Homework 1: Eigendigits

Jodie Simkoff

jms9562 - jsimkoff@utexas.edu

September 18, 2017

1 Introduction

Image recognition involves correctly identifying images as belonging to a certain class, based on analysis of their features. Since this typically means an analysis of the pixel data, the problem can quickly become high-dimensional and computationally expensive. As such, it is often useful to project the pixel data to a lower dimensional subspace before implementing a classification algorithm on training and test data. In this work, a program was implemented to correctly identify the value of handwritten digits. Principal component analysis (PCA) was used for for dimensionality reduction and k-Nearest Neighbors for classification.

2 Methods

In order to make the classification problem more tractable, principal component analysis was first used to identify the directions of maximum variance of the $x \times n$-dimensional data set. Here $x = 784$ is the number of pixels in each image, while n is the number of training images used to construct the basis; n was varied to explore its effect on performance as discussed in the Results.

To construct the subspace for classification, the covariance matrix of the mean-centered data was computed. Since in most cases the number of images used to construct the basis was greater than the pixel size ($n > x$), this amounted to identifying

$$\mathbf{\Lambda} = eig(\mathbf{A}\mathbf{A}^T) \tag{1}$$

although $\mathbf{A}^T\mathbf{A}$ can also be used to identify the largest eigenvalues and their eigenvectors. Here $\mathbf{A}$ is the $x \times n$ matrix of vectorized pixel data.

The largest T eigenvalues in $\mathbf{\Lambda}$ were identified and their corresponding eigenvectors used to construct the basis for the subspace since these directions explain the majority of the variance. Once the basis $\mathbf{B}$ was constructed, a new (mean-centered) pixel data vector $\mathbf{v}$ was projected using $\mathbf{v}_p = \mathbf{B}\mathbf{v}$. The images can also be reconstructed using $\mathbf{v}_r = \mathbf{v}_p^T\mathbf{B}$.

Different training data set sizes n and number of eigenvectors T were considered. The goodness of the basis was judged in part by the ability to reconstruct the projected images. Sample images of the eigenvectors and reconstructed images are presented in Section 3.1.

Once the basis was constructed, a k-nearest neighbors model was built using the *fitcknn* function in MATLAB. The weight was based on the inverse distance. Then, the test data (consisting of two sets of 5000 images) was classified using this model and the results were scored against the true values of the images. Different values for the nearest number parameter k were considered and the results are presented in Section 3.2.

3 Results

3.1 Dimensionality reduction

Both the number of training data vectors n and the number of principal components T affect the how well the basis can represent the images and thus how well the classification will perform. Although more data and more eigenvectors must improve performance, to keep the problem computationally tractable it would be best to keep these numbers reasonably small. In practice, a data set size of about $n = 5000$ and number of principle components $T = 20$ perform well. Plotted in Figure 1 are the original and reconstructed images for the first 10 training images using such a scheme.

Figure 1: First row: original images, second row: reconstructed images after projection to 784x10-dimensional subspace constructed using 5000 training images

Plotted in Figure 2 are the first 20 eigenvectors. Note that the first 10 eigenvectors correspond somewhat to the general form of some of the 10 digit classifications in the data set. Those that follow are less physically meaningful and harder to interpret, but as shown in the following section still represent an important set of features of the images.

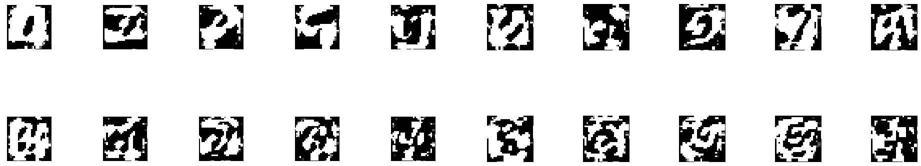

Figure 2: Eigenvectors corresponding to 20 largest eigenvalues of data covariance matrix: 5000 training images

