Eigendigits classification

Pandian Raju (pandian@cs.utexas.edu, UT ID: pr22695)

February 2017

1 Introduction

The project aims at classifying handwritten digits using dimensionality reduction and Principal Component Analysis (PCA), given a training set and test set. Training set consists of a set of (60000) images where each image is a 28*28 matrix comprising of the features of the image. The variance among the images in the training set is derived in terms of the covariance matrix and the eigen vectors of the covariance matrix form the new cooordinate system. The main idea in doing this is that the classification works best when the images are projected to the new coordinate system (of eigenvectors). These new eigendigits are displayed, training set is projected to the new system and the test set is predicted for the actual label given the test image and hence the accuracy is calculated. This experiment also aims at doing a detailed analysis of the parameters used in the classification like the number of eigen vectors, size of the training set, k-NN size and the effects of varying these parameters on the accuracy.

2 Methods

As far as the experiments are concerned, each experiment is repeated for 5 to 10 times (depending on the run time of the experiment) and the average of them is chosen to be the final value. This ensures that the results are not distorted due to random coincidences.

2.1 Eigendigits and new eigenspace

- The first step in finding the eigendigits is representing the variance among the digits given in the training set. For this, a set of images (of size E, say) from the training set is chosen (28*28*E) and it is converted to a 2D matrix of dimensions (784*E), where each column represents an image and each row represents a feature in the image.

- The mean of the images is calculated and is subtracted from each image in the set (say, the resultant matrix is A of size 784*E). The covariance matrix is given by AA^T and the normalized eigenvectors of the covariance matrix form the new coordinate system. But since the dimension of AA^T is 784*784, it can be computationally costly to find the eigen vectors. Hence, the eigen vectors of A^TA (which is of dimension E*E) are found and a simple trick of multiplying the eigen vectors with A gives the eigen vectors of the covariance matrix, J'.

- The eigen vectors are then sorted by the value of the eigenvalues in descending order since the ones with higher values represent the maximum variance between the digits. Top n eigen vectors from this list can be chosen to represent the new coordinate system and the accuracy varies depending on n.

- The new coordinate system (eigenspace) is found by normalizing the eigen vectors found above (J`) and let J denote the normalized new coordinate system.

2.2 Projecting to the eigenspace and reconstruction

- Once the new coordinate system is found, any image (of dimension 784*1) can be projected to the new coordinate system by multiplying with the transpose of new coordinate system J (of dimension 784*E) resulting in a vector of dimension E*1. Effectively, we have reduced the dimension (number of features) of the image from 784 to E.

- To reconstruct an image which is projected to the eigenspace, the projected image (E*1) is multiplied with the eigenspace J (784*E) to give back the original image (784*1). Note that there will be some loss of clarity in the reconstructed image since we approximated the image to an eigen space of lesser dimension and then reconstructed back.

2.3 Predicting test labels for the test set

- A set of images from the training set is chosen and projected to the new eigenspace J.

- To test how well the algorithm works, a set of images from the test set is chosen and projected to the new eigenspace J.

- For each of the test image, k-NN classification algorithm is applied to find which of the training images are very close to the test image and the same is chosen as the test label.

- The accuracy of the algorithm is obtained by the percentage of test images which were correctly labelled by the algorithm.

2.4 k-NN classification

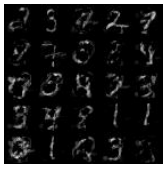
- For each of the test image, the distance between the test image and each of the training set image is found and the corresponding training set images are sorted by increasing order of distance.

- Typically, Euclidean distance is used to measure the similarity between images but cosine distance and manhattan distance are also evaluated for the k-NN classification.

- Depending of the size of k in k-NN, the k nearest neighbours for each test image are chosen and the most frequent training set label among them is labelled as the label for the test image by the algorithm.

