Robust Boosting for Functional Regression

July 22, 2021

1 Introduction

Nowadays, functional data has become more and mode widely available and has applications in a vast number of fields. Practitioners frequently observe data that can be described as functions defined on a domain (of time for instance). This kind of variables is inherently infinite-dimensional but are usually measured discretely.

In the context of functional data analysis (FDA), a common problem is to estimate a regression model when the explanatory variable is functional. We assume that $(X_1, Y_1), ..., (X_n, Y_n)$ are independent and identically distributed realizations of the pair (X, Y), where $Y \in \mathbb{R}$ is the response variable and $X \in L^2(\mathcal{I})$ is a square integrable function defined on an interval $\mathcal{I}$, and consider the multi-index model

$$Y = r(\langle X, \alpha_1 \rangle, ..., \langle X, \alpha_p \rangle) + \epsilon, \tag{1}$$

where $\langle X, \alpha_j \rangle = \int_{\mathcal{I}} X(t)\alpha_j(t)dt$, $\alpha_j \in L^2(\mathcal{I})$ are projection coefficients, $r : \mathbb{R}^p$ is an unknown smooth function, and ϵ is independent of X.

Previously, we found an approximation to r in (1) using `TFBoost`, which yielded an estimator given by:

$$\hat{r}_1(\langle X, \beta_{1,1} \rangle, ..., \langle X, \beta_{1,K} \rangle) + ... + \hat{r}_M(\langle X, \beta_{M,1} \rangle, ..., \langle X, \beta_{M,K} \rangle). \tag{2}$$

The `TFBoost` algorithm used a “functional multi-index trees” to compute each r_j, where each tree was constructed with K projections $(\langle X, \beta_{j,1} \rangle, ..., \langle X, \beta_{j,K} \rangle)$ which we refer to as “indices”.

In principle, `TFBoost` applies to data without outliers, which are atypical observations that deviate from the bulk of the data. In practical situations, however, measurements in the response and the functional predictor may either or both contain outlying values. Borrowing the terminology from Rousseeuw and Van Zomeren (1990) for linear regression, we refer to (X_i, Y_i) as an “leverage point” if X_i deviates from the majority of the data. This is only in regard to the outlyingness of X and does not take the response Y into consideration. If a leverage point (X_i, Y_i) deviates greatly from the X-Y plane that corresponds to the majority of the data, we call it a “bad leverage point”. If we were to use the true model (the one that describes most data points) to predict Y_i, then a bad leverage point will have an extreme residual value. On the other hand, if a leverage point (X_i, Y_i) is well represented by the true model, we call it a "good leverage point". A point (X_i, Y_i) that is not a leverage point and does not follow the true model is called a "vertical outlier".

It is well-known atypical data points can seriously affect the fit of the regression function. These points are (X_i, Y_i) pairs that follow a model different from the one that represents the majority of the data, including "bad leverage points" and "vertical outliers" depending on the outlyingness of X_i. For functional predictors, the outliers in X_i can exhibit different properties: some present different shapes than the vast majority of the data and others have extreme values in some part of or across the domain. The former are usually referred to as "shape outliers" and the latter as "magnitude outliers" (Hyndman and Shang, 2010; Arribas-Gil and Romo, 2014; Dai et al., 2020).

Methods for functional regression in the presence of outliers have been developed in the past decade. Not many proposals exist and most are focused on linear models with vertical outliers. For the very few papers that considered bad leverage points, they either studied shape outliers or magnitude outliers, but not both.

In what follows, we present a robust `TFBoost` algorithm for functional regression and assess its performance on data with different contaminations. In the simulation study, we compare our estimator with existing proposals on robust functional regression and functional additive regression estimators in a wide range of settings, which cover cases without outliers and cases with bad leverage shape outliers, bad leverage magnitude outliers, or vertical outliers.

2 Related work

The impact of outliers has been extensively studied in the context of multivariate regression and numerous robust methods have been developed, especially for linear models. In the functional context, this topic is less explored.

