Rotation Summary with Dr. Tarpey

Hanchao Zhang

May 10, 2020

1. Introduction

The effects of outliers in the functional data analysis are well aware by statisticians in recent years, there have been some works of literature on finding outliers in the functional data. However, most of the literature focused on the detection of the extreme values in the functional data in a univariate case. For example, *Manuel Febrero Environmetrics 2008; 19: 331-345* were focusing on the detection of extreme values that is abnormally large or small compared with the rest of the values. *Ana Arribas-Gil Biostatistics, Volume 15, Issue 4, October 2014* were focusing on the detection of shape outliers (defined by different shape from the rest of the sample). Our method of outlier detection and imputation finds a way to detect the outlier due to the misfunction of the medical devices or detaching of the devices from the patients, which not necessarily produce extreme values or differences in shapes, but a graduate changes of the trend.

Our method was inspired by real data acquired from the Intensive Care Unite. The data was generated from the medical device that attached to the patients' hand or head and measured the oxygen level and carbon dioxide level during the CPR procedure.

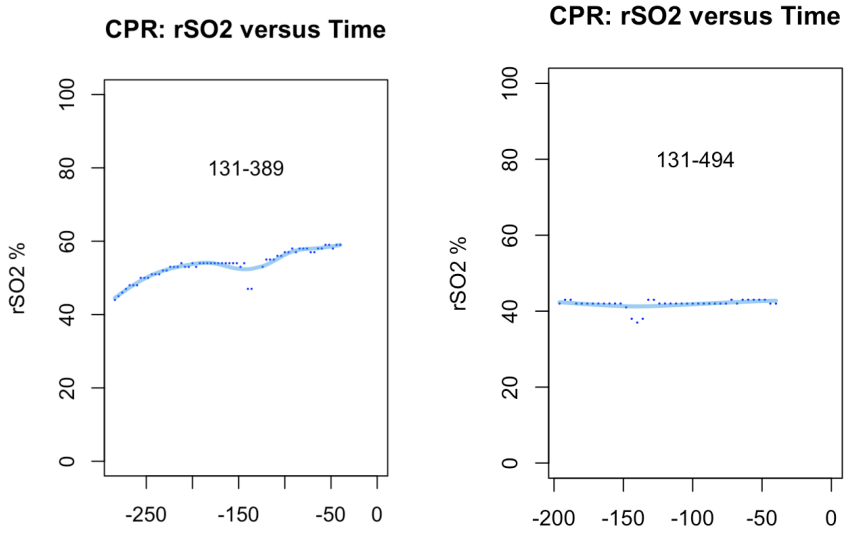
2. Obejective

We proposed a voting method to detect different types of outliers in the real data. We then evaluate the algorithm using some simulated data.

3. Types of the outliers in the real data

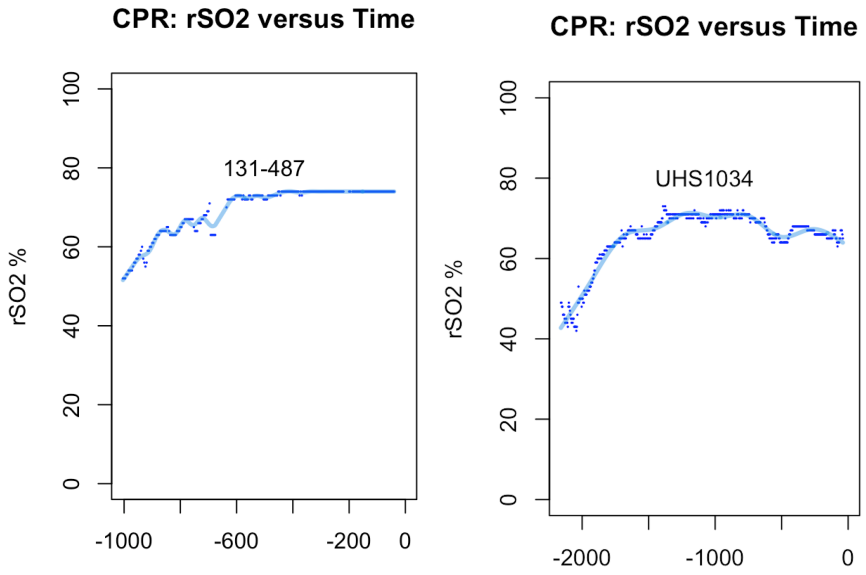
Misfunction and detaching of the devices during the CPR may produce different kinds of outliers in terms of their shape, values, and trend. We attached some of the typical examples of the outliers in the real data.

