STA 314H1S: Problem Set 2

Except for question 4, the questions on this assignment are practice for the quiz on Friday, and are not to be handed in.

For question 4, please bring a printout of your output together with the source code. You will be asked to submit this printout at the end of the quiz.

1. (*Textbook Exercises*) Answer questions 1-5 from section 6.6.

2. (*Regularization*) Recall the Ridge regression estimate $\hat{\boldsymbol{\beta}}_{reg} = (\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\boldsymbol{y}$, where $\mathbf{X} \in \mathbb{R}^{n\times p}$ and $\boldsymbol{y} \in \mathbb{R}^n$.

   (a) Explain why the Ridge regression estimate is uniquely defined even when $p \geq n$ (*hint:* you need to show the matrix inverse in the expression of $\hat{\boldsymbol{\beta}}_{reg}$ is well defined).

   (b) When some features are perfectly correlated, will $\hat{\boldsymbol{\beta}}_{reg}$ still be well-defined?

   (c) True or False?

      i. Ridge regression can be used for model selection purposes.

      ii. Both Ridge and LASSO regression have closed-form solutions.

      iii. In terms of bias, regularized models likely perform better than the OLS model.

      iv. In terms of variance, regularized models likely perform better than the OLS model.

      v. Naive use of the elastic net method often induces high bias, because of the double penalty.

   (d) Now, assume the features $\mathbf{X}$ have been standardized to have zero mean and unit variance. Also, assume different features are uncorrelated (i.e. $\frac{1}{n}\sum_{i=1}^{n} x_{ij}x_{ik} = 0 \; \forall j \neq k$). Derive the Ridge regression estimate as a function of the OLS estimate.

   (e) Based on your answer from (d), what is the shrinkage factor of the Ridge regression estimate? Your answer should be a function of λ.

3. (*PCA*) Assume we want to carry out PCA on a set of p features $\mathbf{x} \in \mathbb{R}^p$, where the true distribution of the features is unknown. However, we collected n samples of these features, the design matrix $\mathbf{X} \in \mathbb{R}^{n\times p}$. These features are already centered, which means $\sum_{i=1}^{n} x_{ij} = 0 \; \forall j \in 1, ..., p$.

   (a) Write out the sample estimate of the covariance matrix $\Sigma = \text{Var}(\mathbf{x})$. Your answer should involve a simple matrix product.

   (b) Prove your answer in (a) is always a PSD, using the definition of PSD matrix.

   (c) Prove your answer in (a) is always a PSD, using the SVD of $\mathbf{X}$.

   (d) Show that the kth sample PC in this case, is just the kth eigenvector of the matrix $\mathbf{X}^T\mathbf{X}$.

(e) Assume the eigenvalues of $\mathbf{X}^T\mathbf{X}$ are $\{d_i\}_{i=1}^p$. What is the (sample) variance of the kth PC? What about the proportion of variance explained (PVE)? Show your derivations.

(f) Show that two different PCs will be uncorrelated (in sample). Can we conclude that the two PCs are independent?

4. (*Computation*) In this question, we are going to implement different model selection & regularization methods, using the cross-validation approach. We assume $n = 1000$ observations with $p = 500$ are generated from the following regression model:

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\boldsymbol{y} \in \mathbb{R}^n$, $\mathbf{X} \in \mathbb{R}^{n\times p}$, $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$, $\boldsymbol{\beta} \in \mathbb{R}^p$. Each feature vector $\mathbf{x} \in \mathbb{R}^p$ is assumed to follow a standard multivariate normal distribution. The vector $\boldsymbol{\beta}$ is independent of other components in the model, with each element $\beta_i \overset{iid}{\sim} N(0, 0.01)$.

(a) Compute the value of σ such that the feature vector $\mathbf{x}$ is expected to explain 50% of the variance in $\boldsymbol{y}$.

(b) Using your answer from (a), simulate $n = 1000$ observations, and then randomly split them into a training set with size 700 and a test set with size 300.
Some starter code to help you get started:

```r
library(glmnet)
library(tidyverse)
set.seed(1234)
## Simulate 1000 obs, 500 features
n <- 1000; p <- 500
## Assume all features are iid normal
X <- #TODO
## Simulate feature coefficients from normal
beta <- #TODO
y <- #TODO
full_data <- cbind(X, y) %>% as.data.frame()
training_data <- #TODO
test_data <-  #TODO
```

(c) We will first implement the Ridge regression method to this simulated dataset. Complete the following function, which takes the training set (`data`), the number of folds in the cross-validation (`k`) and the hyperparameter λ, and returns the mean MSE estimated from the CV procedure (*Note:* The function `createFolds` from the package `caret` can help you to create a partition of the indices of your training data).
Some starter code to help you get started:

```r
library(caret)
k_fold_cv_ridge <- function(data, k, lambda){
  mse_vec <- numeric(length = k)
  indx <- #TODO
  p <- #TODO
  kfolds <- createFolds(#TODO)
  for (i in 1:k) {
    selected_training_data <- #TODO
    selected_X <- selected_training_data[,1:p]
    selected_y <- selected_training_data[,(p+1)]
    selected_validation <- #TODO
    selected_validation <- as.matrix(selected_validation)
    mod_ridge <- glmnet(#TODO)
    mod_pred <- #TODO
    mse_vec[i] <- #TODO
  }
  mean(mse_vec)
}
```

(d) Use the function from (c) to implement 20-fold CV, and find the optimal choice of λ over the interval $[0, 10]$. Plot the mean MSE versus λ.
Some starter code to help you get started:

```
lambda_vec <- seq(0,10, by = 0.1)
mse_vec <- c()
for (i in 1:length(lambda_vec)) {
  mse_vec[i] <- k_fold_cv_ridge(#TODO)
}
best_lambda <- #TODO
plot(mse_vec~lambda_vec, ylab = "MSE", xlab = "lambda", type = 'o')
abline(v = best_lambda, col = "red")
```

(e) Modify your answers above for (c) and (d); Perform the same kind of 20-fold cross-validation for LASSO regression and plot the mean MSE versus λ.

(f) Compare the selected LASSO model and the selected Ridge model using the MSE estimated from CV. Also, fit these two models using the training data, and assess their MSE in the test set. Which one performs better? Is that aligned with your expectation?

(g) Compute the (sample) PCs of the p features and plot the screeplot. Note that you don't need to center your features since you know $\mathbb{E}(\mathbf{x}) = \mathbf{0}$ (*hint:* the function `eigen` in R does the eigendecomposition for you).

(h) Fit a linear model using the first ten (sample) PCs and assess its prediction accuracy by computing the MSE in the test set.