**Pattern Recognition**

**Assignment #1**

**Facial-recognition**

**Name:**  Mohamed Ibrahim El-Metwaly

**ID:** 3804

**Name:** Mohamed El-Sayed Abdelrehim

**ID:** 4055

Linear Discriminant Analysis and K-Nearest Neighbors for Facial Recognition

Data Preprocessing

Four hundred images of faces for forty different people were loaded into a data matrix with the first column of that matrix representing the labels for images that were unfolded such that each row represents an image. Then we loaded the data matrix into a python dictionary to for easy access of images of each label. After that we split the data into a training set and a testing set.

 We first performed the LDA + K-Nearest Neighbors algorithm on a scaled training data one time, and on the training data as it is the other time. There was an improvement in the accuracy of the predicted labels when we skipped the scaling, namely 92% without scaling and 89.5% with scaling.

Calculations

To calculate the between-class matrix (Sb), the total mean of the data and the mean of the data specific to each label were subtracted from each other. Then the Sbfor each label was calculated by multiplying that difference by its transpose then by the number of samples in having that label and calculating the sum of that on all labels.

For the within-class matrix (Sw) the mean specific to each label the data having that label were subtracted from each other, that difference was multiplied by its transpose and then the sum of that over all labels is the within-class matrix.

The best new space on which to project the data for maximum separation between classes is found by calculating the dominant eigenvectors of

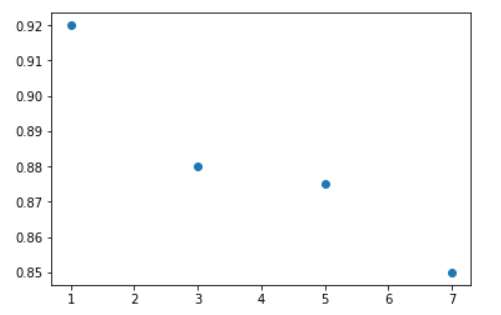
(Sw−1 .  Sb)

, but finding the inverse of the within-class matrix was somewhat tricky because we couldn’t be sure whether it’s invertible or not so we used the pseudoinverse which is the same as the inverse if the matrix is invertible and an approximation of the inverse if the matrix is non invertible.

Finally, the test data and the training data were projected using the same eigenvectors and the resulting datasets were given as an input to a K-Nearest Neighbors classifier

K-Nearest Neighbors Classifier

After tuning the classifier for multiple values of K we found that taking only one neighbor results in the best accuracy score (92%). Increasing the value of K significantly degraded the accuracy score.



Principal Component Analysis and K-Nearest Neighbors for Facial Recognition

Data Preprocessing

Four hundred images of faces for forty different people were loaded into a data matrix with the first column of that matrix representing the labels for images that were unfolded such that each row represents an image. Then we split the data into a training set and a testing set.

 We first performed the PCA + K-Nearest Neighbors algorithm on a scaled training data one time, and on the training data as it is the other time. There was an improvement in the accuracy of the predicted labels when we skipped the scaling.

Accuracy for every value of alpha separately

-without scaling

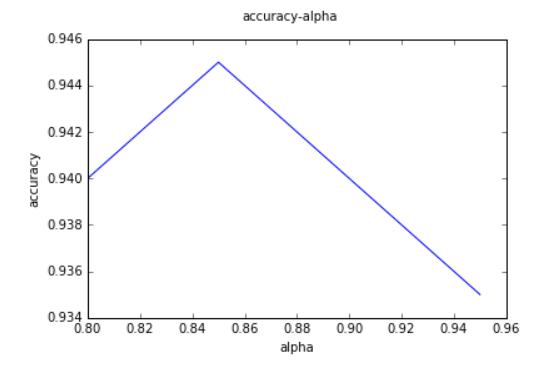
Alpha -> Accuracy

0.8 -> 94%

0.85 -> 94.5%

0.9 ->94%

0.95 ->93.5%



-with scaling

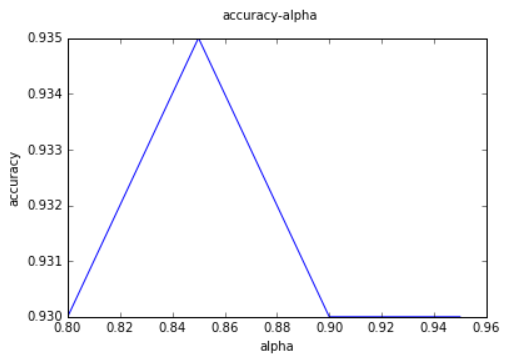
Alpha -> Accuracy

0.8 -> 93%

0.85 -> 93.5%

0.9 ->93%

0.95 ->93%



Calculations

First we get the mean of each column in the training set, this will give us a vector (10304,1), then we subtract this mean from the training set to get a dataset of zero mean ,After that we get the covariance matrix which is a symmetric matrix so to get the eigenvectrors and eigenvalues we will use the “ numpy.linalg.eigh() “.

Finally, the test data and the training data were projected using the same eigenvectors and the resulting datasets were given as an input to a K-Nearest Neighbors classifier  .

K-Nearest Neighbors Classifier

After tuning the classifier for multiple values of K we found that taking only one neighbor results in the best accuracy score (94.5%) note that we take alpha=0.85. Increasing the value of K significantly degraded the accuracy score.