Consider a general model for functional regression

$$Y = F(X) + \epsilon, \tag{3}$$

where $X \in L^2(\mathcal{I})$, $Y \in \mathbb{R}$, ϵ is independent form X, and the regression function F: $L^2(\mathcal{I}) \to \mathbb{R}$. Traditionally, one defines F as the conditional mean $E(Y|X)$ which is the minimizer of

$$E((Y - F(X))^2).$$

When the training data contain outliers, the conditional mean is easily affected and thus can produce misleading results. To deal with this problem, one can instead model the conditional median. The task then becomes to estimate F that minimizes

$$E(|Y - F(X)|).$$

A few proposals on functional quantile regression include the estimation of conditional median as a particular case. Cardot et al. (2005) and Kato (2012) considered methods that connected the conditional quantile of the response with a linear function of the functional variable. Ferraty et al. (2005) and Crambes et al. (2008) modelled the conditional quantile in a nonparametric way, extending the functional Nadaraya-Watson estimator introduced by Ferraty and Vieu (2002). Chen and Müller (2012) took an indirect approach by modelling and inverting the conditional distribution function. More recently, methods to estimate the conditional quantile have been developed for functional local linear regression (Kaid and Laksaci, 2017; Al-Awadhi et al., 2019), functional single index regression (Sang and Cao, 2020), support vector machines (Crambes et al., 2013), and partial functional regression (Qingguo and Kong, 2017).

Some other proposals adopted outlier-resistant loss functions to construct M-type estimators. ......(unfinished......q)

3 Methodology

We address here the problem to estimate the regression function $F : X \to Y$, where the explanatory variable $X \in L^2(\mathcal{I})$ and the response $Y \in \mathbb{R}$. Define the target function

$$F = \operatorname*{argmin}_G L(Y, G(x)),$$

over joint distribution of (X, Y) where L is a pre-specified loss function, we estimate F by adapting the robust gradient boosting procedure (`RRBoost`) proposed by Ju and Salibián-Barrera (2021).

The generalization of `RRBoost` to functional data requires using functional regressors as the base learners. A convenient choice is the type B tree adopted by our previously proposed `TFBoost`. We recall the two stage estimating procedure of `RRBoost` below and use type B trees as base learners to approximate the negative gradients in both stages:

- Stage 1:compute an S-type boosting estimator with high robustness but possibly low efficiency;
- Stage 2:compute an M-type boosting estimator initialized at the function and scale estimators obtained in Stage 1.

We call the resulted estimator `RTFBoost` and outline the algorithm as follows:

Algorithm 1: RTFBoost algorithm

Input : A data set $(\mathbf{x}_i, y_i), i \in \mathcal{I}_{\text{train}}$
The number of stage 1 iterations T_1
The number of stage 2 iterations T_2
The maximum depth of type B trees d
The number of random directions of type B trees P
Basis $\boldsymbol{\psi} = \{\psi_1, ..., \psi_d\}$
Shrinkage parameter ν

Initialize: $\hat{F}_0(\mathbf{x})$

Stage 1 :

for $t = 1 : T_1$ **do**

$\hat{\sigma}_n(\hat{F}_{t-1}) = \{\sigma : \frac{1}{|\mathcal{I}_{\text{train}}|} \sum_{i \in \mathcal{I}_{\text{train}}} \rho_0\left(\frac{y_i - \hat{F}_{t-1}(x_i)}{\sigma}\right) = \kappa\}$

$C_t = \left[\sum_{i \in \mathcal{I}_{\text{train}}} \psi_0\left(\frac{y_i - \hat{F}_{t-1}(x_i)}{\hat{\sigma}_n(\hat{F}_{t-1})}\right)\left(\frac{y_i - \hat{F}_{t-1}(x_i)}{\hat{\sigma}_n(\hat{F}_{t-1})}\right)\right]^{-1}$

$g_{t,\ell} = -C_t \psi_0\left(\frac{y_\ell - \hat{F}_{t-1}(x_\ell)}{\hat{\sigma}_n(\hat{F}_{t-1})}\right)$

