Eigendigits classification lab report

DIYUAN DAI, EID: DD33653,

Department of Computer Science, The University of Texas at Austin

1 1. INTRODUCTION

Data dimensionality reduction is an important part in the field of machine learning. The so-called dimensionality reduction is to use a certain mapping method to map data points from a high-dimensional space to a low-dimensional space. Its essence is to learn a mapping function. Currently, vector expressions are mostly used. The mapping function may be explicit or implicit, and may be linear or non-linear.

The reason for using dimensionality reduction data representation is because the original high-dimensional space may contain redundant information, as well as noise information, which will cause errors in the prediction of the data and reduce the accuracy; At the same time, the method can also reduce the complexity and improve the efficiency of calculation. In many algorithms, the dimensionality reduction algorithm has become a part of data preprocessing. This project is a handwritten digits classifier using dimensionality reduction and Principal Component Analysis (PCA), given a training set and test set of images.

2 METHOD

2.1 Eigenvector Decomposition

- Firstly, a set of images from training set is chosen and transformed into a 2D matrix of shape (n*784). Each row of the matrix represents a image of shape (28*28).
- Then, centralize all the images, which is every row by subtracting the average and denote the resultant matrix A.
- The covariance matrix V is calculated by $AA^{\intercal}$. However, this computation might be costly and inefficiently since $AA^{\intercal}$ is a 784*784 matrix. The alternative plan is to

calculate $A^{\intercal}A$, which is only n*n and multiply the result by A to give out the eigen vectors of original system, V.

- Matrix of all eigen vectors, V is sorted in a descending order of the corresponding eigen values. A certain amount, m of top eigen vectors are selected and form a new matrix P, which is the principle components matrix of the samples.

2.2 Projecting to new coordinate system and reconstruction

- Finally, the original images are projected into corresponding eigen space using the principle components matrix P. Here, the dimension of original image data is reduced from 784 to m successfully.
- Since each image is projected to the new eigenspace. We may simply multiplied the projected image with eigenspace matrix (m * 784) to reconstruct the original image.
- However, be noted that since only top m eigen vectors of original data are selected, we are tracing that m principle components. There might be some loss for imformation during this dimensionality reduction process. The reconstructed images may not be that clear and recognizable.

2.3 K-Nearest Neighbor Classification

- Using the transform function, any images in the data set can be projected into new system consisted of main components by multiply matrix P in front of the images. A set of images is chosen from test images set and put into the eigenspace. By comparing the nearness of each components between training set and test set, we can label the test images using the same label as the closest k training images. This is KNN classification and in this project, I use euclidean distance to calculate the difference between training image and test image.
- Finally, by checking the real labels of the test set and the predicted labels of test set, accuracy of the classifier is calculated.

3 RESULTS

3.1 Reconstruction

Figure 1 and figure 5 show the original image of randomly picked samples. Figure 2 and figure 6 shows centralized images of the picked samples. Figure 3 and figure 7 the top 20 eigen vectors while m = 20 and 100 correspondingly. Figure 4 and figure 8 shows the

