Course: CS391L Machine Learning
UTEID: my3852

Use PCA and KNN For Handwritten Digits Recognition

Anna Mengjie Yu[1,2]*

1. Department of Integrative Biology, The University of Texas at Austin, Austin, USA
2. Department of Statistics and Data Sciences, The University of Texas at Austin, Austin, USA

* annayu2010@gmail.com

Abstract

Handwritten digit recognition is an active topic in OCR applications and pattern recognition research. In this project, Principal Component Analysis (PCA) was used to extract digit information and project digits to low dimension spaces, followed by *k*-nearest neighbor classifier (KNN) to predict unknown digits. A combination of ks (ranges from 1 to 100), number of training images (ranges from 500 to 40000), and number of top eigenvectors (ranges from 10 to 782) were used. The highest recognition accuracy achieved is 96.76% at k=1 and number of training iamges = 40000. The effect of k, number of training image and number of top eigenvectors will be discussed.

Keywords: PCA, KNN classifier, Handwritten digit recognition

Introduction

Digit recognition is dealt in many fields such as postal mailing sorting, bank check processing, *etc*. Various approaches have been proposed to improve the accuracy of digit recognition accuracy, such as support vector machine (SVM), a binary classifier, and neural classifier, where the parameters of neural networks are optimized in discriminative supervised learning to separate patterns of different digit classes.

Principal Component Analysis (PCA) is a popular method for dimension reduction and information extraction. It uses orthogonal transformation to convert observations to linearly uncorrelated variables, *i.e.* principal components, then calculates the eigenvalues and corresponding eigenvectors of the covariance matrix. K-nearest neighbor classifier (KNN) classifies unknown by relating the unknown to the known using distance function. KNN is a brute-force computation of all pairs of points in the dataset. If k = 1, it simply assigns the unknown to the class of the nearest neighbor, also called the nearest neighbor algorithm.

In this project, PCA is used to reduce image dimensions and extract features. KNN with Euclidean distance metric is used to predict testing images through training image sets.

Method

The data used in this experiment were downloaded from Dr. Dana Ballard's webpage, also accessible from MNIST database (http://yann.lecun.com/exdb/mnist/). The data consist of 60,000 training images, and 10,000 testing images.

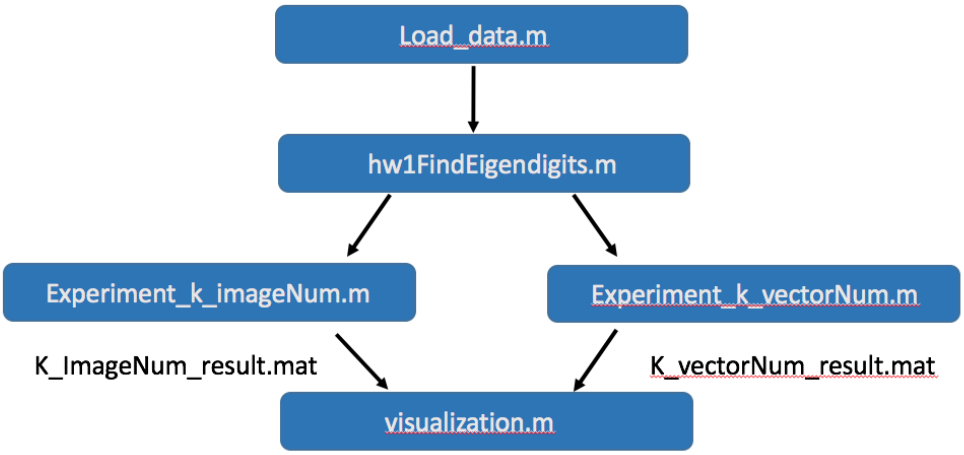


Figure 1. Workflow of matlab code for feature extraction, testing, and visualization

In hw1FindEigendigits.m, it normalizes the matrices by subtracting the column vector mean, then calculates the covariance matrices. Eigenvector matrices are returned with the corresponding eigenvalues sorted in descending order, and normalized to be unit vectors. The testing images were reconstructed using the equations: $I' = V * V^T * (I - m)$, where V is the eigenvector matrix and m is the average column vector of the training data.