Sample $\mathcal{P} = \{\mathbf{p}_1, ..., \mathbf{p}_P\}$

$h_t = \operatorname{argmin}_{h \in \mathcal{H}} \sum_{i \in |\mathcal{I}_{\text{train}}|} \left(g_{t,i} + h\left(\mathbf{p}_1^T \langle x_i \boldsymbol{\psi} \rangle, ..., \mathbf{p}_P^T \langle x_i \boldsymbol{\psi} \rangle\right)\right)^2$

$\alpha_t = \operatorname{argmin}_\alpha \hat{\sigma}\left(\hat{F}_{t-1} + \alpha h_t\right)$

$\hat{F}_t(x) = \hat{F}_{t-1}(x) + \alpha_t h_t(x)$

end

$\hat{\sigma} = \hat{\sigma}_n(\hat{F}_{T_1})$

Stage 2 :

for $t = 1 : T_2$ **do**

$g_{t,i} = -\frac{1}{\hat{\sigma}_n} \psi_1\left(\frac{y_i - \hat{F}_{T_1+t-1}(\mathbf{x}_i)}{\hat{\sigma}_n}\right)$

Sample $\mathcal{P} = \{\mathbf{p}_1, ..., \mathbf{p}_P\}$

$h_t = \operatorname{argmin}_{h \in \mathcal{H}} \sum_{i \in |\mathcal{I}_{\text{train}}|} \left(g_{t,i} + h\left(\mathbf{p}_1^T \langle x_i \boldsymbol{\psi} \rangle, ..., \mathbf{p}_P^T \langle x_i \boldsymbol{\psi} \rangle\right)\right)^2$

$\alpha_t = \operatorname{argmin}_\alpha \sum_{i \in \mathcal{I}_{\text{train}}} \rho_1\left(\frac{y_i - \hat{F}_{T_1+t-1}(x_i) - \alpha h_t(x_i)}{\hat{\sigma}}\right)$

$\hat{F}_{T_1+t}(x) = \hat{F}_{T_1+t-1}(x) + \alpha_t h_t(x)$

end

Output : $\hat{F}_{T_1+T_2}(x)$

3.1 Early stopping

Same as for `RRBoost`, we use an early stopping rule to determine T_1 and T_2 in Algorithm (1) in order to mitigate overfitting. We monitor the validation loss:

$$L_{1,\text{val}}(t) = \hat{\sigma}_{\text{val}}(\hat{F}_t),$$

for stage 1 where $\hat{\sigma}_{\text{val}}$ satisfies

$$\frac{1}{|\mathcal{I}_{\text{val}}|} \sum_{i \in \mathcal{I}_{\text{val}}} \rho_0\left(\frac{y_i - \hat{F}_{t-1}(x_i)}{\hat{\sigma}_{\text{val}}}\right) = \kappa,$$

and

$$L_{2,\text{val}}(t) = \sum_{i \in \mathcal{I}_{\text{val}}} \rho_1 \left(\frac{y_i - \hat{F}_{T_1+t}(x_i)}{\hat{\sigma}} \right)$$

for stage 2 where $\hat{\sigma} = \hat{\sigma}_n(\hat{F}_{T_1})$.

We let $T_{1,\max}$ and $T_{2,\max}$ be the maximum number of iterations allowed in each stage. The early stopping times for stage 1 and stage 2 are defined as

$$T_1 = \underset{t=1,\ldots,T_{1,\max}}{\operatorname{argmin}} L_{1,\text{val}}(t)$$

and

$$T_2 = \underset{t=1,\ldots,T_{2,\max}}{\operatorname{argmin}} L_{2,\text{val}}(t)$$

respectively.

3.2 Initialization

Since the loss functions involved in `RTFBoost` are typically non-convex, a reliable initialization step is required to avoid reaching a local optima with a large objective value. We suggest using a type B tree computed with the L1 loss, which we call "multi-index LAD tree". This generalizes `LADTree` recommended for `RRBoost` from multivariate variables to a functional variable.

