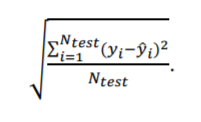
Intro 2 Machine Learning HW04

I start by reading the dataset and dividing it into data (X) and label (y) and then into training and test.

I set the min value to 0. My max bin width is three. So, my max value should be the smallest multiple of 3 that is larger than the max x value. ceiling(max(data\_set$x)/3) \* 3 gives me that max value. I also have a data\_interval list that is btw min and max values incremented by 0.01.

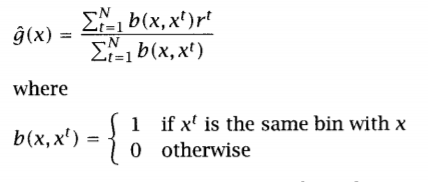
I have a function called plot\_points\_and\_legends that draws the training and test points with blue and red and also draws the legend for the colors. I also have a function called plot lines that draws the horizontal and vertical lines for each bin/interval. It takes p\_head and and interval as input. Interval is the list of x points and p\_head is the list of y values for the lines to be drawn.

I also have a function for calculating the RMSE based on the formula:

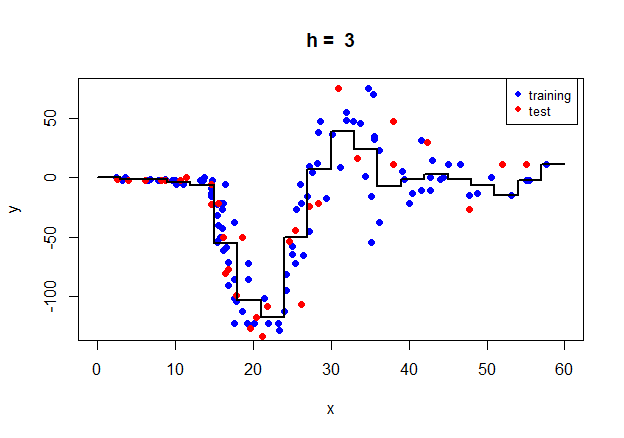


In this formula, yi’s are already known with the y\_test list. To calculate yi estimate, I need to find which interval/bin the x\_test(i) value falls on. I find it with ceiling((x\_test[i]-min\_value) / interval\_width). Now that I know which bin/interval my data point is in, I can estimate its y value by getting the value for that interval/bi from the p\_head list. I find it find p\_head[i]. I now know yi and yi estimate. I take the squares of all the errors and take the square root of the mean of the error squares.

For the regrossogram, I specify the bin width and find the borders based on this bin width. I need to calculate the p\_head : I calculate each bin by getting the mean of the y\_train of the x values that are between the borders of the bin. This is based on the formula:

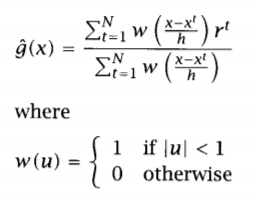


After this, I draw the points and the legend. Then, I draw the lines by supplying the plot\_lines function with p\_head and borders variables. I calculate the RMSE by supplying the RMSE function with the bin\_width, p\_head and bin\_width (for interval width) variables. My results are below:

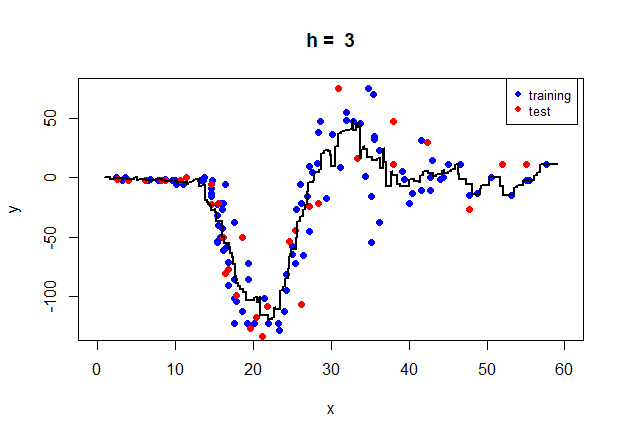


|  |
| --- |
| [1] "Regressogram => RMSE is 24.7260 when h is 3" |
| |  | | --- | |  | |

For the running means smoother, I set bin\_width to 3. I already have the data interval. I need to calculate the p\_head : I calculate each interval by getting the mean of the y\_train of the x values that are between 0.5\*bin\_width proximity of the data interval point. This is based on the formula:

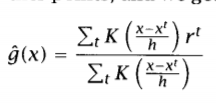


After this, I draw the points and the legend. Then, I draw the lines by supplying the plot\_lines function with p\_head and data\_interval variables. I calculate the RMSE by supplying the RMSE function with the bin\_width, p\_head and 0.01 (for interval width) variables. My results are below:

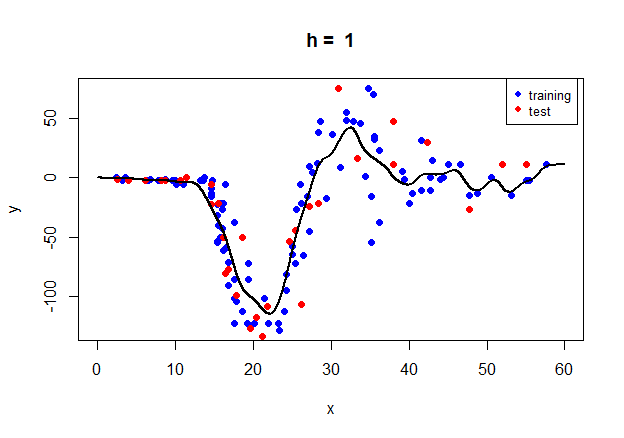


|  |
| --- |
| [1] "Running mean smoother => RMSE is 23.8403229051725 when h is 3" |
|  |
| |  | | --- | |  | |

For the kernel, , I set bin\_width to 1. I already have the data interval. I need to calculate the p\_head : I find the kernel values for each u value of the training set. U is found by (interval\_point-training\_point)/ bin\_width. Kernel is found by setting mean to 0 and standard deviation to 1 in the gaussian distribution. I find the p\_head by dividing the sum of kernel(i)\*y\_train(i) by the sum of kernel(i). This is based on the formula:



After this, I draw the points and the legend. Then, I draw the lines by supplying the plot\_lines function with p\_head and data\_interval variables. I calculate the RMSE by supplying the RMSE function with the bin\_width, p\_head and 0.01 (for interval width) variables. My results are below:



|  |
| --- |
| [1] "Kernel smoother => RMSE is 24.1624643592717 when h is 1" |
|  |
| |  | | --- | |  | |