3.1 Consecutive jumps during the measuring time



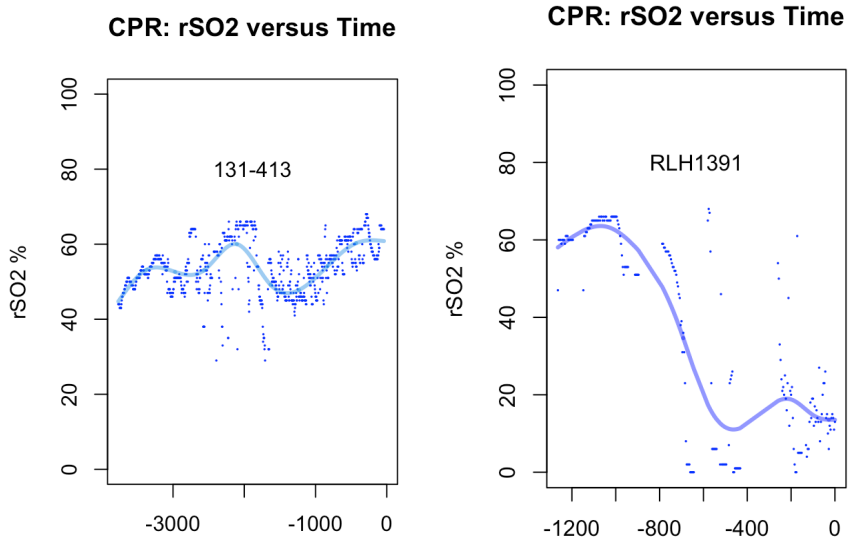
This kind of jump in the data is not the most common one but the one that is easiest to solve. Since there are only a few points that lie away from the trend of the data, we could simply remove it by fitting a non-parametric regression using kernel and splines, and then remove the points that at preset times away from the μ at that time point.

3.2 Nonconsecutive jumps during the measuring time



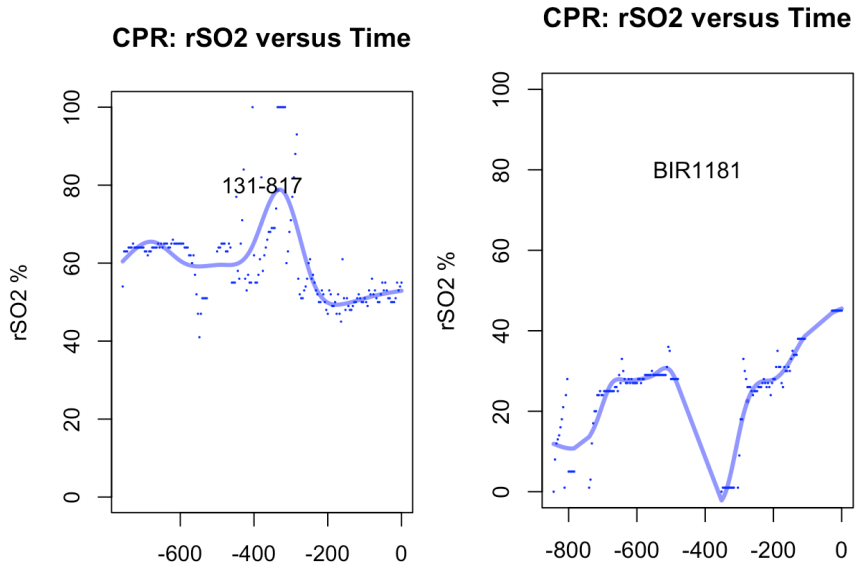
There are two examples of nonconsecutive jumps during the measuring time, which are more common ones in the real data. Both curves on the left and right show clear patterns of the trend. Even though the number of outliers is more than the first scenario, the general trends of the data were not corrupted by the outliers. The outliers can be found and detected by fitting smooth curve, such as a none-parametric spline with a small number of the knots, and applying the previous standard to remove the outliers

