Homework 2

Due Thursday 2/10

Solutions for theoretical problems may be typed or handwritten and scanned. Solutions for applied solutions should include R code and output – try to use `R markdown`. Homework must be submitted on Blackboard as one file.

Mu Hu, Yuheng Hu, Baldur Hua and Hanxing Jiang will lead HW discussion.

1. (20 points) *Exploring Training vs. Test Error in Simulated Data – EoSL Figure 7.1.*

   This problem uses the same simulated data set-up as HW1 Q1. You will be calculating training and test error using k-NN for 100 simulated training datasets.

   As in HW1 Q1(a), generate a dataset of 100 observations for class $y = 1$ with x_1 drawn from $\mathcal{N}(0, 1)$ and x_2 drawn from $\mathcal{N}(1, 1)$ using `rnorm` in R. Next generate a dataset of 100 observations for class $y = 0$ with x_1 drawn from $\mathcal{N}(1, 1)$ and x_2 drawn from $\mathcal{N}(0, 1)$. Combine these two datasets to form a dataset of 200 observations with variables $y \in (1, 0), x_1$, and x_2.

   Repeat the data generating process 100 times to create 100 *training* datasets. Repeat the data generating process another 100 times to create 100 *test* datasets.

   (a) Using the first training dataset, implement 20 versions of k-NN with $k = (1, 2, \ldots, 19, 20)$ using the `knn` function in package `class`. Obtain the predicted class from each of the 20 `knn` objects for the training data and one of the test datasets. Plot the training error $\overline{\text{err}}$ (in light blue) and test error $\text{Err}_{\mathcal{T}}$ (in pink) as a function of k (from 20 to 1), where error is defined as misclassification rate (0-1 loss). Comment on the shape of the curves and the similarities/differences between the two. Hint: To create your version of EoSL Figure 7.1, you want the range of the x-axis to go from low complexity to high complexity (technically, from small to large degrees of freedom). In the case of k-NN, the least "complex" case is k=20 and the most "complex" case is k=1 (technically, the degrees of freedom is N/k).

   (b) Repeat part (a) for the other 99 training and 99 test datasets. In the same plot, you will have have 100 pink test error curves and 100 light blue training error curves. Comment on how all the light blue lines compare to each other and how all the pink lines compare to each other.

   (c) You will estimate the expected test error, Err, and expected training error, $\text{E}(\overline{\text{err}})$, with averages over the 100 replications. Find the average test and average training error across all 100 simulated datasets at each value of k. Plot the average training error (in blue) and the average test error (in red) as a function or k on your plot from part (b). Comment on the shape of these average curves and the similarities/differences between the two.

   (d) What k would you choose to best fit this simulated data? Explain your choice in the context of training vs. test data.

(e) Explain in words how this study of simulated data illustrates the bias-variance tradeoff. Hint: The bias-variance tradeoff is inherent in the value of test error (by the bias-variance decomposition). Models with low complexity tend towards high bias but low variance, whereas models with high complexity tend towards low bias but high variance. How do you see this balance play out in the shape of your estimated test error curve?

2. (40 points) *Comparing Test Error Estimates for a Linear Regression Model.*

You will be modeling COVID-19 case rates in the U.S. as a function of characteristics of the geographic area. County-level COVID-19 case counts (as of 1/25/22) are obtained from `https://usafacts.org/visualizations/coronavirus-covid-19-spread-map/`. The target (a.k.a. dependent, output) variable is the covid case rate for the county

$$100*(covid\_count \ / \ Tot\_Population\_ACS\_14\_18)$$

where the denominator is the total population of the county estimated from the 5-year (2014-2018) American Community Survey from the U.S. Census Bureau.

Geographic and demographic characteristics of each county are obtained from the Census Bureau's 2020 Planning Database (PDB). For now we will use the following predictors:

- $pct\_URBANIZED\_AREA\_POP\_CEN\_2010$: percent of urbanized land,
- $pct\_Males\_ACS\_14\_18$: percent males,
- $pct\_Pop\_65plus\_ACS\_14\_18$: percent 65+ years old,
- $pct\_Inst\_GQ\_CEN\_2010$: percent in group quarters (e.g. dorms),
- $pct\_Hispanic\_ACS\_14\_18$: percent Hispanic origin,
- $pct\_NH\_Blk\_alone\_ACS\_14\_18$: percent non-Hispanic Black,
- $pct\_Prs\_Blw\_Pov\_Lev\_ACS\_14\_18$: percent in poverty.