(a) Top 25 eigendigits

(b) Original image



(c) Reconstructed image

Figure 1: Eigenspace of dimension 25

(a) Top 25 eigendigits

(b) Original image

(c) Reconstructed image

Figure 2: Eigenspace of dimension 200

3 Results

3.1 Eigendigits and reconstruction

- Figures 1, 2 and 3 show the top 25 eigen digits (in decreasing order of eigen values), sample 25 original digits (from test/training set) and the digits reconstructed after having been projected to the eigenspace for different number of eigen vectors chosen - 25, 200 and 500 respectively.
- The eigendigits contain the main variances among the digits which are used to classify the digits. It can be seen that the eigendigits are very feeble for 25 eigen vectors while it is clear and encodes much variance for 200 and 500 eigen vectors.
- The reconstructed image for 25 eigen vectors is very distorted because only the top 25 features is chosen and projecting the image to the eigenspace loses all other features. On the other hand, the reconstructed images for 200 and 500 eigen vectors are pretty clear since those cover most of the important features of the digits.
- There is a slight difference between reconstructed images for 200 and 500 images in that the images for 200 eigen vectors are slightly less clear than the images for 500 eigen vectors. So, depending on the level of accuracy required, the number of eigen vectors can be chosen accordingly.

3.2 Effect of number of eigen vectors on accuracy

- Figure 4 shows the effect of varying the number of eigen vectors on the accuracy obtained.
- It can be seen that the accuracy increases linearly with the number of eigen vectors till some threshold and then remains (approximately) stable. Some of the peaks observed are for 35, 150, 350 and 450 eigen vectors.

(a) Top 25 eigendigits

(b) Original image

(c) Reconstructed image

Figure 3: Eigenspace of dimension 500

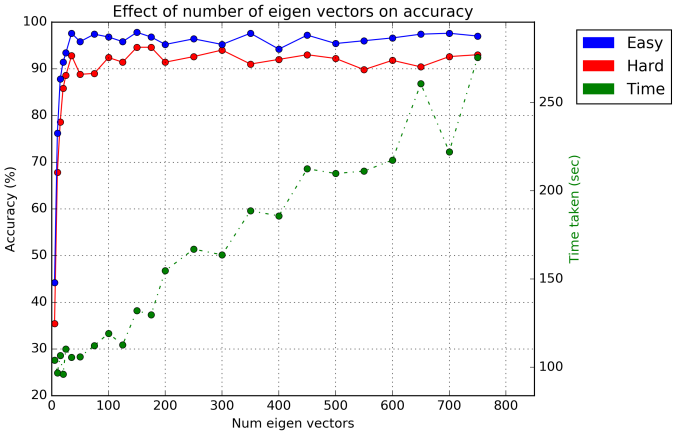


Figure 4: Training set: 5000 and k-NN: 5

- The time taken to complete the classification also grows linearly with the number of eigen vectors chosen. So, depending on the performance and accuracy that need to be achieved, the number of eigen vectors can be chosen (in this case, 150 suits well since it takes lesser time and gives good accuracy of around 98% for easy test cases)

3.3 Effect of training set size on accuracy

- Figures 5 and 6 show the effect of training set size on the accuracy obtained for smaller and larger training set sizes respectively.
- It can be seen that the accuracy is very low (below 80%) for training set sizes below 750. As the size of training set increases, the accuracy also improves.
- The accuracy reaches a peak of around 98% at training set size of 60000.
- As the training set size increases, the experiment becomes computationally expensive (as denoted by the linear increase in time taken).
- Depending on the level of accuracy desired and the performance guarantee required, the training set size can be chosen accordingly.

