Aykut Aykut
0034039
November 8th, 2018
Comp421 Homework04

In this homework, I was asked to implement 3 non-parametric regression algoritms. In order to do that , I followed these 17 steps:

1) I read the dataset into memory as *data*. The data has 133 rows and 2 columns which are 'x' and 'y'. Each row corresponds to a data point.

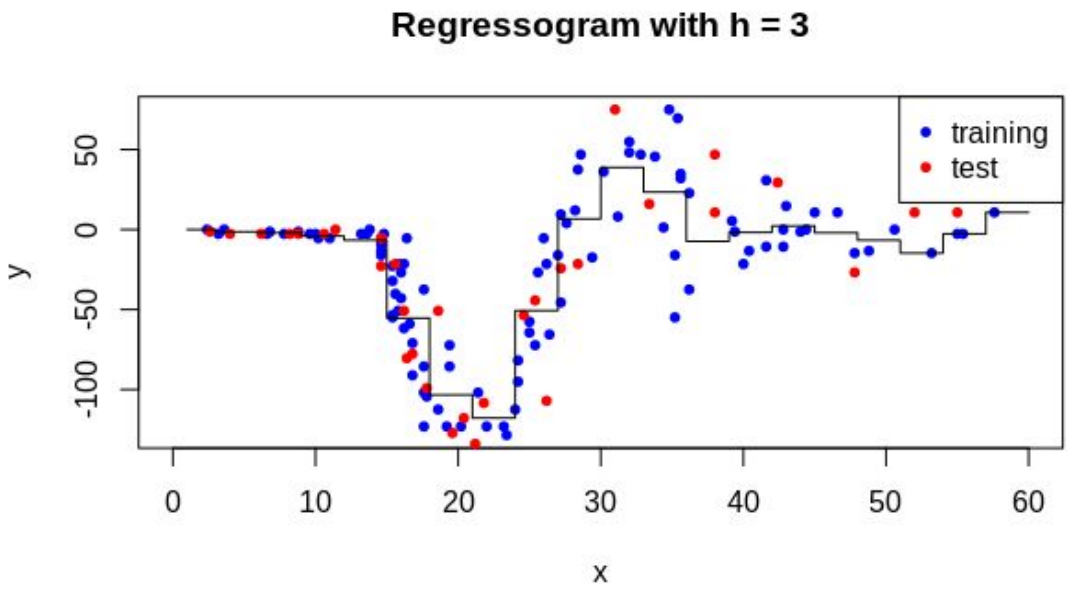
2) Train-test split: I splitted the data into 2 groups, train and test. First 100 rows are in *train_data* and last 33 rows are in *test_data.*

3) I defined *regressogram* function with *data, bin_width, origin* and *x_max* parameters using the formula below. In *regressogram* function, first I found *bin_number* of each data point by checking their *x* value, then grouped the data according to *bin_number* column and calculated mean of *y* values in each group. Lastly, I checked whether each bin has at least 1 point in itself or not, and if there is a bin with no point in it, I manually assign 0 value for mean of that bin. At the end, I returned *bin_means* data frame ordered according to *bin_number* column.

$$\widehat{g}(x) = \frac{\sum_{i=1}^{N} b(x, x_i) * y_i}{\sum_{i=1}^{N} b(x, x_i)} \quad where\ N = 100\ and\ b(x, y) = [x\ and\ y\ are\ in\ the\ same\ bin] * 1$$

4) I set the *regressogram_bin_width, regressogram_origin* and *regressogram_x_max* parameters as follows:
regressogram_bin_width <- 3
regressogram_origin <- 0
regressogram_x_max <- 60

5) I applied *regressogram* function on *train_data* with *bin_width = regressogram_bin_width, origin = regressogram_origin, x_max = regressogram_x_max* parameters and plotted *train_data, test_data* and regressogram of the *train_data* in the same figure. Regressogram plot of *train_data* with h = 3 is as follows:



6) I calculated root mean squared error of regressogram for *test_data* using the formula below and printed on the console.

$$rmse = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \widehat{y}_i)^2}{N}}$$

```
> cat('Regressogram => RMSE is', test_data_regressogram_rmse,
+    'when h is', regressogram_bin_width)
Regressogram => RMSE is 24.726 when h is 3
```

7) I defined *w* function with *x* parameter for *rms* function using the following formula:
$w(x) = [|x| \leq 0.5] * 1$

8) I defined *rms* function with *data_interval, data* and *bin_width* parameters for running mean smoother. In *rms* function, the formula below is applied to each element of *data_interval* list and its result is returned as a data frame along with corresponding *data_interval* values.

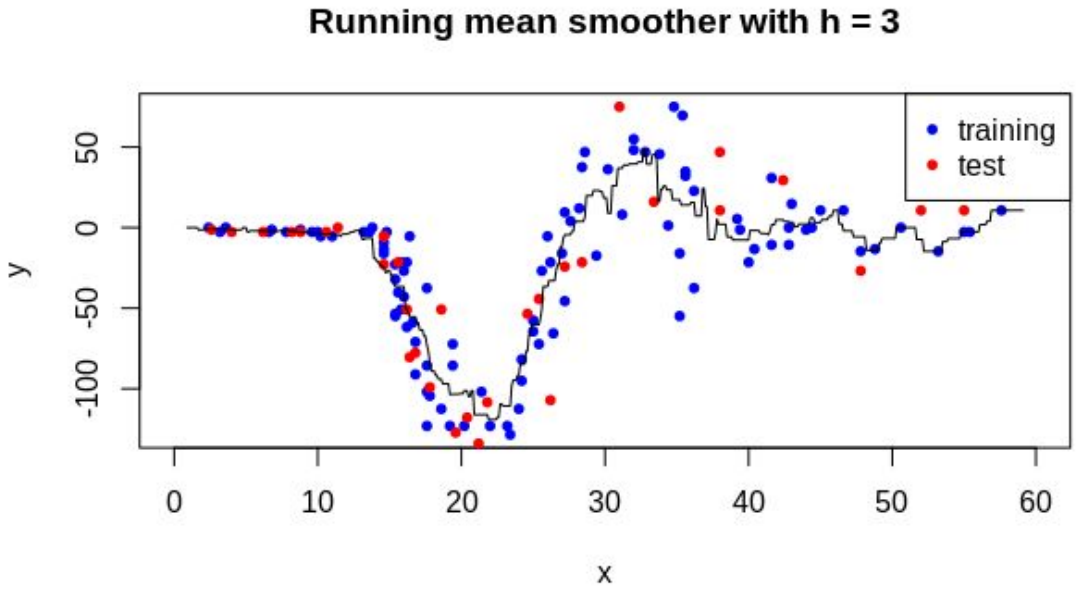
$$\widehat{g}(x) = \frac{\sum_{i=1}^{N} w(\frac{x - x_i}{h}) * y_i}{\sum_{i=1}^{N} w(\frac{x - x_i}{h})} \text{ where } N = 100 \text{ and } h \text{ is the bin width which is } 3$$

9) I set the *rms_bin_width, rms_origin, rms_x_max* and *rms_data_interval* parameters as follows:

```
rms_bin_width <- 3
rms_origin <- 0
```

```
rms_x_max <- 60
rms_data_interval <- seq(from = rms_origin, to = rms_x_max, by = 0.1)
```

10) I applied *rms* function on *train_data* with *data_interval = rms_data_interval, bin_width = rms_bin_width* parameters and plotted *train_data, test_data* and running mean smoother of the *train_data* in the same figure. Running mean smoother plot of *train_data* with h = 3 is as follows:



11) I calculated root mean squared error of running mean smoother for *test_data* and printed on the console.

```
> cat('Running Mean Smoother => RMSE is', test_data_rms_rmse,
+    'when h is', rms_bin_width)
```

