Functional Regression Model with Bike Data

Liela Meng[1]
[1]Division of Biostatistics, UC Davis

1553 Words Student ID:917843295
Keywords: functional regression model, model-based boosting, classification
10th March, 2021

Abstract

In this analysis, two years of daily data from the bike-sharing system was analyzed to predict the total count of bicycles per hour with a function-on-function regression model and to investigate the classification performance via fitting a scalar-on-function regression model with a binary response. Besides, curves measured with significant low depths were viewed as outliers and dropped from modeling.

1 Introduction

The bike-sharing systems provide a service for either registered membership or casual customers to rent a bike. If we can predictor the number of the bike in demands based on known predictors, the analysis can help redistribute bicycles. Besides, this data set is provided on the UCI Mashing Learning Repository website [1] and was utilized in an article to classify a day as a working day or not based on the count of the rented bike and other predictors.

Thus, after excluding outliers, this analysis intends to predict the number of demands with several functional or scalar predictors and to investigate a functional regression model's classification performance with binary response.

2 Background

The data set contains the hourly count of rental bikes between 2011 and 2012 in the Capital bike-share system with the functional (temperature, wind speed, humidity, etc.) and scalar covariates (working day, season, weekday, etc.).

For the purpose of this analysis, 731 days (except for two outliers) hourly data were utilized in the analysis and the selected variables are hour index, day index, hourly temperature, hourly humidity, hourly wind speed, working day indicator, season index, and weekday index.

3 Methods

3.1 Outlier detection

Depth is a way of measuring the centrality of a point such that points near the center have higher depth. Fraiman and Muniz extended the notion of depth in Euclidean space to

functional data and introduced functional data depth in 2001 , which measures how long a curve remains in the middle of the whole trajectories. [2]

Assuming the whole set of curves has been drawn from the same stochastic process, then curves having a significant low depth can be viewed as outliers. (see Appendix A and [2] for more details about the procedure of detecting outliers.)

3.2 Functional regression model with FDboost

Greven and Scheipl discussed two approaches to estimate covariate effects for functional regression model: based on mixed models framework or component-wise gradient boosting. [3] In this analysis, I used R package *FDboost* which implements the latter approach.

Functional regression model

Let the random variable $Y(t)$ be the functional response and covariate set $X = \{Z, X(s)\}$ include both scalar and functional variables. In particular, Z denotes a scalar covariate set and $X(s)$ denotes a functional covariate set.

Let i be the date index and t as the time index. Modeling the expectation of response by an additive regression model: $E(Y_i(t)|X_i = x_i) = h(x_i, t) = \sum_j h_j(x_i, t)$. We can see, response $Y_i(t)$, linear predictor $h(x_i, t)$ and additive effects $h_j(x_i, t)$ are functions of time t. (See Appendix A and [3] for more details about procedures to estimate coefficients in *FDboost*.)

In this analysis, both response and covariates are observed on a common grid of evaluation points $s/t = 1, 2, \ldots, 24$. Assume linear functional effect of working day indicator that varies smoothly over time, and non-linear effect of temperature and humidity with the form $\int x(s)\beta(s)ds$. Based on stability selection (selection of influential variables or model components with error control), choosing three variables with upper bound for the per-family error rate as 1, the covariates were selected in the final model are temperature, humidity, and working day indicator.

Hence, Y is the count of rented bikes per hour; $Z = 0$ if that day is a weekend or holiday. X includes the normalized temperature in Celsius and humidity per hour. Since the response variable is a count, instead of fitting the model directly using $h(x_i, t)$, a log link function would be used: $log(E(Y_i(t)|x_i)) = log(\mu(t)) = h(x_i, t)$, $Y_i(t) \sim Poisson(\mu(t))$. The final model to estimate the count of bike is:

$$\begin{aligned} log(E(Y(t)|x)) = \beta_0(t) &+ \int_s x_{temp}(s)\beta_{temp}(s,t)ds \\ &+ \int_s x_{hum}(s)\beta_{hum}(s,t)ds + Z_{work}\beta_{work}(t) \end{aligned}$$

Similarly, we have a scalar-on-function model for the binary response variable with logit link function.

Working Day prediction model

$$logit(E(Z|x)) = \beta_0 + \int_s x_{temp}(s)\beta_{temp}(s)ds + \int_s x_{hum}(s)\beta_{hum}(s)ds + \int_s x_{count}(s)\beta_{count}(s)ds$$

Estimation by gradient boosting

Boosting was recognized as a model fitting technique to iteratively improve base learners' predictive performance. Model-based boosting allows for a component-wise fitting of additive terms and can define each covariate's effects separately in different base learners. Component-wise gradient boosting minimized the loss function via gradient descent in a step-wise procedure and iteratively selecting only one base learner in each boosting step. (see appendix A and [3] for more details)

Three tuning parameters affect the resulting estimation and prediction performance: the number of boosting iteration m_{stop}, the step-length v, and the specification of base learners (number of knots, degree of freedom, smoothing parameter). In the package *FDboost*, it uses fixed step-length and base-learner-specific tuning parameters for all iterations. Hence, only the number of iterations would determine the fitted model, and the optimal stopping iteration was determined by empirical risk estimation using a specified resampling method.