To fit a multi-index LAD tree, we first sample P random directions $\mathbf{p}_1, ..., \mathbf{p}_P$ that satisfy $\|\mathbf{p}_j\| = 1$ and $p_{j,1} > 0$. Given a basis $\boldsymbol{\psi} = \{\psi_1, ..., \psi_d\}$, we calculate the candidate indices $\{\mathbf{p}_1^T \langle x_i, \boldsymbol{\psi} \rangle, ..., \mathbf{p}_P^T \langle x_i, \boldsymbol{\psi} \rangle\}$. At each node, a multi-idnex LAD tree finds the split that minimizes the average absolute value of the residuals in a step-wise manner. Specifically, let $i \in \mathcal{I}_o$ be all the observations at a node o and $R_1(j,s) = \{i | \mathbf{p}_j^T \langle x_i, \boldsymbol{\psi} \rangle \leq s\}$ and $R_2(j,s) = \{i | \mathbf{p}_j^T \langle x_i, \boldsymbol{\psi} \rangle > s\}$ be the left and right regions of the split made on the j-th index, we search for the split by solving

$$\min_{j \in \{1,\ldots,P\}} \left\{ \min_s \left(\min_{z_1} \sum_{i \in \{\mathcal{I}_o \cap R_1(j,s)\}} |y_i - z_1| + \sum_{i \in \{\mathcal{I}_o \cap R_2(j,s)\}} |y_i - z_2| \right) \right\}.$$

This procedure is repeated until the tree reaches the maximum depth (d_{init}) or minimum number of observations per node (m_{init}).

To select the parameters (d_{init} and m_{init}) of the initial tree, we first fit `RTFBoost` using the median of the responses as the initial fit:

$$\hat{F}_0(x) = \text{median}(\{y_i | i \in \mathcal{I}_{\text{train}}\}),$$

and flagged potential outliers ($\mathcal{I}_f$) as those with residuals deviating from their median by more than 3 times their MAD. Then we fit `RTFBoost` with multi-index LAD tree for each combination of (d_{init}, m_{init}) under consideration, and chose the one that performs the best on the validation set, where the performance is evaluated as the absolute deviation of residuals of non-flagged validation data points:

$$\sum_{i \in \mathcal{I}_{\text{val}}/\mathcal{I}_{\text{f}}} \left| y_i - \hat{F}_{T_1+T_2}(x_i) \right|.$$

4 Simulation

4.1 Data generation

- We generated data sets $D = \{(x_i, y_i), i = 1, ..., N\}$, consisting of a predictor $x_i \in \mathcal{L}_2$ and a scalar response y_i that follow the model:

$$y_i = r(x_i) + \rho\epsilon_i, \tag{4}$$

  where the errors ϵ_i are i.i.d, r is the regression function, and $\rho > 0$ is a constant that controls the signal-to-noise ratio (SNR):

$$\text{SNR} = \frac{\text{Var}(r(X))}{\text{Var}(\rho\epsilon)}.$$

- To sample the functional predictors x_i, we considered the model:

$$x_i(t) = \mu(t) + \sum_{p=1}^{4} \sqrt{\lambda_j}\xi_{ij}\phi_j(t), \tag{5}$$

  where $\mu(t) = 2\text{sin}(t\pi)\text{exp}(1-t)$, $\lambda_1 = 0.8, \lambda_2 = 0.3, \lambda_3 = 0.2$, and $\lambda_4 = 0.1$, $\xi_{ij} \sim N(0, 1)$, and ϕ_j are the first four eigenfunctions of the "Mattern" covariance function $\gamma(s, t)$ with parameters $\rho = 3, \sigma = 1, \nu = 1/3$:

$$\gamma(s,t) = C\left(\frac{\sqrt{2\nu}|s-t|}{\rho}\right),\ C(u) = \frac{\sigma^2 2^{1-\nu}}{\Gamma(\nu)} u^\nu K_\nu(u),$$

  where $\Gamma(.)$ is the Gamma function and K_ν is the modified Bessel function of the second kind. For each subject i, we evaluate x_i on a dense and regular grid $t_1, ..., t_{100}$ equally spaced in $\mathcal{I} = [0, 1]$.

