STAT 380 – k-fold Cross Validation (Lecture 10)

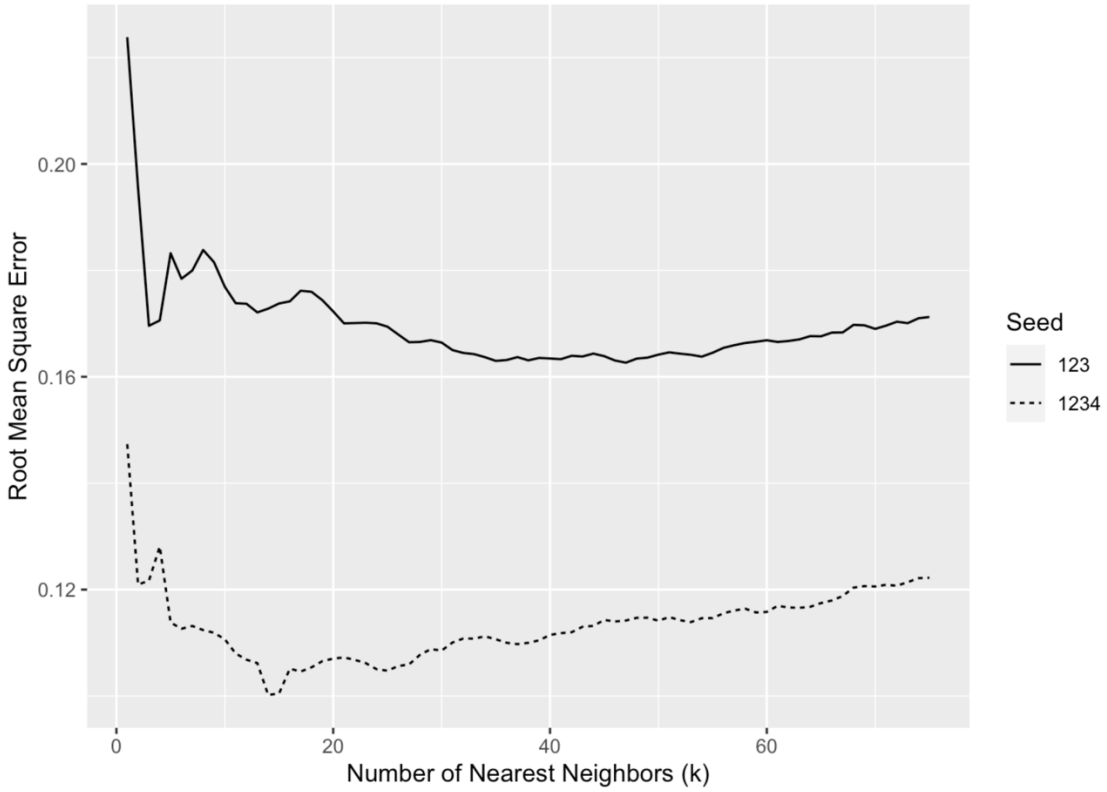
RECALL: We've been using something called the holdout (or validation) set approach in which we randomly divide the data into a training set and validation/holdout set (which we've called the testing set). We fit the model on the training set and see how well it generalizes to the new data in the testing set. For supervised learning problems with a quantitative response, this is often done by examining the RMSE for the testing set.

NOTE: Problems with this method:

1. Only a subset of the observations—those that are included in the training set rather than in the validation set—are used to fit the model. Since statistical methods tend to perform worse when trained on fewer observations, this suggests that the holdout set error rate may tend to overestimate the test error rate for the model fit on the entire data set.

2. The validation estimate of the test error rate can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the holdout set. This is especially true in small datasets.

EXAMPLE 1: In the last lecture, we used 2 seeds and compared the results while choosing *k* in the kNN algorithm. The plot is shown below. Notice the impact of the seed.



k-Fold Cross Validation

IDEA: We can use *k*-fold Cross Validation (*k*-fold CV) as an alternative to the validation/holdout set approach. Although *k*-fold CV is more computationally demanding, it reduces the effect of the random split. This is particularly true for small datasets.

NOTE: An excellent video explain *k*-fold CV may be found at:
https://www.youtube.com/watch?v=fSytzGwwBVw

Definition: *k*-fold Cross Validation (*k*-fold CV) involves randomly dividing the set of observations into *k* groups, or folds, of approximately equal size. The first fold is treated as a validation set, and the method is fit on the remaining *k* - 1 folds. The mean square error, denoted MSE_1, is then computed on the observations in the held out fold. This procedure is repeated *k* times; each time, a different group of observations is treated as a validation set. This process results in *k* estimates of the test error, $MSE_1, MSE_2, \ldots, MSE_k$. The *k*-fold CV estimate is then computed by averaging these values:

$$CV_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

NOTE: The figure below, Figure 5.5 from ISLR, illustrates the process for $k = 5$.

