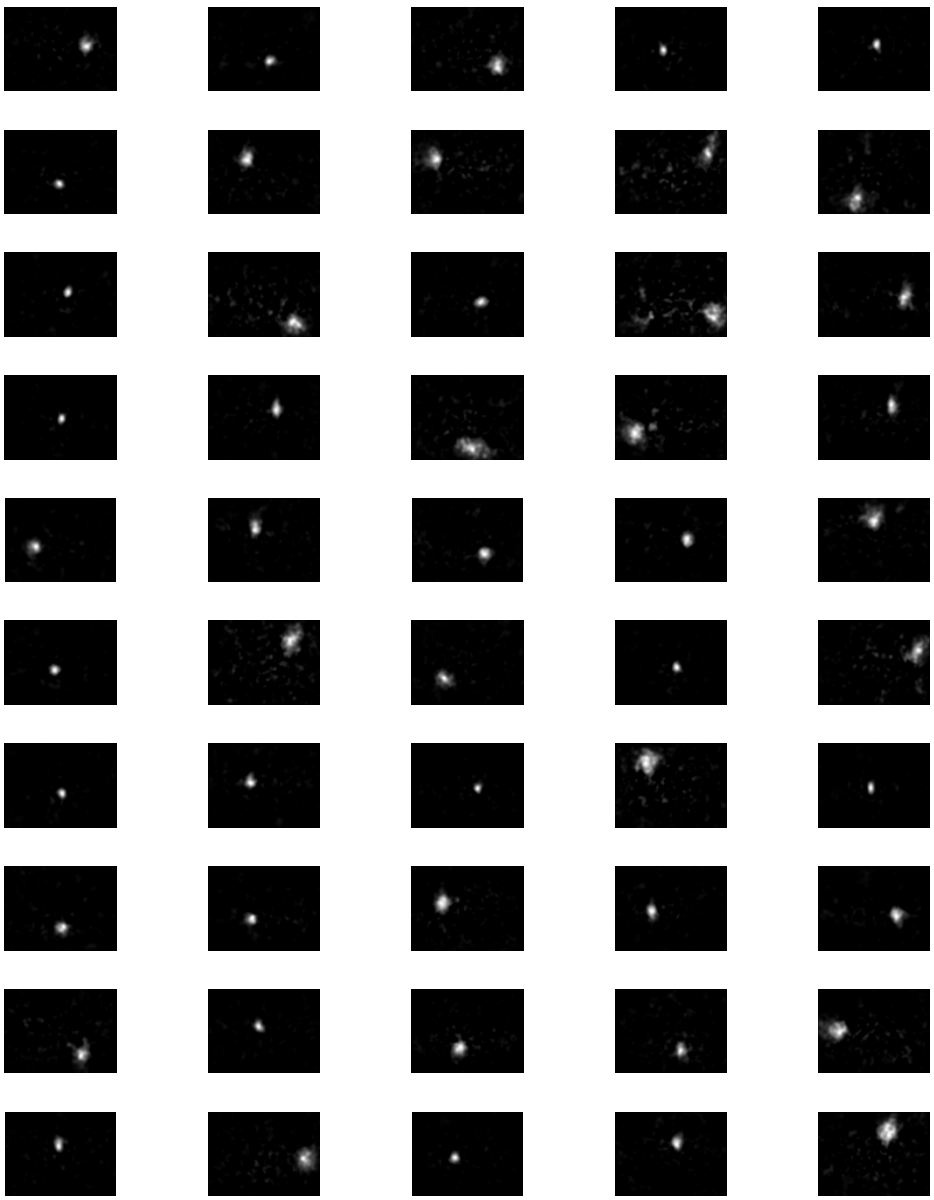

Figure 3.4: 50 NMF basis for 3rd run

Chapter 4

Linear Discriminant Analysis

As mentioned in previous chapters, PCA is an unsupervised feature extraction method, which does not take in feature information. In this chapter, Linear Discriminant Analysis (LDA) is presented as a supervised feature extraction method. LDA aims to find the best projection directions in which the between-class scatter S_B is maximized while the within-class scatter S_W is minimized. Physically, it helps to group the similar feature together while finding a projection to reduce the feature dimension. The definitions of S_B and S_W are as follows:

