BIOS 7720: Applied Functional Data Analysis

Lecture 12: Generalized Function on Scalar Regression

Andrew Leroux

April 27, 2021

Logistics

- HW 2 due yesterday (solutions posted tomorrow)
- HW 3 posted tomorrow
- Final group projects
  - Presentations 5/11 and 5/13
  - Rubric for write up to be posted tomorrow
  - Write up due 5/20

GFoSR

$$g(E[\mathrm{y}_i(s)|\mathrm{x}_i, \boldsymbol{b}_i]) = f_0(s) + f_1(s)\mathrm{x}_i + b_i(s)$$
$$b_i(s) \sim \mathrm{GP}(0, \boldsymbol{\Sigma}_b)$$
$$g(\cdot) \text{ Link function}$$

GFoSR

$$g(E[\mathrm{y}_i(s)|\mathrm{x}_i, \boldsymbol{b}_i]) = f_0(s) + f_1(s)\mathrm{x}_i + b_i(s)$$
$$b_i(s) \sim \mathrm{GP}(0, \boldsymbol{\Sigma}_b)$$
$$g(\cdot) \text{ Link function}$$

- No residuals
- How to estimate this model?

GFoSR: NHANES

- Previously we modelled the NHANES activity data as Gaussian
- Instead suppose we're interested in estimating probability of having an activity count above a certain threshold
- Let $Z_{ij}(s) = 1(Y_{ij}(s) \geq 100)$
    - $i = 1, \ldots, N$ indicates participant
    - $j = 1, \ldots, J_i$ denotes day of observation
    - $Y_{ij}(s)$ is the activity count for participant i on day j at minute s
- $\boldsymbol{Z}_{ij}$ is a binary functional outcome

GFoSR: NHANES

- For computational savings
  - Bin data into 20 minute intervals
  - Only consider ages 5-25
  - Sample $N = 200$ participants
- For simplicity
  - Only consider Saturdays (no multilevel structure)

GFoSR: NHANES

- Model:

$$g(E[\mathrm{Z}_i(s)|\mathrm{Age}_i, \mathrm{Male}_i, \boldsymbol{b}_i]) = f_0(s) + f_1(s)\mathrm{Age}_i + f_2(s)\mathrm{Male}_i + b_i(s)$$
$$= f_0(s) + f_1(s)\mathrm{Age}_i + f_2(s)\mathrm{Male}_i + \sum_{k=1}^{K} \xi_{ik}\phi_k(s$$

- We're binning the data so $Z_i(s)$ here is the total active minutes in a particular 20 minute interval
- $g(\cdot)$ is the logit function
- $E[\mathrm{Z}_i(s)] = \Pr(Z_i(s) = k)$ for $k = 0, \ldots, 20$
- Estimation follows [Scheipl et al., 2015]
    - ϕ_k no longer orthogonal
    - ξ_{ik} no longer uncorrelated
    - Imposes constraint $\sum_{i=1}^{N} b_i(s) = 0$ for all s

GFoSR: NHANES Data Preparation

```r
set.seed(9454785)
## read in the data
df <- readr::read_rds(here::here("data","data_processed",
                                 "NHANES_AC_processed.rds"))
## subset to only "good" Saturdays, participants age 5-25
df <- filter(df, DoW %in% "Saturday",
             good_day %in% 1,
             Age <= 25)
## subset to 200 randomly selected individuals
df <- df %>% sample_n(size=200)
## extract activity counts (Y), create binary RV (Z)
Y <- as.matrix(df[,paste0("MIN",1:1440)])
Y[is.na(Y)] <- 0
Z <- apply(Y >= 100, 2, as.numeric)
## bin the data into 20 minute intervals
N    <- nrow(Y)
tlen <- 20
nt   <- ceiling(1440/tlen)
inx_cols <- split(1:1440, rep(1:nt, each=tlen)[1:1440])
Z_bin <- vapply(inx_cols, function(x) rowSums(Z[,x,drop=FALSE]), numeric(N))
## add binned data back into our data frame
df[["Z_bin"]] <- Z_bin
```