3.3 Jumps that corrupt the trend of the data

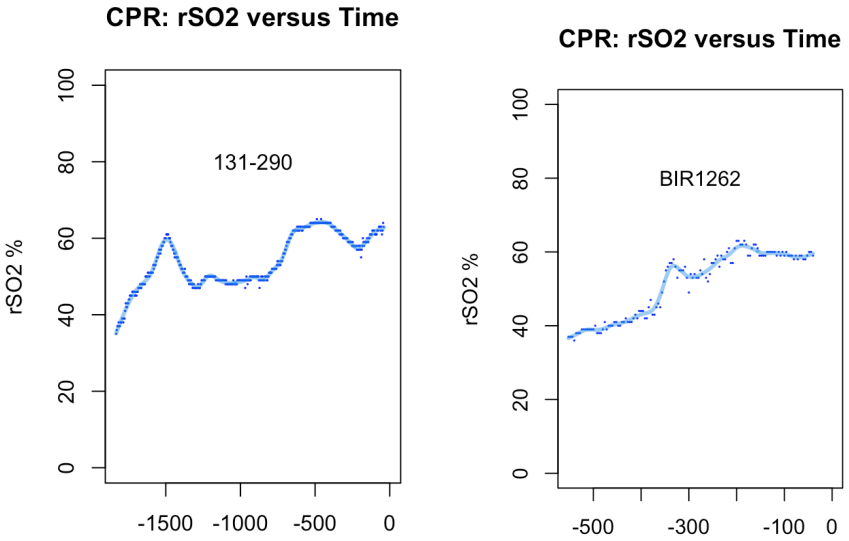


These kinds of jumps are the ones that hard to deal with. First of all, the true trend is hard to identify. Moreover, the observed values were all pretty far away from some expected value after fitting a smooth function.

3.4 Unexpected jumps due to a large number of outliers



The two examples above show one type of unexpected jumps; some extreme values generated the jumps. These outliers are easy to detect using the window screening method (one of the models in the voting methods that we proposed). The voting method is similar to a k fold cross-validation. However, the validation set is not randomly selected but a preset number of consecutive points in the data.



These two examples of the unexpected jumps due to a large number of outliers is different from the previous ones. The outliers that cause the jump are not extreme values but the value that gradually deviates away from the general trend. The left one has two unexpected peaks around the x-axis at - 1500 and -500. The right one has two unexpected peaks around the x-axis at - 400 and - 200.

4. Voting methods to detect and impute the outliers

4.1 Non-parametric splines (1st model in the voting methods)

These are the methods that we can use to remove the outliers in the first two scenarios.

1. Fit a non-parametric spline with an arbitrary number of knots using the whole dataset $\mathcal{D}$
2. Obtain the fitted value y^* and the standard deviation sd at each time points
3. Index the outliers $\mathrm{I}_k^{np} = (|\hat{y}_k - y_k^*| > n \times sd_k)$, n is a arbitrary number, where $\hat{y}$ belongs to the observed

dataset $\mathcal{D}$ and $k = 1, 2, 3, \cdots n$, and n is the number of the observation in the dataset $\mathcal{D}$

4. Calculate the reduced sum of the squares $MSE^{np} = \sum_{k=1}^{n} (\hat{y} - y^*)^2 \times I_k^{np}$

4.2 Window process (2nd model in the voting methods)

1. Create a vector W that represents the length of windows that will be used in the window precess, $0 < |W| < n$(where n is the total observation in the data)
2. Set the window length to W_i (where $0 < i \leq |W|$) and start a cross-validation like process

   2.0. Create a vector I_i^{wp} with all 0 in the vector, such that $|I_i| = n$

   2.1. removed the set of observation $\mathcal{A}$ from the whole dataset $\mathcal{D}$ and denote it as $\mathcal{D}_{ij}$, and denote the complementary set of $\mathcal{D}_{ij}$ as $\mathcal{D}_{ij}^c$, where $\mathcal{A} = \{\omega : \omega \in [x_j, x_{j+W_i}], x_k \in \mathcal{D}\}$, and $k = 1, 2, 3, \cdots, n$

   2.2. fit a non-parametric spline using $\mathcal{D}_{ij}$, denote it as $\mathcal{M}_{ij}$