- We considered five regression functions:
  - $r_1(X) = \int_{\mathcal{I}} \left(\sin\left(\frac{3}{2}\pi t\right) + \sin\left(\frac{1}{2}\pi t\right)\right) X(t)dt$,
  - $r_2(X) = (\xi_1 + \xi_2)^{1/3}$, where $\xi_1 = \int_{\mathcal{I}}(X(t) - \mu(t))\psi_1(t)dt$ and $\xi_2 = \int_{\mathcal{I}}(X(t) - \mu(t))\psi_2(t)dt$ are projections onto the first two FPCs (ψ_1 and ψ_2) of X with mean $\mu(t) = E(X(t))$,
  - $r_3(X) = 5\text{exp}\left(-\frac{1}{2}\left|\int_{\mathcal{I}} x(t)\log(|x(t)|)dt\right|\right)$,
  - $r_4(X) = 5\text{sigmoid}\left(\int_{\mathcal{I}} X(t)^2\text{sin}(2\pi t)dt\right)$, where $\text{sigmoid}(u) = 1/(1 + \exp(-u))$, and

- $r_5(X) = 5\left(\sqrt{\left|\int_{\mathcal{I}_1}\cos(2\pi t^2)X(t)dt\right|} + \sqrt{\left|\int_{\mathcal{I}_2}\sin(X(t))dt\right|}\right)$, where $\mathcal{I}_1 = [0, 0.5]$ and $\mathcal{I}_2 = (0.5, 1]$.

- For clean data (C_0), we generated ϵ_i in (4) from $N(0,1)$ and selected ρ that corresponds to SNR = 5.

  For contaminated data, we sampled 10% training samples as outliers and let the set of their indices be $\mathcal{O}$. The outliers belong to one of the five types introduced below. For $j \in \mathcal{O}$,

$$y_i = r(x_i) + \eta_i$$

  - C_1: *Shape outliers*

    In (5), $\xi_{j,2} \sim N(20.5)$, $\mu(t) = 1+2\cos(2t\pi)$, and the other parameters are the same as the model for clean data,

  - C_2: *Magnitude outliers (curve-type)*

    $x_j = 2\tilde{x}_j + 5$, where $\tilde{x}_j$ follows the model for clean data,

  - C_3: *Magnitude outliers (point-type)*

    Randomly sample 10 points form $t_1, ..., t_{100}$ and denote them as $t_{j,o_1}, ..., t_{j,o_{10}}$. For $k = 1, ..., 10$,

$$x_j(t_{j,o_k}) = \tilde{x}_j(t_{j,o_k}) + \eta_{j,o_k},$$

    where $\eta_{j,o_k} \sim 0.5N(10, 0.25) + 0.5N(-10, 0.25)$, where $\tilde{x}_j$ follows the model for clean data,

  - C_4: *Magnitude outliers (interval-type)*

    Randomly sample one interval from intervals $[t_1, ..., t_{10}]$, ...,$[t_{91}, ..., t_{100}]$, and denote the interval as $t_{j,o}, ..., t_{j,o+9}$
    For $k = 0, ..., 9$,

$$x_j(t_{j,o+k}) = \tilde{x}_j(t_{j,o+k}) + \eta_{j,o+k},$$

    where $\eta_{j,o+k} \sim N(10, 0.25)$, and $\tilde{x}_j$ follows the model for clean data

  - C_5: *vertical outliers*

$$x_j = \tilde{x}_j,$$

    where $\tilde{x}_j$ follows the model for clean data.

  For contaminated settings C_1,..., C_5, we generate η_i from

$$0.5N(10, 0.25) + 0.5N(-10, 0.25)$$

4.2 Model comparison

For each setting, we used 100 independently generated datasets and compared the performance of the following methods:

- `FPPR`: functional projection pursuit regression (Ferraty et al., 2013),
- `FGAM`: functional generalized additive models (McLean et al., 2014),
- `MFLM`: Sieve M-estimator for a semi-functional linear model (Huang et al., 2015),
- `RFSIR`: robust functional sliced inverse regression (Wang et al., 2017)
- `RFPLM`: robust estimation for semi-functional linear regression models (Boente et al., 2020),
- `TFBoost(L2)`: tree-based functional boosting with L2 loss,
- `TFBoost(LAD)`: tree-based functional boosting with LAD loss,
- `RTFBoost`: proposed robust tree-based functional boosting.