One thing to notice is that boosted models can not provide classical formal inference due to variable selection (among each iteration in boosting algorithm) and shrinkage of coefficient estimates. (see [4] for details regarding component-wise gradient boosting algorithm utilized in the package *FDboost*.)

4 Results

4.1 Outlier

Based on Fraiman Muniz depth and the Bootstrap procedure, we can see there are the outliers for two years' count data when setting cut-point as 0.5 and sampling 200 samples with smoothing parameter 0.05: date 2012-07-04 and 2012-09-08.

Figure 3 shows the two outliers and we can see those colorful outliers have a different shape than the rest of the curves. Hence, I subtracted those two days for followed analysis.

4.2 Functional regression model

Function-on-function model

Setting the dimension of the basis that is used for estimating the offset as 15, and using cubic B-spline on 10 knots with a first difference penalties for temperature and humidity, the effects of those two predictors are shown below.

In Figures 1 and 2, red color denotes positive association while blue denotes negative association between covariate and the count of bike in the left perspective plots. To have a closer look at the relationship in another way, we can look at the contour plots in the right panel.

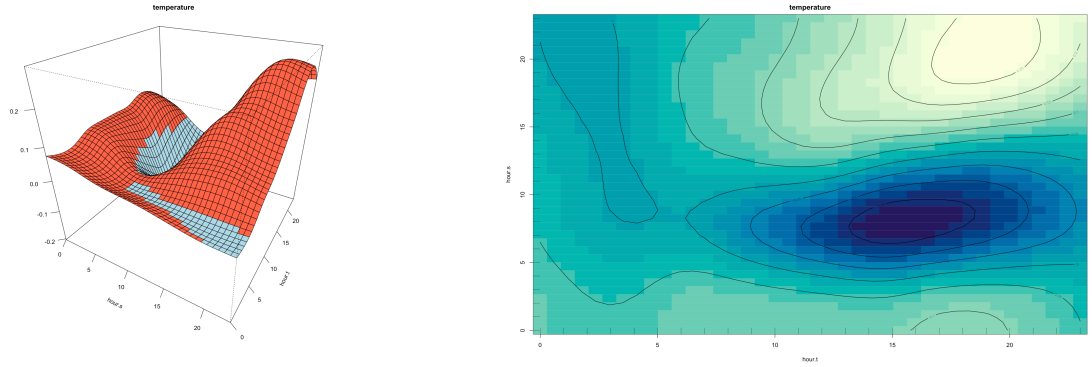

Fig. 1. Left: Perspective plot of temperature. Right: Contour plot of temperature

For the contour plots in Figures 1 and 2, lighter color denotes a larger coefficient estimation and vice versa. We can view green as no effect, white as a relatively high positive effect, and purple as a negative effect.

For temperature in Figure 1, one illustration for interpreting the temperature contour plot: for the time point t at 3 PM, the temperature within the time point from 5 AM to 11 AM negatively associated with the count of the bike, but the expected count is positively related with temperature from 2 PM to later on.

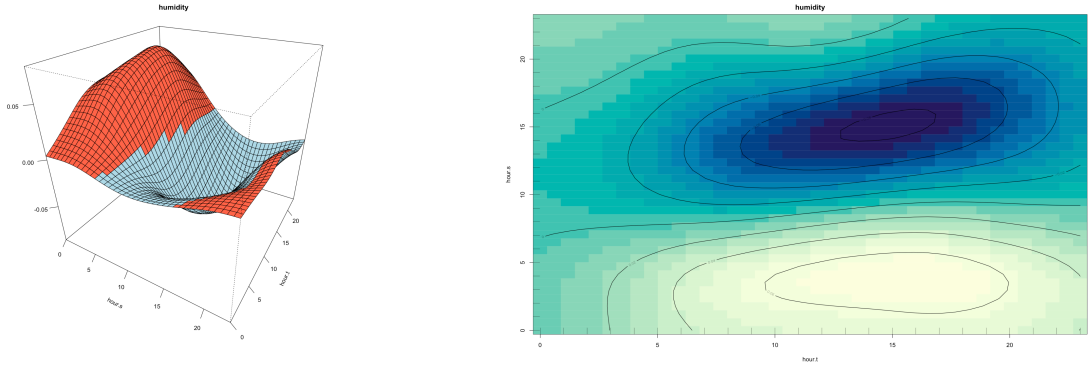

Fig. 2. Left: Perspective plot of humidity. Right: Contour plot of humidity

As for humidity in Figure 2, for example, the effect of humidity on the count of the bike at 3 PM is highly negatively associated with humidity within 12 PM to 5 PM. It makes sense since high humidity might indicate raining and people tend to avoid riding a bike when rainy.

