Homework 2

Statistics S4240: Data Mining
Columbia University
Due Thursday, July 23

For your .R submission, submit a file for question 2, 5, and 6 labeled `hw02_q2.R`, and so on. The write up should be saved as a .pdf and under 8MB.

DO NOT submit .rar, .tar, .zip, .docx, or other file types.

Problem 1. (35 Points) In this problem we will manually go through all of the steps for PCA. Basic computations like finding the eigenvalues for a matrix may be done using R.

a. (2 Points) Load `hw02_q1_p1.csv`. Find the column means and the row means for the data. What do these values tell us about this data set?

b. (3 Points) Center the data and and find the empirical covariance matrix, $\hat{\Sigma}$. This should be a 5-by-5 matrix. What do the diagonal values of the covariance matrix tell us about this data set? What do the off diagonal elements tell us about this data set?

c. (5 Points) Give the eigenvalues and associated eigenvectors of $\hat{\Sigma}$. Why does this matrix have the same left eigenvectors as right eigenvectors?

$$\mathbf{x}_{left}^T \hat{\Sigma} = \lambda \mathbf{x}_{left}^T, \quad \hat{\Sigma}\mathbf{x}_{right} = \lambda \mathbf{x}_{right}$$

d. (5 Points) Give all of the loadings and all of the scores for the data.

e. (5 Points) Plot the proportion of variance captured against the number of components included. How many components should we include and why?

f. (5 Points) Load `hw02_q1_p2.csv`. This has 5 new observations in the original coordinates. Give their scores.

g. (5 Points) Now use only the first two scores to represent the observations from the previous part. What are the coordinates of the projections in the original space, $\mathbf{x}'$? What is their Euclidean distance from the original data points?

h. (5 Points) Define the error of a point as

$$d(\mathbf{x}', \mathbf{x}) = \mathbf{x}' - \mathbf{x},$$

which is a 5-dimensional vector of errors. In what direction is $d(\mathbf{x}', \mathbf{x})$ for the 5 new points? Why do you think this is?

Problem 2. (65 Points) We will continue working with the Yale Faces B data set from the last homework with the goal of representing the images using PCA. We will use four lighting conditions, `P00A+000E+00`, `P00A+005E+10`, `P00A+005E-10`, and `P00A+010E+00`, which are closest to straight on lighting. We will use the `pixmap` library to manipulate the data. Load this library and make sure that the folder `YaleCropped` is in your working directory.

a. (10 Points) Load the views `P00A+000E+00`, `P00A+005E+10`, `P00A+005E-10`, and `P00A+010E+00` for all subjects. Convert each photo to a *vector*; store the collection as a matrix where each row is a photo. What is the size of this matrix?

b. (10 Points) Compute a "mean face," which is the average for each pixel across all of the faces. Subtract this off each of the faces. Display the mean face as a photo in the original size and save a copy as .png. Include this in your write up.

c. (10 Points) Use `prcomp()` to find the principal components of your image matrix. Plot the number of components on the x-axis against the proportion of the variance explained on the y-axis.

d. (10 Points) Each principal component is a picture, which are called "eigenfaces." Display the first 9 eigenfaces in a 3-by-3 grid. What image components does each describe? (Note: pixmapGrey() is fairly flexible and will automatically rescale data to have min 0 and max 1. You can do this manually or allow pixmapGrey() to do it.)

e. (15 Points) Use the eigenfaces to reconstruct `yaleB05_P00A+010E+00.pgm`. Starting with the mean face, add in one eigenface at a time until you reach 24 eigenfaces. Save the results in a 5-by-5 grid. Again, starting with the mean face, add in five eigenfaces at a time until you reach 120 eigenfaces. Save the results in a 5-by-5 grid. Include both of these in your write up. How many faces do you feel like you need until you can recognize the person?

f. (10 Points) Remove the pictures of subject 01 from your image matrix (there should be four pictures of him) and recenter the data. Rerun `prcomp()` to get new principal components. Use these to reconstruct `yaleB01_P00A+010E+00.pgm`. Do this by subtracting off the mean face and projecting the remaining image onto the principal components. Print the reconstructed image. Does it look like the original image? Why or why not?

Problem 3. (20 Points) James 3.7.3

Problem 4. (20 Points) James 3.7.4

Problem 5. (20 Points) Load the data set `hw02_q5.csv`.

a. (5 Points) Use the function `dist()` to produce a matrix of distances between all pairs of points. Distances should be computed for the two-dimensional input points $x = [x_1, x_2]$ (y is the output variable). Print the results.

b. (5 Points) Use the first data point as the testing set and the rest of the data as a training set. Implement kNN regression using the distance matrix from (a) for $k = 1, 2, \ldots, 10$. This algorithm should predict the y value of the first data point (with some error). Compute the mean squared error for the testing set and the mean squared error for the training set for each value of k; denote these values as $MSE^{k,1}_{test}$ and $MSE^{k,1}_{train}$.

c. (5 Points) Rerun part (b). For each data point: use the ith data point as a testing set, the remaining data as a training set, and run kNN for $k = 1, 2, \ldots, 10$ for observations $i = 2, 3, \ldots, n$.

For each value of k compute a mean squared error as follows:

$$MSE^k_{train} = \frac{1}{n}\sum_{i=1}^{n} MSE^{k,i}_{train}$$

$$MSE^k_{test} = \frac{1}{n}\sum_{i=1}^{n} MSE^{k,i}_{test}.$$

d. (5 Points) The results from part (c) are called *leave one out cross-validation* error. They are commonly used for estimating prediction error and selecting model parameters. Use these results to pick the optimal value for k. Should you make your choice based on MSE^k_{train} or MSE^k_{test}, and why? What is the optimal choice of k, and why?

Problem 6. (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.