GFoSR: NHANES Data Preparation

```
## remove unnecesarry minute columns
df <- dplyr::select(df, Age, SEQN, BMI, Z_bin, Gender)
## create factor variable for ID
df <-
  df %>%
  mutate(ID=factor(SEQN))
## create long format data frame
N <- nrow(df)
sind <- seq(0,1,len=nt)
df_long <-
  data.frame("Z" = as.vector(t(df$Z_bin)),
             "Age" = rep(df$Age, each=nt),
             "Gender" = rep(df$Gender, each=nt),
             "sind" = rep(sind, N),
             "ID"=rep(df$ID, each=nt))
df_long$Male <- as.numeric(df_long$Gender %in% "Male")
df_long$Z_n <- tlen - df_long$Z
```

GFoSR: NHANES Model Estimation

```
fit_marginal <- bam(cbind(Z,Z_n) ~
                    s(sind, bs="cc",k=10)+
                    s(sind,by=Age,bs="cc",k=10) +
                    s(sind,by=Male,bs="cc",k=10),
                  family="binomial", data=df_long,
                  method="fREML", discrete=TRUE)
gfosr_time_st <- Sys.time()
fit_gfosr <- bam(cbind(Z,Z_n) ~
                    s(sind, bs="cc",k=10)+
                    s(sind,by=Age,bs="cc",k=10) +
                    s(sind,by=Male,bs="cc",k=10) +
                    ti(ID, sind, bs=c("re","cr"),mc=c(TRUE,FALSE),k=c(5,5)),
                 family="binomial", data=df_long,
                 method="fREML", chunk.size = 10000, discrete=TRUE)
gfosr_end_st <- Sys.time()
difftime(gfosr_end_st, gfosr_time_st, units="mins")

## Time difference of 1.783896 mins
```

GFoSR: NHANES Model Results

- We can extract coefficients in the usual way
- *mgcv::predict.gam()* with type='terms'
- Returns estimates of
  - $\hat{f}_0(s)$ subject to identifiability constraint (i.e. missing the constant term)
  - $\hat{f}_1(s)\text{Age}_{\text{pred}}$
  - $\hat{f}_2(s)\text{Male}_{\text{pred}}$
  - $\tilde{b}_{i\ \text{pred}}(s)$ (GFoSR fit only)

GFoSR: NHANES Model Fixed Effect Results

```r
## get estimated coefficients
df_pred <- data.frame(sind=sind, Age=1, Male=1, ID=df_long$ID[1])
coefs_marginal <- predict(fit_marginal, newdata=df_pred, type='terms',se.fit=TRUE)
coefs_gfosr    <- predict(fit_gfosr, newdata=df_pred, type='terms',se.fit=TRUE)
```

GFoSR: NHANES Model Fixed Effect Results

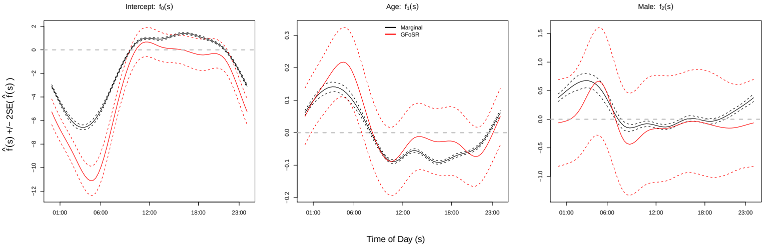
```
str(coefs_marginal)

## List of 2
##  $ fit   : num [1:72, 1:3] -1.31 -1.77 -2.24 -2.69 -3.12 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:72] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:3] "s(sind)" "s(sind):Age" "s(sind):Male"
##  $ se.fit: num [1:72, 1:3] 0.0637 0.0626 0.064 0.0711 0.0834 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:72] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:3] "s(sind)" "s(sind):Age" "s(sind):Male"

str(coefs_gfosr)

## List of 2
##  $ fit   : num [1:72, 1:4] -1.49 -2.03 -2.52 -2.97 -3.4 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:72] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:4] "s(sind)" "s(sind):Age" "s(sind):Male" "ti(ID,sind)"
##  $ se.fit: num [1:72, 1:4] 0.555 0.548 0.537 0.525 0.516 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:72] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:4] "s(sind)" "s(sind):Age" "s(sind):Male" "ti(ID,sind)"
```