$$S_B = \sum_{i=1}^{C} P_i(m_i - m)(m_i - m)^T \tag{4.1}$$

$$S_W = \sum_{i=1}^{C} P_i S_i \tag{4.2}$$

where $P_i = n_i/N$ is an estimate of the prior probability for class i, m_i is the class mean vector, m is the total mean vector and S_i is the class covariance. The projection directions calculated by LDA will have maximal discriminant power. In other words, data after such projections can be best separated. Hence, LDA is widely used in data classification and more commonly, dimensional reduction catering to classification.

4.1 Computation of LDA

There are two common methods to compute LDA. The first method is to

1. Compute the eigenvalue decomposition of $S_W^{-1}S_B = U^{-1}\Lambda U$.
2. The first p columns in U (first p eigen vectors of $S_W^{-1}S_B$) form the Fisher vectors.

The second method is to

1. Solve the Generalized Eigenvalue Decomposition problem, $S_B\theta_i = \lambda_i S_W\theta_i$.
2. θ_i for $i = 1, 2, 3, ..., p$ are the first p Fisher vectors.

For the ease of implementation, the first method is utilized and realized in MATLAB. The new reduced-dimension feature will then be

$$y_i = (U_{d,p})^T(x_i) \tag{4.3}$$

A well-known problem of LDA is the computational cost. In this section of the project, the original dimension of the images are 6912 pixels, with only 400 samples. Also, since there are much lesser sample than the feature dimension, the constructed scatters matrix will be singular. To solve these problem, we first reduce the dimension of the feature to 110 by utilizing PCA as discussed in previous chapter, then only LDA algorithm is applied to the reduced-dimension feature. In this way, suppose the transformation matrix of PCA and LDA are $(P_{d,110})^T$ and $(W_{110,p})^T$ respectively, then the overall transformation matrix will be

$$(\Theta_{d,p})^T = (P_{d,110}W_{110,p})^T. \tag{4.4}$$

4.2 Fisherfaces Display

Similar to the eigenfaces of PCA algorithm, the columns of $\Theta_{d,p}$ form the fisherfaces. Fig. 4.1 shows the first ten fisherfaces of the LDA algorithm, corresponding to the largest 10 eigenvalues of $S_W^{-1}S_B$. Unlike the eigenfaces shown in the PCA chapter, fisherfaces are much less like human faces. Noted that since the class number is 4, only 3 eigenvalues will be positive. From the 4th eigenvalue onwards, complex eigenvalues and eigenvectors might be obtained. In order to show the next 7 fisher faces, only the real values of the eigenvectors is used.

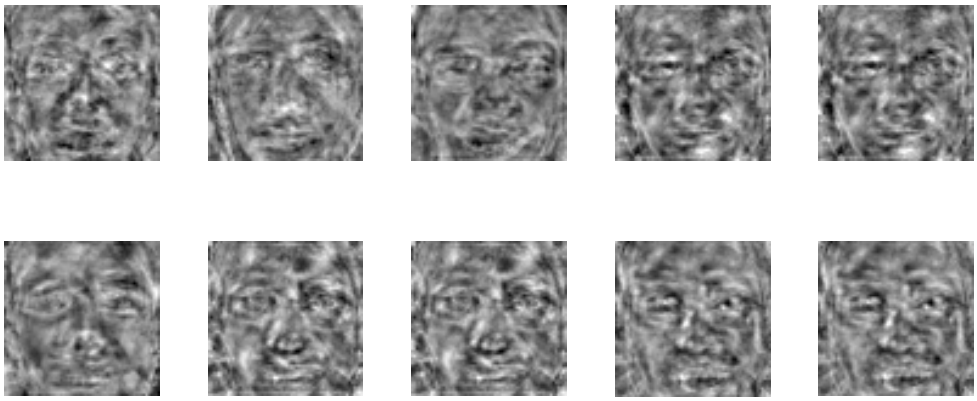

Figure 4.1: First 10 fisherfaces

4.3 NN Classification

Similar to what has been shown in the PCA chapter, NN classification will be utilized upon the LDA feature extraction. The classification result from $p = 1$ to $p = 100$ is plotted in

Fig. 4.2.