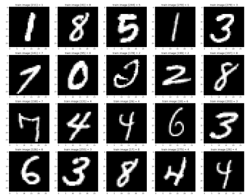


Fig. 1. original samples

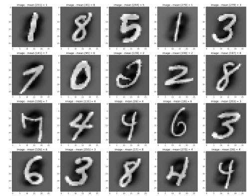


Fig. 2. original samples, mean subtracted

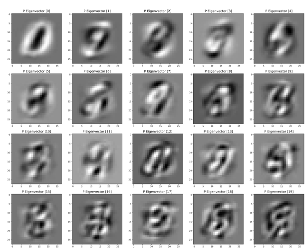


Fig. 3. Top 20 eigenvectors

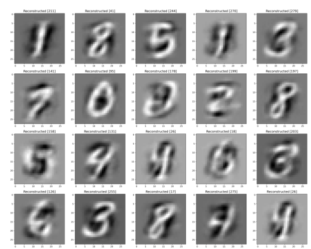


Fig. 4. Reconstructed images from dim-20 eigenspace

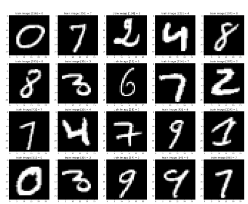


Fig. 5. original samples

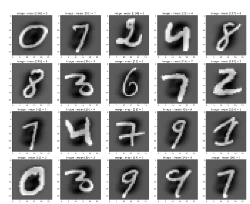


Fig. 6. original samples, mean subtracted

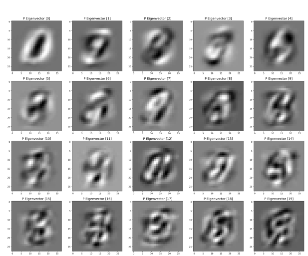


Fig. 7. Top 20 out of 100 eigenvectors

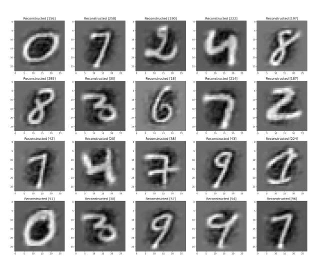
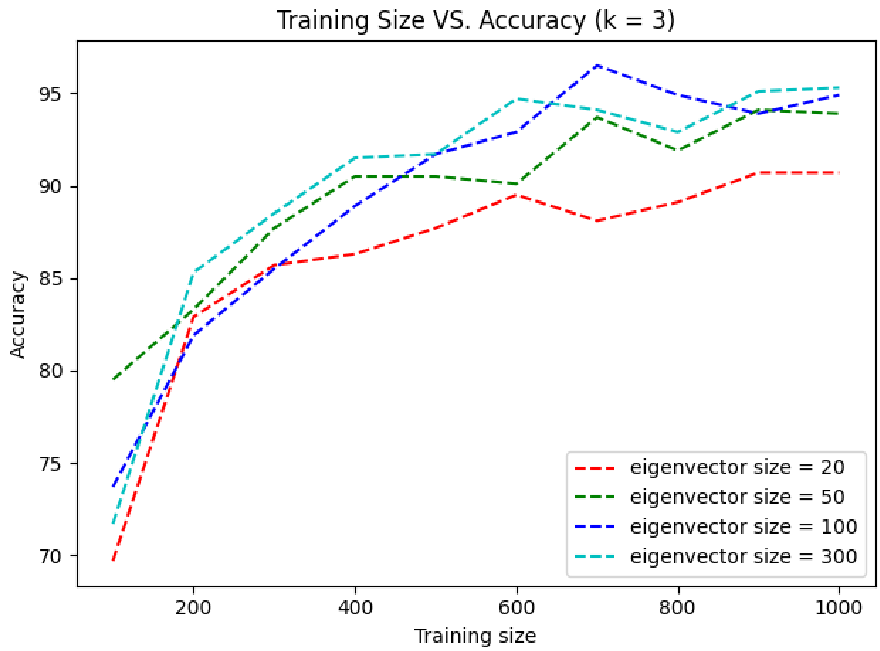