In figure 4, the red dash line corresponds to working days. For working days, around commute time, we can see two peaks (7 AM and 6 PM). As for non-working days, it has only one peak around 1 PM and the two trajectory seems inverse. This pattern indicates working

day plays a crucial role in determining the time of large demands of bikes and is consistent with what I discovered in project 1 with smoothing.

Scalar-on-function model

Fitting the scalar-on-function regression model to classify a day into a working day or non-working day based on the count of bikes, temperature, and humidity. Table 1 shows the prediction result and the misclassification rate is relatively low (1.5%).

Table 1
Summary of predicted classification vs observed classification

	observe (notwork)	observe (work)
predict (not work)	232	6
predict (work)	5	488

5 Discussion

Though the effects of temperature, humidity, and working day are as expected and the classification performance of the second model is well. This analysis has several limitations:

First, as I mentioned in section 3, one major disadvantage of boosting approach is the lack of formal inference. [3] However, the R package *refund* allows the user to formally test pre-specified hypotheses. Thus, I recommend conducting such tests in future research to test whether the corresponding predictor has a significant effect on response.

Besides, the optimal number of iteration was supposed to determined via resampling methods like cross-validation or bootstrapping to achieve the minimized empirical risk. Considering a trade-off between computing time and flexibility of each base learner, I decided to use step-length as $v = 0.001$ and a number of iterations and as $m_{stop} = 100$ since the model becomes more complex with more boosting iteration and this number is not the optimal number provided by the algorithm. However, based on plots for estimations of coefficients, the number of iteration seems to do not have a huge impact on the response when compared with $m_{stop} = 500$, I suggest further investigation regarding the effect of the number of iteration.

Lastly, the Poisson distribution assumes equality between variance and mean. But it assumption might not holds thus I fitted another model - negative binomial model and the intercept (Figure 5) seems similar, hence the model we obtained might be more appropriate than we expected.

6 Acknowledgement

I would like to thank Professor Hans-Georg Müller and Han Chen for their help in writing this report.

7 References

1. Fanaee-T, H. & Gama, J. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence,* 1–15 (2013).
2. Manuel G., W. G. *Outlier detection in functional data by depth measures, with application to identify abnormal NOx levels*
3. Sarah Brockhaus David Rügamer, S. G. *Boosting Functional Regression Models with FDboost* https://www.jstatsoft.org/article/view/v094i10.
4. Benjamin Hofner Andreas Mayr, e. a. *Model-based Boosting in R: A Hands-on Tutorial Using the R Package mboost* https://cran.r-project.org/web/packages/mboost/vignettes/mboost_tutorial.pdf.

Appendix A

The Fraiman and Muniz (FM) depth:

$$FMD(y_i) = \int D(y_i(t))dt,$$

where $D(x_i(t))$ is the univariate depth of the point $y_i(t)$: $D(y_i(t)) = 1 - |\frac{1}{2} - F_t(y_i(t))|$, and $F_t(x_i(t))$ is the empirical cumulative distribution function of the values of the curves $y_1(t), \ldots, y_n(t)$.

Estimating effects of covariates

The additive effect $h_j(x_i, t)$ are linearized using a basis representation while Kronecker product of marginal bases represent the effects: $h_j(x_i, t) = b_j(x_i, t)^T\theta_j = (b_j(x_i)^T \otimes b_Y(t)^T)\theta_j$.

Gradient booting

For functional response, the algorithm compute the loss at each point t and integrate it over the domain of the response. In this case, we have count data, thus boosting aims at minimizing the Poisson loss function: $\sum_{i=1}^{N} \int [h(x_i, t) - y_i(t)log(h(x_i, t))]^2 dt$.