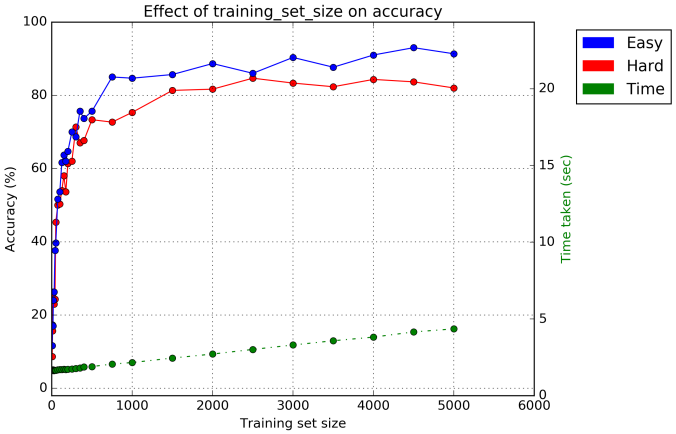


Figure 5: Num eigen vectors: 75 and k-NN: 5

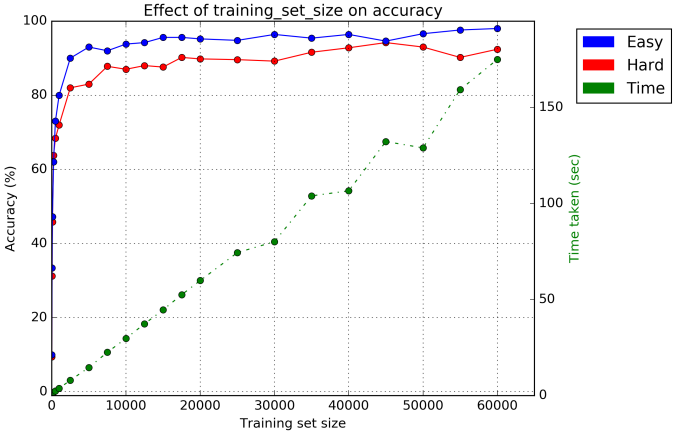


Figure 6: Num eigen vectors: 250 and k-NN: 5

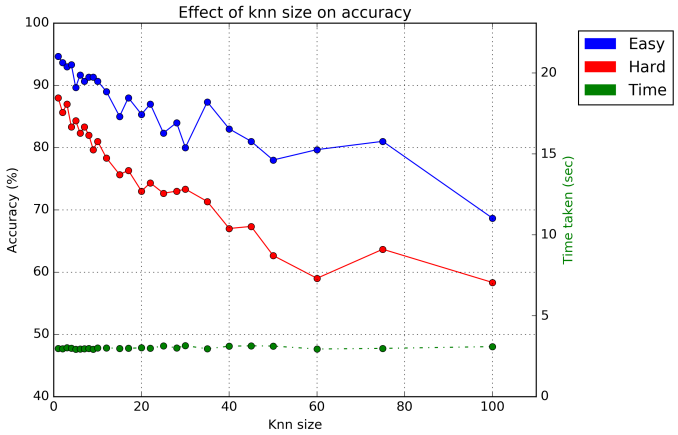


Figure 7: Training set: 5000 and Num eigen vectors: 100

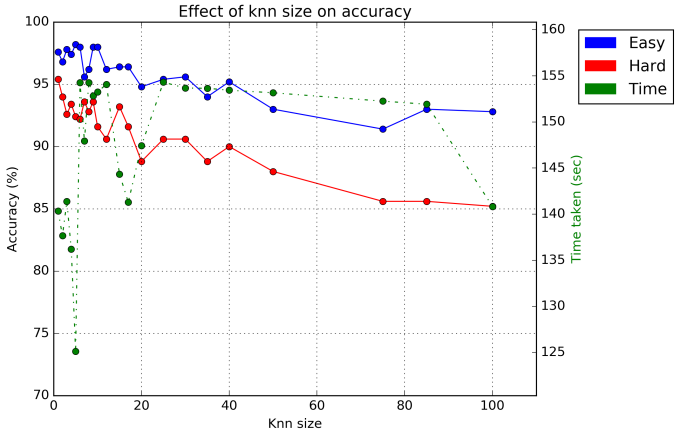


Figure 8: Training set: 50000 and Num eigen vectors: 500

3.4 Effect of k-NN size on accuracy

- Figures 7 and 8 show the effect of k-NN size (number of nearest neighbours) on accuracy obtained.
- It can be observed that the accuracy drops as the size of k-NN increases. The value of k-NN value for which accuracy is highest varies depending on the training set size and number of eigen vectors chosen.
- According to these sets of parameters, the optimal value of k in k-NN seems to be between 1 and 5, depending on other paramters.
- The (computational) performance of the algorithm is not affected much by the k-NN size (as inferred by the time graph).

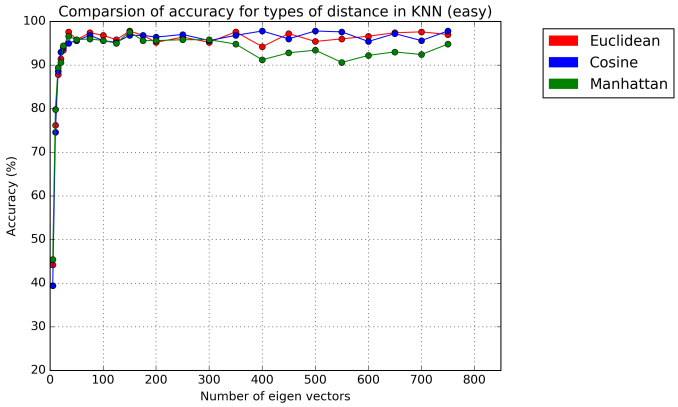


Figure 9: Num eigen vectors vs Accuracy (easy)

4 Discussion

4.1 Comparison between types of distances used in k-NN classification

