Face Recognition by using Linear Discriminant Analysis

Linear Discriminant Analysis is also a very powerful tool for features extraction as well dimensionality reduction. In LDA our main intension is to find the transformation matrix (W), by which we can project the data on such direction where they will be linearly separable. Transformation matrix depends on two things, which is basically the beauty of LDA (1) Within class scatter matrix (SW) and (2) Between class scatter matrix (SB). Within class scatter matrix tells about the how the data distributed within the class, more specifically we can say that how the data is distributed along the means, we can measure this by calculation the variance that is:

$$SW = \sum_{i=1}^{C} \sum_{i} \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Where C represents the number of classes, and $\sum$ represents the variance.

$$\sum_{i} = (V - \mu_i) * (V - \mu_i)^t \ldots\ldots\ldots (2)$$

Where μ is the mean of each class, and V is the population belongs to that class i.

While Between class scatter matrix tells about how the data is distributed between the classes. This can be calculated by

$$SB = \sum_{i=1}^{C} \sigma_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Where $\sigma_i = (\mu_i - M) * (\mu_i)^t \ldots\ldots\ldots\ldots (4)$

Here M is the overall mean, we can say that mean of all classes.

We have to select a criterion function (J) which maximizes the between class and minimize the within class scatter. This is because if the inter class (SB) difference is more then we can easily distinguish between two, while if the intra class (SW) difference is less that means the samples belongs to that class are more similar in nature.

$$J = (SW)^{-1} * SB \ldots\ldots\ldots\ldots (5)$$

The problem lies in this approach is, if the sample size is less than the dimension it will leads to singularity in within class matrix, and we cannot calculate the inverse of this. This problem is called the small sample size problem (SSS). For avoiding this problem we have to first use

the PCA (principal component analysis) to reduce the dimension and then we can use the LDA on the projected faces of PCA.

Training Steps:

1. Apply PCA on the given data and make a database of projected faces $(\mathrm{PF})_{k*P}$, 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 $(\mu_i)_{k*1}$and mean of the Projected faces $(\mathrm{M})_{k*1}$
4. Calculate the within class scatter matrix $(\mathrm{SW})_{k*k}$, and between class scatter matrix $(\mathrm{SB})_{k*k}$ 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 $(\mathrm{W})_{k*m}$ vectors of using these k bests.
9. Generate the fisher faces (FF) by projecting the projected faces by this transformation matrix W.

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

Testing Steps:

1. Read a test image and make it as a vector, we are making it column vector here because we are following that convention. Say $(\mathrm{Test\_img})_{mn*1}$
2. Do mean zero of the test image, say $(\mathrm{Mean\_img})_{mn*1}$
3. Calculate the projected Eigen Face (PEF) by Project it with the Eigen faces (EF) that we have constructed by using PCA.

$$\mathrm{PEF}_{k*1} = (\mathrm{EF})^t{}_{k*mn} * (\mathrm{Mean\_img})_{mn*1}$$

4. Now do the final projection (Projected Fisher Test Image) with the help of feature vectors $(W)_{k*m}$ to this Projected Eigen Face.

$$(\text{Projected\_Fisher\_Test\_Img})_{m*1} = (W)^{t}{}_{m*k} * (PEF)_{k*1}$$

5. At the end you have Projected_Fisher_Test_img , a column vector (m*1) and Fisher Faces (FF) having m*p dimension, find the distance between each column of the Fisher Faces to the Projected_Fisher_Test_img, you will get p distances, whatever the minimum assign that test image to that class.

Note: use 60 % face data for training and 40 % data for testing. Keeping that thing in mind that (Training set) ∩ (Testing Set) = ∅.

Assignment:

1. ***Camper the classification results of both PCA and LDA, and comment that, who suits better for face recognition.***
2. ***Apply both Euclidean and mahalanobis distance and evaluate the performance of both the techniques, also comment which distance better suits for face data.***