Machine Learning for Computer Vision Coursework 1

Hao Ding (CID: 00734091), Tanat Sanpaveeravong (CID: 00733970),
Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ
Email: hd1812@ic.ac.uk, ts3912@ic.ac.uk

PART 1: DATA PARTITION & PCA

The provided data set contains the faces of 52 people (classes) with 10 variations for each person. The data was partitioned based on the K-fold cross validation method where the number of folds chosen is 10. For each person, 1 out of 10 variations will be randomly selected as a test image while the rest are put into the training set. Therefore, for each fold, there will be 468 images in the training set and 52 images in the testing set. Such procedure is repeated 10 times so that there are 520 images in the testing set and thus, a reasonable evaluation of the algorithm's performance can be obtained.

K-fold cross-validation has the advantage of using all the available data set in both the training and testing procedure. This helps to maximise the model performance when the data is sparse. However, this approach has a considerably high computational complexity. As such, a trade-off between computation time and performance will need to be considered before choosing a method for data partition.

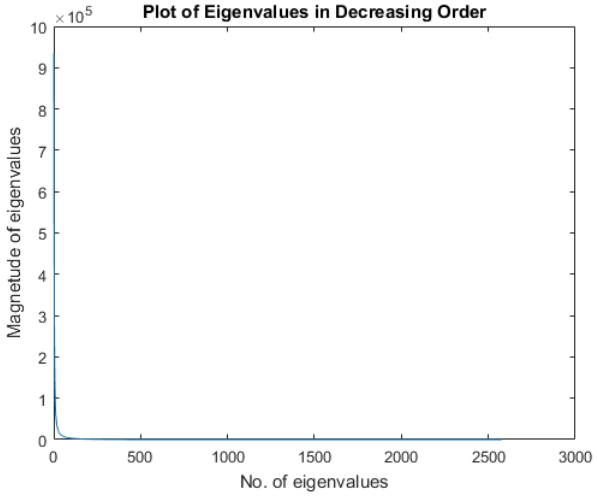

Fig. 1. Magnitude of eigenvalues plotted against No. of eigenvalues (for when $S = \frac{1}{N}AA^T$).

In order to apply Principal Component Analysis (PCA) to the data, the eigenvectors (principle components) and eigenvalues of the data covariance matrix have to be computed. This

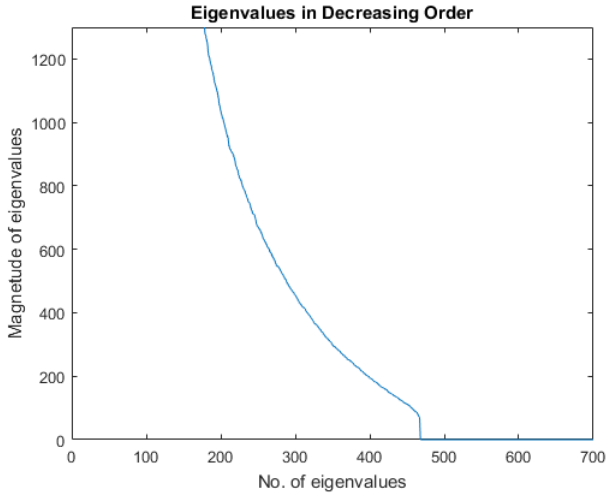

Fig. 2. A magnified version of Magnitude of eigenvalues plotted against No. of eigenvalues (for when $S = \frac{1}{N}AA^T$).

covariance matrix S is given by

$$S = \frac{1}{N}AA^T \tag{1}$$

where A is a matrix made up of all the normalised face vectors. The covariance matrix S has the dimension of 2576×2576. This means that the corresponding eigenvectors will have the dimension of 2576×1 and the total number of eigenvalues that can be obtained is 2576.

It is important to note that A is not a full-rank matrix. In this case, A has a rank of 468, which is the number of training data. By considering relationships given in (2), the rank of the covariance matrix S is 468.

$$rank(A^T) = rank(A) = rank(AA^T) = rank(A^TA) \tag{2}$$

From the perspective of linear algebra, the number of eigenvalues that can be solved for depends on the rank of the matrix. This implies that only 468 eigenvalues of S will be non-zero. This deduction can be further justified by looking at Figure 1 and 2. It can be observed in Figure 2 that the eigenvalue drops to approximately zero after 468.