Appendix B - Plot

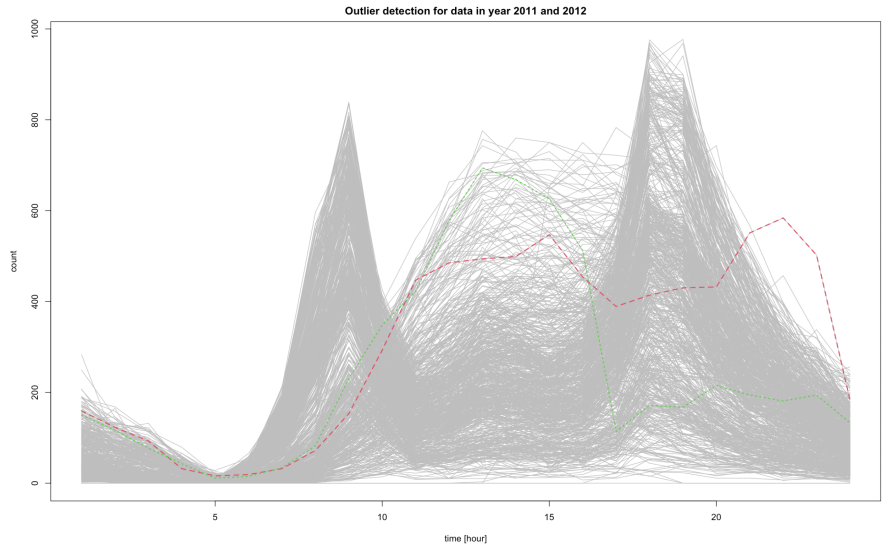

Fig. 3. Two outliers for functional data

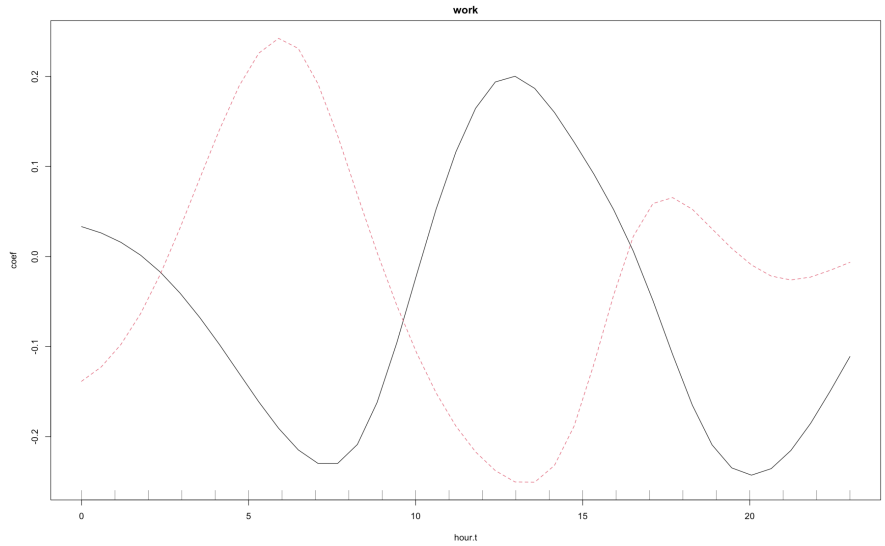

Fig. 4. Coefficient estimation (work) of the function-on-function regression model for the count of bike

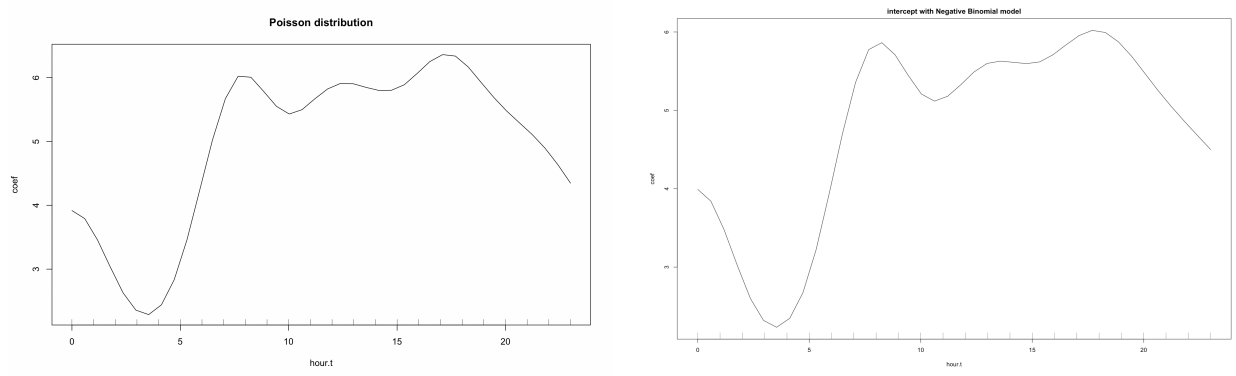

Fig. 5. Left: Intercept for Poisson model. Right: Intercept for Negative Binomial model

