BIOS 7720: Applied Functional Data Analysis

Lecture 9: Function on Scalar Regression

Andrew Leroux

April 8, 2021

FoSR

- Thus far we've discussed regression models where you have a function as a predictor
- Now we'll discuss what changes when you have a function as your outcome

FoSR

- Consider our physical activity data

$$\text{LAC}_i(t) = f_0(t) + f_1(t)\text{Age}_i + \epsilon_i(t)$$
$$\epsilon_i(t) \overset{\text{iid}}{\sim} N(0, \sigma_\epsilon^2)$$

- Recall from the lecture 4 in-class exercise that the assumptions of this model were violated (correlation within individuals)
- How to account for this correlation?

FoSR

- Consider our model

$$
\begin{aligned}
\mathrm{LAC}_i(s) &= f_0(s) + f_1(s)\mathrm{Age}_i + b_i(t) + \epsilon_i(s) \\
b_i(s) &\sim \mathrm{GP}(0, \boldsymbol{\Sigma}_b) \\
\epsilon_i(s) &\overset{\mathrm{iid}}{\sim} N(0, \sigma_\epsilon^2)
\end{aligned}
$$

- The assumption on b_i is the same as we used in fPCA

FoSR

$$\text{LAC}_i(s) = f_0(s) + f_1(s)\text{Age}_i + \sum_{k=1}^{K} \xi_{ik}\phi_k(s)\xi_{ik} + \epsilon_i(s)$$

$$b_i(t) \sim \text{GP}(0, \mathbf{\Sigma}_b)$$

$$\epsilon_i(s) \overset{\text{iid}}{\sim} N(0, \sigma_\epsilon^2)$$

- How to estimate ϕ?
- Iterative procedure

FoSR: Simulating Data

```r
set.seed(19840)
# simulation settings
N  <- 200  # number of functions to simulate
ns <- 100   # number of observations per function
sind <- seq(0,1,len=ns) # functional domain of observed functions
K  <- 4     # number of true eigenfunctions
lambda <- 0.5^(0:(K-1))    # true egenfunctions
sig2 <- 2  # error variance
# set up true eigenfunctions
Phi <- sqrt(2)*cbind(sin(2*pi*sind), cos(2*pi*sind),
                     sin(4*pi*sind), cos(4*pi*sind))
# simulate coefficients
# first, simulate standard normals, then multiply by the
# standard deviation to get correct variance
xi_raw <- matrix(rnorm(N*K), N, K)
xi     <- xi_raw %*% diag(sqrt(lambda))
# simulate b_i(s) as \sum_k \xi_ik \phi_k(t)
bi <- xi %*% t(Phi)
```

FoSR: Simulating Data

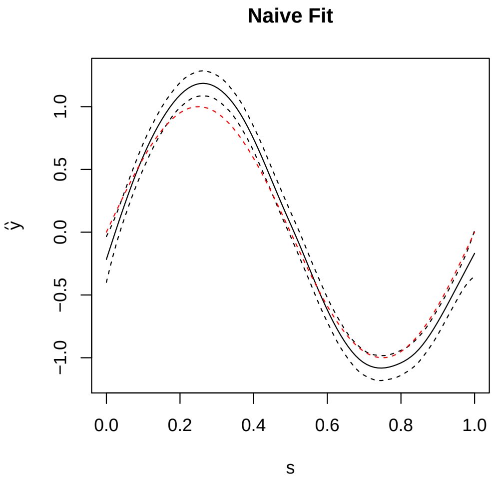
```r
## define f(s)
f <- function(s) sin(2*pi*s)
## get f(s) for all i, s
## fS is an N x ns matrix with rows repeated
fS <- kronecker(matrix(f(sind), 1, ns), matrix(1, N, 1))
x  <- rnorm(N)
## get f(s)*x for each individual
fX <- fS * kronecker(matrix(x, N, 1), matrix(1, 1, ns))
## simulate the outcome
y <- bi + fX + matrix(rnorm(N*ns, sd=2), N, ns)
```

FoSR: Ignore correlation

```
df_fit <-
  data.frame(
    id = factor(rep(1:N,each=ns)),
    y = as.vector(t(y)),
    bi = as.vector(t(bi)),
    x = rep(x, each=ns),
    id = rep(1:N, each=ns),
    sind = rep(sind, N),
    phi1 = rep(Phi[,1], N),
    phi2 = rep(Phi[,1], N),
    phi3 = rep(Phi[,1], N),
    phi4 = rep(Phi[,1], N)
  )
head(df_fit)
```

```
##   id          y        bi         x id.1       sind       phi1       phi2
## 1  1 -0.4223359 0.8028229 -2.192767    1 0.00000000 0.00000000 0.00000000
## 2  1 -2.3254423 1.0359615 -2.192767    1 0.01010101 0.08969497 0.08969497
## 3  1  2.9544945 1.2651368 -2.192767    1 0.02020202 0.17902876 0.17902876
## 4  1  3.2506201 1.4872519 -2.192767    1 0.03030303 0.26764168 0.26764168
## 5  1  3.4911148 1.6992696 -2.192767    1 0.04040404 0.35517689 0.35517689
## 6  1  3.1149040 1.8982600 -2.192767    1 0.05050505 0.44128193 0.44128193
##         phi3       phi4
## 1 0.00000000 0.00000000
## 2 0.08969497 0.08969497
## 3 0.17902876 0.17902876
## 4 0.26764168 0.26764168
```