Figure 3 shows the mean face and eigenfaces corresponding to the given data set. The eigenvalues of eigenface 1-6

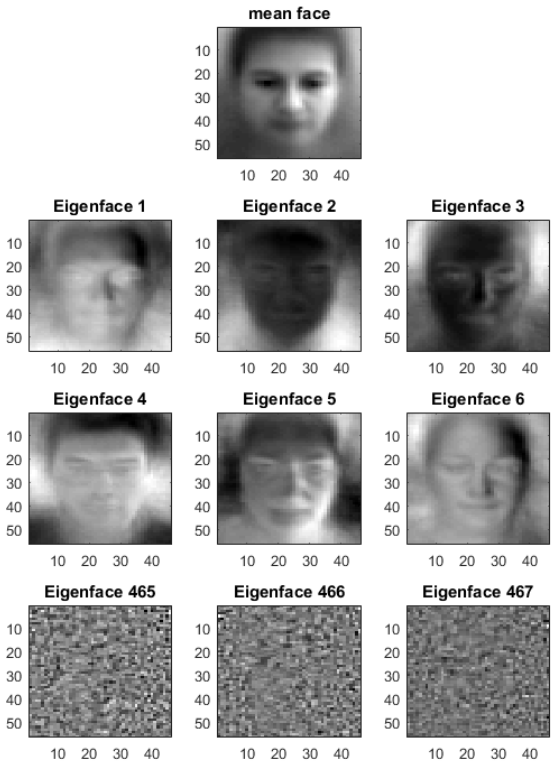

Fig. 3. Mean face and eigenfaces (for when $S = \frac{1}{N}AA^T$)

are much more significant than those of eigenface 465 - 467. Eigenfaces with greater eigenvalues displays the shape of 'ghost face' while this shape becomes less obvious for eigenfaces corresponding to smaller eigenvalues. These findings match our expectation since eigenfaces describe the characteristics that distinguish one face from another.

The number of images selected for face recognition depends on many factors. To choose the number of principal components K, the threshold ε should be arbitrarily set on the total variance v, which is represented by equation (3). This condition is illustrated in equation (4).

$$v = n \cdot (\lambda_1 + \lambda_1 + ...\lambda_n) \tag{3}$$

where λ_1 to λ_n are all the eigenvalues computed from S.

$$\frac{n \cdot (\lambda_1 + \lambda_1 + ...\lambda_m)}{v} > \varepsilon \tag{4}$$

where m is the smallest number of eigenvectors that satisfies the condition. In this part, ε is set to 0.95. Thus, 129 eigenvectors are used in order to include up to 95% of the total information. Adding more eigenvectors reduce the reconstruction error of training images, but leads to higher computational complexity. Additionally, using more eigenvectors increases the risk of overfitting the model.

PART 2: AN ALTERNATIVE APPROACH FOR PCA

An alternative method for computing the covariance matrix S is to use the approach shown in equation (5). This gives 468 eigenvalues which is equal to the rank of A minus one.

$$S = \frac{1}{N}A^T A \tag{5}$$

From Figure 5, it can be observed that the eigenfaces constructed from this alternative approach are similar to the ones which are constructed from the original form of covariance matrix. However, the intensities are shown to vary for some eigenfaces even after normalisation. It can be seen that some eigenfaces are exactly the opposite of the others. This is caused by the process that computes the eigenvectors. The eigenvectors of the matrix $S = \frac{1}{N}A^T A$ and $S = \frac{1}{N}AA^T$ will always lie in the same eigenspace but the eigenvectors corresponding to the same eigenvalue can take the opposite direction to each other. This results in the 'opposite' eigenfaces.

Now, let us look at the eigenvalues obtained from this this new S. These values are the same as the values obtained from the $\frac{1}{N}AA^T$ covariance matrix (although the first method gives 2576 eigenvalues, only the first 468 values are non-zero). This implies that the eigenvectors will also be the same since they are computed using the same eigenvalues. To prove the point, consider the matrix $A^T A$ with dimension $N \times N$.

$$A^T Av = \mu v \tag{6}$$

where μ is the eigenvalue and v is the eigenvector

Multiplying both sides of the equation (6) by matrix A gives

$$AA^T Av = \mu Av \tag{7}$$

$$SAv = \mu Av \tag{8}$$

Let $\mathbf{u} = Av$

$$S\mathbf{u} = \mu\mathbf{u} \tag{9}$$

From the relationship shown in equation (9), it can be concluded that $A^T A$ and AA^T have the same eigenvalues. This conclusion can be further justified by comparing Figure 2 and 4.

