Functional Data Regression

Madhur Bansal (210572)

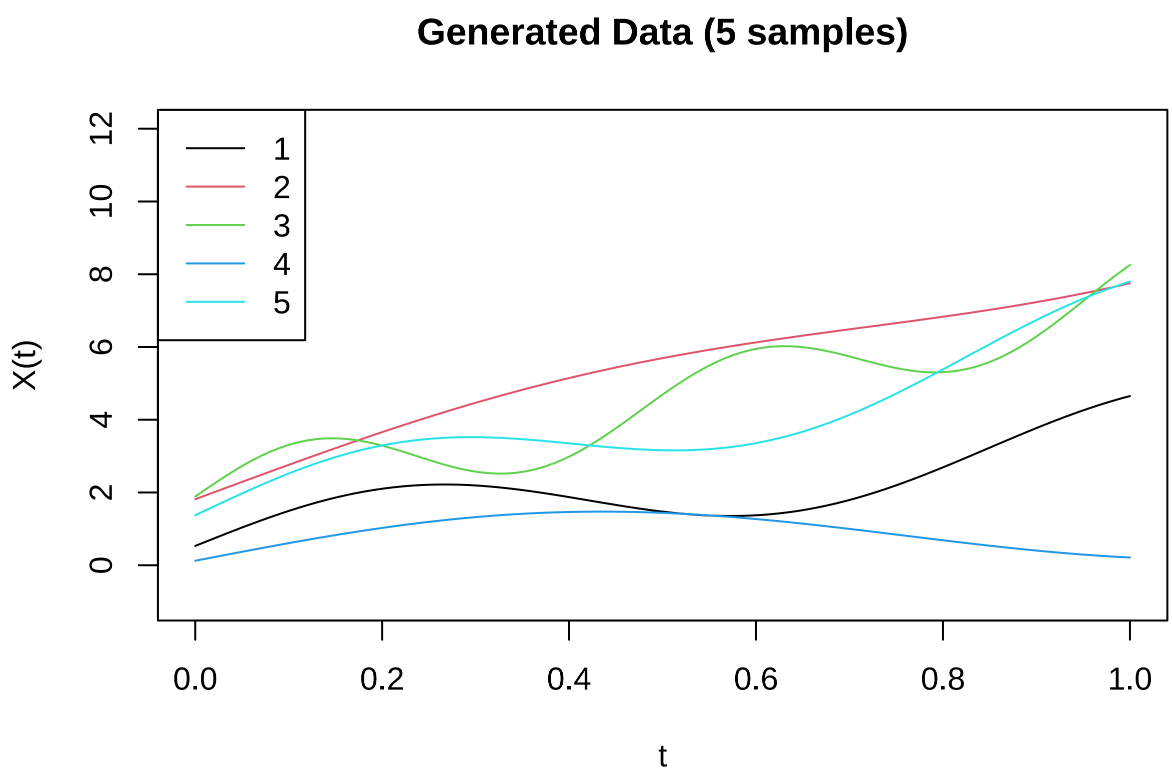
2024-03-14

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 = 100) 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]  8.204814 41.079967 33.139107  1.021508 26.661513 25.103260
```

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

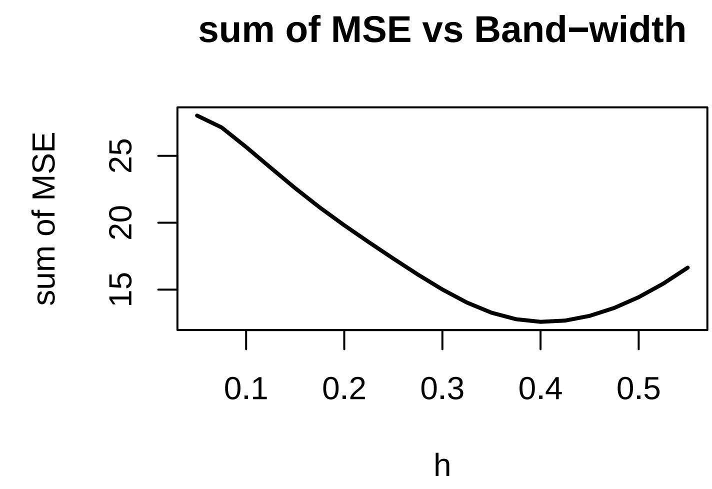
$||.||$ is the norm on $L_{2[0,1]}$ space

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: 0.4"
```

