Functional Data Regression

Madhur Bansal (210572)

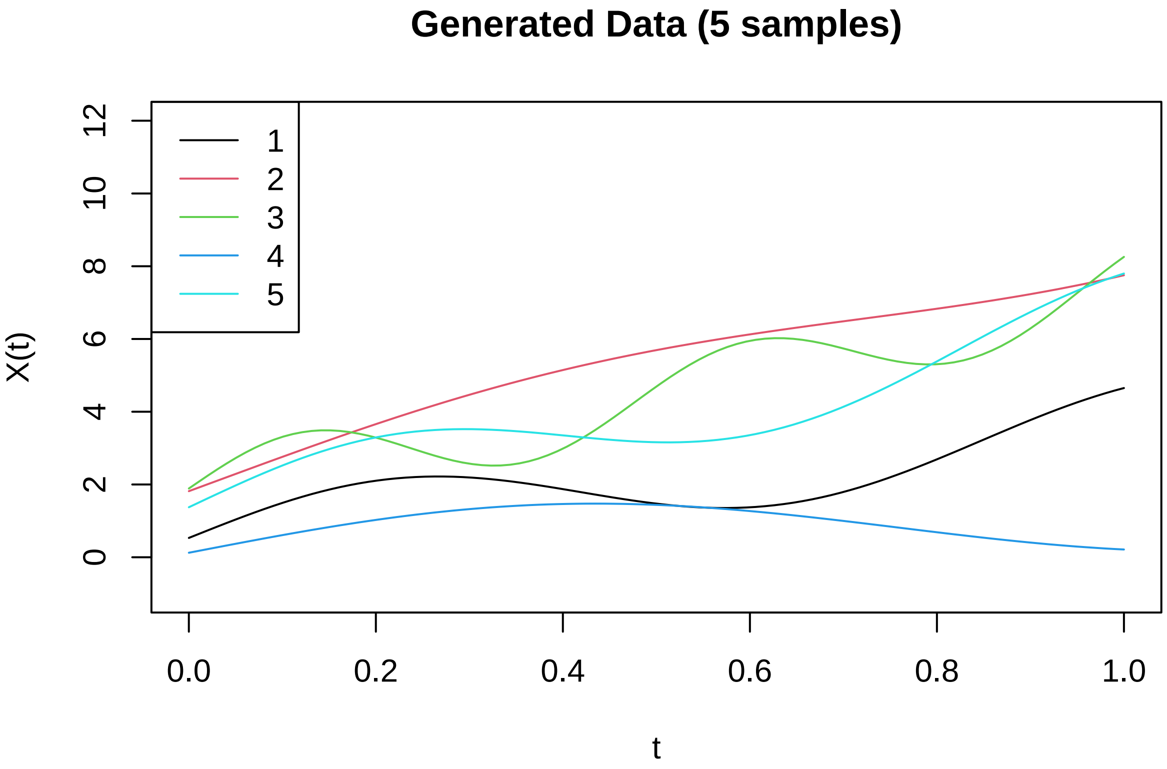
2024-03-14

Simulating the data

We construct a random variable $X_{(t)}$ from $L^2_{[0,1]}$ space using the following:

$$X_{(t)} = (c_1 * e^t) + sin(c_2 * 10t) + (c_3 * 2t)$$

where c_1, c_2, c_3 are random variables drawn from $uniform(0, 2)$. We generate (n = 300) samples of $X_{(t)}$ as our given data. Here is the plot of 5 samples from the generated data.



We simulate real Y using the following:

$$Y = m(X_{(t)}) + \epsilon$$

where $m(X_{(t)}) = \int_0^1 X^2_{(t)}(sin(t) + cos(t))dt$ and $\epsilon \sim N(0, 1)$

Here are the real values of y_i for above $X^i_{(t)}$ in respective order:

```
## [1]  6.970074 43.629690 31.696077  1.947331 24.870695 22.454314
```