One obvious advantage of using the $A^T A$ covariance matrix instead of AA^T matrix is the reduction in computational complexity and hence the computation time. Since the eigenvalue decomposition process has the complexity of $\mathcal{O}(n^3)$ for a matrix of dimension $n \times n$, in this question the computational complexity of decomposing S with dimension 2576×2576 is approximately 167 times of S with dimension

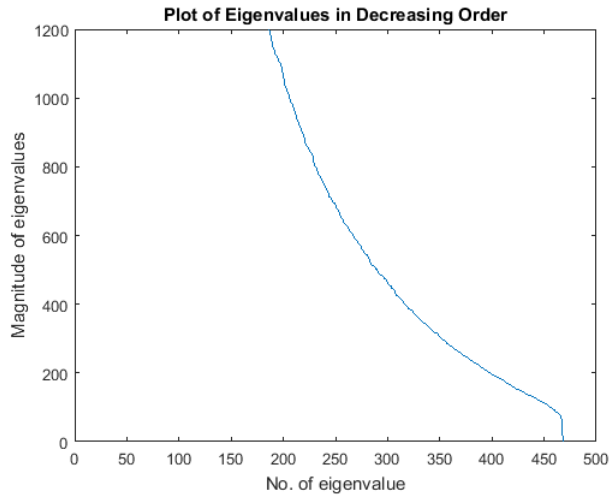

Fig. 4. A magnified version of Magnitude of eigenvalues plotted against No. of eigenvalues. (for when $S = \frac{1}{N} A^T A$)

468×468. Both $A^T A$ and AA^T have exactly the same non-zero eigenvalues and their eigenvectors can be converted to each other. Therefore, $A^T A$ is considered to be a more efficient approach.

PART 3: FACE IMAGE RECONSTRUCTION

In order to show the results obtained from the reconstruction face image, 3 classes from both the training and testing images are reconstructed using different numbers of eigenvecters (468, 100 and 10). These results are shown in Figure 7 and Figure 8 respectively. Let us begin by looking at the training images. Theoretically, using all 468 eigenvectors in the reconstruction process should lead to an exact reproduction of the original training image. As can be observed from the second column of Figure 7, these images exactly match their corresponding original images in the first column. When the number of eigenvectors used is reduced, the amount of information contained in the reconstructed images also reduce. Thus, these images are not as sharp and clear as those reconstructed using more eigenvectors. When the number of eigenvectors used is reduced to 10, the reconstructed training images become so blurry that they are almost unidentifiable. In this case, it is almost impossible to tell the difference between that of face 1 and face 2.

As for the testing images, using all the eigenvectors in the reconstruction process does not result in an exact reproduction of the original image. The images in the second column of Figure 8 are not as clear and sharp as those in the first column, which are the original testing images. One possible explanation for this is that the testing images contain some features that are not present in the training images, for example, different shades of lighting or face angles. Therefore, this makes it difficult for a reproduced testing