4.3 Implementation details

- For `FPPR`, we implemented the method based on the code to fit a functional single index model shared by the authors of Ferraty et al. (2013).
- For `FGAM`,

`TFBoost(L2)` were computed using decision trees of the maximum depth equal to 1, 2, 3, and 4 as base learners

References

Al-Awadhi, F. A., Kaid, Z., Laksaci, A., Ouassou, I., and Rachdi, M. (2019). Functional data analysis: local linear estimation of the l1-conditional quantiles. *Statistical Methods & Applications*, 28(2):217–240.

Arribas-Gil, A. and Romo, J. (2014). Shape outlier detection and visualization for functional data: the outliergram. *Biostatistics*, 15(4):603–619.

Boente, G., Salibian-Barrera, M., and Vena, P. (2020). Robust estimation for semi-functional linear regression models. *Computational Statistics & Data Analysis*, 152:107041.

Cardot, H., Crambes, C., and Sarda, P. (2005). Quantile regression when the covariates are functions. *Nonparametric Statistics*, 17(7):841–856.

Chen, K. and Müller, H.-G. (2012). Conditional quantile analysis when covariates are functions, with application to growth data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):67–89.

Crambes, C., Delsol, L., and Laksaci, A. (2008). Robust nonparametric estimation for functional data. *Journal of Nonparametric Statistics*, 20(7):573–598.

Crambes, C., Gannoun, A., and Henchiri, Y. (2013). Support vector machine quantile regression approach for functional data: Simulation and application studies. *Journal of Multivariate Analysis*, 121:50–68.

Dai, W., Mrkvička, T., Sun, Y., and Genton, M. G. (2020). Functional outlier detection and taxonomy by sequential transformations. *Computational Statistics & Data Analysis*, 149:106960.

Ferraty, F., Goia, A., Salinelli, E., and Vieu, P. (2013). Functional projection pursuit regression. *Test*, 22(2):293–320.

Ferraty, F., Rabhi, A., and Vieu, P. (2005). Conditional quantiles for dependent functional data with application to the climatic" el niño" phenomenon. *Sankhyā: The Indian Journal of Statistics*, pages 378–398.

Ferraty, F. and Vieu, P. (2002). The functional nonparametric model and application to spectrometric data. *Computational Statistics*, 17(4):545–564.

Huang, L., Wang, H., Cui, H., and Wang, S. (2015). Sieve m-estimator for a semi-functional linear model. *Science China Mathematics*, 58(11):2421–2434.

Hyndman, R. J. and Shang, H. L. (2010). Rainbow plots, bagplots, and boxplots for functional data. *Journal of Computational and Graphical Statistics*, 19(1):29–45.

Ju, X. and Salibián-Barrera, M. (2021). Robust boosting for regression problems. *Computational Statistics & Data Analysis*, 153:107065.

Kaid, Z. and Laksaci, A. (2017). Functional quantile regression: local linear modelisation. In *Functional Statistics and Related Fields*, pages 155–160. Springer.

Kato, K. (2012). Estimation in functional linear quantile regression. *The Annals of Statistics*, 40(6):3108–3136.

McLean, M. W., Hooker, G., Staicu, A.-M., Scheipl, F., and Ruppert, D. (2014). Functional generalized additive models. *Journal of Computational and Graphical Statistics*, 23(1):249–269.

Qingguo, T. and Kong, L. (2017). Quantile regression in functional linear semiparametric model. *Statistics*, 51(6):1342–1358.

Rousseeuw, P. J. and Van Zomeren, B. C. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical association*, 85(411):633–639.

Sang, P. and Cao, J. (2020). Functional single-index quantile regression models. *Statistics and Computing*, pages 1–11.

Wang, G., Zhou, J., Wu, W., and Chen, M. (2017). Robust functional sliced inverse regression. *Statistical papers*, 58(1):227–245.