   2.3. predit the data $\mathcal{D}_{ij}^c$ using $\mathcal{M}_{ij}$, denote the predicted dataset as $\widetilde{\mathcal{D}_{ij}^c}$

   2.4. let $\hat{y}_k$ and $\tilde{y}_k$ are eletments in $\mathcal{D}_{ij}$ and $\widetilde{\mathcal{D}_{ij}^c}$ respectively, let sd_k represent the standard deviation for $\tilde{y}_k$ and $I_{ij}^{wp} = I_{ij}^{wp}(|\hat{y}_k - \tilde{y}_k| > n \times sd_k)$, where $k = 1, 2, 3, \cdots, j$

   2.5 repeat the step 2.1 to 2.4 until the end of the window approach the last observation
3. calculate the reduced sum of squares $MSE_i^{wp} = \sum_{k=1}^{j} (\hat{y}_k - \tilde{y}_k)^2 I_{ik}$ where $k = 1, 2, 3, \cdots, j$
4. Go back to step one and repeat for all preset window length and obtain the MSE_i^{wp} and I_i^{wp} for each window length $W_i \subset W$

4.3 Voting MSE of from the two models above

The voting method is a combined method of the two previous models. By using the voting method, we can get the best parameter by grid searching, which was arbitrary in the previous step.

For each length of the windows, we can compute a total reduced MSE contructed by MSE_1, MSE_2, and MSE_3

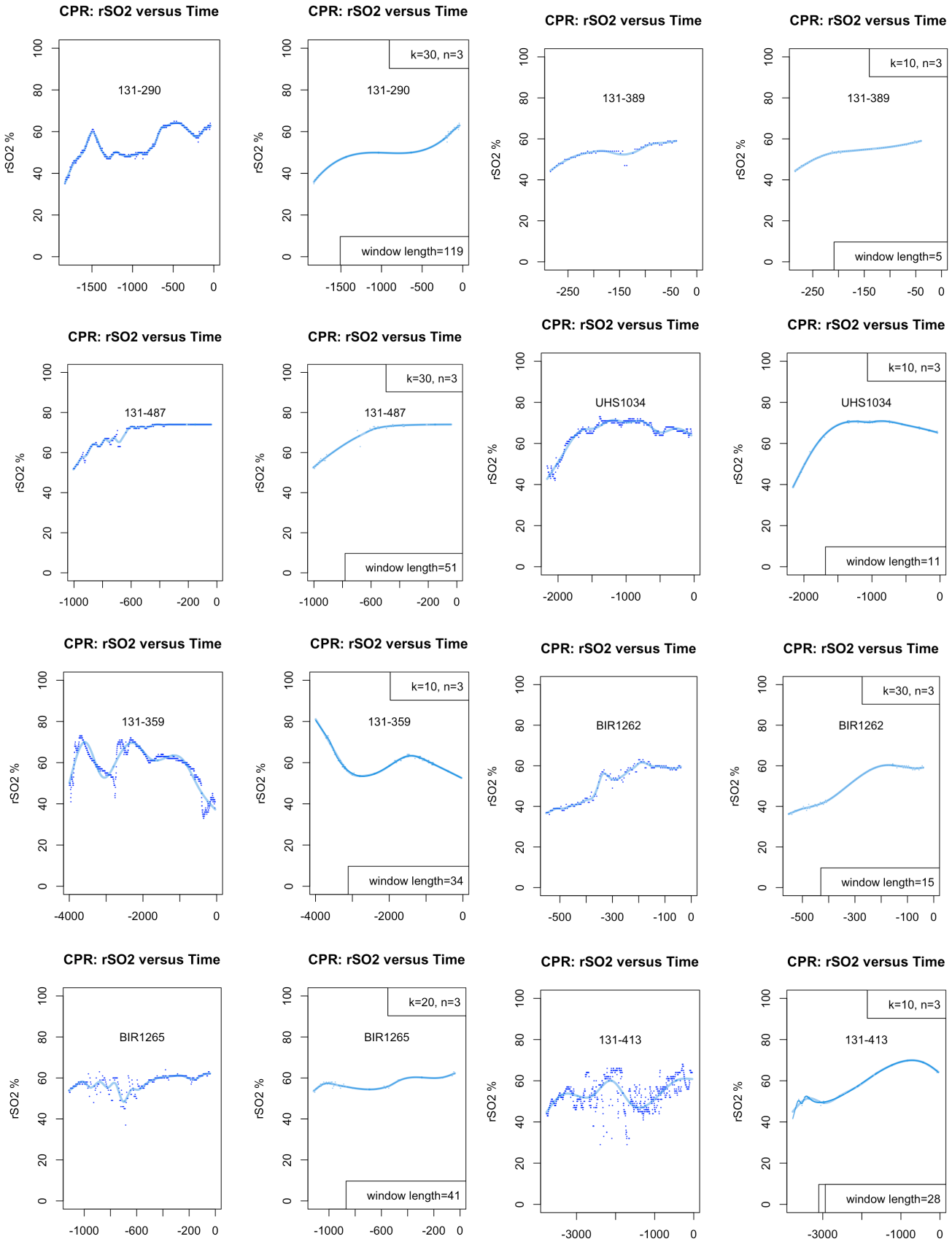
1. We denote $\text{MSE}_{i1} = \frac{1}{2} \times (\frac{\sum_{k=1}^{n} (\hat{y}_k - y_k^*)^2 \times I_k^{np} \times I_{ik}^{wp}}{\sum_{k=1}^{n} I_k^{np} \times I_{ik}^{wp}} + \frac{\sum_{k=1}^{n} (\hat{y}_k - \tilde{y}_k)^2 \times I_k^{np} \times I_{ik}^{wp}}{\sum_{k=1}^{n} I_k^{np} \times I_{ik}^{wp}})$
2. Denote $\text{MSE}_{i2} = \frac{\sum_{k=1}^{n} (\hat{y}_k - y_k^*)^2 \times I_k^{np} \times (1 - I_{ik}^{wp})}{\sum_{k=1}^{n} I_k^{np} \times (1 - I_{ik}^{wp})}$
3. Denote $\text{MSE}_{i3} = \frac{\sum_{k=1}^{n} (\hat{y}_k - \tilde{y}_k)^2 \times (1 - I_k^{np}) \times I_{ik}^{wp}}{\sum_{k=1}^{n} (1 - I_k^{np}) \times I_{ik}^{wp}}$
4. Calculate $\text{MSE}_i = \text{MSE}_{i1} - \text{MSE}_{i2} - \text{MSE}_{i3}$
5. Repeat the step 1 to 4 for all window length, find the length that maxmized the MSE as our preferred window length