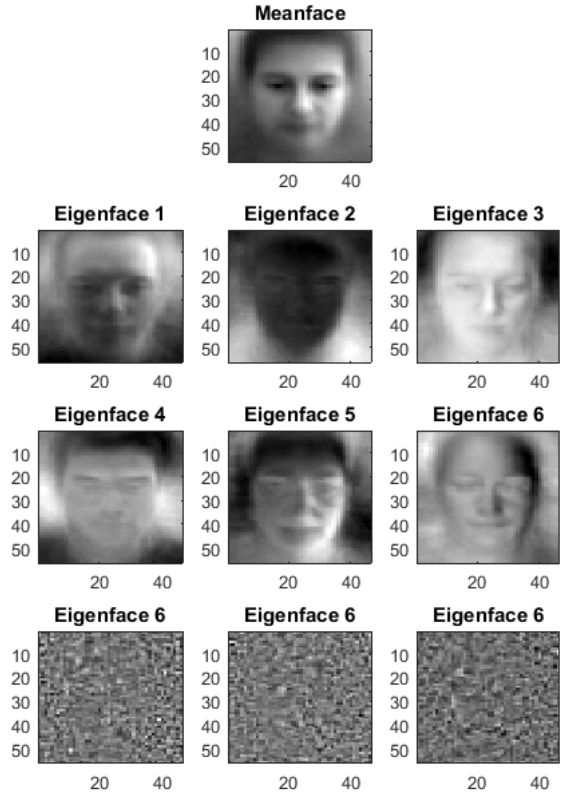

Fig. 5. Mean face and eigenfaces (for when $S = \frac{1}{N} A^T A$)

image, which is constructed using features (information) from the training images, to match the original image exactly. Again, as in the case of the training images, reducing the number of eigenvectors used leads to a reduction in sharpness and clearness of a reconstructed image.

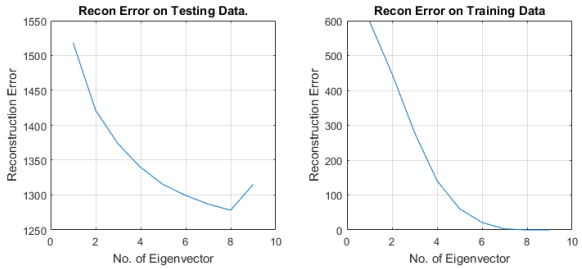

Fig. 6. Reconstruction Error for testing data (Left) and training data (Right)

To further illustrate the relationship between the number of eigenvectors used and the accuracy, the error is plotted against the number of eigenvectors in Figure 6. For the training images, the error decreases exponentially when the

number of eigenvectors goes up, reaching almost zero at when the number of eigenvectors used is approximately 7. For the testing images, the error decreases exponentially as the number of eigenvectors increase but it does not converge to zero. Instead, the error curve goes up as the number of eigenvectors used surpasses the optimal value of 8. This finding is consistent with the underlying theory and observations made earlier. The testing images cannot be perfectly constructed to match the original image and so the error will always exist. The rise in error is mostly caused by the overfitting of the model. Increasing the number of eigenvectors beyond the optimal value does not only lead to overfitting, this act also incur higher computational cost. Since the magnitude of the error decreases exponentially, increasing the number of eigenvectors beyond a certain range will not result in a significant reduction in the magnitude of error. Therefore, in order to prevent over-fitting and limit the computation time, not all eigenvalues should be used. A trade-off between the error and these two factors should be considered when choosing an optimal number of eigenvectors to be used. From Figure 6 an optimal number is 8.

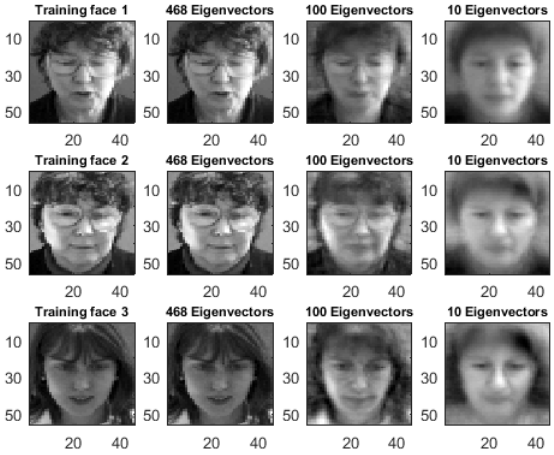

Fig. 7. Reconstruction of 3 traning images using different numbers of eigenvectors

PART 4: FACE RECOGNITION

PCA-based face recognition can be performed on the training and testing data set uisng the Nearest Neighbor (NN) classification method and another alternative method, which will be discussed in detailed.

Nearest Neighbor classification:

Let us first consider the Nearest Neighbor classification. The test image is projected onto the eigen-subspace spanned by the training data. The Euclidean distance between its projection and each of the projections corresponding to the training data will be measured and compared. The test image will get assigned to the same class as the training image whose projection lies the closest to the test image's projection.

