ST563 – Homework 1 Solution

Problem 1 (30 points)

Consider the model: for $i = 1, \ldots, n$,

$$Y_i = f(x_i) + \epsilon_i,$$

where $\epsilon_i \sim N(0, \sigma^2)$ independent over i. Assume that $x_1, \ldots, x_n$ are non-random. Suppose we apply KNN regression method, with a pre-specified value K, to estimate $f(\cdot)$ as a fixed point x_0.

(a) Compute the variance of $\widehat{f}(x_0)$. [Hint: Recall, $\widehat{f}(x_0)$ is just an average of Ys.]

(b) Using your answer in (a), can you verify that more flexible procedures tend to have more variance?

(c) Suppose we run two KNN regressions, one with $K = 10$ and the other with $K = 30$. For each model, we compute training MSE, and test MSE using an independent test sample. Which regression will have lower training MSE? Which one will have lower test MSE? Explain you answers.

Solution of Problem 1

(a) Compute the variance of $f(x0)$. [Hint: Recall, b f(x0) is just an average of Ys.]

$$Var(\hat{f}(x_0)) = Var(\frac{1}{K}\sum_{i=1}^{K} y_i) = \frac{1}{K^2}Var(\sum_{i=1}^{K} y_i) = \frac{1}{K^2}\sum_{i=1}^{K} Var(y_i) = \frac{\sigma^2}{K}.$$

(b) Using your answer in (a), can you verify that more flexible procedures tend to have more variance?

From (a) we know that $Var(\hat{f}(x)) \propto 1/K$, which indicates that more flexible (with a smaller K) KNN models tend to have more variance. Finally, the estimation of variance will go to σ^2 when $K = 1$.

(c) Suppose we run two KNN regressions, one with $K = 10$ and the other with $K = 30$. For each model, we compute training MSE, and test MSE using an independent test sample. Which regression will have lower training MSE? Which one will have lower test MSE? Explain you answers.

We have two models, and one with $K = 10$ is more flxible than another one with $K = 30$. We know that the traning MSE declines monotonically as flexibility increases, thus the one with **K = 10 has lower training MSE**. However, testing MSE initially declines as the level of flexibility increases, but it will start to incrase again after some point. If K corresponde with that "change point" is smaller than 10, then the KNN with **K = 10 has lower test MSE**; If the "change point" means a K lager than 30 then the KNN with **K = 30 has lower test MSE**; Similarly, we **cannot compare these two models** when the change point of K is between 10 and 30.

Problem 2 (50 points)

Consider the `Boston` data discussed in lectures. Now we want to build a prediction model for `medv` based on the remaining numeric variables in the dataset, excluding `chas`.

(a) Use the KNN regression method to build a predictive model where the hyperparameter is tuned using 5-fold cross-validation?

(b) Estimate the test error of your model using holdout method.

(c) Predict `medv` for the following new datapoint:

```
## # A tibble: 1 x 12
##    crim    zn indus  chas   nox    rm   age   dis   rad   tax ptratio lstat
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl> <dbl>
## 1 0.257     0  9.69     0 0.538  6.21  77.5  3.21     5   330    19.0  11.4
```

(d) How much does `medv` change if `lstat` changes from 5 to 10, while keeping the other variables fixed at their median value?

(e) Do you expect to see the same amount of change in `medv` for each increase of 5 units in `lstat`? Explain.

Solution of Problem 2

(a) Use the KNN regression method to build a predictive model where the hyperparameter is tuned using 5-fold cross-validation.

The model building method is shown below. Figure 1 shows the RMSE profile for tuning process.

```
# prep
set.seed(1)
library(ISLR2)
library(caret)
# Exclude CHAS
Boston_data <- Boston[,-4]
# tuning grid
kgrid <- expand.grid(k=c(1:50))
# Specify 5-fold CV
cv<-trainControl(method="cv",number=5)
# Train the model
KNN_fit <- train(medv~.,
               data = Boston_data,
               method = "knn",
               tuneGrid = kgrid,
               trControl = cv)
plot(KNN_fit)
```

```
k_opt <- KNN_fit$bestTune$k
k_opt
```

```
## [1] 5
```

The best KNN model is found with $K = 5$. Note that change of the seed may give different values of K. Now we refit the model on the entire data with optimal K.

```
# Final predictive model
final_knn_fit <- train(medv~.,
               data = Boston_data,
```