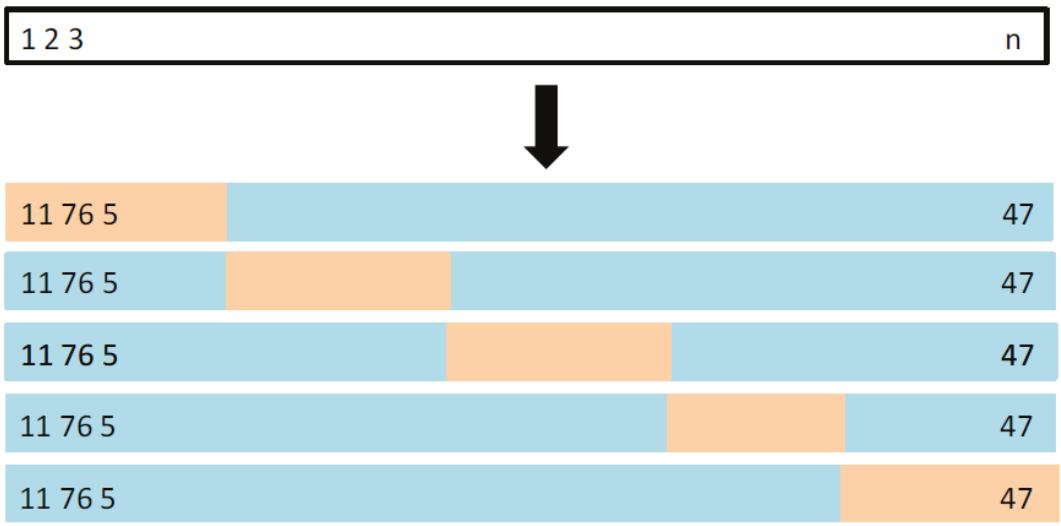


FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

EXAMPLE 2: Using the L08_bmd.csv, we want to perform *10*-fold CV. There are many ways to do this in R, but we will walk through the logic of one approach. Read in the dataset, create indicators for medication, and scale the variables age, weight, and medication indicators.

a. After reading in the dataset, creating indicator variables, and scaling the data, we can assign each observation a "fold." The following code assigns each observation in the dataset a fold number (a value from 1-10). Unfortunately, the folds values are in order.

```
num_folds <- 10
folds <- cut(x = 1:nrow(bmd), breaks = num_folds, labels = FALSE)
head(folds, 50)
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3
## [39] 3 3 3 3 3 3 3 3 3 3 3 3
```

```
table(folds)
```

```
## folds
##  1  2  3  4  5  6  7  8  9 10
## 17 17 17 17 17 16 17 17 17 17
```

b. We don't want to split our data in order (i.e., there is no reason the first 17 rows should be the first fold), so we will randomly permute the values in folds using a random seed of 123.

```
#Randomly permute the fold assignments
set.seed(123)
folds <- sample(folds)
set.seed(NULL)

#Notice impact or random permutation
folds
```

```
##   [1] 10  1  3  7  3 10 10 10  6  6 10  6  9  6  5  2  1 10  5  5 10  7  7  5  9
##  [26]  2  7 10  9  5  2  4  8  9  2  5 10  9  4  6 10  2  3  2  5  3  3 10  4  6
##  [51]  1  7  6  1  7  8  6  7  3  2  9 10  9  1  1  8  8  4  2  6  9  9  6  3  3
##  [76]  2  1  7  9  4 10  5  4  9  8  3  1  3  9  7  9  8  3  1  4  5  4  6  2  9
## [101]  7 10 10  5  2  3  1 10  8  6  8  8  4  1  3  3  5  2  6  7  4  1  4  7  7
## [126]  7  5  7  1  8  8  9  5  4  8  4  2  5  1  2  8  2  6  6  9  1  2  9  4  6
## [151]  3  1  3  5  7  5  3  4  7  8  8  4  8  4  8  5  2  1 10
```

c. Write a for loop that will cycle through the folds. For each iteration of the for loop, you should remove the data associated with one fold, build a regression model for predicting bmd using age, weight, and medication indicators on the remaining folds, compute the RMSE on the held out fold, and store the RMSE.

d. Find the 10-fold CV MSE ($CV_{(10)}$). (0.12966)

e. How many times did we build a regression model during the 10-fold CV process?

NOTE: One of the advantages of *k*-fold CV is that every data point is used both as a training point (k-1 times) and as a validation (testing) point (once).

EXAMPLE 3: Rewrite the code from Example 2 so that $k = n$.

a. Find the *n*-fold CV MSE. (0.100251)

b. How many times did we build a regression model during this process?

NOTE: The process in Example 3 is known as Leave One Out Cross Validation (LOOCV) and it is very computationally demanding; however, in the case of linear regression, there are shortcut formulas that only require us to build the model once. In particular:

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2$$