FoSR: Ignore Correlation



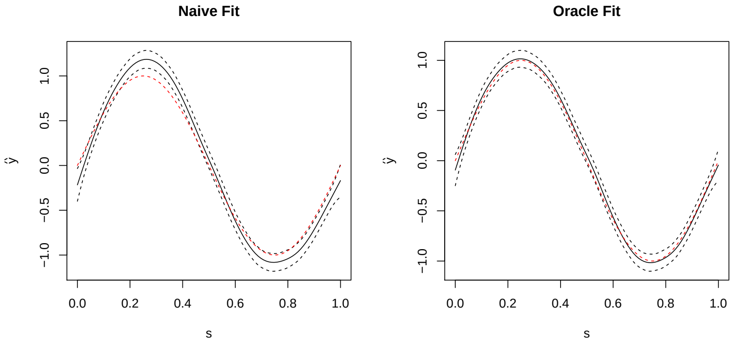
FoSR: Oracle Fit

- What if we “knew" $b_i(s)$ for each person?
- We could treat this as an offset and consider it “fixed"

FoSR: Oracle Fit

```
fit_oracle <- gam(y ~  s(sind, k=20, bs="cr") + s(sind, by=x,bs="cr", k=20),
                  data=df_fit, method="REML", offset=df_fit$bi)
```

FoSR: Oracle Fit



FoSR: Oracle Fit

- In practice this is the same as defining

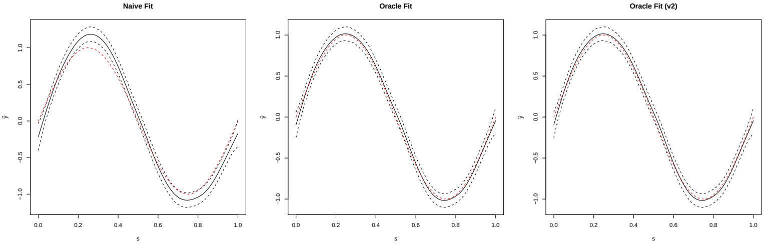
$$y_i^* = y_i - b_i(s)$$

- And regressing y_i^* on x assuming iid normal errors

FoSR: Oracle Fit

```
df_fit$ystar <- df_fit$y - df_fit$bi
fit_oracle2  <- gam(ystar ~  s(sind, k=20, bs="cr") + s(sind, by=x,bs="cr", k=20),
                    data=df_fit, method="REML")
```

FoSR: Oracle Fit



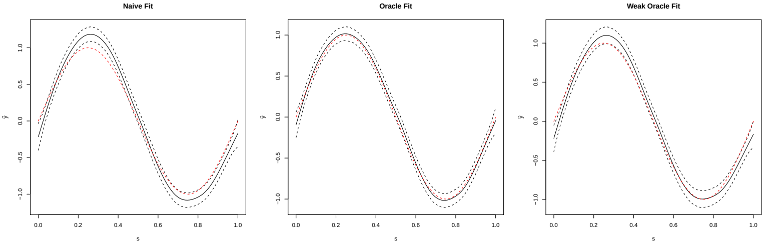
FoSR

- In practice we never observe $b_i(s)$
- What if instead we knew ϕ_k?
- Recall that under the fPCA framework ϕ_k are orthogonal
- And ξ_{ik} are independent

FoSR: "Weak" Oracle

```r
## note that I'm using only the first eigenfunction
## for computational efficiency
fit_oracle_phi <- bam(y ~ s(sind, bs="cr",k=20) + s(sind, bs="cr",k=20, by=x) +
                        s(phi1, by=id,bs="re"),
                      data=df_fit, method="fREML",discrete=TRUE)
```

FoSR: “Weak" Oracle



FoSR

- In practice we also never observe $\phi_k(s)$
- How could we estimate them from the data?
- Iterative procedure:
  1. Estimate the working model under independence
  2. Fit fpca to the residuals
  3. Extract the eigenfunctions
  4. Fit the “weak" oracle model
  5. Repeat 1-4 as necessary