Appendix C - R code

```r
library(readr)
library(fdapace)
library(tidyverse)
library(plyr)
library(dplyr)
library(fda)
library(lubridate)
library(ggplot2)
library(fdapace)
library(splines)
library(Matrix)
library(fds)
library(rainbow)
library(MASS)
library(pcaPP)
library(RCurl)
library(fda.usc)
library(mboost)
library(parallel)
library(stabs)
library(FDboost)

##### two years data - data__all
    ------------------------------------------------

# outlier
    ---------------------------------------------------------------------
hour <- read.csv("hour.csv")
# filter
data_all0_all <- hour %>% mutate(date=ymd(dteday)) %>%  dplyr::select(
    hr,cnt,date)
data_all <- data_all0_all %>% pivot_wider(
  names_from = date,
  values_from = cnt,
  values_fill = list(cnt = 0)
) %>% as.data.frame()
# set up
data_all1 <- t(data_all) [-1,]%>% as.data.frame()
data_all_fdata_all <- fdata(data_all1)
plot(optim.np(data_all_fdata_all)$fdata_all.est)
# detecting outliers
outliers.depth.pond(data_all_fdata_all,nb=200) # outliers "2012-07-04"
    "2012-09-08" # dep.out 9.707107 9.336198
# plotting
data0 <- hour %>% mutate(date=ymd(dteday)) %>%  dplyr::select(hr,cnt,
    date)
data <- data0 %>% pivot_wider(
  names_from = date,
  values_from = cnt,
  values_fill = list(cnt = 0)
) %>% as.data.frame()
# set up
data1 <- t(data) [-1,]%>% as.data.frame()
data_fdata <- fdata(data1)
plot(optim.np(data_fdata)$fdata.est)
```

```r
# detecting outliers
outliers.depth.pond(data_fdata,nb=200) # 18,26,27
# plot outliers
plot(data_fdata,col="grey",main="Outlier detection for data in Jan 2011",
     xlab = "time [hour]", ylab = "count" )

out1 <- data0 %>% filter(date== "2012-07-04"| date=="2012-09-08")
out2 <- out1 %>% pivot_wider(
  names_from = date,
  values_from = cnt,
  values_fill = list(cnt = 0)
) %>% as.data.frame()
out <- t(out2) [-1,]%>% as.data.frame()
data_fdata_out <- fdata(out)
plot(data_fdata,col="grey",main="Outlier detection for data in year 2011 and 2012",
     xlab = "time [hour]", ylab = "count" )
lines(data_fdata_out ,lwd=2,lty=2:4,col=2:4)

# set up dataset
# ------------------------------------------------------------
# filter
data_all0_all <- hour %>% mutate(date=ymd(dteday))%>%
  filter(date %in% c("2012-07-04","2012-09-08")== FALSE)
# count
count <- data_all0_all %>%  dplyr::select(hr,cnt,date) %>%
  mutate(hr=paste("hour",hr))  %>% pivot_wider(
  names_from = date,
  names_prefix = "date_",
  values_from = cnt,
  values_fill = list(cnt=0)
) %>% as.data.frame() %>% dplyr::select(-hr)
# temp
temp <- data_all0_all %>%  dplyr::select(hr,temp,date) %>%
  mutate(hr=paste("hour",hr))   %>% pivot_wider(
  names_from = date,
  names_prefix = "date_",
  values_from = temp,
  values_fill = list(temp=0)
) %>% as.data.frame() %>% dplyr::select(-hr)
# work
work <- data_all0_all %>%  dplyr::select(hr,workingday,date) %>%
  mutate(hr=paste("hour",hr))%>% pivot_wider(
  names_from = date,
  names_prefix = "date_",
  values_from = workingday,
  values_fill = list(workingday=0)
)  %>% filter(hr=="hour 1") %>% dplyr::select(-hr) %>% as.matrix()
work <- t(work) %>% as.data.frame() %>% mutate(work=as.factor(V1)) %>%
    mutate(work = work
                                          %>%fct_recode(
                                          "neither weekend nor holidate" = "1",
                                          "weekend nor holidate" = "0"
                                        )) %>% dplyr::select(work)  %>%
    as.matrix()
# wind
```

```
wind <- data_all0_all %>%  dplyr::select(hr,windspeed,date) %>%
  mutate(hr=paste("hour",hr))  %>% pivot_wider(
  names_from = date,
  names_prefix = "date_",
  values_from = windspeed,
  values_fill = list(windspeed=0)
) %>% as.data.frame() %>% dplyr::select(-hr)
# week
week <- data_all0_all %>%  dplyr::select(hr,weekday,date)  %>%
  mutate(hr=paste("hour",hr))  %>% pivot_wider(
  names_from = date,
  names_prefix = "date_",
  values_from = weekday,
  values_fill = list(week=0)
) %>% filter(hr=="hour 1") %>% dplyr::select(-hr) %>% as.matrix()
week <- t(week) %>%
  as.data.frame() %>% mutate(week=as.factor(V1)) %>%
  mutate(week = week %>% fct_recode( "Sundate" = "0",
                                     "Mondate" = "1",
                                     "Tuesdate" = "2",
                                     "Wednesdate" = "3",
                                     "Thursdate" = "4",
                                     "Fridate" = "5",
                                     "Saturdate"= "6")) %>% dplyr::
    select(week) %>%
  mutate(week=week %>% fct_relevel("Saturdate", "Sundate"))  %>% drop_
    na() %>% as.matrix()
# date
date <- data_all0_all %>% dplyr::select(hr,weekday,date)%>%
  mutate(date=paste("date",date)) %>% pivot_wider(
  names_from = hr,
  names_prefix = "hr_",
  values_from = weekday,
  values_fill = list(week=0)
)  %>% dplyr::select(date) %>% mutate(date=as.factor(date)) %>% as.
    matrix()
# weather
weather <- data_all0_all %>%  dplyr::select(hr,weathersit,date)  %>%
  mutate(hr=paste("hour",hr))  %>% pivot_wider(
  names_from = date,
  names_prefix = "date_",
  values_from = weathersit,
  values_fill = list(weathersit=1)
) %>% dplyr::select(-hr)  %>% as.matrix()
# hum
humidity <- data_all0_all %>%  dplyr::select(hr,hum,date)  %>%
  mutate(hr=paste("hour",hr))  %>% pivot_wider(
  names_from = date,
  names_prefix = "date_",
  values_from = hum,
  values_fill = list(hum=0)