- The k-NN classification algorithm uses some metric, typically Euclidean distance, to measure the similarity between the test set and the training set. Additional types of distances like the cosine similarity and Manhattan distance are also evaluated and corresponding accuracies obtained are compared for the same test and training parameters.
- Figures 9 and 10 show the comparision between types of distances used in k-NN classification for easy and hard test cases.
- One clear outcome of the comparison is that Manhattan distance performs relatively poor in terms of the accuracies obtained.
- Although cosine and Euclidean plots have peaks at different points and they are very close to each other in terms of accuracy, cosine seems to be slightly better than Euclidean.
- Cosine can be computationally costlier than Euclidean and so factoring in that and the level of accuracy desired, a particular distance type can be chosen.

4.2 Tuning number of eigen vectors

- Figure 11 shows the plot between training set size and accuracy for different values of number of eigen vectors chosen.
- It can be seen that the accuracy is very low for 10 eigen vectors, but beyond 50 eigen vectors, there is no much difference in the accuracies obtained.
- The peak is at 500 eigen vectors for training set of size 5000 (although the difference is very low).

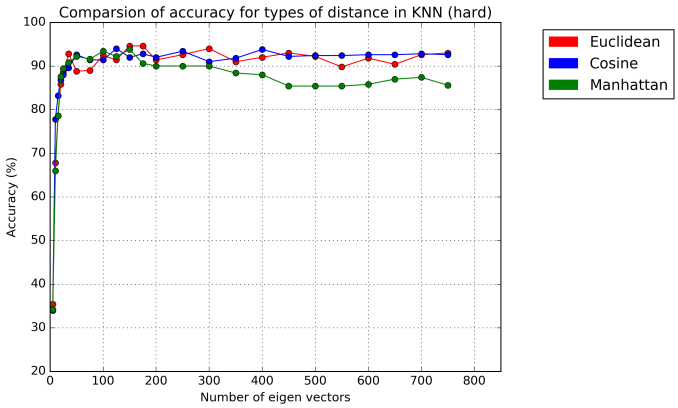


Figure 10: Num eigen vectors vs Accuracy (hard)

