CS 4/791Y Mathematical Methods for Computer Vision

Fall 2002 - Dr. George Bebis

Programming Assignment 1

Due date: 10/22/2002

In this project, you will compare the Principal Components Analysis (PCA) technique with the Linear Discriminant Analysis (LDA) technique on a face recognition problem. You will be using the ORL face database in your experiments. This database contains 400 92x112 gray-scale images from 40 subjects (10 images per subject). The views are approximately upright frontal and there is variation in lighting, facial expression (open/closed eyes, smiling/not smiling) and glasses (dark homogeneous background). I have put the images in the following directory: */image/FaceImages/ORL/NormFaces* (this directory is mounted on all the SUN machines in the Computer Vision Laboratory). There are 40 subdirectories under this directory, one per subject. The images in this subdirectories are in "pgm" format and number from 1 to 10.

1. Create Training/Test Sets: once you have aligned the face images, you will need to divide them into three sets: training, test1, and test2. You will be using the training set images for computing the PCA and LDA components, no images from the test set should be used for this. The test images will be kept aside to be used for testing and comparing PCA with LDA. To create the training set, select all the odd-numbered images for subjects 1-35 (175 images). To create set test1, select all the even-numbered images for subjects 1-35 (175 images). To create set test2, select both the odd- and even-numbered images for subjects 36-40 (50 images).

2. Implement the PCA (eigenface) approach: read carefully and understand the steps of the eigenface algorithm. Your program should run in two modes: (i) training and (ii) testing. In training mode, your program needs to read in the training images and compute the average face and eigenfaces. Then, it needs to decide how many eigenfaces to keep (this could be done in an interactive mode with the user). It will then project each training image into the eigenspace and compute its representation in this space (i.e., the coefficients of projection Ω^k, k=1,2,..,M, where M is the number of training images). Finally, your program needs to store into a file the coefficients Ω^k, the average face, and the eigenfaces.

In testing mode, your program will read in the coefficients Ω^k, the average face, and the eigenfaces. It will then use the images from sets test1 and test2 to test the performance of the PCA approach. Using test1, we want to see how well PCA performs on people who have already their images in the training set (i.e., to find out if the method can tolerate changes to illumination, facial expression etc.) Using test2, we want to see how well PCA performs on people whose images are not stored in the database at all (i.e., intruders). Given a new image, your program needs to project it in the eigenspace and compute the coefficients of projection Ω. To recognize the face in the input image, you need to find the closet match Ω^k to Ω (let's call $e_k = \|\Omega^k - \Omega\|$). To be able to reject intruders, you would have to threshold e_k (i.e., accept the match only of $e_k < T$). You would have to determine the threshold experimentally (what would be a good choice for T? elaborate on this in your report).

Very Important: to make sure that your program works correctly, try the following: given an image I, (i) project it onto the space of eigenfaces, (ii) reconstruct it using all eigenfaces (let's call the reconstructed image $\hat{I}$), (iii) compute $I - \hat{I}$. This difference should be very small (ideally zero) or your code does not work correctly. Do not proceed unless you have been able to verify the above.

3. Implement the LDA (fisherface) approach: follow the same guidelines for implementing and testing the LDA approach.

4. Results and Comparisons: you need to report the results from each approach separately (both on sets test1 and test2). When evaluating face recognition systems, we are interested in reporting the following results: (i) *errors*, (ii) *false positives*, (iii) *false negatives*. Test cases from set test1 that were recognized incorrectly (i.e., min error e_k was less than the threshold T but face identity assigned is incorrect) will be referred to as *errors*. Test cases from set test1 that were not recognized at all (i.e., min error e_k was greater that the threshold T) will be referred to as *false negatives*. Finally, test cases from set test2 that were not rejected (i.e., min error e_k was less than the threshold T) will be referred to as *false positives*. Report these errors using percentages for easy comparisons. Tabulate your results and make it easy for the reader to understand them.

When comparing PCA with LDA, it would be interesting to see whether the two methods make the same errors (e.g., do both methods reject the same faces?) If they make different errors, you might be able to combine them, creating a hybrid approach with superior performance (**extra credit**). Another way to get **extra credit** is by implementing a nice user interface.

Hints: You need to turn in a report (hard copy or on-line) documenting your implementation, choice of parameters, results, and comparisons. Your grade will be determined by large based on the quality and organization of your report. Organize your report into different sections and include lots of visuals. Do not expect to get a good grade if you just throw in a few numbers and tables without actually discussing the results you have obtained.