We can also repeat the previous step for different non-parametric splines, and the one that maximized the MSE will be used to detect the outliers.

5. Imputation

We developed 7 different ways of imputation. Currently, we would prefer to impute the outliers with the fitted value of the newly fitted non-parametric splines using the data without outliers detected in the previous voting method.

6. Result of the voting method



The first and second columns are the original data; the second and fourth columns are the one after applying the voting methods and imputing outliers using the imputation method mentioned above.

In general, the voting methods removed the outliers and kept the trend we need in the functional data regression

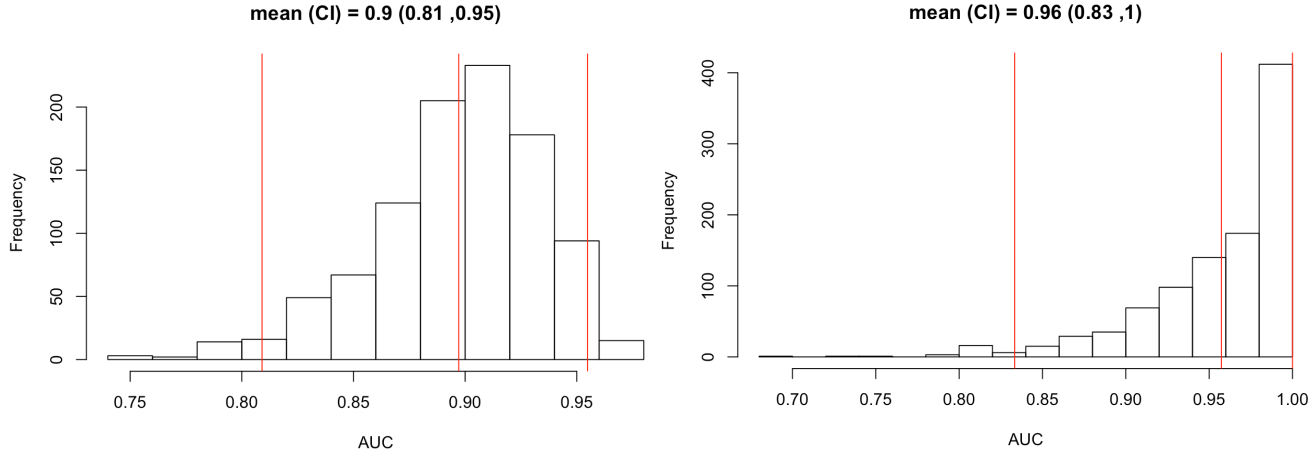
and other analyses using cumulated and aggregated measures of the values.