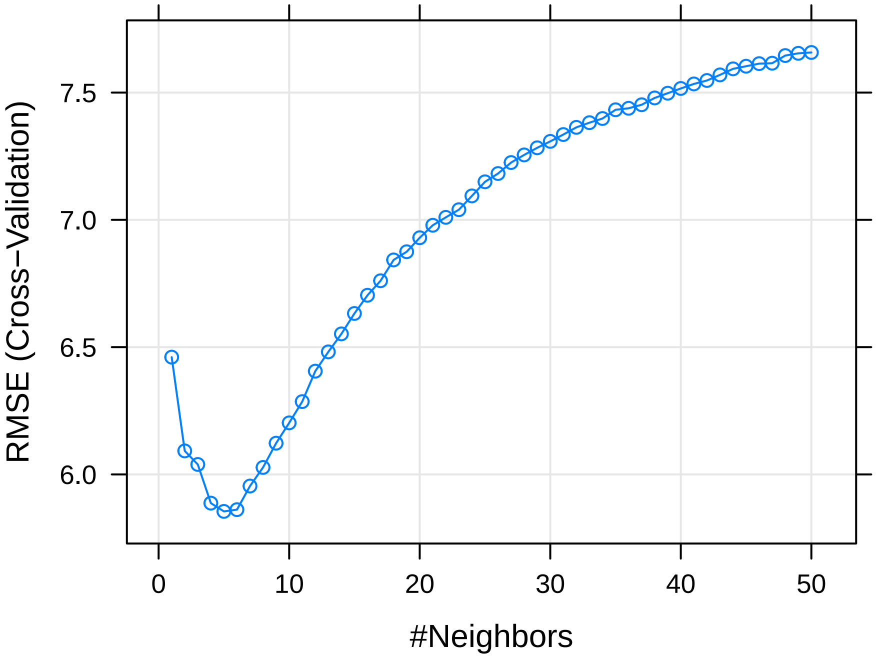

Figure 1: RMSE profile for parameter tuning.

```
               method = "knn",
               tuneGrid = expand.grid(k = k_opt),
               trControl = trainControl(method = "none"))
```

(b) Estimate the test error of your model using holdout method.

To estimate the test error, we first create a holdout set, and a training set. We apply the entire process in part (a) to the training set, and compute test error using the holdout set. For better estimation of test error, we repeat this process $B = 10$ times and take average of the test errors.

```
# get 10 hold out partitions (80%)
B <- 10
Boston_hold <- createDataPartition(Boston_data[ ,1],
                                   times = B,
                                   p = 0.8,
                                   list = TRUE,
                                   )

# test for each partitions
MAE <- MSE <- K_opt <- rep(0, B)

for (ii in 1:B){
  # get train vs test sets
  curr_ind <- Boston_hold[[ii]]
  Boston_train <- Boston_data[curr_ind, ]
  Boston_test <- Boston_data[-curr_ind, ]

  # tuning
  kgrid <- expand.grid(k = c(1:50))
  cv<-trainControl(method = "cv",
                   number = 5)
  KNN_fit <- train(medv~.,
               data = Boston_train,
               method = "knn",
               tuneGrid = kgrid,
               trControl = cv)

  # results
  K_opt[ii] <- KNN_fit$bestTune$k
  # train on training set
  KNN_opt <- train(medv~.,
               data = Boston_train,
               method = "knn",
               tuneGrid = expand.grid(k=K_opt[ii]),
               trControl = trainControl(method = "none"))
  # test on test set
  yhat <- predict(KNN_opt, Boston_test)
  # get MAE & RMSE
  MAE[ii] <- mean(abs(Boston_test$medv - yhat))
  MSE[ii] <- mean((Boston_test$medv - yhat)^2)
}

## Result
summary(MSE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   22.99   27.79   32.19   33.89   37.60   57.24
```

```
summary(MAE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.513   3.614   3.856   3.932   4.034   5.135
```

The estimated test MSE is the mean of MSE over the 10 repeats, 33.89. The estimated test MAE is the mean of MSE over the 10 repeats, 3.93

(c) Predict medv for the following new datapoint:

We need to use the final prediction model in part (a).

```
# New data point
new_data <- data.frame(crim = 0.257,
                       zn = 0,
                       indus = 9.69,
                       nox = 0.538,
                       rm = 6.21,
                       age = 77.5,
                       dis = 3.12,
                       rad = 5,
                       tax = 330,
                       ptratio = 19,
                       lstat = 11.4)
# Prediction
y_pred <- predict(final_knn_fit, newdata = new_data)
y_pred
```

```
## [1] 21.64
```

(d) How much does medv change if lstat changes from 5 to 10, while keeping the other variables fixed at their median value?

```
# get medians and change lstat
newpoints <- t(apply(as.matrix(Boston_data),2, median))
newpoints <- rbind(newpoints, newpoints)
newpoints[,11] <- c(5, 10)
# predict
y_preds <- predict(final_knn_fit, newpoints)
# result
y_preds
```

```
## [1] 31.74 24.16
```

```
# change in prediction
y_preds[2] - y_preds[1]
```

```
## [1] -7.58
```

We have a change of -7.58 when `lstat` changes from 5 to 10.

(e) Do you expect to see the same amount of change in medv for each increase of 5 units in lstat? Explain.

No. The KNN regression is a non-linear model – the response for 5 units change in the predictors will depend on the baseline value. For example, let's change `lstat` from 10 to 15.

```
newpoints[,11] <- c(10, 15)
# predict
y_preds <- predict(final_knn_fit, newpoints)
# result
y_preds
```

```
## [1] 24.16 20.40
```

```
# change in prediction
y_preds[2] - y_preds[1]
```

```
## [1] -3.76
```

The change in response is now -3.76.

Problem 3 (20 points)

Do Problem 2 in Chapter 2.4 in “An Introduction to Statistical Learning”, second edition".

Solution of Problem 3

(a) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary.

Regression problem; interested in inference; $n = 500$ and $p = 3$.

(b) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

Classification problem; interested in prediction; $n = 20$ and $p = 13$.

(c) We are interested in predicting the % change in the USD/Euro exchange rate in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the USD/Euro, the % change in the US market, the % change in the British market, and the % change in the German market.

Regression problem; interested in prediction; $n = 52$ and $p = 3$.