Elif Küçük
0040851
COMP 421
HW 04 Report.

COMP/INDR 421/521 HW 04: Nonparametric Regression– Report

This file briefly explains each function and variable defined in the project. The defined function names and variables are written in bold. The code also has the comment lines to make understating easier.

Question 1

In order to separate data into two sets of vector with random order I used sample function of R and chosen 100 indices for training data. To separate test from the training data I used setdiff function of R and took the indices for test data.

In the HW we store the training and test data in **data_train** and **data_test** respectively.

Question 2

In order to learn a regressogram I defined **bind_interval** which is two-column vector that stores bind start and ending x values.

In the function **reg_est** for particular x value we are estimating the g(x) using the Regressogram equation that is shown below.

$$\widehat{g(x)} = \frac{\sum_{t=1}^{N} b(x, x^t) * r^t}{\sum_{t=1}^{N} b(x, x^t)} \; where \; b(x, x^t) = 1 \; id \; x^t is \; the \; same \; bin \; with \; x \; and \; t = 1{:}N$$

By defining a **data_interval** starting from **minimum_value** to **maximum_value** incremented by 0.1, we are estimating **p_head** for each data point in data_interval by function **reg_est**. I used lines function to draw data_interval vs p_head on the plot.

Running the code below we get the regrossogram plot.

```
p_head_plot<-sapply(data_interval, reg_est)
plot(data_train[,1],  data_train[,2], type = "p", pch = 20, col = "RED",
    ylab = "density", xlab = "x")
points(data_test[,1], data_test[,2], type = "p", pch = 20, col = "BLUE" )
lines(data_interval, p_head_plot, type = "l", lwd = 2, col = "black")
```

Elif Küçük
0040851
COMP 421
HW 04 Report.

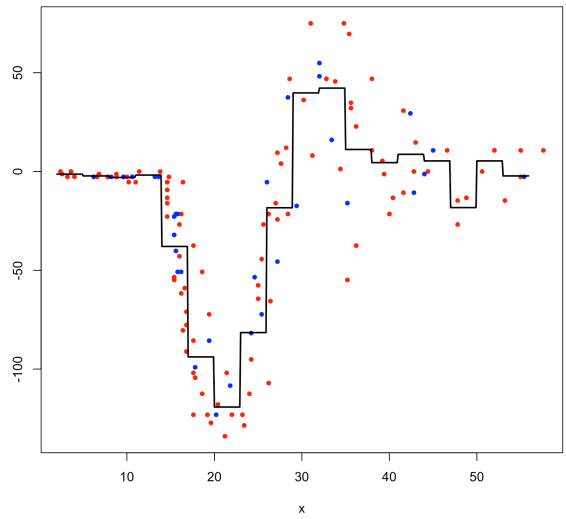

Figure 1

Question 3 - 5 - 7

For Question 3, 5 and 7 we are calculating the mean squared error. To make the coding more efficient I defined a function **find_mse(p_h,s)** which takes the **p_head** for estimated value and s (string) to print which method we are using to estimate.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_{test}} (y_i - \widehat{y_i})}{N_{test}}}$$

The output of RMSE for Question 3 is :

```
> p_head<-sapply(data_test[,1], reg_est)
> find_mse(p_head, "Regressogram")
[1] "Regressogram=> RMSE is 19.098845 when h is 3"
>
```

Figure 2

Question 4

To learn Running Mean Smoother I defined two functions, **running_mean_weight** which corresponds to w function in below equation and **running_mean_smoother** which evaluates g(x) for given input.

The rest is similar to learning and drawing plot in Question 2 and 3. With the given **data_interval** by using the **running_mean_smoother** function we evaluate **p_head** and draw to plot using lines function.

$$\widehat{g(x)} = \frac{\sum_{t=1}^{N} w\left(\frac{x - x^t}{h}\right) * r^t}{\sum_{t=1}^{N} w\left(\frac{x - x^t}{h}\right)} \quad where\ w(u) = 1\ if\ |u| < 1\ and\ 0\ o.w.\ \ t = 1{:}N$$

Elif Küçük
0040851
COMP 421
HW 04 Report.

