BIOS 7720: Applied Functional Data Analysis

Lecture 13: Multilevel fPCA

Andrew Leroux

May 4, 2021

Logistics

- HW 3 due 5/13
- Final group projects
  - Presentations 5/11 and 5/13
  - Write up due 5/20

Multilevel Data

- Multilevel data arises when data are sampled within units, potentially at multiple levels
- Example: Scools
  - One level
    1. Sample schools within a school district
  - Two levels
    1. Sample schools within a school district
    2. Sample classrooms within schools
  - Three levels
    1. Sample schools within a school district
    2. Sample classrooms within schools
    3. Sample students within classrooms

Multilevel Data

- Multilevel data can be modeled using either fixed or random effects
- One level data (Gaussian)
  - Fixed: ANOVA
  - Random: Random intercept model
- Some considerations
  - Target of inference
    - These specific groups (fixed/random)
    - The population from which these groups are drawn (random)
  - Data density: number of observations within groups
    - Moderate/large (fixed/random)
    - Small (random)

Multilevel Data

- Denote schools as i, classrooms as j, and students as k. Suppose we want to model students' standardized test scores
  - School average $\bar{y}_{i..}$
  - Class average $\bar{y}_{ij.}$
  - Student's individual scores y_{ijk}
- One level: sample schools

$$\bar{y}_{i..} = \beta_0 + \epsilon_i$$
$$\epsilon_i \sim N(0, \sigma_\epsilon^2)$$

- Two levels: sample classrooms within schools

$$\bar{y}_{ij.} = \beta_0 + b_i + \epsilon_{ij}$$
$$b_i \sim N(0, \sigma_b^2), \qquad \epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$$

- Three levels: sample students within classrooms, within schools

$$y_{ijk} = \beta_0 + b_i + \nu_{ij} + \epsilon_{ijk}$$
$$b_i \sim N(0, \sigma_b^2), \qquad \nu_{ij} \sim N(0, \sigma_\nu^2), \qquad \epsilon_{ijk} \sim N(0, \sigma_\epsilon^2)$$

Multilevel Data

- Two levels: sample classrooms within schools

$$\bar{y}_{ij.} = \beta_0 + b_i + \epsilon_{ij}$$
$$b_i \sim N(0, \sigma_b^2), \qquad \epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$$

- Variance in the data

$$\begin{aligned} \mathrm{Var}(\bar{y}_{ij.}) &= \mathrm{Var}(b_i + \epsilon_{ij}) \\ &= \sigma_b^2 + \sigma_\epsilon^2 \end{aligned}$$

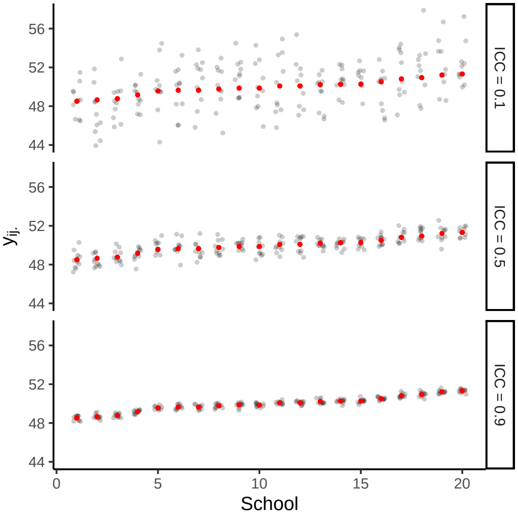
- Intraclass correlation

$$\mathrm{Cov}(y_{ij}, y_{lk}) = \begin{cases} \sigma_b^2 & i = l, j \neq k \\ 0 & i \neq l \end{cases}$$
$$\rho(y_{ij}, y_{ik}) = \sigma_b^2 / (\sigma_b^2 + \sigma_\epsilon^2)$$

Multilevel Data

```
N <- 20 # number of schools
J <- 10 # number of classrooms within schools
sig2_b <- 0.5 # variance of school average
sig2_e <- 0.5 # variance of class average deviation
beta_0 <- 50  # overall mean
set.seed(12010)
# simulate data
bi     <- rnorm(N, mean=0, sd=sqrt(sig2_b))
y      <- beta_0 +
          kronecker(bi, rep(1,J)) +
          rnorm(N*J, mean=0, sd=sqrt(sig2_e))
```