In Experiment_k_imageNum.m, a certain number of training images [500, 1000, 2000, 4000, 8000, 20000, 40000] were randomly selected from the total training set, and the prediction accuracy were tested with a combination of ks [1, 5, 10, 20, 40, 100]. All the recognition accuracies were calculated at the selection of top 100 eigenvectors.

In Experiment_k_vectorNum.m, a combination of different top eigenvectors [10, 50, 100, 200, 300, 400, 500, 600, 700, 782] were selected at the fixed training image number 4000. Accuracies were calculated for a combination of ks [1, 5, 10, 20, 40, 100].

Result

1. Accuracy on variable training image numbers

From Figure 2 and Supplementary Table 1, we can see that the prediction accuracy increases as the number of training image increases. The prediction accuracy increases rapidly when the number of training images increases from 500 to 4000, and accuracy increase slows down as more training images were added. From Supplementary Table 1, we can see that the highest accuracy is achieved with the largest number of training images.

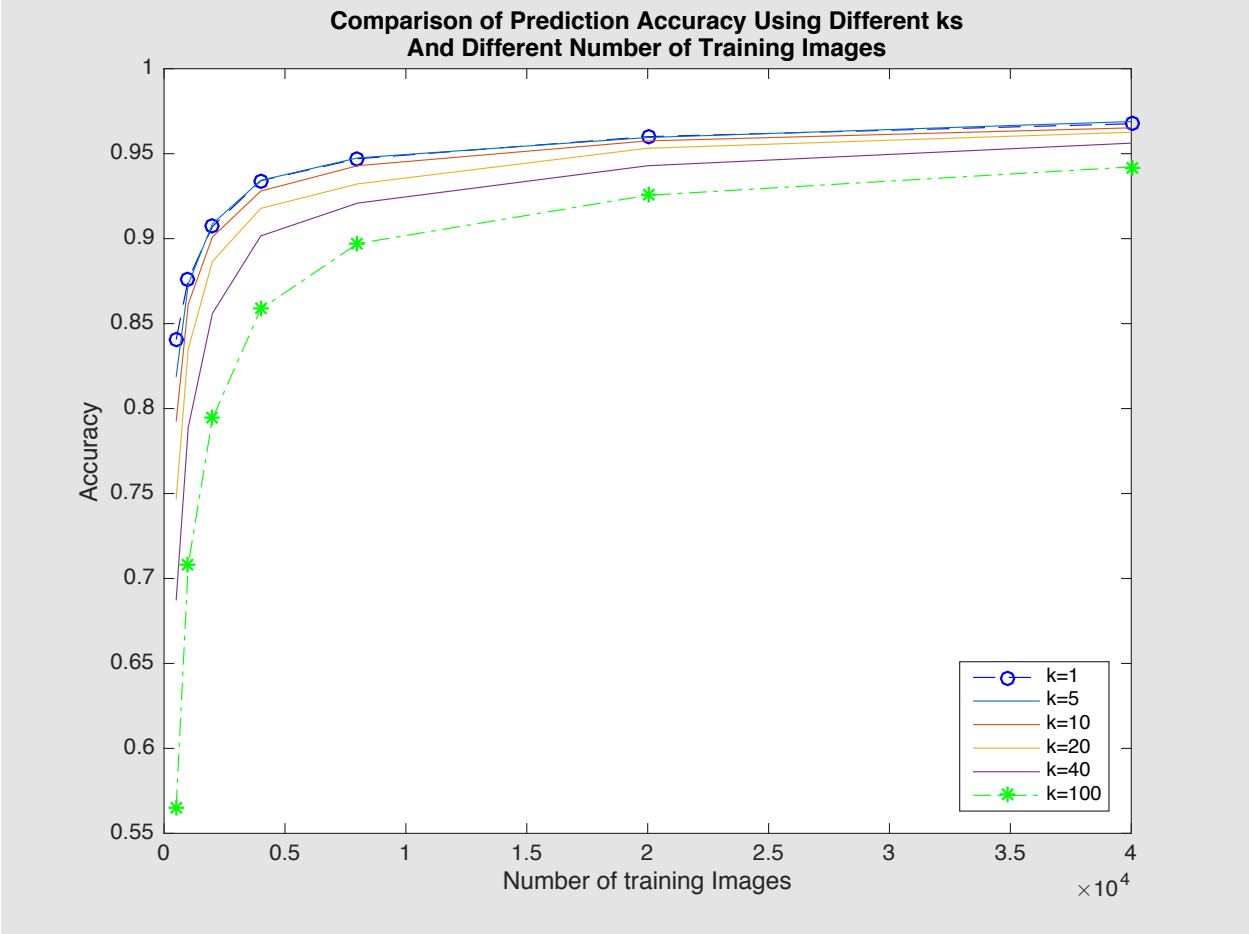


Figure 2. Accuracies in different training image number and k combinations

2. Accuracy on variable top eigenvector numbers

From Figure 3 and Supplementary Table 2, we can see that when the top N eigenvector number increase from 10 to 100, the accuracy increases, however, as more eigenvectors included, the accuracy stays stable or drops slightly.

3. Accuracy on variable k values

From Figure 2 and Figure 3, we can see the general trend that the bigger the k value, the lower the predicted accuracy. In Figure 2, we can see that the predicted accuracy for k=1 and k=5 is very similar for different number of training images. In Figure 3, at top eigenvector number being 50, k = 5 achieved a slightly higher accuracy than k=1. In both figures, k = 100 achieved the lowest accuracy.

