Assignment 3

Face Recognition using PCA

Problem Statement

We intend to perform face recognition. Face recognition means that for a given image you can tell the subject id. Our database of subject is very simple. It has 40 subjects. Below we will show the needed steps to achieve the goal of the assignment.

1. Download the Dataset and Understand the Format

a. ORL dataset is available at the following link.
https://www.kaggle.com/kasikrit/att-database-of-faces/

b. The dataset has 10 images per 40 subjects. Every image is a grayscale image of size 92x112.

2. Generate the Data Matrix and the Label vector

a. Convert every image into a vector of 10304 values corresponding to the image size.

b. Stack the 400 vectors into a single Data Matrix D and generate the label vector y. The labels are integers from 1:40 corresponding to the subject id.

3. Split the Dataset into Training and Test sets

a. From the Data Matrix D400x10304 keep the odd rows for training and the even rows for testing. This will give you 5 instances per person for training and 5 instances per person for testing.

b. Split the labels vector accordingly.

4. Classification using PCA

a. Use the pseudo code below for computing the projection matrix U. Define the alpha = {0.8,0.85,0.9,0.95}

b. Project the training set, and test sets separately using the same projection matrix.

c. Use a simple classifier (first Nearest Neighbor to determine the class labels).

d. Report Accuracy for every value of alpha separately.

e. Can you find a relation between alpha and classification accuracy?

ALGORITHM 7.1. Principal Component Analysis

PCA ($\mathbf{D}, \alpha$):

$\mu = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$ // compute mean

$\mathbf{Z} = \mathbf{D} - \mathbf{1}\cdot\mu^T$ // center the data

$\mathbf{\Sigma} = \frac{1}{n}(\mathbf{Z}^T\mathbf{Z})$ // compute covariance matrix

$(\lambda_1, \lambda_2, \ldots, \lambda_d) = \text{eigenvalues}(\mathbf{\Sigma})$ // compute eigenvalues

$\mathbf{U} = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_d) = \text{eigenvectors}(\mathbf{\Sigma})$ // compute eigenvectors

$f(r) = \frac{\sum_{i=1}^{r}\lambda_i}{\sum_{i=1}^{d}\lambda_i}$, for all $r = 1, 2, \ldots, d$ // fraction of total variance

Choose smallest r so that $f(r) \geq \alpha$ // choose dimensionality

$\mathbf{U}_r = (\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_r)$ // reduced basis

$\mathbf{A} = \{\mathbf{a}_i \mid \mathbf{a}_i = \mathbf{U}_r^T\mathbf{x}_i, \text{for } i = 1, \ldots, n\}$ // reduced dimensionality data

5. Bonus

1.Use different Training and Test splits. Change the number of instances per subject to be 7 and keep 3 instances per subject for testing. compare the results you have with the ones you got earlier with 50% split.

2. Report is done using latex

Note: Each bonus is worth one mark

6. Deliverables

You are required to deliver the following:

- Your code.
- Report

7 Notes

- You should write your code in python.
- You should work with groups of 3.