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:

```
##         Real y Estimated y       Error
## [1,] 25.039944   25.519573 -0.47962918
## [2,] 21.278169   20.465545  0.81262407
## [3,]  1.686397    2.611027 -0.92463012
## [4,]  8.952435    9.773870 -0.82143463
## [5,] 18.974028   18.906697  0.06733104
## [6,] 14.837879   14.011128  0.82675184
```

```
## [1] "MSE =  1.08949042918219"
```

```
## [1] "R2 =  0.989799998756462"
```

Conclusion

The implementation of the Nadaraya-Watson estimator for functional regression was successful, with the model achieving a Mean Squared Error (MSE) of 1.09 and an R-squared (R^2) score of 0.98. These results show that the model performs very well in terms of accuracy and explains almost all of the variability in 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:

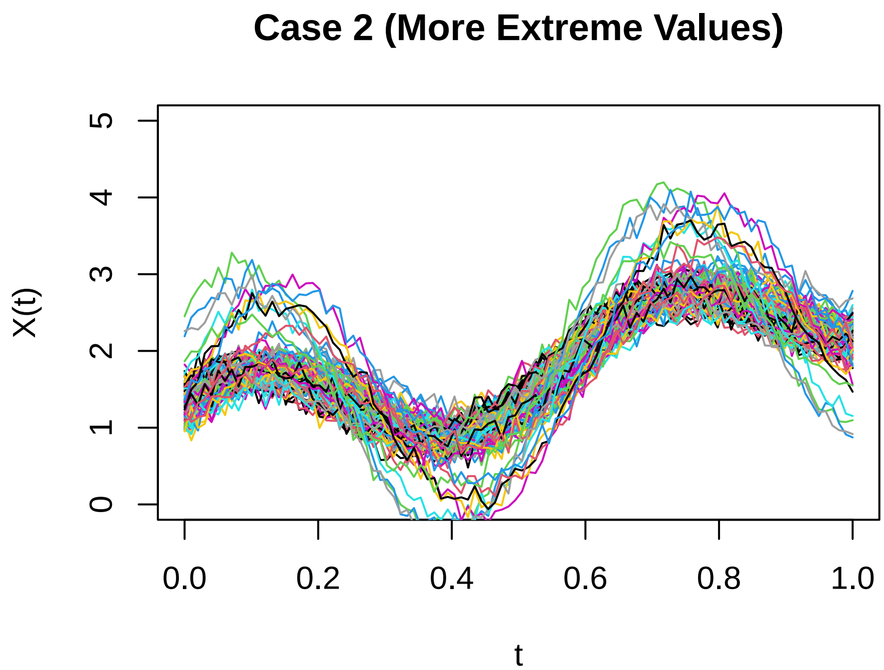
$$X_{(t)} = 0.5(c_1 sin(10.c_1.t) + c_2 sin(10.c_2.t) + c_3 sin(10.c_3.t)) + c_4$$