Fig. 8. Reconstructed images from dim-100 eigenspace

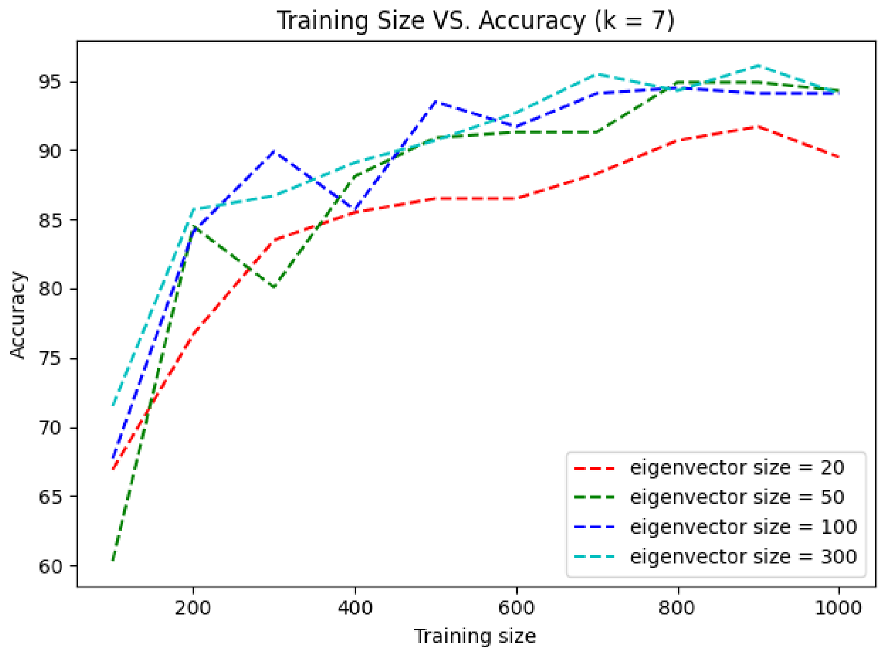


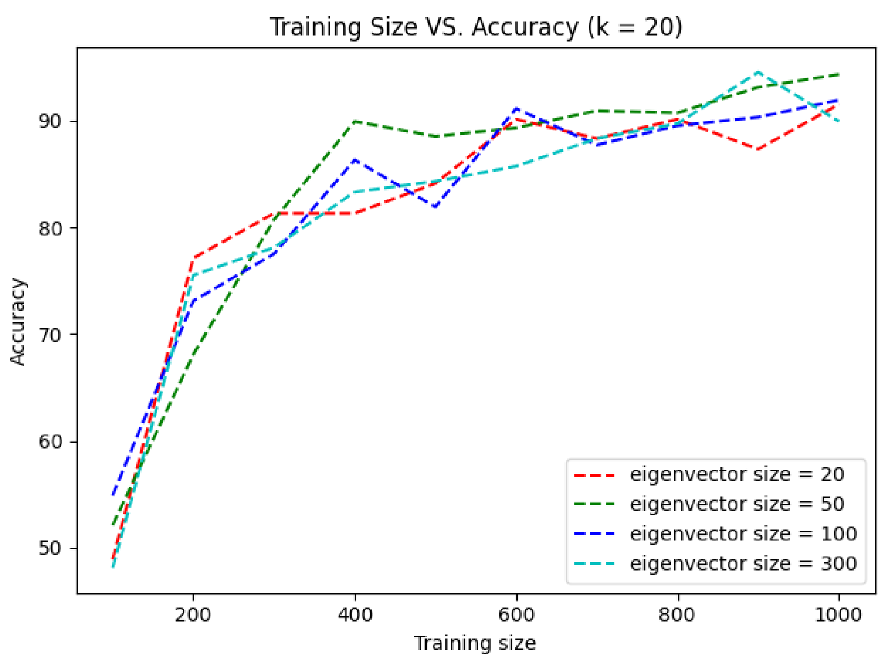
reconstructed images from 20-dimension eigenspace and 100-dimension eigenspace.

We can obviously find that the model can reconstruct some almost-recognizable digits from an eigenspace of a dimension of 20. However, the reconstructed images are much better while using more eigenvectors, from a higher dimensional eigenspace.

3.2 Effect of number of eigenvectors and training set size

The performance of the classifier is shown in Figure 9. This experiment is carried 10 times, under a parameter of K = 3, test set = first 3000 images, and average results are taken and ploted. We can see the model can reach a good prediction accuracy after 400 samples training. Also, generally, larger eigenspace, which means more eigenvectors used, better predicted labels statistics. However, after m = 50, the accuracy does not change much.


Training Size VS. Accuracy (k = 7)
Accuracy
Training size
200
400
600
800
1000
60
65
70
75
80
85
90
95
eigenvector size = 20
eigenvector size = 50
eigenvector size = 100
eigenvector size = 300


Training Size VS. Accuracy (k = 20)
Accuracy
Training size
200
400
600
800
1000
50
60
70
80
90
eigenvector size = 20
eigenvector size = 50
eigenvector size = 100
eigenvector size = 300

Another notable variance is the value of K used in KNN classifying algorithm. As far as I observed, bigger K value will lead to a slightly convergence on the difference between the accuracy of the results using different numbers of eigenvectors. Also, the accuracy of the model decreases significantly after k value > 10.

4 SUMMARY

According to the above explanation of the mathematical principles of PCA, we can understand some of the capabilities and limitations of PCA. PCA essentially takes the direction with the largest variance as the main feature, and "discorrelates" the data in each orthogonal direction, that is, makes them have no correlation in different orthogonal directions.

The experiments have shown dimensionality reduction under this circumstance is good enough after 50 eigen vectors while the original data has a dimension of 784. The classifier reach a good accuracy after 700 training samples and the accuraccy of the model does not change significantly after m > 100.