ML Assignment 1

Neeharika Immaneni - NI2452

neeharika@utexas.edu

1 Introduction

The goal of the assignment was to use principal component analysis (PCA), project images on the reduced eigen space and use k-nearest neighbours model to see the features captured by the projections. The data from the given .mat file was loaded into separate vectors in MATLAB. The eigen vectors was computed for the training images and the top n eigen vectors were considered as the principal components and the model was tested on the reduced eigen space. From the projected space, few images were projected back to their original size to be reconstructed. Many experiments were done which have been elaborated in detail below.

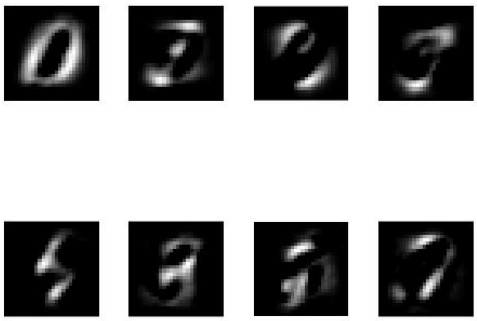
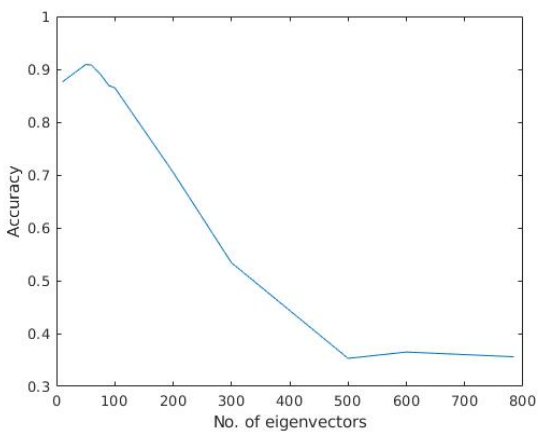


Figure 1: Top 8 eigen vectors

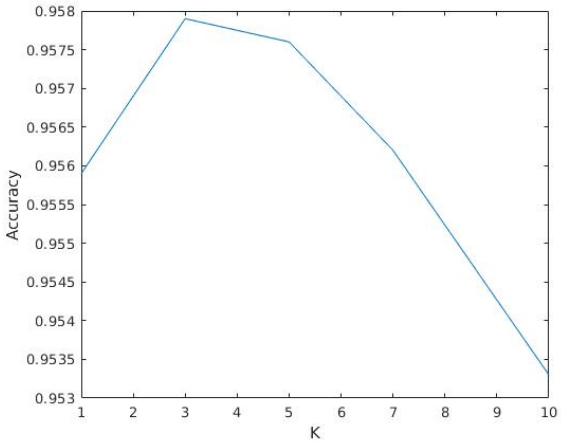
2 Method

The input data was converted into vectors, each 28x28 image was represented as a column vector of 784x1. For computation of the eigen vectors, the co-variance of the input data was computed after mean subtraction from the raw input data. The eigen values and vectors are found for the co-variance matrix, which were sorted in ascending order by default. To get the top-n vectors, the eigen vectors were sorted in descending order corresponding to their eigen values and the vector was normalized.

$$\begin{aligned} Input(I_{shifted}) &= Input(I_{raw}) - Mean(M) \\ EigenSpace(Z) &= Input(I_{shifted}) * V \\ Input(I_{recon}) &= (Z * V^T) + M \end{aligned} \tag{1}$$



(a) No. of Eigen vectors vs accuracy



(b) K-variants vs accuracy

Figure 2: Accuracy

The steps involved in projecting the images on the reduced eigen space can be understood from the formulas above. The eigen vectors (V) are generated for the given set of input images. For testing, the mean vector (M) is subtracted from all the test images(I) and this is multiplied by the eigen vector matrix, which gives us the projected eigen space. Reducing the number of eigen vectors (taking the top n only) results in a reduced projection space (Z). This vector is multiplied by it's transpose and the mean vector is added to get back the original image (re-constructed image).

A KNN classifier is defined with the training images in the reduced eigen space as the input to the classifier, which learns to classify the digits. The KNN classifier is tested to classify digits in the test set and the accuracy is determined.

3 Experiments

Various experiments were performed by varying the number of training images and the number of principal components chosen (eigen vectors). The top eight eigen vectors were visualized as shown in Figure 1.