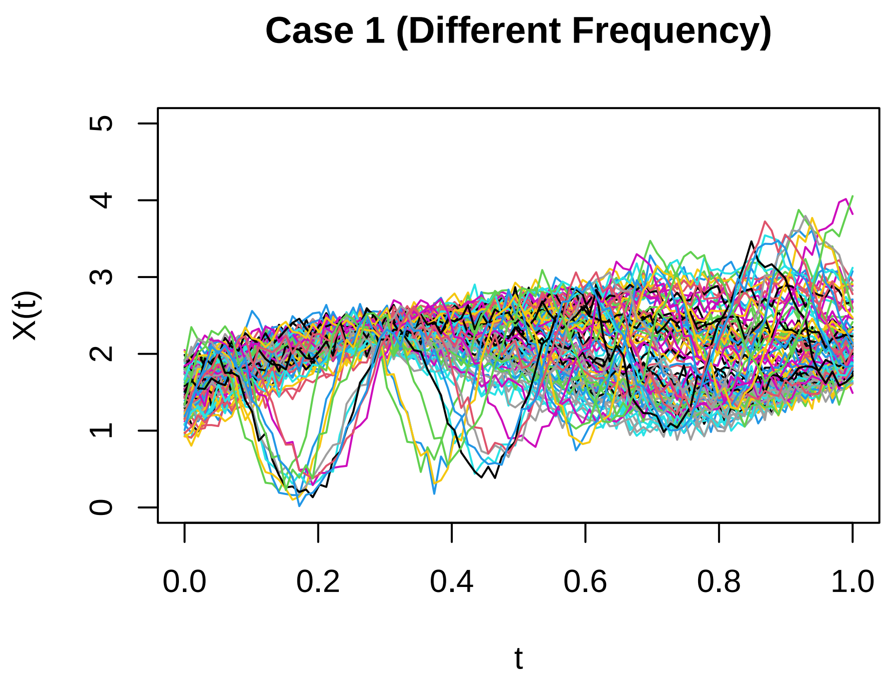
where c_i are random constants from $Unif(0, 1)$.

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)
X(t)
5
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0
t


Case 1 (Different Frequency)
X(t)
5
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0
t

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.

Then, I apply the following transformation to the data. This centers the data and allows to identify the outliers having different trend from rest of the data.

$$T(X_i(t)) = X_i(t) - \frac{1}{T}\sum_{j=1}^{T} X_i(j) \forall i = 1(1)n$$

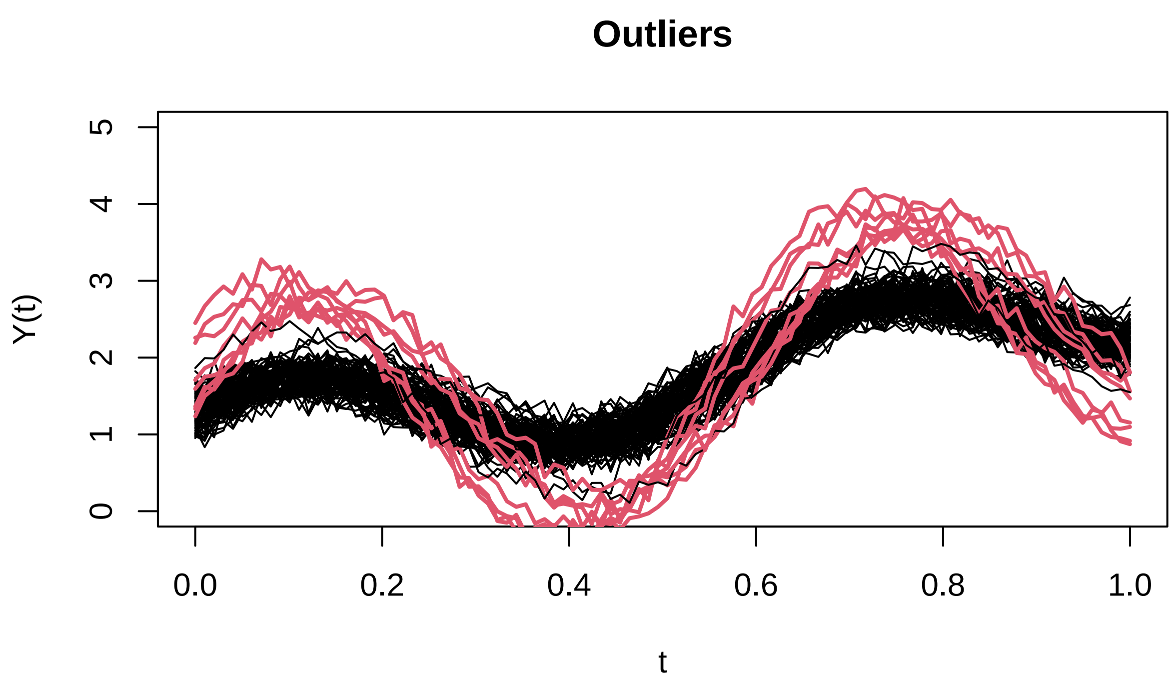
After applying the transformation, I again find the outliers using 95% region (decribed above). Then, finally I apply a transformation to normalize the centred data. This helps in detecting the outliers which may have different shape than the data.