Functional Linear Regression

Initially I predict the response variable using in-built functional linear regression function *fdRegress(.)*. The only reason to predict using the Linear Regression model is to get a baseline for model comparison. It tries to find β such that:

$$Y_i = \beta_0 + \int_0^1 X_i(t)\beta(t)dt + \epsilon_i$$

Both $X_i(t)$ and $\beta(t)$ can be expanded using orthonormal basis functions in L_2 space, say K:

$$X_i(t) \approx \sum_{k=1}^{K} c_{ik}\phi_k, \beta(t) \approx \sum_{k=1}^{K} b_k\phi_k$$

Expanding the respective terms, we get:

$$Y_i = \beta_0 + \sum_{k=1}^{K} b_k c_{ik}$$

Proposed Estimator (Based on Nadaraya-Watson Estimator)

I am using something similar to Nadaraya-Watson estimator for estimating m. The proposed estimator is as follows:

$$\hat{m}(X_{(t)}) = \frac{\Sigma_{i=1}^{n} K(||X - X^i_{(t)}||_{L_2}/h)y_i}{\Sigma_{i=1}^{n} K(||X - X^i_{(t)}||_{L_2}/h)}$$

where:

$K(.)$ is a valid kernel. I have used standard Normal $N(0,1)$

h is the band-width

$||.||$ is the norm on $L_{2[0,1]}$ space

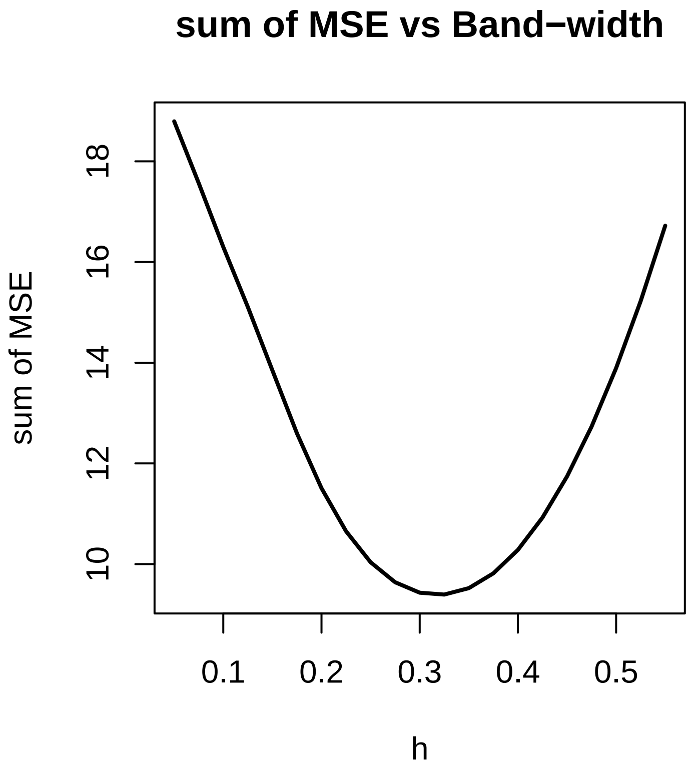
Choosing the Bandwidth (h)

In order to choose the appropriate band-width for the simulated sample data, I am using K-fold cross validation.

Here are the steps I followed:

1. I generate a grid of band-width containing 21 values - $h = \{0.05, 0.075, ...0.55\}$
2. Then I perform K-fold cross validation (with $K = 4$), and calculate the sum of the MSE obtained from cross-validation, for every h in the grid.

3. Chose the h which gives the lowest sum of MSE from the K cross validation results.



```
## [1] "Chosen Band-width: 0.325"
```

Estimation for new data

Now, we generate 100 new $(X_{(t)}, y)$ in order to evaluate the performance of our estimator.