From the result plotted, we find the accuracies increases to about 40% as compared to the PCA algorithm. Also, we observe that the accuracies drop rapidly after the first few p. Similar to what have mentioned in the previous section, the maximum dimension of the reduced-dimension feature is equal to the number of class minus 1. In this case, $p = 3$ should be the optimum point for LDA feature extraction. This result is more or less reflected in the simulation where the best classifying result is obtained near to $p = 3$. In overall, LDA feature extraction has better performance than the PCA feature extration. Comparing with PCA, LDA has the advantages of utilizing label information as well as a more reasonable optimization target for classification, by maximizing between-class scatter and minimizing within-class scatter. PCA only maximizes the overall data variation and does not discriminate class differences. This is the main shortcoming of PCA, but LDA is introduced to solve this issue.

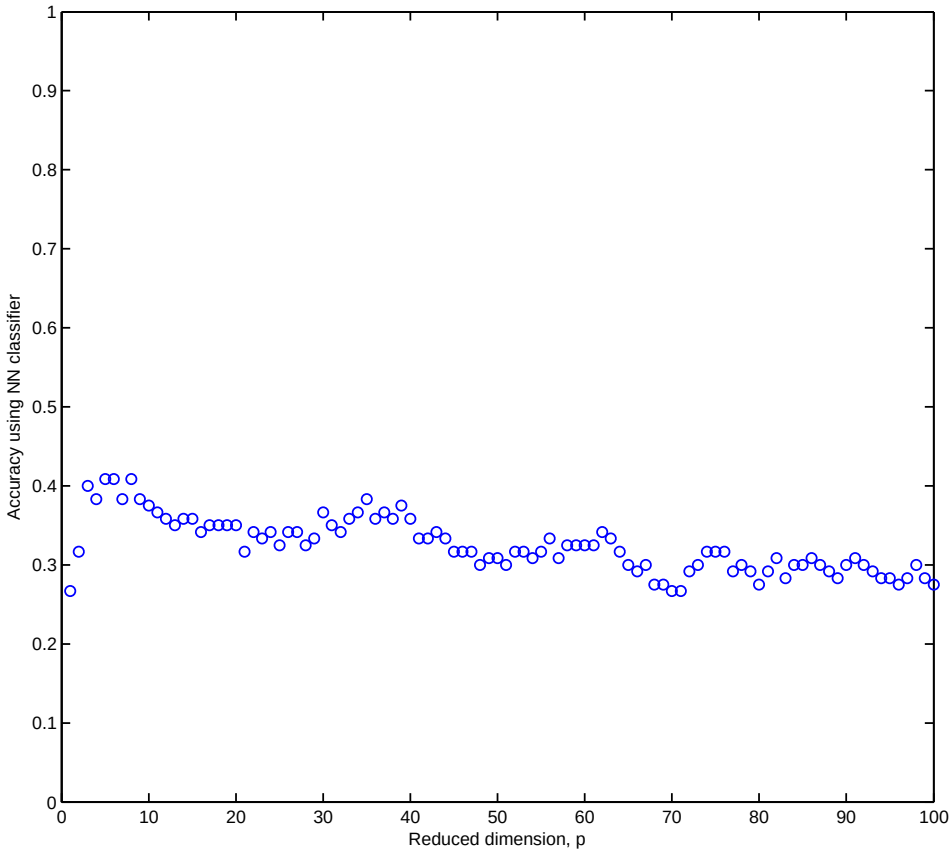

Figure 4.2: Accuracies of NN classifier upon LDA

Chapter 5

Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. It is a type of generative models. GMM parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model. In this project, the EM algorithm will be used to train the GMM for the saliency map problem. To model the saliency map using the GMM structure, a few assumptions are made [1]:

