Soft Computing:ISCO630E
Report

Assignment 6.1

Anant Chaturvedi(IIT2016506)

Question Description

1. Face recognition using PCA
    i. Hints:

    Create face dataset using your mobile phone for your face as well as faces of 9 other friends.Create multiple variants (at least 5) of each faces with different view angles.

Introduction

We have to create a dataset comprising of Images of 9 people taken from 7 different angle.Now given a new Image we have to predict the person of whom the image is taken of.

RESULTS

75% of the data was used for training while the remaining was used for testing the predictor.

The Accuracy on the Training data is around :- 77.79%

The Accuracy on the Testing data is around :- 77.79%

Because there is a random distribution of training and testing data upon running the program every time thus results may differ from time to time.

Concepts Used

Steps involved in training:

1. Generate the face database:

   Each face image is represented in the form a matrix having m rows and n columns, where each pixel (x,y) such that x∈m, and y∈n shows pixel location of the image as well as the direction.

   For the simplicity we are assuming each face image as a column vector, if we have p images then the size of the face database will be mn*p.

   Let's say face database is denoted as $(\text{Face\_Db})_{mn*p}$

2. Mean Calculation:

   Calculate the mean of each observation

   $$M_i = \sum_{i=1}^{mn} \sum_{j=1}^{p} Face\_Db(i,j)$$

   here mean vector will have the dimension $(M)_{mn*1}$

3. Do mean Zero

   Subtract mean face from each face image, let's say this mean zero face data as Δ, and calculated as

   $$(\Delta\ (i))_{mn*p} = (\text{Face\_Db}(i))_{mn*p} - (M)_{mn*1}$$

   Where i∈1,2,3 ……., p.

4. Calculate Co-Variance of the Mean aligned faces (Δ)

   Here there is a slightly variation in calculation of covariance, generally we prefer to calculate the covariance of data by:

   $$C = \sum_{i=1}^{n} (X_i - \overline{X})\ ((Y_i - \overline{Y}))^t$$

Where $\overline{X}, \overline{Y}$ *are the mean of* X_i *and* Y_i, and C is the covariance matrix. If we will follow the same convention on face data we will get.

$$C\,(mn,mn) = \sum_{z=1}^{mn} \sum_{y=1}^{mn} \sum_{i=1}^{p} (\Delta(z,i) - M_z) * (\Delta(z,i) - M_{y)^t}$$

Here we will get mn direction, which is very hard to compute, store and process. It also increases the program complexity, hence in 1991 Turk and Peterland [1] two researches suggested a new way to calculate the co-variance that is basically known as surrogate covariance, that is:

$$C\,(p,p) = \sum_{z=1}^{mn} \sum_{y=1}^{mn} \sum_{i=1}^{p} (\Delta(z,i) - M_{y)^t} * \ (\Delta(z,i) - M_z)$$

Hence here will get only p * p dimension, which is easy to compute and process, the idea behind computing the surrogate covariance suggested by turk and peterland that, these are only the valid direction where we will get maximum variances, and rest of the directions are insignificant to us. Menas these are direction where we will get the eigenvalues and for rest we will get eigenvalues equal to zero.

5. Do eigenvalue and eigenvector decomposition:

   Now we have covariance matrix $(C)_{p*p}$, *find out the eigenvectors and eigenvalues.*
   Let we have eigenvector $(V)_{p*p}$ *and eigenvalues* $(\lambda)_{p*p}$.

6. Find the best direction (Generation of feature vectors)

   Now select the best direction from p directions, for this sort the eigenvalues in the descending order and decide a k value, which represents the number of selected eigenvectors to extract k direction from all p direction. On the basis of k value we can generate the Feature vector $(\Psi)_{p*k}$.

7. Generating Eigenfaces:

   For generating the eigenfaces (Φ) project the each mean aligned face to the generated feature vector.

$$(\Phi)_{k*mn} = (\Psi)^{t}{}_{k*p} * (\Delta)^{t}{}_{p*mn}$$

8. Generate Signature of Each Face:

   For generating signature of each face (ω), project each mean aligned face to the eigenfaces.

$$(\omega)_{k*i} = (\Phi)_{k*mn} * (\Delta)_{mn*i}$$

Where $i \in 1, 2, 3, \ldots, p$. hence ω will have the size k * p.