Here are some of the estimates made for the newly generated data:

Estimation using Linear Regression:

```
##         Real y Estimated y     Error
## [1,] 19.382151  21.5130396 -2.130888
## [2,]  1.664833   0.3223428  1.342490
## [3,]  1.566500   3.3098935 -1.743394
## [4,]  6.425769   5.4234245  1.002345
## [5,] 13.571157  22.6553290 -9.084173
## [6,] 19.895903  21.7855517 -1.889649
```

```
## [1] "MSE =  9.8529470947704"
```

Estimation using proposed estimator

```
##         Real y Estimated y       Error
## [1,] 19.382151   19.679719 -0.29756778
## [2,]  1.664833    1.526623  0.13821023
## [3,]  1.566500    1.922560 -0.35606053
## [4,]  6.425769    6.391972  0.03379736
## [5,] 13.571157   13.480416  0.09074011
## [6,] 19.895903   19.457724  0.43817868
```

```
## [1] "MSE =  0.673709591846734"
```

Conclusion

Both estimators successfully predicted the dependent variable; however, the Nadaraya-Watson estimator outperformed the linear model slightly. This indicates that the linear model may not be the most appropriate fit for the data.

Outlier Detection

Madhur Bansal (210572)

2024-03-15

The objective is so generate simulated data along with some outliers. Then we have to propose a method detect the outliers and estimate the proportion of outliers in the data.

Simulating Data:

We generate a random $X_{(t)}$ from $L^2_{[0,1]}$ space using the following:

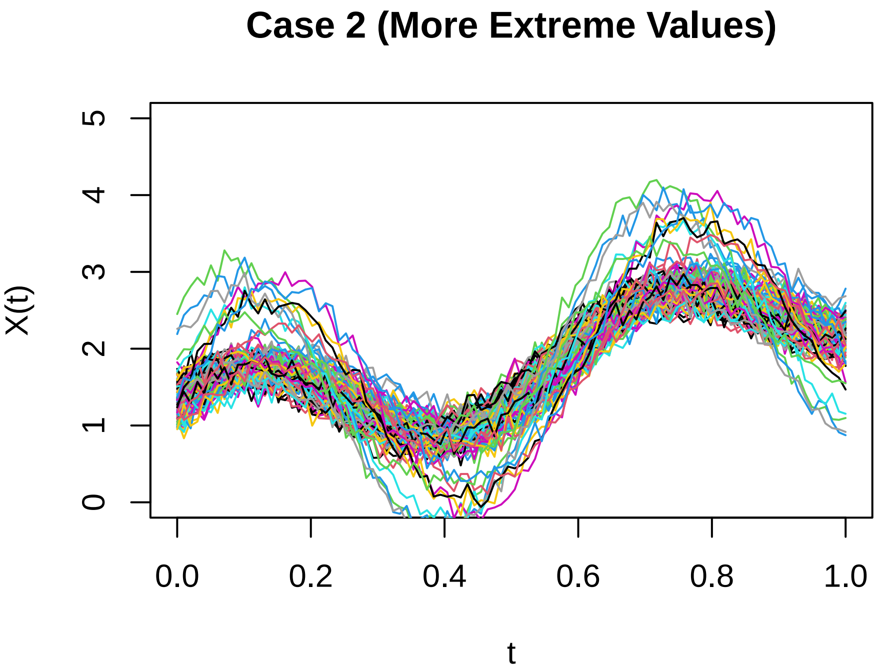
$$X_{(t)} = 0.5(c_1 sin(10.c_1.t) + c_2 sin(10.c_2.t) + c_3 sin(10.c_3.t)) + c_4$$

where c_i are random constants from $Unif(0,1)$.

