Q1. Eigenfaces

1. The total data set is randomly partitioned into 80% training data and 20% testing data. 80% training data is large enough to train our models. 20% test data is enough to evaluate the performance of different models with different hyperparameters. However, this random split does not take into account the label of each face, which means that theoretically it could happen that a person has no faces left in the training set. Therefore, a better option is to split data randomly into training and testing subset within each class, and then combine the partitioned subsets of each class into a larger training and testing set separately. From the results, there are no obvious differences in model performance between these two split methods since the special case almost never happens. Another reason is that PCA as an unsupervised learning maximizes total projected data variance and LDA maximizes the ratio of projected between-class and within-class scatter. Therefore, the performance of these two methods do not depend on the number of faces of a particular person to train, which means that even if a person has only one face left in the training set, it does not put the models at a disadvantage for recognizing that person in the testing set.

The number of eigenvalues (eigenvectors) is the same as the number of pixels per image D (2576) because the covariance matrix S is D×D. The number of eigenvectors with non-zero eigenvalues equals the number of data in the training set minus one (N - 1 = 415). In PCA optimization, the projected data variance along each eigenvector equals the corresponding eigenvalues and this number should be maximized to achieve larger data variance after projection. Hence the total number of eigenvectors used for PCA (Mpca) should not exceed the number of non-zero eigenvalues (N -1), otherwise the projected variance is zero along those eigenvectors with zero eigenvalues. The larger Mpca, the more information of original data space is kept after projection and the results of classification or reconstruction should be better, but more memory is used and it will take more time for classification or reconstruction.

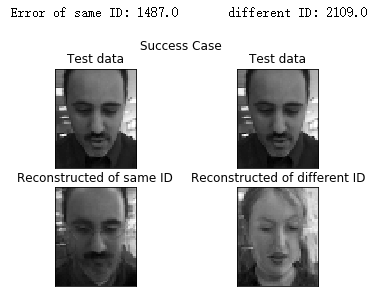
1. The eigenvalues computed using low-dimensional method are identical to those computed without low-dimensional method. However, the corresponding eigenvectors are not identical, but have the following relationship (u = Av). The computation cost (memory and time) of low-dim method is much smaller than the normal method if D>>N, because AAT is D×D, but ATA is only N×N. Moreover, although AAT is D×D, rank (AAT) is always N – 1 which means that it has no more than N - 1 nonzero eigenvalues, hence the computation of zero eigenvalues and their corresponding eigenvectors is a waste of time and memory for PCA. This can be overcome by using low-dim method which only gives N eigenvalues and among them N-1 are non-zero, hence saving time and memory. However, the computed eigenvectors using low-dim method cannot be used as eigenfaces directly, but should be multiplied by A and then normalized, hence extra computation effort is needed. In summary, if D>>N, low-dim method is much preferable but if D < N, low-dim method is not useful. The above results always hold when N is changed.

Q1. Applications of eigenfaces

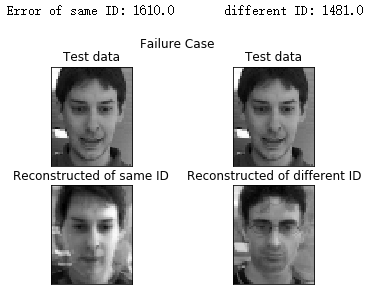
1. The reconstruction error of each faces is computed, and the average of these error become smaller if Mpca is larger. This is reasonable since more PCA bases learnt implies higher dimension of PCA subspace, hence more information of original data space is kept after projection.
2. Comparison between NN-classification and alternative method.

The accuracy of AM (or reconstruction error) does not change too much as the number of bases learnt per class changes. Even if AM chooses only one eigenvector to span each eigenspace (i.e. linear subspace), the accuracy of AM is still better than that of NN with the largest Mpca. The reason is that AM takes into account the labels of each face when training its model, making it more discriminative than NN. In other words, AM generates an eigenspace for each class, which will definitely capture the features of that particular person more accurately, than just project all training data from different identities onto the same eigenspace.

The success case below shows that AM recognizes the test data because it has smaller reconstruction error than the images reconstructed from all other different identities.



The failure case below shows that AM regards the test person as the one with different identity in the lower-right position because it has smaller reconstruction error than the image reconstructed from the same identity.



In summary, the reconstruction error can be used as a discriminative feature due to better accuracy than NN and reasonable computational cost. However, the performance of AM seems to be not related with its hyperparameter, which means that the best result is still not good enough compared with other discriminative methods like LDA. Moreover, since the number of images available to train a subspace is limited (no more than 10 in this coursework), the reconstruction error is relatively larger than NN, which can have smaller reconstruction error by increasing the number of bases learnt (up to N -1 = 415 bases available).