You can read about the data at `https://www.census.gov/topics/research/guidance/planning-databases.html` and get the detailed description of the predictors at `https://www.census.gov/content/dam/Census/topics/research/2020StateandCountyPDBDocumentation_V2.pdf`.

The dataset combining the two sources, *covid_data_pdb.csv*, is posted on Blackboard.

Using the `lm` function in R, fit the linear regression model for the log of the covid case rate as a function of all 7 predictors.

(a) Using only basic R functions*, obtain and report the estimate of test error, assuming squared error loss, using each of the following:

- AIC,
- BIC,

- 10-fold CV,
- 5-fold CV,
- LOOCV,
- bootstrap,
- leave-one-out bootstrap, and
- .632 bootstrap.

* You are required to program all the test error estimation methods using for loops, random sampling, etc. That is, you must program the methods from scratch. Do NOT use the special functions that calculate the estimates directly in one function call (e.g. `AIC`, `BIC`, `train`, `boot`). You may, however, use elements stored within your `lm()` object such as `fitted.values` and `residuals`.

(b) Calculate the training error and compare to the test error estimates. Do you think training error is a good estimate for generalization error for this model? Why or why not? What characteristics of the data and model (e.g. p, N, linear model) do you think contribute to any differences between the training error and the estimates of test error? Comment on the expected optimism.

(c) Compare the test error estimates you obtained from the different methods to each other. Do you find differences between in-sample and extra-sample test error estimates? Why or why not? What do your results suggest about the effect of varying the type of CV (i.e. $k = 10,\ 5,\ 1$) or bootstrap? What does this suggest about the bias/variance associated with the model?

3. (30 points) *Comparing Estimates and Inference for a Simple Linear Regression Model.*

Consider the simple linear regression of *log covid case rate* on $pct\_Inst\_GQ\_CEN\_2010$.

(a) Carry out maximum likelihood estimation using the `lm` function in R. Find the MLE $\hat{\beta}_{pct\_Inst\_GQ\_CEN\_2010}$ along with its estimated standard error and 95% confidence interval. Provide an interpretation of the MLE and the 95% confidence interval.

(b) Plot *log covid case rate* vs. $pct\_Inst\_GQ\_CEN\_2010$ and overlay the estimated regression line from part (a).

(c) Using parametric bootstrap* with $B = 1000$, find the parametric bootstrap estimate $\hat{\beta}_{pct\_Inst\_GQ\_CEN\_2010}$ and its estimated standard error and 95% confidence interval. Provide an interpretation of this slope parameter estimate and its 95% confidence interval.

(d) Plot *log covid case rate* vs. $pct\_Inst\_GQ\_CEN\_2010$ and overlay the estimated regression line based on $\hat{\beta}_{intercept}$ and $\hat{\beta}_{pct\_Inst\_GQ\_CEN\_2010}$ estimated from parametric bootstrap in red. Also plot each of the 1000 estimated regression lines on the same plot in pink. Comment on the similarities/differences between the 1000 regression lines.

(e) Repeat part (d) and (e) using nonparametric bootstrap* with $B = 1000$.

(f) Compare the estimates, standard errors and confidence intervals from the three approaches. Why do you think the estimates are similar or different? Why do you think

the largest estimated standard error is the largest? Why do you think the widest confidence interval is the widest?

* You may check your work with the relevant `R` packaged commands (e.g. `boot`) but your solutions must use base `R` commands such as loops and random sampling.

4. (10 points) *Cross-Validation for Choosing k in k-NN.*

Generate one dataset as in Q1 with 1000 total observations (500 from each class). For each value of $k = 1, \ldots, 100$, find the test error (assuming 0-1 loss) estimated using 5-fold cross-validation. Plot the 5-fold CV test error vs. k. Which k would you choose to fit this data and why? How does this choice compare to your choice from Q1? How is 5-fold CV different from doing one iteration of the methid in Q1 (i.e. doing part (a) of Q1 only)?

Note: we will be doing a lot of this during the semester – cross-validation is a very common way to choose "tuning" parameters in machine learning approaches.