1. There are 8 components, $\omega_1, \omega_2, ..., \omega_8$
2. Component ω_j has an associated mean vector μ_j
3. Each component generates data from a Gaussian model with mean, μ_j, and covariance matrix, Σ_j
4. Assume each data point is generated according to the following recipe:
   (a) Pick a component at random based on the priors, $P(\omega_j)$
   (b) The data point is generated according to Gaussian distribution, $N(\mu_j, \Sigma_j)$

5.1 Expectation-Maximization algorithm

Expectation-maximization (EM) is a method for finding maximum likelihood estimate of parameters in statistical model, where the model depends on the unobserved latent variables. In summary, EM is an iterative method which alternates between performing an expectation (E) step and a maximization (M) step, with

1. E-step computes the expectation of the log-likelihood evaluated using the current estimate for the latent variables
2. M-step computes parameters maximizing the expected log-likelihood from the E-step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E-step

For a general GMM model, the component means μ_j, component covariance matrices Σ_j and component priors P_j are all dynamic, which means all of them will be updated for every EM iteration as shown above. The hidden variables z_i^j carries the belonging information to each class.

Based on the lecture notes in [1], to find the maximum likelihood (ML) estimates for μ_j, Σ_j and P_j provided that z_i^j are known, we have parameter estimation results expressed as:

$$\mu_j(t+1) = \frac{\sum_i z_i^j x_i}{\sum_i z_i^j} \tag{5.1}$$

$$\Sigma_j(t+1) = \frac{\sum_i z_i^j [x_i - \mu_j(t+1)][x_i - \mu_j(t+1)]^T}{\sum_i z_i^j} \tag{5.2}$$

$$P_j(t+1) = \frac{\sum_i z_i^j}{N} \tag{5.3}$$

On the other hand, if μ_j, Σ_j and P_j are known, expectation of z_i^j can be obtained as:

$$E[z_i^j] = \frac{\left((2\pi)^{d/2}|\Sigma_j|^{1/2}\right)^{-1} \exp\left(-\frac{1}{2}(x_i-\mu_j)^T\Sigma_j^{-1}(x_i-\mu_j)\right) P_j}{\sum_{l=1}^{8}\left((2\pi)^{d/2}|\Sigma_l|^{1/2}\right)^{-1} \exp\left(-\frac{1}{2}(x_i-\mu_l)^T\Sigma_l^{-1}(x_i-\mu_l)\right) P_l} \tag{5.4}$$

There are several implementation problems when using EM for GMM. One is its high computational cost. When then dimensionality d of the data vector is high, the algorithm will take a long time to iterate, with very high dimension data need to be computed. A reasonable solution is to use PCA to extract a set of reduced-dimensional features from the data before carrying out GMM. In this section, the reduced dimension by PCA is set as 40 which is a good compromise between accuracy and computational time. Also, it solves the singular problem similar to previous chapter.

Next is the initialization of the GMM algorithm. In an ideal case, the centroid of the initialization should be in each of the class. If the initialization of μ_j, Σ_j and P_j are not reasonable enough, the outcome maybe undesirable. For this project, μ_j is initialized by first running K-means clustering, to obtain an initial centroid from each class.

5.2 Simulation Results

Simulation result is shown in Fig. 5.1. The figure shows the 8 centers/means of the saliency map after 500 iteration runs on the EM algorithm. To check whether the solutions are global optima, a second simulation is run again. The result is shown in Fig. 5.2. It seems that the 8 centers are the same, only with different position shown in the figure. It shows that after 500 iteration, the result are almost converged to the optimum point.