3.1 Varying number of eigen vectors

Classification was initially performed by considering all the eigen vectors which resulted in very low accuracy. The vector was trimmed down step-wise and classification was done to determine the accuracy with only top n vectors and the accuracy was plotted against the number of vectors considered and the results can be seen in Figure 2a. It can be observed the accuracy reaches around 90% when the top-50 eigen vectors are considered and then drastically reduces, which explains that they are mostly garbage values with respect to the class label. We can see that of the 784 features in the image (28x28), considering the top 50-75 features alone will help us achieve high accuracy.

3.2 K-Nearest Neighbor (KNN) Search

From the input images on the projected space, a k-nearest neighbor classification model was built, which classifies images into one of the ten classes [0,1,2,3,4,5,6,7,8,9] using the euclidean distance metric. As shown in table 1, high accuracy for achieved when k=3, irrespective of the training and test data size. By keeping these parameters as a constant, the model was trained on 30000 training images, top 50 eigen

Training data	Test data	K-3	K-5
10000	1000	0.909	0.913
10000	3000	0.9096	0.9576
10000	5000	0.9112	0.9068
30000	10000	0.9579	0.9576
60000	10000	**0.9665**	0.9656

Table 1: Accuracy for varying K values and training points

Training data	Test data	K-1	K-3	K-5	K-7	K-10
30000	10000	0.9559	0.9579	0.9576	0.9562	0.9533

Table 2: Accuracy of KNN model by varying K

vectors were chosen and was tested for varying values of K and the graph plotted can be see in Figure 2b ans the corresponding values tabulated in Table 2.

3.3 Varying number of data points

Experiments were done by increasing the number of training points starting from 100, till 60000 (=total number of training images provided) and it can be seen from Figure 4a that as the training data increases, the accuracy of the model increases. The top 50 eigen vectors were considered and this was kept constant. With 10000 images, the model reaches 90% accuracy after 30K, it hardly improves and almost remains the same.

3.4 Image reconstruction

A random image was chosen from the data set and tried reconstructing it by projecting it back to the original space, re-shaping to its original dimensions and the results can be seen in Figure 3. When all the 784 features are retained, it is able to reconstruct a sharp image which looks very similar to the original and as we reduce the eigen vectors, the image becomes blurry.

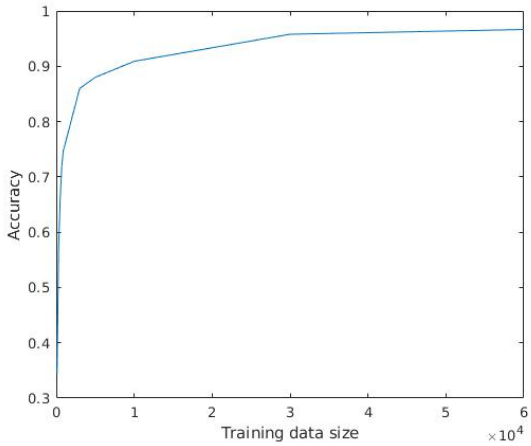
4 Discussions and Conclusion

A maximum accuracy of **0.9665** was achieved with a KNN model of k=3, trained on 60000 images and tested on 10000 images with the top 50 eigen vectors chosen.

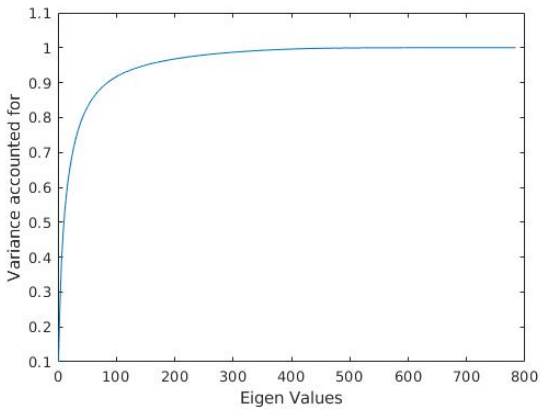
The cumulative sum of the normalized eigen values were plotted again the variance accounted for as shown in Figure 4b. We can see that the graph increases rapidly and the first 200 values account for around 90% of the variance in the data set after which it increases sluggishly.

(a) 100 EV (b) 300 EV (c) 500 EV (d) All 784 EV

Figure 3: Image reconstruction by varying the number of principal components



(a) Training data size vs accuracy



(b) Eigen values vs Variance

Figure 4: Accuracy

Though the data set size was huge, we observe that the model learns and achieves 90% accuracy very quickly with small value of K.