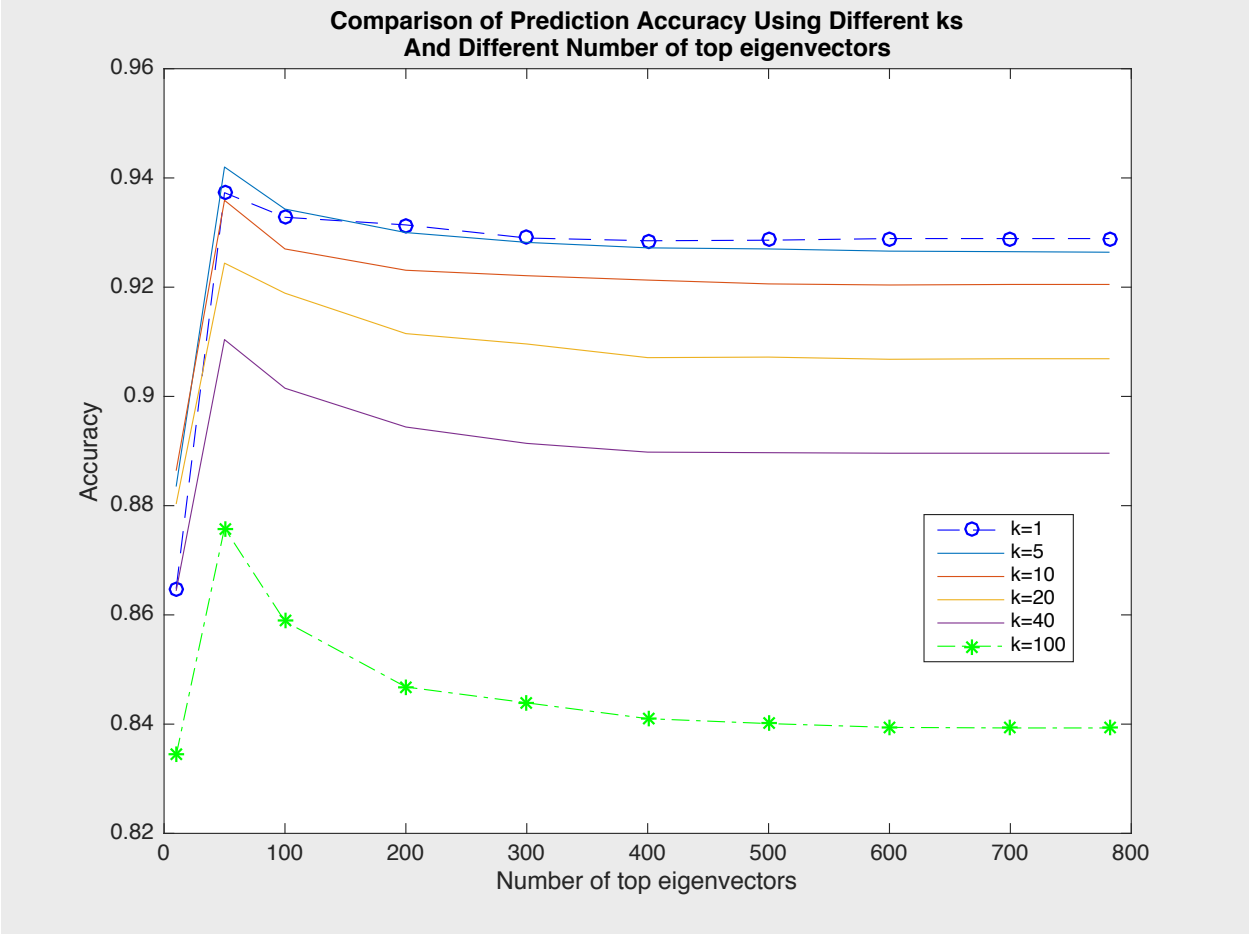


Figure 3. Accuracies in different top eigenvector number and k combinations

Discussion

In this experiment, a various combination of training image numbers, ks and top n eigenvector numbers were tested to evaluate their effect in handwritten digit recognition accuracy. We see the trend that the larger number of training images the higher the accuracy. However, very large the training image set will make the computation expensive.

In PCA orthogonal linear transformation, the great variance by the projection of the data lies on the first principal component, the second greatest variance on the second coordinate, and so on. From our results, the prediction accuracy increases as the top number of eigenvectors increase from 50 to 100, then the accuracy stays stable or drops slightly. This shows the majority of useful information is captured in first 100 top eigenvectors, and more eigenvectors of less importance might introduce noise, rather than signal.

The increase of k will reduce the effect of noise on the classification, however, large values of k makes the boundaries between classes less distinct. That is why we see a trend of decreasing accuracy when k gets bigger. KNN is a brute-force computation approach to

measure the pairwise distance between all points. In large category prpoblems such as Chinese character recognition, a parallel approach combined with OpenMP or MPI will reduce the cost of computation time.