Running the code below we get the Figure 3 as output:

```
p_head<- sapply(data_interval, running_mean_smoother, data_train)

plot(data_train[,1], data_train[,2], type = "p", pch = 20, col = "RED",
    ylab = "density", xlab = "x")
points(data_test[,1], data_test[,2], type = "p", pch = 20, col = "BLUE" )
lines(data_interval, p_head, type = "l", lwd = 2, col = "black")
```

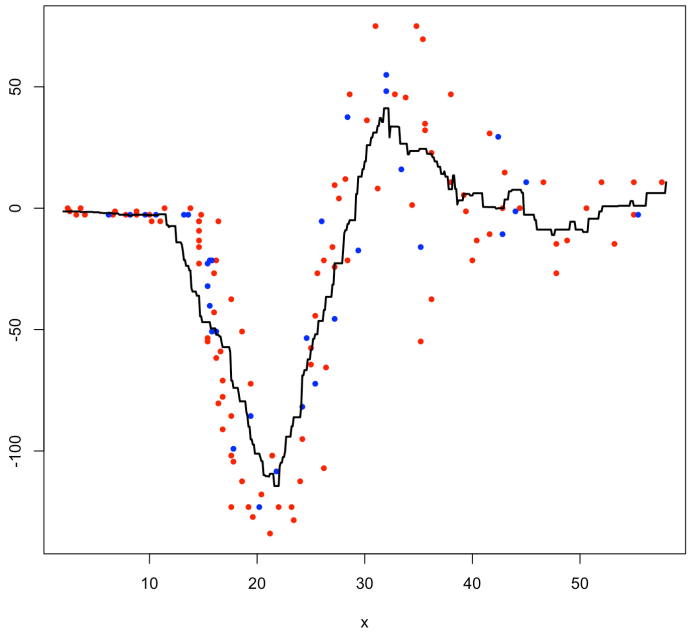

Figure 3

Question 5:

For running mean smoother we evaluate the mean squared error as explained in Question #3. In the Figure 4 you can see the result.

```
> #Question 5
> p_head<-learn_running_mean_smoother_fun(data_test[,1], data_train)
> find_mse(p_head, "Running Mean Smoother")
[1] "Running Mean Smoother=> RMSE is 19.755201 when h is 3"
```

Figure 4

Question 6:

Similarly for Kernel Smoother I defined two following functions: **Kernel** which is K function in below equation and **kernel_est** which is the full equation below. **kernel_est** estimates the g(x) given a input x.

Elif Küçük
0040851
COMP 421
HW 04 Report.

$$\widehat{g(x)} = \frac{\sum_{t=1}^{N} K\left(\frac{x - x^t}{h}\right) * r^t}{\sum_{t=1}^{N} K\left(\frac{x - x^t}{h}\right)} \quad where\ K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-u^2}{2}\right) \quad t = 1{:}N$$

Running the code below we get the plot (Figure 5) as output.

```
p_head<-kernel_est(data_interval)
plot(data_train[,1],  data_train[,2], type = "p", pch = 20, col = "RED",
    ylab = "density", xlab = "x")
points(data_test[,1], data_test[,2], type = "p", pch = 20, col = "BLUE" )
lines(data_interval, p_head, type = "l", lwd = 2, col = "black")
```

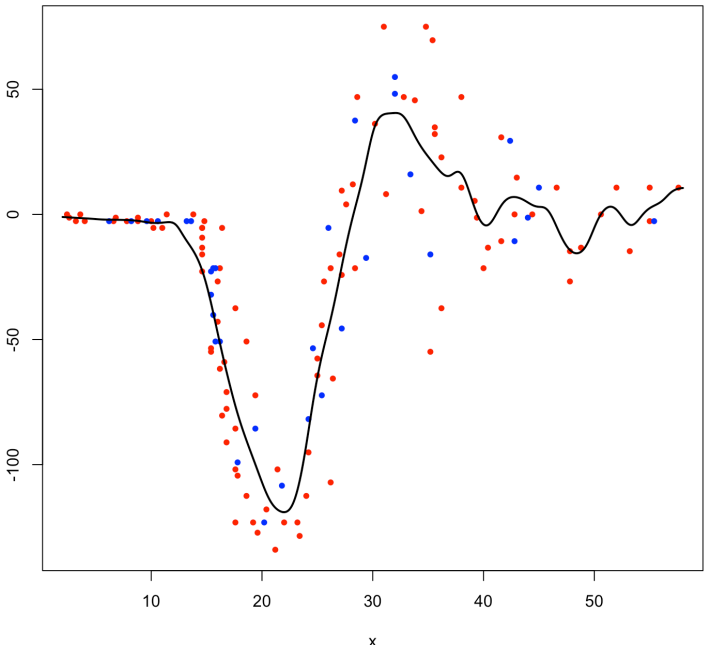

Figure 5

Question 6:

For kernel smoother we evaluate the mean squared error as explained in Question # 3. In the Image 6 you can see the result.

```
> find_mse(kernel_est(data_test[,1]), "Kernel Estimator")
[1] "Kernel Estimator=> RMSE is 18.056611 when h is 3"
>
```
Figure 6