Quantitative Foundations Project 4 : Dimensionality Reduction Classification

Nilesh Kumar, Saad Hassan

December 2019

1 Problem Description

In this project, we were given a database of 400 facial images from 40 subjects (10 per subject). We were asked to load these images to Matlab and use PCA to extract *eigenfaces*. Afterwards, we had to employ linear classification methods for image reconstruction, face recognition, and face identification.

2 Train and Test Set

The train set is same as instructed while the test set contains thirty (30) Cifar images on top of the given images.

3 Using PCA to Extract EigenFaces

3.1 How do the leading eigenfaces look like as an image?

Leading eigenfaces look like general features of human face, you can see a human face template in initial eigenfaces.

(a) First Eigenface

(b) Second Eigenface

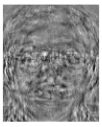
(c) Third Eigenface

Figure 1: How do Eigenfaces looks like as an image?

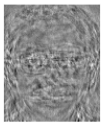
3.2 How does the importance of the eigenfaces decrease?

The last few eigenfaces, when sorted according to eigen values and plotted as images do not provide a lot of information as seen below. Furthermore, as you go down in eigenfaces their contribution to overall variance of the data decreases. You can also see the decreasing importance trend in the reconstructed images, lower eigenfaces contribute a small amount of information in reconstruction.

(a) 100th eigenface



(b) 150th eigenface



(c) 200th eigenface

Figure 2: Examples of eigenfaces with very little information and lot of noise

4 Face reconstruction with PCA

4.1 Observe the difference between reconstructed and original images, as the number of eigenfaces used in reconstruction increase?

It can be seen that the difference between the original and the reconstructed decreases as you use more and more eigenfaces. Also, after certain number of eigenfaces the contribution of the eigenfaces becomes smaller and smaller.

(a) 10 eigenfaces (b) 50 eigenfaces (c) 100 eigenfaces (d) 150 eigenfaces (e) 200 eigenfaces (f) 250 eigenfaces

Figure 3: Effect of using different number of eigenfaces on the reconstructed face of participant 1

(a) 10 eigenfaces (b) 50 eigenfaces (c) 100 eigenfaces (d) 150 eigenfaces (e) 200 eigenfaces (f) 250 eigenfaces

Figure 4: Effect of using different number of eigenfaces on the reconstructed face of participant 2

(a) 10 eigenfaces (b) 50 eigenfaces (c) 100 eigenfaces (d) 150 eigenfaces (e) 200 eigenfaces (f) 250 eigenfaces

Figure 5: Effect of using different number of eigenfaces on the reconstructed face of participant 3 using SVD

4.2 How many eigenfaces are required to recover an original face with reason-able errors?

This decision can be informed by looking at the images and qualitatively accessing if the images are reasonable. However, we can also leverage the information from variance metric. For this data set, 200 eigenfaces yield an original face with reasonable error. Figure 6 shows total variance plotted against number of eigenfaces.. It also depends on the task that we have at hand. For example, classification task 1 requires less details about faces as compared to task 2.

5 Classification

5.1 Face recognition

We calculated the PCA space on the training data. We then transformed test data on the basis of calculated PCA space on training data. We also normalized test data on the basis of parameters from training data. Our test data consisted of 120 faces and 30 non faces. Our training data set did not contain any of the non-face images. The main challenge of this task was to classify face and non-face without looking at any face images in train set. We tried to solve this problem such that at test time we try to look at the confidence of the classifier and if the classifier is not confident about image being a face image then it is a non-face image. The method explained in class does not do that because the scores are not normalized (cannot be compared) so we used methods that give probabilities (or some comparable numbers) in output. These methods included **KNN**, **K-means**, **SVM**, and **decision tree**, and **softmax**. All other methods except for KNN and Kmeans did not prove to be helpful for this task because they predicted non-face images as face images with same confidence. KNN classified all samples correctly after using threshold value for distance (mostly due to stark difference in face and non-face images), and K-means also perfectly divided face and non-face in two clusters. If the distance between closest neighor chosen was greater than threshold (decided by validation) then it was counted as non-face.

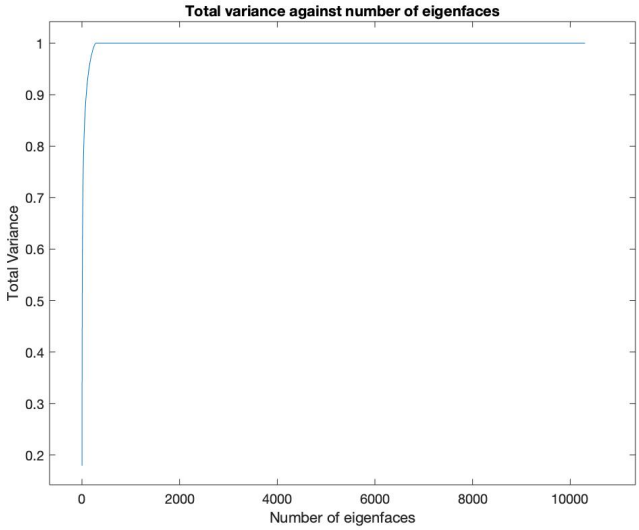


Figure 6: Total variance plotted against eigenfaces.

Table 1 - Task accuracies				
Sr.	Method	Task 1	Task 2	Task 2 modified
1	Linear Regression	0.80	0.93	0.53
2	KNN	0.94	0.90	0.56
3	K-means	1	-	-
4	Decision Tree	0.80	0.67	0.40
5	Softmax	0.80	0.96	0.56
6	SVM	0.80	0.97	0.57

5.2 Face identification

We tried to model this problem in the same way where thirty-five (35) known faces are classified with a confidence and if the confidence is low it counts as an unknown face. We tried to use sklearn methods to get probabilities of the predictions but like the first problem the probabilities did not provide any confidence in the prediction. However, we were able to somewhat get reflection confidence by decisionScores and were able to get five examples (unknown face) out while using softmax and that can be made better by using more validation to get a more accurate number. Overall, if we do not model task 2 as 35 known faces and remaining as unknown face, most of the classifier perform well. We have not shown quantitative results for kmeans because we were not sure if it was the best way to explain it. But, if we look at most of the pairs fall in the same cluster and the last 50 images (10 for each subject), most of the divisions of the ten images have majority of one cluster. So, it seems to give good performance.

```
Columns 1 through 15
  22   22   32   32   40   34   27   34    6    6    3    3   35   35   16
Columns 16 through 30
  16   30   30   11   11   32   32   27   40   39    9   33   33    7    7
Columns 31 through 45
  38   38   28   28   39    9   36   29   18   34   17   17   13   24   18
Columns 46 through 60
  18    8   19   12   27   14   14   32   32   23   29    2    2    8    8
Columns 61 through 75
  25   25   21   21   31   17    5    5   27   37   26   19   19   19   19
Columns 76 through 90
  26   19   19   19   19   23   23   15   23   15   15   15   23   15   23
Columns 91 through 105
   1    1    1    1    1    1    1    1   18    1   13   13   20   24   24
Columns 106 through 120
  13   24   20   20   13    4   12    9   10   12   39    4   39    4   39
```

Figure 7: Output of K-means

6 Conclusion

It should be quite clear now that methods that provide good confidence scores of their predictions can better solve these kind of problems. Additionally, more data (non-face images) is needed to make task 1 more difficult to solve. Specifically, the non-face images that can look close to human faces.