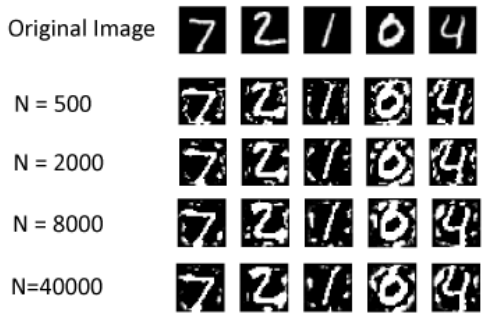
Supplementary

Supplementary Table 1: Prediction Accuracy using different ks and training image numbers (topN eigenvector = 100)

Accuracy	imageNum	500	1000	2000	4000	8000	20000	40000
K = 1		0.8406	0.8760	0.9071	0.9341	0.9470	0.9600	0.9676
K = 5		0.8187	0.8731	0.9087	0.9343	0.9475	0.9595	0.9689
K = 10		0.7923	0.8615	0.9010	0.9280	0.9429	0.9575	0.9652
K = 20		0.7464	0.8351	0.8866	0.9178	0.9322	0.9532	0.9626
K = 40		0.6871	0.7889	0.8560	0.9016	0.9209	0.9429	0.9562
K = 100		0.5654	0.7080	0.7941	0.8586	0.8971	0.9255	0.9422

Supplementary Table 2: Prediction Accuracy using different ks and top eigenvector numbers (training image number = 4000)

Accuracy	topN	10	50	100	200	300	400	500	600	700	782
K = 1		0.8648	0.9373	0.9328	0.9314	0.9290	0.9285	0.9286	0.9289	0.9289	0.9289
K = 5		0.8835	0.9420	0.9343	0.9300	0.9282	0.9272	0.9270	0.9266	0.9265	0.9264
K = 10		0.8864	0.9359	0.9270	0.9231	0.9221	0.9213	0.9206	0.9204	0.9205	0.9205
K = 20		0.8803	0.9244	0.9189	0.9115	0.9096	0.9071	0.9072	0.9068	0.9069	0.9069
K = 40		0.8644	0.9104	0.9015	0.8944	0.8914	0.8898	0.8897	0.8896	0.8896	0.8896
K = 100		0.8343	0.8759	0.8588	0.8468	0.8439	0.8410	0.8401	0.8394	0.8393	0.8393



Supplementary Figure 1. Reconstruction of The Testing Images with Different Training Image Numbers