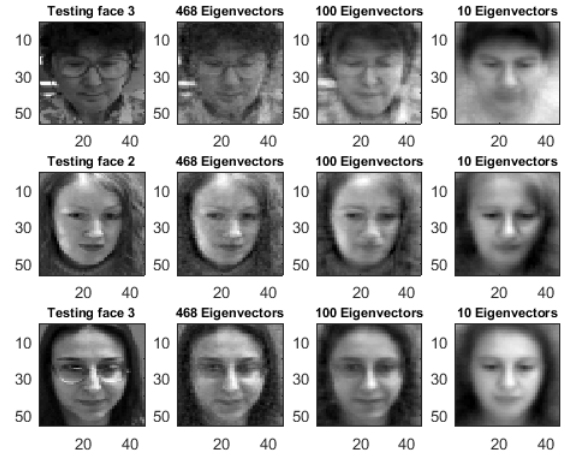

Fig. 8. Reconstruction of 3 testing images using different numbers of eigenvectors

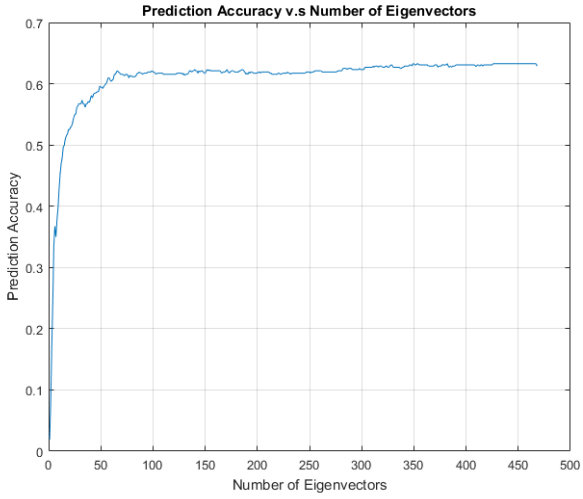

Fig. 9. Nearest Neighbor prediction accuracy against eigenvector number

Figure 9 shows the estimation accuracy plotted against the number of eigenvectors used. The estimation accuracy is obtained by calculating the average accuracy of the 10-fold cross-validation. The maximum accuracy is found to be approximately 0.63 when the number of eigenvector is 349. Note that the marginal improvement in prediction accuracy is significant when the number of eigenvectors is small. As the number gets larger, the accuracy will approach a certain limit. It should be noted that a very large number of eigenvectors will likely cause overfitting and at the same time, increase the computational complexity.

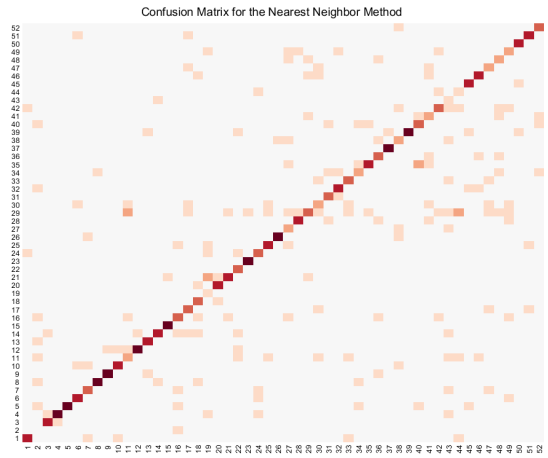

Fig. 10. Confusion Matrix for the Nearest Neighbour Method

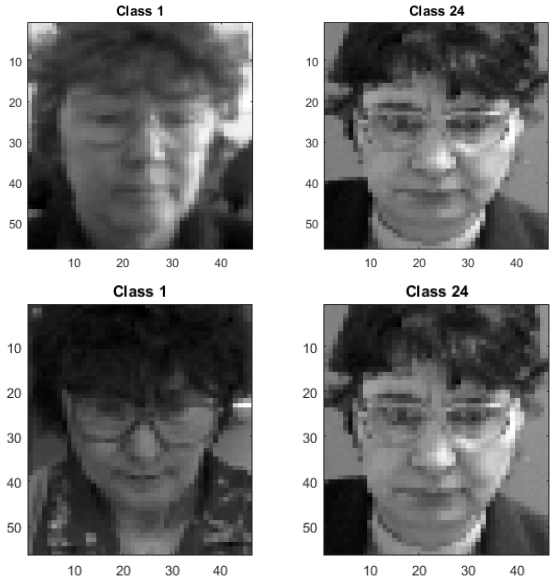

Fig. 11. Examples of misclassification associated with the Nearest Neighbour method (Top) and Alternative method (Bottom).