$$T(X_i(t)) = \frac{X_i(t)}{||X_i(t)||_{L_2}} \forall i = 1(1)n$$

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. I also referenced this blog: https://www.lancaster.ac.uk/stor-i-student-sites/harini-jayaraman/anomaly-detection-in-functional-data in order to improve my implementation.

Case 1

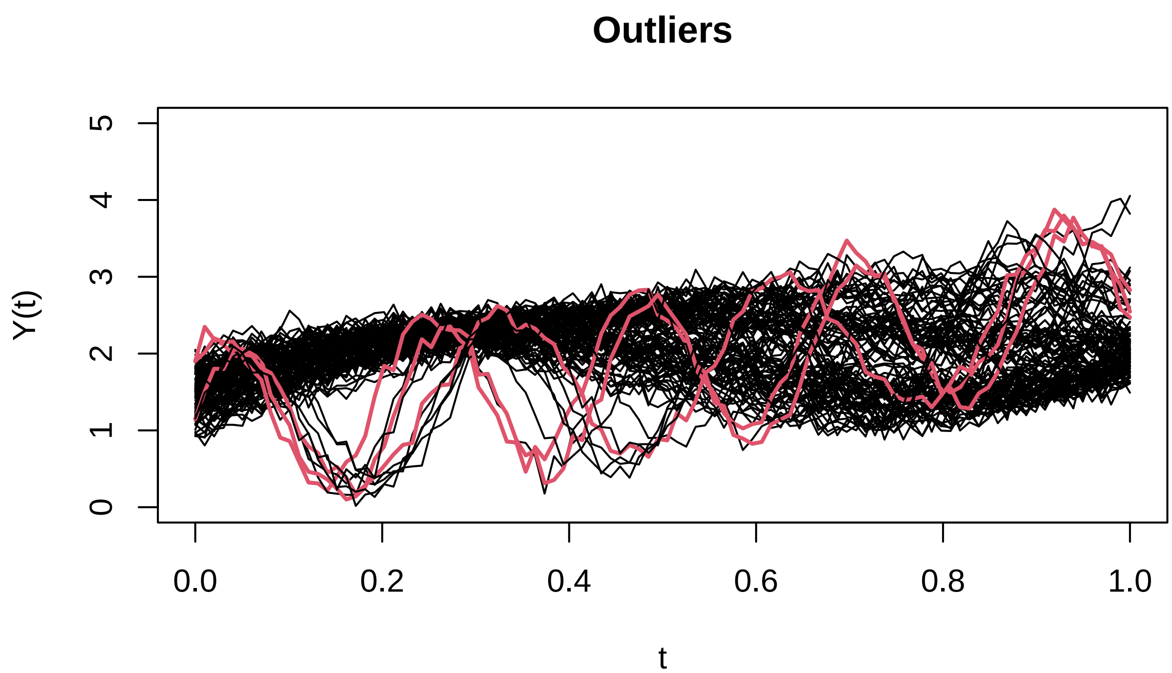
The method works well in this case. It is able to identify most of the outliers correctly.



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

Case 2

In this case also, the method was able to identify most of the outliers.




```
##           predicted_outlier
## is_outlier  0  1
##          0 90  0
##          1  7  3
```

Conclusion

The method successfully identified most of the outliers in two scenarios: functions with more extreme values and those with higher frequency patterns. This shows that the approach is effective in detecting different types of anomalies in the data.