

Assignment : IEC20171047

(1) PCA (2) LDA

QUESTION DESCRIPTIONS

(1) Face recognition using PCA : Create face dataset using your mobile phone for your face as well as faces of your friends.. Create multiple variants (at least 5) of each faces with different view angles.

(1) Face recognition using LDA : Create face dataset using your mobile phone for your face as well as faces of your friends. Create multiple variants (at least 5) of each faces with different view angles

[QUESTION 1]

INTRODUCTION

We have to create a dataset comprising of Images of 16 people taken from different angles. Now given a new Image we have to predict the person of whom the image is taken of using PCA.

RESULTS

The dataset was divided into 80% training and 20% testing.

Accuracy of the transformed testing data using Principal Component Analysis: 80.0% => The results can vary because of the random distribution of training and testing data.

```

In [105]: correct=0
          for i in range(X_test.shape[0]):
              if predictions[i]==Y_test[i]:
                  correct+=1
          print("correct Predictions: " , (correct) , " Out of: " , (X_test.shape[0]))
          print("percentage accuracy: " , (correct / X_test.shape[0]) * 100)

correct Predictions: 16 Out of: 20
percentage accuracy: 80.0

```

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 $x \in m$, and $y \in n$ 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 $mn \times p$.

Let's say face database is denoted as $(Face_Db)_{mn \times p}$

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 $(M)_{mn \times 1}$

3. Do mean Zero

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

$$(\Delta(i))_{mn \times p} = (Face_Db(i))_{mn \times p} - (M)_{mn \times 1}$$

Where $i \in 1, 2, 3, \dots, p$.

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:

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

Where \bar{X}, \bar{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.

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

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:

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

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. Means 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 $(C)_{p \times p}$, find out the eigenvectors and eigenvalues.

Let we have eigenvector $(V)_{p \times p}$ and eigenvalues $(\lambda)_{p \times p}$.

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 $(\Psi)_{p \times k}$.

7. Generating Eigenfaces:

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

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

8. Generate Signature of Each Face:

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

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

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

[QUESTION 2]

INTRODUCTION

We have to create a dataset comprising of Images of 16 people taken from different angles. Now given a new Image we have to predict the person of whom the image is taken of using LDA.

RESULTS

The dataset was divided into 80% training and 20% testing.

Accuracy of the transformed testing data using LDA: 90.0% The results may vary because of the random distribution of training and testing data.

```
In [174]: correct=0
for i in range(X_test.shape[0]):
    if predictions[i]==Y_test[i]:
        correct+=1
print("correct Predictions: " ,(correct) , " Out of: ",(X_test.shape[0]))
print("percentage accuracy: ", (correct / X_test.shape[0]) * 100)

correct Predictions:  18 Out of:  20
percentage accuracy:  90.0
```

CONCEPTS USED

Training Steps:

1. Apply PCA on the given data and make a database of projected faces $(PF)_{k \times p}$, where k is the number of selected principal components and p is the training population.
2. Divide the data into class like if each person have n images then, $number_of_classes = P/n$;
3. Calculate the means of each class $(\mu_i)_{k \times 1}$ and mean of the Projected faces $(M)_{k \times 1}$
4. Calculate the within class scatter matrix $(SW)_{k \times k}$, and between class scatter matrix $(SB)_{k \times k}$ discussed in equation (1) and (4).
5. Use the criterion function suggested in equation (5).
6. Find the Eigen vector and Eigen values of the Criterion function.
7. Now we need to select the best principal components from there, we can select m best values based on the maximum Eigen values.
8. Construct feature $(W)_{k \times m}$ vectors of using these k bests.
9. Generate the fisher faces (FF) by projecting the projected faces by this transformation matrix W .

$$FF_{m \times p} = (W^t)_{m \times k} * (PF)_{k \times p}$$