Face Recognition – well, sort of...

Charles Nicholson, Ph.D.

August 27, 2020

1 Problem Overview

Instead of performing facial recognition for this problem, you will need to use those same techniques as discussed in class to perform handwritten *digit* recognition. Handwritten digits, automatically scanned from envelopes by the U.S. Postal Service have been deslanted and size normalized, resulting in 16 x 16 grayscale images (Le Cun et al., 1990).

The image data has been transformed into text data: 256 variables associated with the grayscale value of every pixel for each image. Each record represents someone's handwriting of a single digit 0 thru 9. These data were kindly made publicly available by the neural network group at AT&T research labs. There is one file available per digit on the course website. The actual image files are available there as well. (Note: the images are interesting artifacts and not necessary for this problem.)

2 Let's get started

The file "ClassDigits.csv" from the course website contains data for 30,000 images, each corresponding to a 28×28 B&W image of handwritten digits, $0, \ldots, 9$. Each digit is represented in 784 dimensions. The data file is a comma separated values (CSV) text file. The first column of the raw data is the label associated with the digit (i.e., if the handwritten digit is a 0, then the label equals 0; if the handwritten digit is a 1, then the label equals 1; etc.) The remaining data are pixel values ranging between 0 and 255.

The overall goal is to compute the eigenvectors of the digit data and use these eigenvectors to create a significantly lower dimensional representation of the handwritten images in order to perform digit recognition. Note: do *not* scale the image data during the PCA analysis. Finally, the image recognition testing will be applied to the 7 observations in the “Class7Test.csv” file.

2.1 Tasks

1. Load the file `ClassDigits.csv`.
2. Compute the eigenvectors of the digit data.
3. Create a JPG image of the mean digit. Name this file `meanDigit.jpg`
4. Reconstruct two training images (image #15 and #100) based on $k = 5, 20,$ and 100 principal components. Name these files `image15-5.jpg, image15-20.jpg, image15-100, image100-5`, . . .”.
5. Choose a value for $k \ll 784$ based on the PCA summary or a screeplot. Using this value of k, for each of the 7 observations in the *test* data, determine the average mahalanobis distance from “digit-space”. Describe the results.
6. Load the file `Class7Test.csv`.
7. For the *test* images, 4, 5, and 6, determine the lowest value of k principal components that you need to correctly identify the 10 digits. The value of k may be different for each test image.

Please note that matrix multiplication in R is performed using `%*%`. The results of this operation is dependent on the the matrices being conformable.

2.2 How to create JPG’s from the data

To create a JPG image of raw numeric data, there are a couple of required steps:

1. First, install and load the `jpeg` library in R.
2. Next, if you have a numeric vector X of $28\times28=784$ values, you need to force this into a 28×28 matrix. Example code which creates a new R object in the right shape is shown here:

   ```
   digitMatrix <- matrix(X,28,28,byrow=TRUE)
   ```

3. Note: the `jpeg` R package expects the data to be between 0 and 1, so it may be necessary to scale the data before going to the next step.
4. Finally, to create the JPG file:

   ```
   writeJPEG(digitMatrix,target="FileName.jpg")
   ```

   where “FileName.jpg” should be replaced with the desired location and filename.