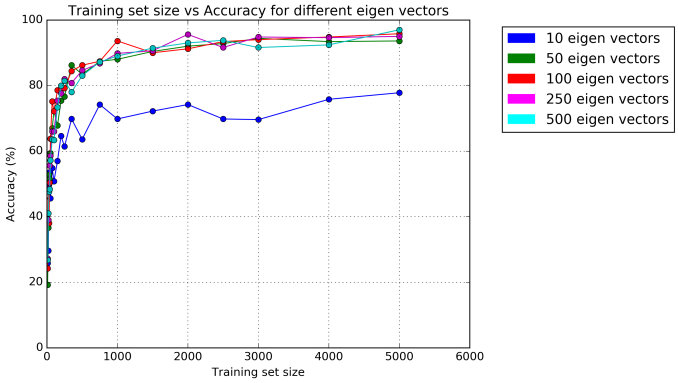


Figure 11: Training set size vs Accuracy for k-NN size 2 (easy)

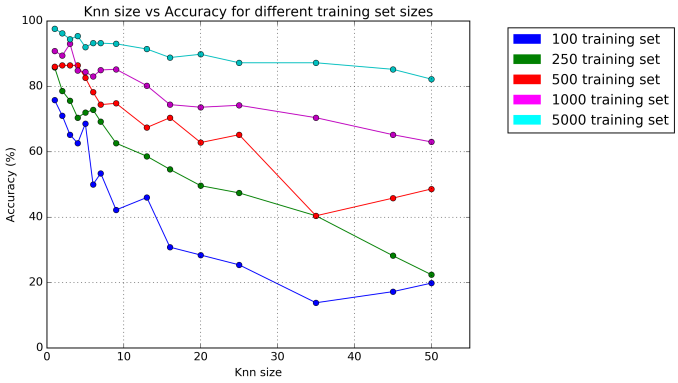


Figure 12: k-NN size vs Accuracy for num_eigen_vectors 100 (easy)

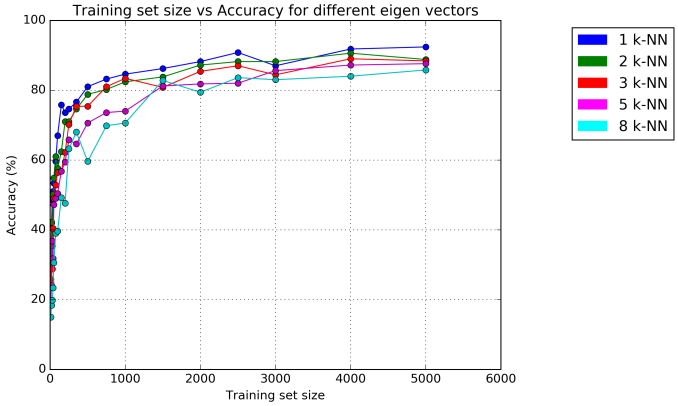


Figure 13: Training set size vs Accuracy for num_eigen_vectors 300 (hard)

4.3 Tuning training set size

- Figure 12 shows the plot of k-NN size vs accuracy for different training set size.
- It can be clearly seen that as the training set size increases, the accuracy also increases. This can be used to tune the training set size depending on the accuracy required.

4.4 Tuning k-NN size

- Figure 13 shows the plot of training set size vs accuracy for different k-NN size.
- The accuracy decreases as the size of k-NN increases and the higher values of accuracies are observed for k-NN size of 1, 2 or 3.
- This can be used to tune the k-NN size used for the classification.

5 Conclusion

By this experiment, various parameters that are used in PCA are analyzed, varied and the results are plotted. By experimenting with various sets of paramters, the set which gives the maximum accuracy can be chosen. Also, the use of dimensionality reduction is apparent in that the accuracy reaches reasonable good value after 50 eigen vectors while the actual dimension is 784, meaning that good accuracies can be achieved computationally cheaper with lesser number of principal components. Also, based on performance required (analyzing the time graph), the training set size can be chosen.