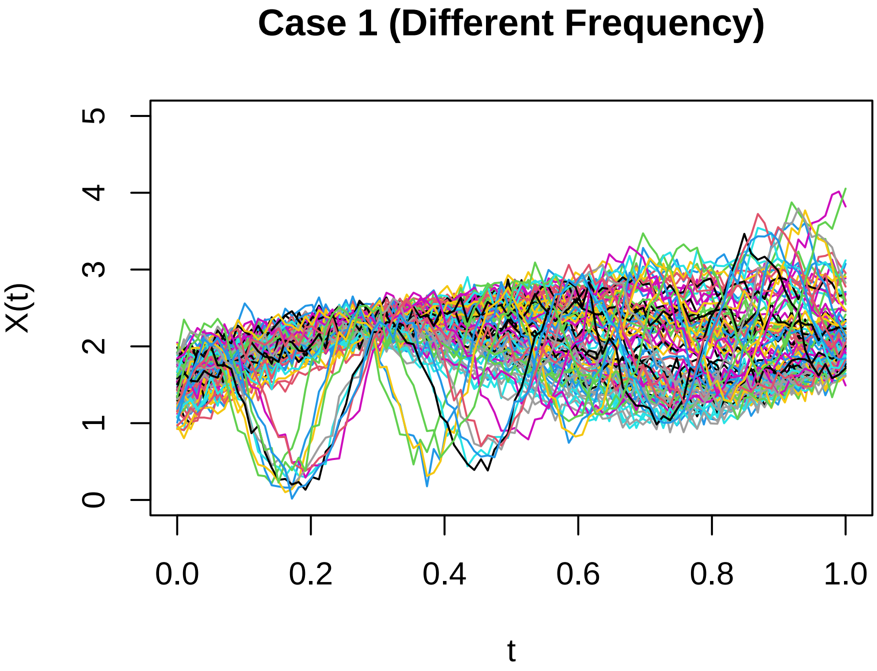
In this analysis, I have generated a total of 100 samples, with exactly 10 of them as outliers. I explore two different cases: Frequency Difference and Scale Difference

- In case 1, the outliers have similar shape to the data but attain more extreme values compared to the data. To introduce this difference, I multiply a random generated value uniform distribution $Unif(1,2)$ to the original function.
- In case 2, the outliers have a different frequency compared to the rest of the data. To achieve this, I multiply the frequency of sinusoidal function with 10*c where c ~ runif(1, 1.5)

Here are the plots of the datasets for the two cases:



Case 2 (More Extreme Values)



Case 1 (Different Frequency)

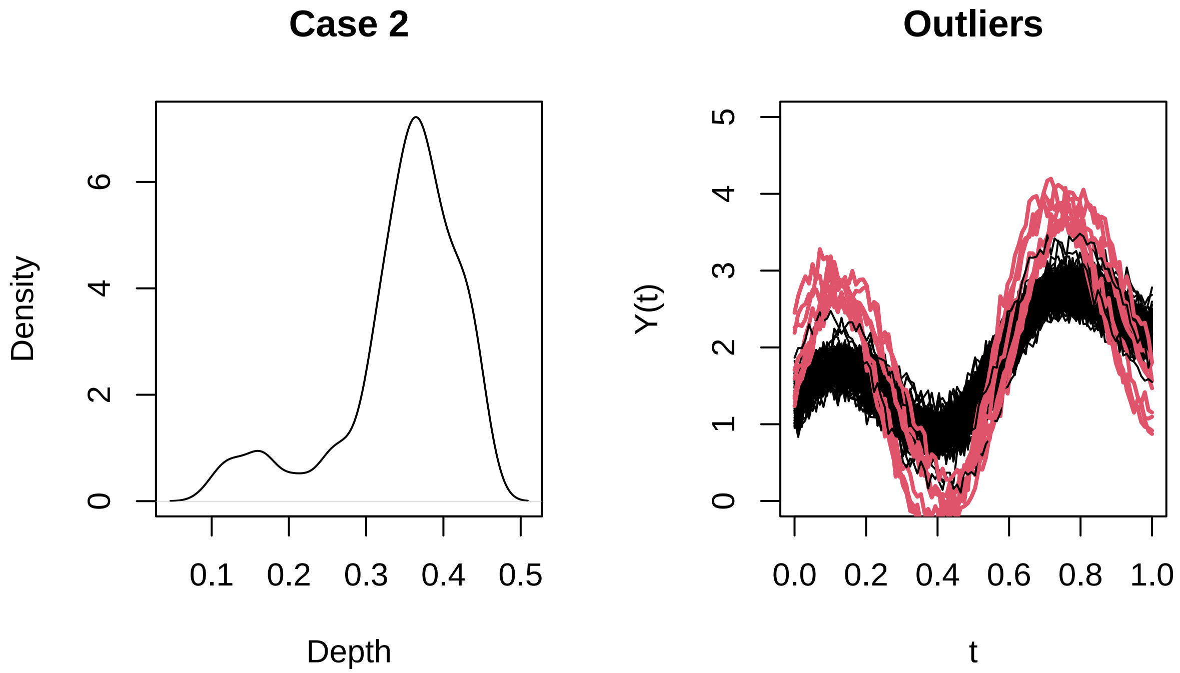
Proposed method:

To detect the outliers in the data, I calculate the depth of the $X_{(t)}$'s in the dataset. Subsequently, then assuming the depth comes from a normal distribution, a cut-off value based on the 95% region for $N(\mu, \sigma^2)$ where $\mu = mean(depth)$ and $\sigma^2 = Var(depth)$. Any samples that lie outside the 95% region, are labelled as outliers.

