Homework 3: Handwritten Digit Recognition (updated)

STAT 665

due Thurs. Feb. 16th 5PM

Update on Feb. 9

Part 2 is now worth 40%. Part 3 (scroll down) has been added, and is worth 30%.

Background

In this homework assignment, we will consider the multiclass classification problem of identifying handwritten digits. The dataset consists of pixel data for 16 by 16 images, each of which contains a single handwritten digit from 0 to 9. Each pixel is either on (1) or off (0).

There are 3 datasets:

- 'digits_train.csv' - a training set
- 'digits_valid.csv' - a validation set
- 'digits_test.csv' - a test set

The training and the validation set each have 257 columns. The first 256 (our predictors) are binary values (0 or 1) indicating whether the pixel is activated. The last column is a 'digit' column with the actual digit that was meant to be represented (the true class label). Of course, the test set does not have this true class label.

For reference, if we look at the 4th row in the training set, you can fill a 16 by 16 matrix one column at a time to obtain the following:

```
matrix(train[4,-257], 16, 16, byrow=TRUE)
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,] 0    0    0    0    0    0    0    1    1    1     1     1     0
##  [2,] 0    0    0    0    1    1    1    1    1    0     1     1     1
##  [3,] 0    0    0    1    1    1    0    0    0    0     0     1     1
##  [4,] 0    0    0    1    1    0    0    0    0    0     0     1     1
##  [5,] 0    0    0    1    1    0    0    0    0    0     0     1     1
##  [6,] 0    0    0    1    1    0    0    0    0    0     1     1     1
##  [7,] 0    0    0    0    1    1    1    1    0    1     1     1     0
##  [8,] 0    0    0    0    0    0    0    0    1    1     1     0     0
##  [9,] 0    0    0    0    0    0    0    1    1    1     0     0     0
## [10,] 0    0    0    0    0    0    1    1    1    0     0     0     0
## [11,] 0    0    0    1    1    1    1    1    0    0     0     0     0
## [12,] 1    1    1    1    1    1    1    1    0    0     0     0     0
## [13,] 1    1    1    1    0    0    1    1    1    1     0     0     0
## [14,] 0    0    0    0    0    0    0    1    1    1     1     0     0
## [15,] 0    0    0    0    0    0    0    0    0    1     1     1     1
## [16,] 0    0    0    0    0    0    0    0    0    0     0     1     1
##       [,14] [,15] [,16]
##  [1,] 0     0     0
##  [2,] 0     0     0
##  [3,] 0     0     0
##  [4,] 0     0     0
##  [5,] 0     0     0
##  [6,] 0     0     0
```

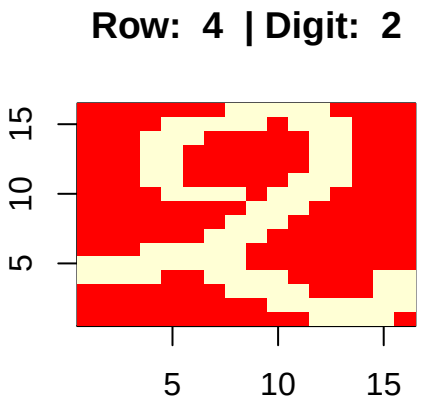
```
##  [7,] 0     0     0
##  [8,] 0     0     0
##  [9,] 0     0     0
## [10,] 0     0     0
## [11,] 0     0     0
## [12,] 0     0     0
## [13,] 0     1     1
## [14,] 0     1     1
## [15,] 1     1     1
## [16,] 1     1     0
```

I've written a function that will plot this for us:

```
plotDigit <- function(k, dat) {
  p <- matrix(as.numeric(dat[k,1:256]),16,16)
  image(x=1:16, y=1:16, p[,16:1], xlab="", ylab="",
        main=paste("Row: ", k, " | Digit: ", dat[k,257]))
}
```

So to see an image of the 4th row, we can do:

```
plotDigit(4, train)
```



Part 1 (30%)

Which digits do you think will be most difficult to distinguish between or classify? Show a few images (filling no more than 1 page) to justify your answer.

You may use my function above to plot, if you prefer.

Part 2 (40%)

Train a k-nearest neighbor and a linear discriminant analysis classifier using your training set. Of course, you may need to find a good value of k, and the validation set could be helpful for that part. You don't have to write your own implementation of any of these, so go ahead and use any pre- written algorithms. However, you should ask before you use any fancy model-selection packages (and I'll likely say no). Show your code.

(Note: Use standard Euclidean distance for k-nearest neighbors classification. And yes, we will use LDA even though the 256 pixels are clearly not meant to be modeled by normal distributions.)

(1) In no more than 1 paragraph, summarize the performance of the 2 approaches. What error rates do you achieve from each?

(2) Examine a confusion matrix for your best model. Which digit(s) is(are) relatively difficult to classify? How does this compare to your initial guesses from Part 1?

(3) In 1-2 sentences, explain why it would be difficult to apply a multinomial logistic regression to this problem.

Finally, create a .csv file called `HW3_netid.csv` containing the predictions for the test set from your best k-nearest neighbors classifier and LDA. The file should contain columns named (in order):

- `knn_pred`: predicted digit from k-nearest neighbors
- `lda_pred`: predicted digit from LDA

That makes for a total of 2 columns and 638 rows, with 1 additional header row. Of course, your rows should correspond to the rows in the test set in the order they appear.

Part 3 (30%): Newly added

In this newly added part of the homework, we will explore the use of multinomial logistic regression with a dimension-reduction pre-processing step, achieved using linear discriminant analysis.

Start by running LDA on your training set to obtain the 9 linear discriminant variables (projection vectors). Then, use the `predict()` function **twice** to project your training data and your validation data using these vectors. Now, you should have 9 potential predictors `LD1`, `LD2`, ..., and `LD9` for use in predicting `digit`.

Consider the following 9 nested multinomial logistic regression models for predicting `digit`:

```
digit ~ LD1
digit ~ LD1 + LD2
...
digit ~ LD1 + LD2 + ... + LD9
```

Which of these models achieves the lowest error on the validation set? Show your work and report the lowest error achieved.

(There is no need to make a prediction for the test set for this part.)

What to Submit

- `HW3_NETID.csv` from Part 2
- A compiled .Rmd or .ipynb file in either Word or PDF format documenting your work from Parts 1 through 3 (including loading the data). As always, you may compile to HTML first, and then save that as a PDF. Please ensure that there is no extraneous output in this file unless you intend to discuss them. For example, you should not be printing entire rows of the datasets.