3.2 Image recognition performance

As discussed in the previous section, the selection of a basis for the pixel data subspace affects the accuracy of the representation formed by projecting the data. This ultimately affects how well the k-nearest neighbor algorithm performs at classification, since the data will be more or less granular based on these parameters. The metric used to assess the classification performance was the percent of numbers successfully identified, PS. Figure 3 shows a plot of PS for both of the test data sets against the number of training images n, for a fixed number of eigenvectors in the basis, $T = 10$. Figure 4 shows a plot of PS for both of the test data sets against the number of eigenvectors in the basis T, for a fixed number of training data, $n = 2000$. Both plots also include computation time on the right y-axis. Note that the majority of the computation time is spent searching the subspace for the nearest neighbor to the newly projected data point (i.e. most of the work is in predicting, rather than in model building). It should also be noted that increasing n and/or T will significantly increase the time required to define the projection subspace; however, these times were not reported since it is assumed this could be done off-line (a priori) in an application.

To get both test data sets success rate above 85%, $n = 5000$ training images are required, which takes about 10 seconds for both test data sets together. It appears that increasing the size of the training data yields diminishing returns above this number, and the computation time may become prohibitive, although it's still relatively fast. For comparison, performing the search on the full-dimensional space for $n = 10000$ requires 459.82 seconds as compared to 14.08 seconds in this subspace.

For a fixed number of training images, the success rate is also highly dependent on the number of eigenvectors used to construct the basis. As seen in Figure 4, significant improvements are obtained by increasing this dimension up to about $T = 20$. Interestingly, the computation time is also near a minimum around this value.