Note: As a measure of depth, we are using Modified Band-Depth (MBD) presented by Sara Lopez-Pintado and Juan Romo. This is the link to the original paper: https://www.jstor.org/stable/40592217

Case 1

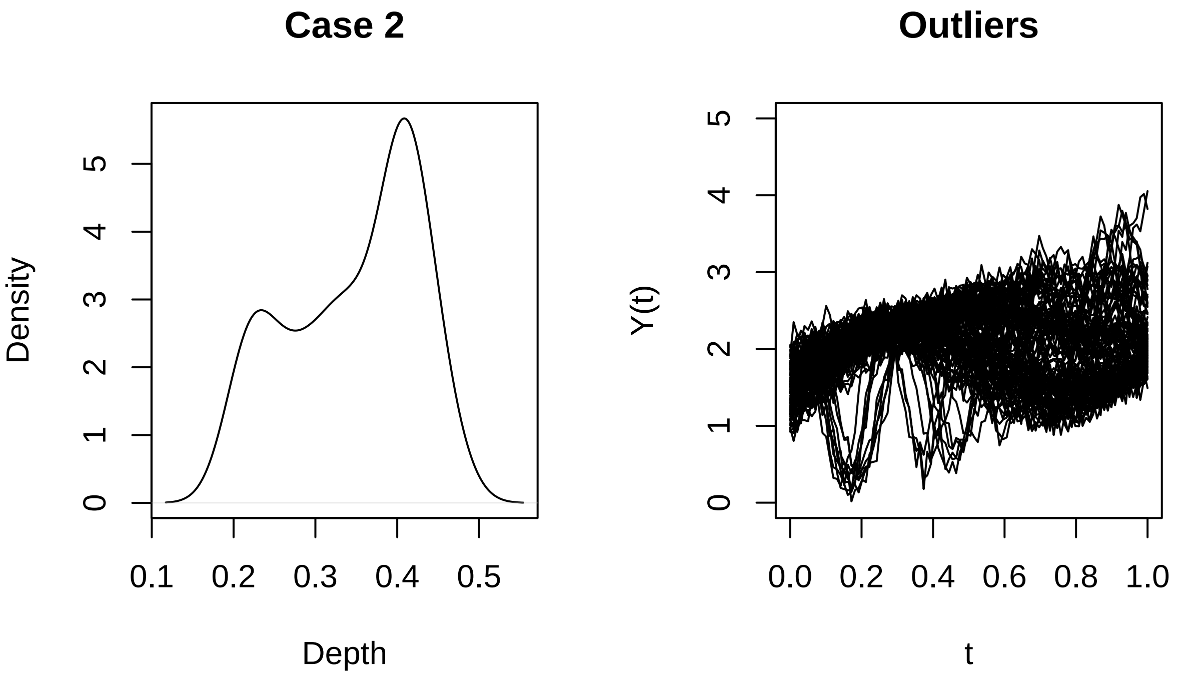
The method works well in this case. It is able to identify most of the outliers correctly.



```
##           predicted_outlier
## is_outlier  0  1
##          0 90  0
##          1  2  8
```

Case 2

In this case, the method was unable to accurately predict the outliers. Due to higher frequency of the random function, the function spends most of the time inside the band. Consequently, these functions, which should be identified as outliers, are instead ranked with a greater depth than non-outliers



```
##            predicted_outlier
## is_outlier  0
##          0 90
##          1 10
```

Conclusion

The method was effective in detecting most outliers with more extreme values. However, it struggled to accurately identify outliers that had higher frequency but were on the same scale. To improve the performance, additional features that account for frequency variations can be included while detecting outliers.