According to the results found in Part 1, the optimal number of eigenvectors is 129, which gives the prediction accuracy of approximately 0.62. Using the number of eigenvectors to determine the time and the memory used, the average time to compute each fold of cross-validation is around 0.06s while the memory used is approximately 6.3 MB.

Figure 10 shows a confusion matrix associated with the NN method where 129 eigenvectors are selected to train the model. The top row of Figure 11 shows an example of misclassification associated with the NN method. The nearest neighbour match the image from Class 1 with an image from Class 24. It can be observed that the two images have high degree of similarity. After projecting them onto the eigen-subspaces, it is found that the Euclidean distance between their projections is the smallest.

Alternative method:

The second method is to reconstruct the test image using the principle subspaces for each class. The class with the minimum reconstruction error is selected as the class of the test image. The estimation accuracy is averaged over the 10-fold cross-validation. The peak accuracy is found to be approximately 0.76 when the number of eigenvector is 8. Similar to the first method, increasing the number of eigenvectors results in a high accuracy at the start but introduces the risk of overfitting when the number is too large. When the number of eigenvector is 8, the average time used to perform one fold of cross-validation is approximately 0.35 second while the memory used is approximately 0.4 MB.

Figure 13 shows the confusion matrix corresponding to the alternative method. It can be seen that the misclassification error is less than that of the first method. An example of misclassification associated with the alternative method is shown in bottom row of Figure 11. Since the alternative method uses the features from each class to reconstruct images, the misclassification indicates that the image can be better reconstructed using features of other classes. Since the facial image can have various brightness and angle, it is possible for different classes to have similar facial basis, which results in an error.

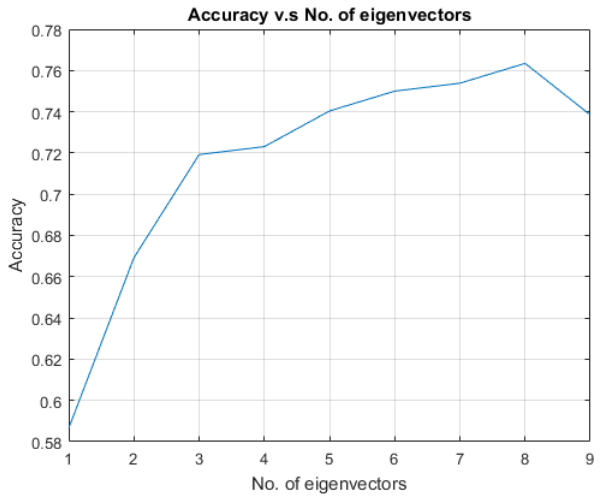

Fig. 12. Accuracy v.s No. of eigenvectors for the alternative method

By comparing the first and the second method, it is obvious that the alternative method gives a better recognition accuracy compared to the Nearest Neighbor approach provided that the number of eigenvectors in each class is sufficiently large. It

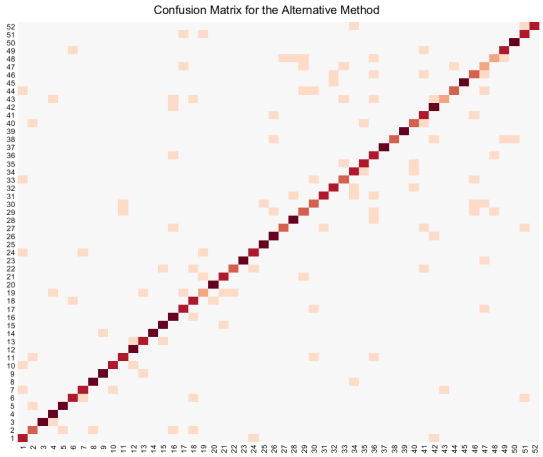

Fig. 13. The Confusion Matrix for the Alternative Method

is also more efficient in term of memory usual than the NN method. This is due to the fact that in every loop, only a small number of eigenvectors is required for training and testing while the first method requires eigenvectors from all classes. However, the alternative method is more time-consuming than the first method since it has to loop through all 52 classes and perform training and testing on each class. For the second method, PCA is only performed once for each fold of cross-validation.