The results shown the density of the 8 different position of human eyes observing the total 400 pictures given in the experiment. It is seems that most of the eye positions are fixed to the center or near center of the images, which is quite reasonable as we usually look

at the center of the images for the view of whole picture. Also, notice that a few means shows high density around 1/3 of the picture position horizontally and vertically. They are corresponding to the golden ratio of the images. It is commonly known to professional photographers that, by placing the object at the golden ratio (about 1/3) horizontally or vertically, it will be more appealing to the people who see the images. I think it somehow proven in this experiment.

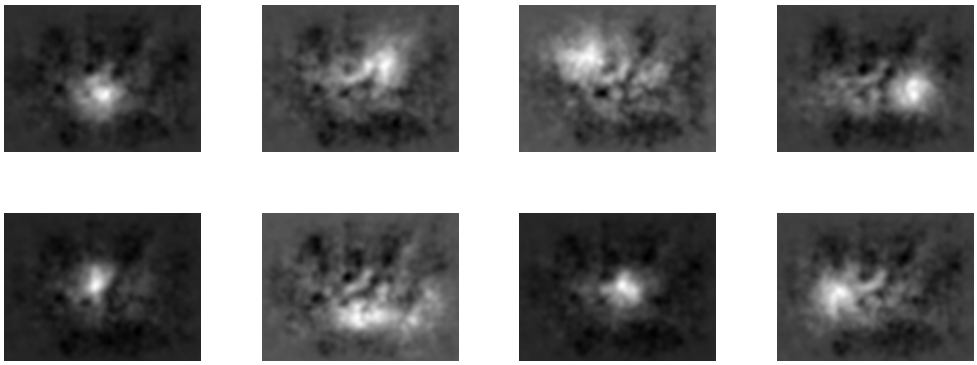

Figure 5.1: 8 centers by EM for GMM after 500 iterations

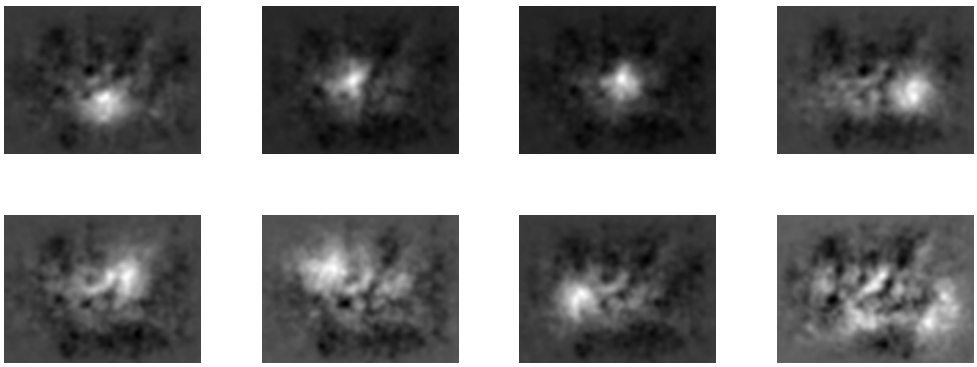

Figure 5.2: 8 centers by EM for GMM after 500 iterations 2nd run

Chapter 6

Conclusion

In this project, two interesting pattern recognition problems was investigated. First is the classification of human faces according to ethnic group, and second is the analysis on the data obtained based on the eye fixation position in an experiment. The main classification method that was used is the Nearest Neighbor (NN) classifier. Important feature is extracted by several feature extraction method, such as the Principal Component Analysis (PCA), the Nonnegative Matrix Factorization (NMF), and the Linear Discriminant Analysis (LDA). Also, the general Gaussian Mixture Model (GMM) is studied to obtain the distribution properties of the database.

Based on the results, it has proven that LDA extraction method can perform better than PCA extraction method since it is a supervised feature extraction. NMF algorithm are found to be only converge to local optima, with high sensitivity towards initial condition. Lastly, GMM algorithm using EM method found to be very powerful in obtaining the means of the data.

Bibliography

[1] Y. Sun and S. Yan, *Lecture notes, EE5907R Pattern Recognition*. 2011.

[2] D. Lee and S. Seung, "Algorithms for non-negative matrix factorizationl,"