7. Improvement in prediction

7.1 AUC in original and smoothed data set (processed by the voting method)

We applied a logistic regression using survival as the outcome and mean oxygen level, APACHE score, CPR duration, and CPR derivatives as the predictor on the full data set and the processed data set, respectively.

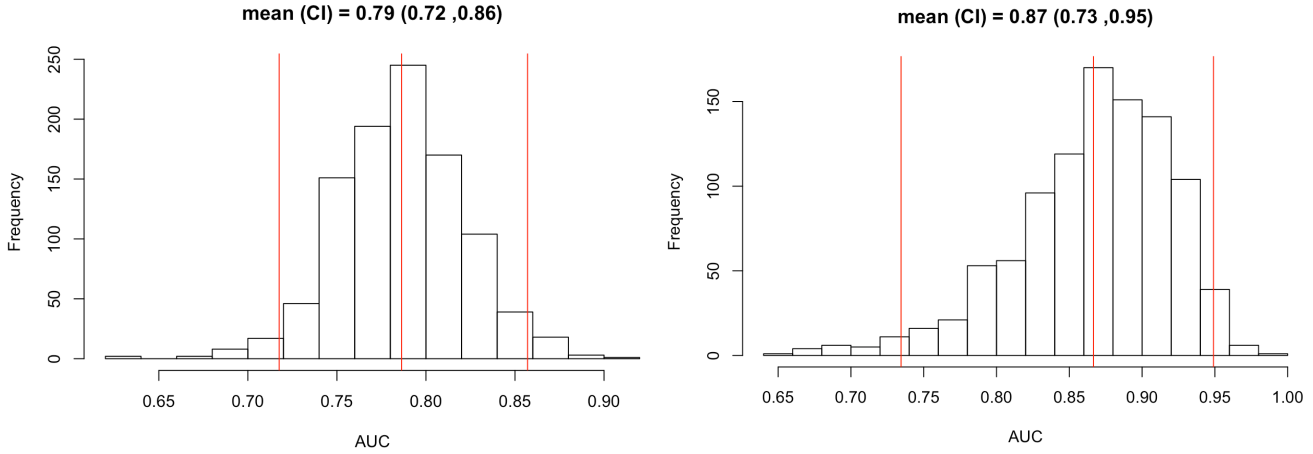
We evaluate the AUC using a 5 fold cross-validation. We repeat the 5 fold cross-validation 1000 times to obtain the empirical distribution of the estimated AUC.



We were able to improve the mean of the AUC by 6 percent. However, outcome survival is a rare outcome, and AUC might not be able to give us a valid estimate.

7.2 AUC in the bootstrapped original and smoothed data set (processed by the voting method)

Since the outcome survival is a rare event, we applied bootstrap on the dataset first to enlarge the size of the data to five times as the original data and then applied the same analysis to estimate the AUC



The bootstrapped sample also shows an increase in the accuracy after removing the outliers. The AUC was increased by 6% after removing outliers using our voting method.

However, since it is a bootstrapped sample, the data used for fitting the model may appear in the validation dataset, the AUC is overestimated. We would like to purpose some other estimation of the prediction accuracy in the future to perform the analysis again.

8. Future works

8.1 Simulation

To be finished, we purpose to simulate the functional data and the outlier generating process. We will apply the voting method on the simulated dataset to see if the method works well.

8.2 R function and packages

I would like to wrap up the method and R function, and test the method on more simulation data and generalize this method in terms of more types of outlier detections.

I would also like to see if there is any other way to improve the method on its computational efficiency.