) %>% dplyr::select(-hr)
# season
season <- data_all0_all %>%  dplyr::select(hr,season,date)  %>%
  mutate(hr=paste("hour",hr))  %>% pivot_wider(
  names_from = date,
  names_prefix = "date_",
  values_from = season,
```

```
  values_fill = list(season=3)
)  %>% filter(hr=="hour 1") %>% dplyr::select(-hr) %>% as.matrix()
season <- t(season) %>% as.data.frame() %>%
  mutate(season=as.factor(V1)) %>%  mutate(season = season %>%
                                             fct_recode("winter" = "1",
                                                        "spring" = "2",
                                                        "summer" = "3",
                                                         "fall" = "4"))
    %>%
  dplyr::select(season) %>% as.matrix()

# data__all
data__all <- list(count = t(count), temp = t(temp),wind=t(wind),
                  work=as.factor(work),week=as.factor(week),
                  date=as.factor(date),
                  humidity=t(humidity),season=as.factor(season))

# define function indices
data__all$hour.t <- 0:23
data__all$hour.s <- 0:23

# center temperature curves:
# data__all$temp <- sweep(data__all$temp, 2, colMeans(data__all$temp))
data__all$humidity <- scale(data__all$humidity,scale = F)
data__all$temp <- scale(data__all$temp,scale = F)
data__all$weather <- scale(data__all$weather,scale = F)
data__all$wind <- scale(data__all$wind,scale = F)

# k min
   ------------------------------------------------------------------------

k_min10 <- FDboost(count ~ 1, timeformula = ~ bbs(hour.t, knots = 11,
   cyclic = TRUE, df=3), offset_control = o_control(k_min = 10), data=
   data__all)
k_min15 <- FDboost(count ~ 1, timeformula = ~ bbs(hour.t, knots = 11,
   cyclic = TRUE, df=3), offset_control = o_control(k_min = 15), data=
   data__all)

par(mfrow=c(2,2))
plot(k_min10,ask=F,main="offset k_min=10")
plot(k_min15,ask=F,main="offset k_min=15") # choose K_min=15

# final model
   ------------------------------------------------------------------
final <- FDboost(count ~ 1
                 + bsignal(temp, hour.s, knots = 10, df = 4)
                 + bsignal(humidity, hour.s, knots = 10,  df = 4)
                 + bols(work,df=1),
                 timeformula = ~ bbs(hour.t, knots = 10,  df=4),
                 offset_control = o_control(k_min = 15),
                 control=boost_control(mstop = 100,nu=0.001),
                 family=Poisson(),
                 data=data__all)
stabsel(final,q=4, PFER = 1)
validateFDboost(final)
fitted(final)
funMRD(final)
funMSE(final)
```

```
funRsquared(final)
# opt No.
folds <- cv(weights = rep(1, final$ydim[1]),type="subsampling",B=10)
opt <- applyFolds(final,folds=folds,grid = 1:100,mc.cores=2)
final <- final[mstop(opt)]
final <- final[10]
## plot the effect of temperature
plot(final, which = 1,pers = TRUE, main = "intercept with Poisson model
    ")
plot(final, which = 2,pers = TRUE, main = "temperature", zlab = "")
plot(final, which = 2,pers = F, main = "temperature", zlab = "",col=hcl
    .colors(20,"YlGnBu"))
plot(final, which = 3,pers = TRUE, main = "humidity", zlab = "")
plot(final, which = 3,pers = F, main = "humidity", zlab = "",col=hcl.
    colors(20,"YlGnBu"))
plot(final, which = 4,pers = TRUE, main = "work")
# residuals
residual <- residuals(final)

# predict - work
    ------------------------------------------------------------

pred0 <- data__all
pred0$count <- scale(data__all$count,scale = F)
predict <- FDboost(work ~ 1
                   + bsignal(temp, hour.s, knots = 10, df = 3)
                   + bsignal(humidity, hour.s, knots = 10,  df = 3)
                   + bsignal(count, hour.s, knots = 10, df = 3),
                   timeformula = NULL,  control=boost_control(mstop =
    1000),
                   data=pred0,
                   family=Binomial())
pred <- predict(predict, type="response")
round_predes <- round(pred)
table(round_predes, as.numeric(pred0$work))


# plotting pred,res,obs
    ----------------------------------------------------

# ind <- sapply(1:31, function(s){ which(data_my$day == ord[s]) })
ind <- rep(1:20,1)
smoothRes <- predict(final)
residual <- residuals(final)
# if( is.null(dim(smoothRes)) ) smoothRes <- matrix(0, ncol = 24, nrow
    = 31)
# smoothRes <- (smoothRes)[ind, ]
# smoothRes <- ( predict(mod4, which=3) )[ind, ]
workOrd <- data__all$work[ind]
fit3 <- (predict(final))[ind, ]
response <- data__all$count[ind, ]
date <- data__all$date
library(maps)
par(mfrow=c(4,5))
for(i in 1:20) {
  plot(1:24, smoothRes[i, ], col = as.numeric(workOrd[i]), type="b",
       main = paste(date[i]),
       cex = 1.2, cex.axis = .8, ylab = "", xlab = "")
```

```
  abline(h = 0, col = 8)
}
# residuals
par(mfrow=c(4,5))
#layout(rbind(matrix(1:28, 4, 8), rep(29, 4), rep(29, 4)))
for(i in 1:20) {
  plot(1:24, residual[i, ], col = as.numeric(workOrd[i]),
       # ylim = range(residual_Jan, response-fit3),
       main = paste(data__all$date[i]),
       cex = 1.2, cex.axis = .8, ylab = "", xlab = "")
  abline(h = 0, col = 8)
}
# observed
par(mfrow=c(4,5))
#layout(rbind(matrix(1:28, 4, 8), rep(29, 4), rep(29, 4)))
for(i in 1:20) {
  plot(1:24, data__all$count[i, ], col = as.numeric(workOrd[i]), type="
    b",
       # ylim = range(residual_Jan, response-fit3),
       main = paste(data__all$date[i]),
       cex = 1.2, cex.axis = .8, ylab = "", xlab = "")
  abline(h = 0, col = 8)
}
# plotPredicted(final_Jan)
```

