**DATA 558**
**SPRING QUARTER 2022**

**Homework # 5**
**Due Via Online Submission to Canvas: Wednesday, June 8 at 5PM**

*Instructions:*

You may discuss the homework problems in small groups, but you must write up the final solutions and code yourself. Please turn in your code for the problems that involve coding. However, code without written answers will receive no credit. To receive credit, you must explain your answers and show your work. All plots should be appropriately labeled and legible, with axis labels, legends, etc., as needed.

**On this assignment, some of the problems may involve random number generation. Be sure to set a random seed (using the command `set.seed()`) before you begin.**

1. *A note before you begin: In this problem, I will ask you to "write out an expression for a linear model." For instance, if I asked you to write out an expression for a linear model to predict an $n$-vector $\mathbf{y}$ using the columns of an $n \times p$ matrix $\mathbf{X}$, then here's what I'd want to see:*

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip} + \epsilon_i,$$

*where $\epsilon_i$ is a mean-zero noise term.*

Suppose we have an $n \times p$ data matrix $\mathbf{X}$, and a continuous-valued response $\mathbf{y} \in \mathbb{R}^n$.

We saw in lecture that the $m$th principal component score vector is a linear combination of the $p$ features, of the form

$$z_{im} = \phi_{1m} x_{i1} + \phi_{2m} x_{i2} + \ldots + \phi_{pm} x_{ip} \tag{1}$$

(e.g. see (12.2) and (12.4) in textbook).

In *principal components regression*, we fit a linear model to predict $\mathbf{y}$, but instead of using the columns of $\mathbf{X}$ as predictors, we use the first $M$ principal component score vectors, where $M < p$.

   (a) Write out an expression for the linear model that we are fitting in principal components regression. Your answer should involve $y_i$, $z_{i1}, \ldots, z_{iM}$, a mean-zero noise term $\epsilon_i$, and some coefficients.

(b) Now plug in Equation 1 from this homework to your answer from (a), in order to express the principal components regression model in terms of $x_{i1}, \ldots, x_{ip}$.

(c) Use your answer from (b) to argue that the principal components regression model is linear in the columns of $\mathbf{X}$.

(d) In light of your answer to (c), is the following claim true? Explain your answer. *Claim: Fitting a linear model to predict* $\mathbf{y}$ *using the first* $m$ *principal components will yield the same fitted values as fitting a linear model to predict* $\mathbf{y}$ *using the columns of* $\mathbf{X}$.

2. We saw in class that $K$-means clustering minimizes the within-cluster sum of squares, given in (12.17) of the textbook. We can better understand the meaning of the within-cluster sum of squares by looking at (12.18) of the textbook. This shows us that the within-cluster sum of squares is (up to a scaling by a factor of two) the sum of squared distances from each observation to its cluster centroid.

   (a) Show *computationally* that (12.18) holds. You can do this by repeating this procedure a whole bunch of times:

   - Simulate an $n \times p$ data matrix, as well as some clusters $C_1, \ldots, C_K$. (It doesn't matter whether there are any "true clusters" in your data, nor whether $C_1, \ldots, C_K$ correspond to these true clusters — (12.18) is a mathematical identity that should hold no matter what.)
   - Compute the left-hand side of (12.18) on this data.
   - Compute the right-hand side of (12.18) on this data.
   - Verify that the left- and right-hand sides are equal. (If they aren't, then you have done something wrong!)

   (b) *Extra Credit:* Show *analytically* that (12.18) holds. In other words, use algebra to prove (12.18).

3. In this problem, you will generate simulated data, and then perform PCA and $K$-means clustering on the data.

   (a) Generate a simulated data set with 20 observations in each of three classes (i.e. 60 observations total), and 50 variables.

   (b) Perform PCA on the 60 observations and plot the first two principal component score vectors. Use a different color to indicate the observations in each of the three classes. If the three classes appear separated in this plot, then continue on to part (c). If not, then return to part (a) and modify the simulation so that there is greater separation between the three classes. Do not continue to part (c) until the three classes show at least some separation in the first two principal component score vectors.

(c) Perform $K$-means clustering of the observations with $K = 3$. How well do the clusters that you obtained in $K$-means clustering compare to the true class labels?

*Hint:* You can use the `table` function in `R` to compare the true class labels to the class labels obtained by clustering. Be careful how you interpret the results: $K$-means clustering will arbitrarily number the clusters, so you cannot simply check whether the true class labels and clustering labels are the same.

(d) Perform $K$-means clustering with $K = 2$. Describe your results.

(e) Now perform $K$-means clustering with $K = 4$, and describe your results.

(f) Now perform $K$-means clustering with $K = 3$ on the first two principal component score vectors, rather than on the raw data. That is, perform $K$-means clustering on the $60 \times 2$ matrix of which the first column is the first principal component score vector, and the second column is the second principal component score vector. Comment on the results.

(g) Using the `scale` function, perform $K$-means clustering with $K = 3$ on the data *after scaling each variable to have standard deviation one.* How do these results compare to those obtained in (b)? Explain.

4. This problem involves the `OJ` data set, which is part of the `ISLR2` package.

(a) Create a training set containing a random sample of 800 observations, and a test set containing the remaining observations.

(b) Fit a support vector classifier to the training data using `cost=0.01`, with `Purchase` as the response and the other variables as predictors. Use the `summary()` function to produce summary statistics, and describe the results obtained.

(c) What are the training and test error rates?

(d) Use the `tune()` function to select an optimal `cost`. Consider values in the range 0.01 to 10.

(e) Compute the training and test error rates using this new value for `cost`.

(f) Repeat parts (b) through (e) using a support vector machine with a radial kernel. Use the default value for `gamma`.

(g) Repeat parts (b) through (e) using a support vector machine with a polynomial kernel. Set `degree=2`.

(h) Overall, which approach seems to give the best results on this data?