GFoSR: NHANES Model Fixed Effect Results

```
head(coefs_gfosr$fit)
```

```
##     s(sind) s(sind):Age s(sind):Male ti(ID,sind)
## 1 -1.489615  0.04987544 -0.064089236    1.226536
## 2 -2.026149  0.06699589 -0.053732784    1.882081
## 3 -2.515904  0.08283845 -0.042926931    2.531829
## 4 -2.970587  0.09763544 -0.029010997    3.169986
## 5 -3.401904  0.11161921 -0.009324302    3.790755
## 6 -3.821564  0.12502207  0.018793834    4.388340
```

GFoSR: NHANES Model Fixed Effect Results



GFoSR: NHANES Model Participant Predictions

- Recall

$$\begin{aligned}\hat{E}[Z_i(s)] &= g^{-1}(g(\hat{E}[Z_i(s)])) \\ &= g^{-1}(\hat{\eta}_i(s))\end{aligned}$$

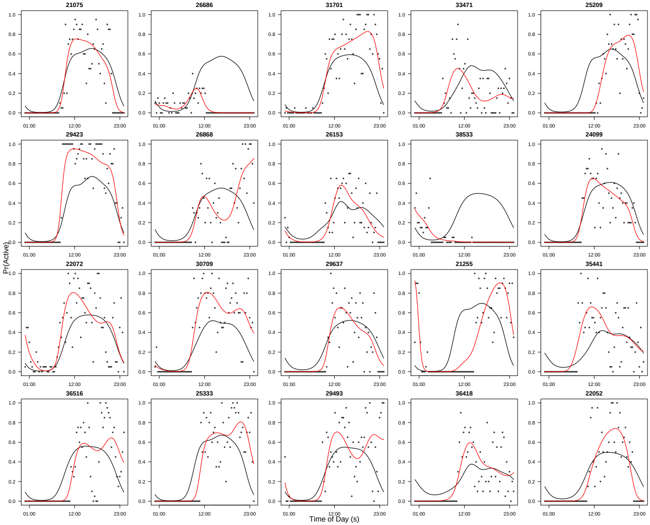
- Inverse logit

$$\begin{aligned}log(\hat{p}_i/(1-\hat{p}_i)) &= \hat{\eta}_i \\ \hat{p}_i &= e^{\hat{\eta}_i}/(1+e^{\hat{\eta}_i}) \\ &= 1/(1+e^{-\hat{\eta}_i})\end{aligned}$$

GFoSR: NHANES Model Participant Predictions

```
## obtain subject predictions using the returned values from mgcv::bam
ests_direct <- fit_gfosr$fitted.values
## obtain subject predictions using predict.bam() with type="terms"
ests_terms  <- predict(fit_gfosr, newdata=df_long, type='terms')
expit       <- function(x) 1/(1+exp(-x))
ests_terms  <- expit(coef(fit_gfosr)[1]+rowSums(ests_terms))
## obtain subject predictions using predict.bam() with type="lpmatrix"
ests_lp     <- predict(fit_gfosr, newdata=df_long, type='lpmatrix')
ests_lp     <- expit(ests_lp %*% coef(fit_gfosr))
```

GFoSR: NHANES Model Participant Predictions



GFoSR: Questions

- What may explain the difference in marginal vs GFoSR estimates? Hint: marginal vs conditional models
- Suppose we are interested in marginal effects. How might we do smoothing parameter selection?
- Is this model estimable using, for example, *lme4::glmer()*? How would you fit this model?

References I

Scheipl, F., Staicu, A.-M., and Greven, S. (2015).
Functional additive mixed models.
Journal of Computational and Graphical Statistics, 24(2):477–501.