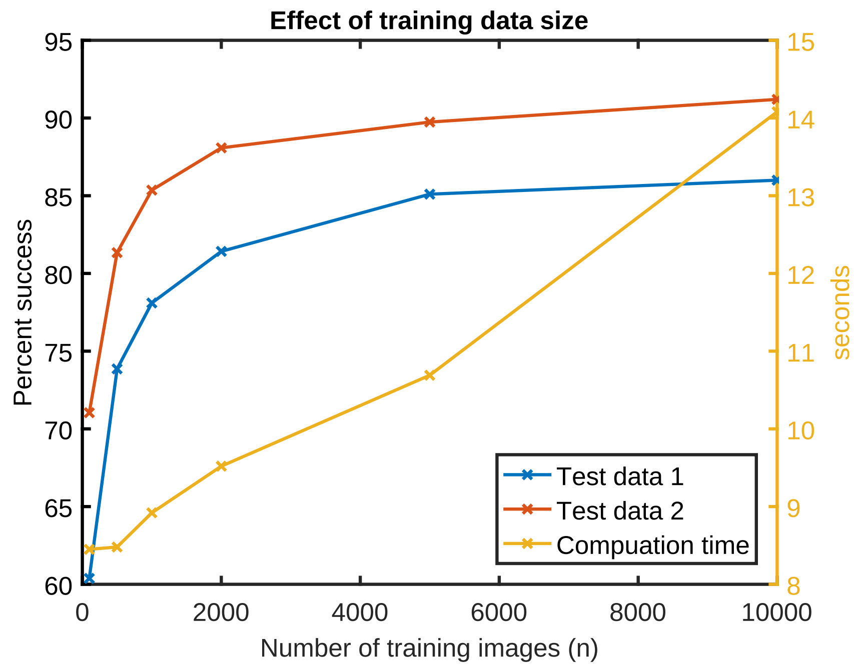

Figure 3: Effect of n; T=20, k=1

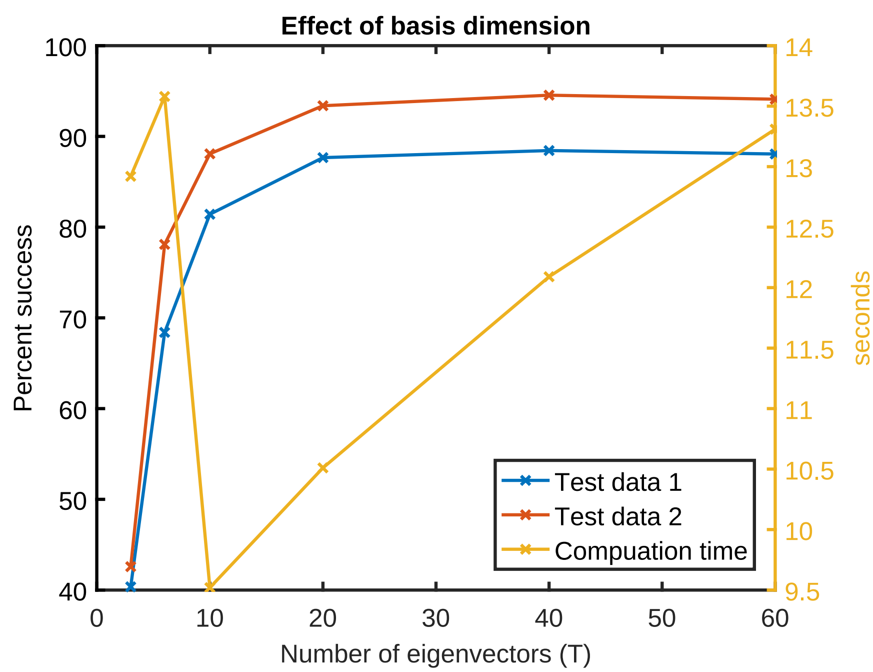

Figure 4: Effect of T; n=2000, k=1

The number of nearest neighbors k is also expected to have a considerable impact on the percent success. Note that in the previous examples, $k = 1$ was used. Next, the value of k was varied while holding constant $n = 2000$ and $T = 20$ since these values provided a good trade-off between speed and performance. The results are plotted in Figure 5. The number of nearest neighbors considered had neglible impact on the computation time so it was not included.

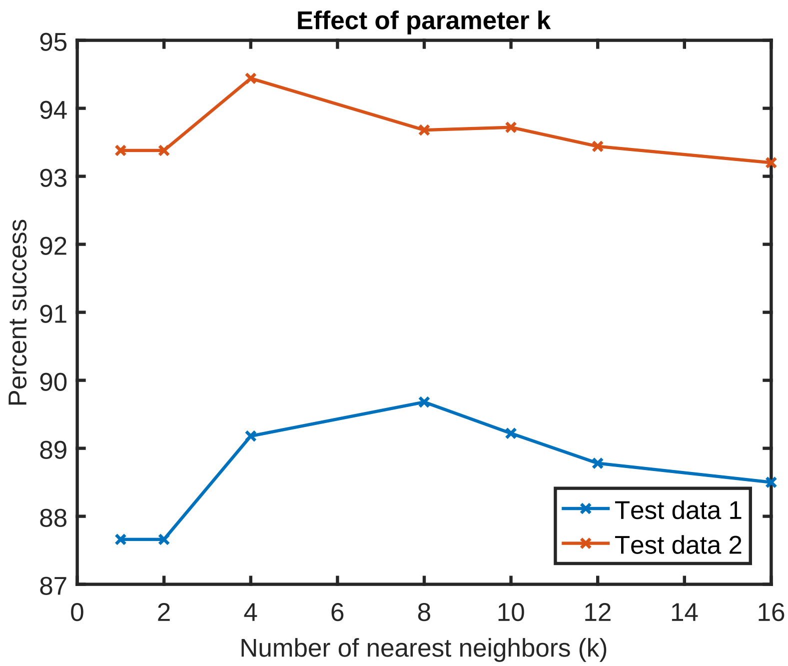

Figure 5: Effect of k; n=2000, T=20

The figure suggests that an optimal value of k is in the range of 4-8 for this data set.

4 Conclusions

A training data set consisting of pixel data for handwritten digit images was used to construct - via principal component analysis - a lower dimensional subspace for the pixel data. Then, a k-nearest neighbors model was used to classify test images by searching the subspace for the closest projected trianing image(s). Reasonable success rates of 89.7% and 93.7% for the respective data sets can be obtianed in about 10 seconds, using 2000 training images, a subspace spanned by the largest 20 principal components of the training data covariance matrix, and a number of nearest neighbors in the range of 4-8.