Bike-sharing Data Analysis
Functional Regression Model

Liela Meng
University of California, Davis, Department of Biostatistics

Graduate Group in BIOSTATISTICS

Introduction

From the UCI Mashing Learning Repository website, variables included in the bike sharing data set are:

- Functional: temperature, humidity, wind speed;
- Scalar: working day, season, weekday;
- Index: hour, date, count of bike.

The aim of this project:

- Modeling the count of bike with function-on-function regression model;
- Classify the day as a working day or not with counts and other covariates.

Methods

Outlier detection:

The Fraiman and Muniz (FM) depth:

$$FMD(y_i) = \int D(y_i(t))dt,$$

where $D(x_i(t))$ is the univariate depth of the point $y_i(t)$: $D(y_i(t)) = 1 - |\frac{1}{2} - F_t(y_i(t))|$, and $F_t(x_i(t))$ is the empirical cumulative distribution function of the values of the curves $y_1(t), \ldots, y_n(t)$.

Function-on-function model:

Let the random variable $Y(t)$ be the functional response and covariate set $X = \{Z, X(s)\}$ include both scalar and functional variables. In particular, Z denotes a scalar covariate set and $X(s)$ denotes a functional covariate set. Let i be the date index and t as the time index.

$$log(E(Y(t)|x)) = \beta_0(t) + \int_s x_{temp}(s)\beta_{temp}(s,t)ds + \int_s x_{hum}(s)\beta_{hum}(s,t)ds + Z_{work}\beta_{work}(t)$$

Scalar-on-function model:

$$logit(E(Z|x)) = \beta_0 + \int_s x_{temp}(s)\beta_{temp}(s)ds + \int_s x_{hum}(s)\beta_{hum}(s)ds + \int_s x_{count}(s)\beta_{count}(s)ds$$

Results

Outlier detection:

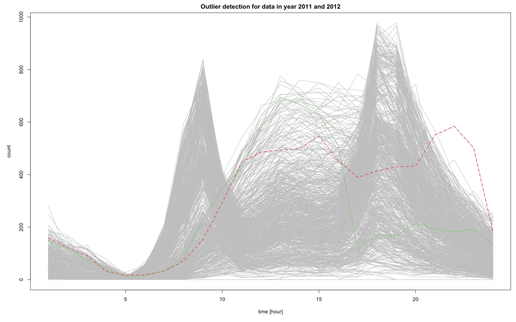

Fig. 1. Two Outliers based on FM depth

Results

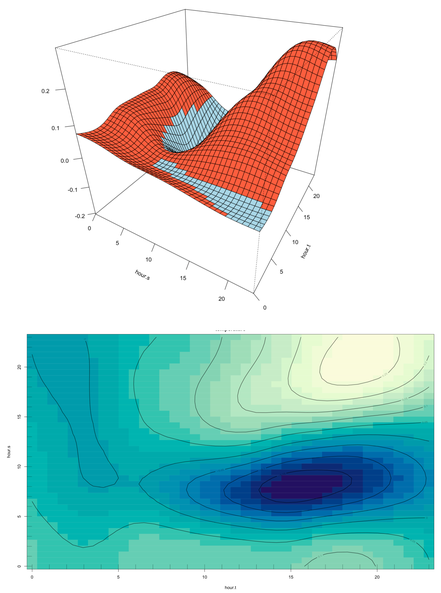

Fig. 2. Perspective plot of temperature. Right: Contour plot of temperature

Figure 2, 3 interpretation:

- Perspective plot: red color denotes positive association while blue denotes negative association in the left contour plot.
- Contour plot: lighter color denotes a larger coefficient estimation and vice versa. We can view green as no effect, white as a relatively high positive effect, and purple as a negative effect.

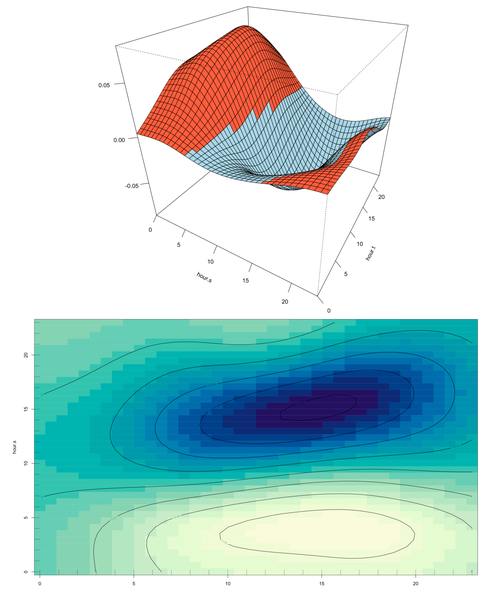

Fig. 3. Perspective plot of temperature. Right: Contour plot of humidity

Results

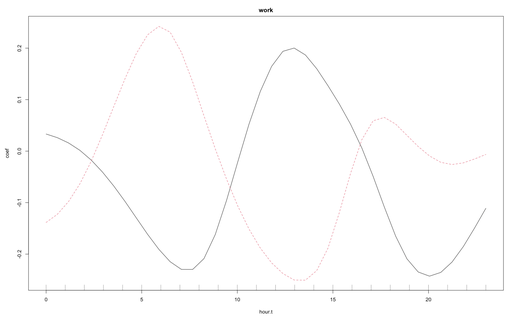

Fig. 4. Coefficient estimation (work) of the function-on-function regression model

Table 1
Summary of predicted classification vs observed classification

	observe (notwork)	observe (work)
predict (not work)	232	6
predict (work)	5	488

Discussion

- Lack of formal inference: the R package refund allows the user to formally test pre-specified hypotheses. Thus, I recommend conducting such tests in future research to test whether the corresponding predictor has a significant effect on response.
- considering a trade-off between computing time and flexibility of each base learner, I decided to use step-length as 0.001 and a number of iterations and as 100. Though based on coefficient estimation plot, the number of iteration seems to do not have a huge impact on the response when compared with 500, I suggest further investigation regarding the effect of the number of iteration.

Acknowledgements

I would like to thank Professor Hans-Georg Mueller and Han Chen for their help in writing this report.
Data was downloaded from UCI Mashing Learning Repository website:
https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

References

[1] Fanaee-T, H. & Gama, J. Event labeling combining ensemble detectors and backgroundknowledge.Progress in Artificial Intelligence,1–15 (2013).
[2] Manuel G., W. G.Outlier detection in functional data by depth measures, with applicationto identify abnormal NOx levels
[3] Sarah Brockhaus David R¨ugamer, S. G.Boosting Functional Regression Models with FDboost.
[4] Benjamin Hofner Andreas Mayr, e. a.Model-based Boosting in R: A Hands-on TutorialUsing the R Package mboost