Homework 3

Statistics W4240: Data Mining
Columbia University
Due Friday, March 4 (Section 01)

For your .R submission, submit a file for questions 3 and 4 labeled `hw03_q3.R` and `hw03_q4.R`, respectively. The write up should be saved as a .pdf of size less than 4MB. **DO NOT** submit .rar, .tar, .zip, .docx, or other file types.

Problem 1. (15 Points) James 3.7.3

Problem 2. (15 Points) James 3.7.4

Problem 3. (30 Points) Load the data set `hw03_q3.csv`. Use the first 10,000 observations as a training set and the last 5,000 observations as a testing set.

a. (10 Points) Write a function `knn_regression(x_train, y_train, x_test, y_test, k_vec)` that takes the arguments: `x_train`, a matrix of training covariates; `y_train`, a vector of training responses; `x_test`, a vector of testing covariates; `y_test`, a vector of testing responses; and `k_vec`, a vector (or scalar) of k-values. The output should be a list of the following elements: `train_mse`, a vector of training MSE for each value of k; and `test_mse`, a vector of testing MSE for each value of k. Use Euclidean distance to find the nearest neighbors. Run this function on the data using only the first covariate, $X1$, and on the same graph plot the MSE for training and testing sets as a function of k for

$$k = \{1, 5, 10, 50, 100, 500, 1000, 5000, 10000\}.$$

Include the graph in your writeup. Notes: `apply()` tends to be faster than looping for applying the same function to each row/column of a large matrix; and the graph may be easier to interpret if you use the plotting argument `log = "x"`.

b. (5 Points) What does the graph from part (a) tell you about the training error as a function of k? What about the testing error? What seems to be the best value for k?

c. (5 Points) Rerun the function from part (a), but varying the number of covariates from 1 to all 100 (always keep the first covariate in your set as it is the only one with signal). For each number of covariates, plot the lowest value of testing MSE. On a second graph, plot the k that produces the lowest testing MSE for each number of covariates. Include both graphs in your writeup.

d. (10 Points) What does the first graph from (c) tell you about about the predictive value of kNN as the number of covariates increases? What does the second graph from (c) tell you about the best neighborhood size as the number of covariates increases?

Problem 4. (40 Points) In this problem, we will use 1NN classification and PCA to do facial recognition.

a. (5 Points) Load the views `P00A+000E+00`, `P00A+005E+10`, `P00A+005E-10`, and `P00A+010E+00` for all subjects in the CroppedYale directory. Convert each photo to a *vector*; store the collection as a matrix where each row is a photo. Give this matrix the name `face_matrix_4a`. For each image, record the subject number and view in a data frame. The subject numbers will be used as our data labels.

   Use the following commands to divide the data into training and testing sets:

```
fm_4a_size = dim(face_matrix_4a)
# Use 4/5 of the data for training, 1/5 for testing
ntrain_4a = floor(fm_4a_size[1]*4/5)
ntest_4a = fm_4a_size[1]-ntrain_4a
set.seed(1)
ind_train_4a = sample(1:fm_4a_size[1],ntrain_4a)
ind_test_4a = c(1:fm_4a_size[1])[-ind_train_4a]
```

   Here `ind_train_4a` is the set of indices for the training data and `ind_test_4a` is the set of indices for the testing data. What are the first 5 files in the training set? What are the first 5 files in the testing set?

b. (5 Points) Do PCA on your training set and use the first 25 scores to represent your data. Specifically, that means creating the mean face from the training set, subtracting off the mean face, and running `prcomp()` on the resulting image matrix. Project your testing data onto the first 25 loadings so that it is also represented by the first 25 scores. Do not rescale the scores. Use 1NN classification in the space of the first 25 scores to identify the subject for each testing observation. In class we discussed doing kNN classification by majority vote of the neighbors; in the 1NN case, there is simply one vote. How many subjects are identified correctly? How many incorrectly? Plot any subject photos that are misidentified next to the 1NN photo prediction.

c. (10 Points) Rerun parts (a) and (b) using the views `P00A-035E+15`, `P00A-050E+00`, `P00A+035E+15`, and `P00A+050E+00` for all subjects in the CroppedYale directory. Give this matrix the name `face_matrix_4c`. For each image, record the subject number and view in a data frame. Use the following commands to divide the data into training and testing sets:

```
fm_4c_size = dim(face_matrix_4c)
# Use 4/5 of the data for training, 1/5 for testing
ntrain_4c = floor(fm_4c_size[1]*4/5)
ntest_4c = fm_4c_size[1]-ntrain_4c
set.seed(2)
ind_train_4c = sample(1:fm_4c_size[1],ntrain_4c)
ind_test_4c = c(1:fm_4c_size[1])[-ind_train_4c]
```

   Do PCA on your training set and use the first 25 scores to represent your data. Project your testing data onto the first 25 loadings so that it is also represented by the first 25 scores. Use 1NN in the space of the first 25 scores to identify the subject for each testing observation. Do not rescale the scores. How many subjects are identified correctly? How many incorrectly? Plot any subject photos that are misidentified next to the 1NN photo prediction.

d. (5 Points) Rerun part (c) with 10 different training and testing divides. Display the number of faces correctly identified and the number incorrectly identified for each. What do these numbers tell us?

e. (10 Points) Compare the results for parts (b) and (c). Are the testing error rates different? What does this tell you about PCA?

f. (5 Points) What happens if we use uncropped photos? Why? Some examples are included in the Files and Resources folder of Courseworks. If you would like to try PCA/kNN on the uncropped photos (not required to answer this question, but recommended), you will need to reduce the image sizes. Photos for subjects 1 to 10 do not currently exist in the uncropped database.