Running Mean Smoother => RMSE is 23.84032 when h is 3

12) I defined *k* function with *x* parameter for *ks* function using the following formula:

$$k(x) = \frac{1}{\sqrt{2*\pi}} * exp(-\frac{x^2}{2})$$

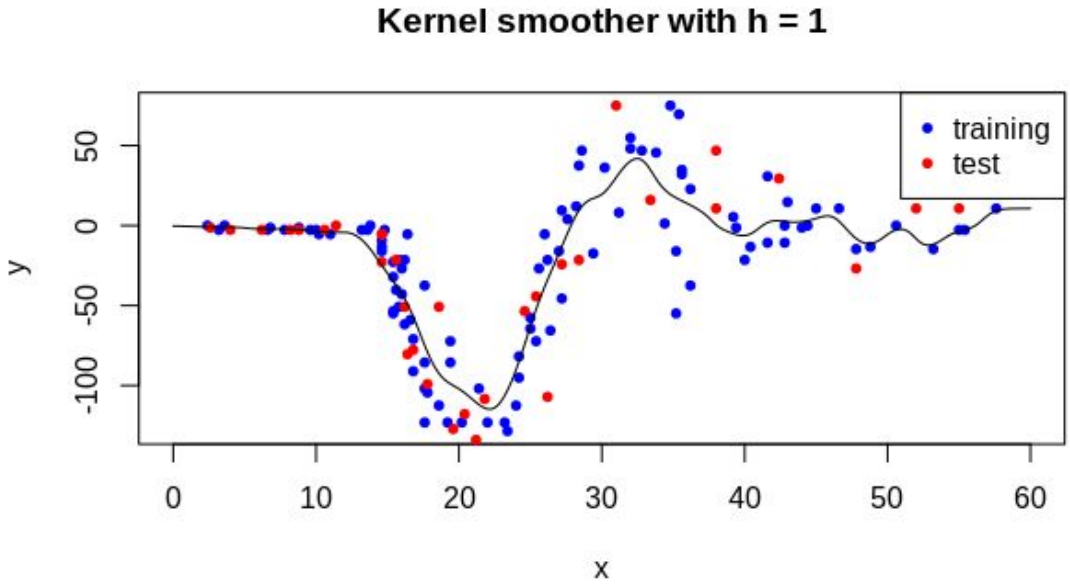
13) I defined *ks* function with *data_interval, data* and *bin_width* parameters for kernel smoother. In *ks* function, the formula below is applied to each element of *data_interval* list and its result is returned as a data frame along with corresponding *data_interval* values.

$$\widehat{g}(x) = \frac{\sum_{i=1}^{N} k(\frac{x-x_i}{h})*y_i}{\sum_{i=1}^{N} k(\frac{x-x_i}{h})}$$ *where* $N = 100$ *and h is the bin width which is* 1

14) I set the *ks_bin_width, ks_origin, ks_x_max* and *ks_data_interval* parameters as follows:

```
ks_bin_width <- 1
ks_origin <- 0
ks_x_max <- 60
ks_data_interval <- seq(from = ks_origin, to = ks_x_max, by = 0.1)
```

15) I applied *ks* function on *train_data* with *data_interval = ks_data_interval, bin_width = ks_bin_width* parameters and plotted *train_data, test_data* and kernel smoother of the *train_data* in the same figure. Kernel smoother plot of *train_data* with h = 1 is as follows:



16) I calculated root mean squared error of kernel smoother for *test_data* and printed on the console.

```
> cat('Kernel Smoother => RMSE is', test_data_ks_rmse,
+    'when h is', ks_bin_width)
Kernel Smoother => RMSE is 24.16725 when h is 1
```

17) I obtained the same results, regressogram plot, root mean squared error of regressogram, running mean smoother plot, root mean squared error of running mean smoother, kernel smoother plot and root mean squared error of kernel smoother with the results given in the homework description.