Multilevel Data



Multilevel Data

- Three levels: sample students within classrooms, within schools

$$y_{ijk} = \beta_0 + b_i + \nu_{ij} + \epsilon_{ijk}$$
$$b_i \sim N(0, \sigma_b^2), \qquad \nu_{ij} \sim N(0, \sigma_\nu^2), \qquad \epsilon_{ijk} \sim N(0, \sigma_\epsilon^2)$$

- Variance in the data

$$\begin{aligned} \mathrm{Var}(y_{ijk}) &= \mathrm{Var}(b_i + \nu_{ij} + \epsilon_{ijk}) \\ &= \sigma_b^2 + \sigma_\nu^2 + \sigma_\epsilon^2 \end{aligned}$$

- Intraclass correlation

$$\mathrm{Cov}(y_{ijk}, y_{lmn}) = \begin{cases} \sigma_b^2 + \sigma_\nu^2 & i = l, j = m, k \neq n \\ \sigma_b^2 & i = l, j \neq m \\ 0 & i \neq l \end{cases}$$
$$\rho(y_{ijk}, y_{imn}) = \sigma_b^2 / (\sigma_b^2 + \sigma_\nu^2 + \sigma_\epsilon^2)$$
$$\rho(y_{ijk}, y_{ijn}) = (\sigma_b^2 + \sigma_\nu^2) / (\sigma_b^2 + \sigma_\nu^2 + \sigma_\epsilon^2)$$

Multilevel Modelling in *R*

```
N <- 20 # number of schools
J <- 10 # number of classrooms within schools
K <- 5  # students within classrooms within schools
sig2_b  <- 0.5 # variance of school average
sig2_nu <- 0.5 # variance of class average deviation
sig2_e  <- 0.5 # variance of student deviation
beta_0 <- 50  # overall mean
set.seed(12010)
# simulate data
bi    <- rnorm(N, mean=0, sd=sqrt(sig2_b))
nuij  <- rnorm(N*J, mean=0, sd=sqrt(sig2_nu))
y     <- beta_0 +
         kronecker(bi, rep(1,J*K)) +
         kronecker(nuij, rep(1,K)) +
         rnorm(N*J*K, mean=0, sd=sqrt(sig2_e))
## combine into a data frame
df <- data.frame("y"=y,
                 "student"=rep(1:K, N*J),
                 "class"=rep(rep(1:J, each=K), N),
                 "school"=rep(1:N, each=J*K))
```

Multilevel Modelling in *R*

```
## load the lme4 package for fitting mixed models
library(lme4)
## get the data frame for the level 2 model
## by averaging across students within classes
df_lvl2 <-
  df %>%
  group_by(school, class) %>%
  summarize(ybar_ij = mean(y)) %>%
  ungroup()
fit_lvl2 <- lmer(ybar_ij ~ 1 + (1|school), data=df_lvl2)
```

Multilevel Modelling in *R*

```
## view summary output
summary(fit_lvl2)

## Linear mixed model fit by REML ['lmerMod']
## Formula: ybar_ij ~ 1 + (1 | school)
##    Data: df_lvl2
##
## REML criterion at convergence: 508.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.90875 -0.57535 -0.01734  0.65615  2.22637
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  school   (Intercept) 0.6252   0.7907
##  Residual             0.5792   0.7610
## Number of obs: 200, groups:  school, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  49.9576     0.1848   270.3
```

Multilevel Modelling in *R*

```r
## calculate ICC
performance::icc(fit_lvl2)

## # Intraclass Correlation Coefficient
##
##      Adjusted ICC: 0.519
##   Conditional ICC: 0.519

## manually calculate
(0.6252/(0.6252 + 0.5792))

## [1] 0.5190966
```

Multilevel Modelling in *R*

```
fit_lvl3 <- lmer(y ~ 1 + (1|school) + (1|school:class), data=df)
summary(fit_lvl3)

## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ 1 + (1 | school) + (1 | school:class)
##    Data: df
##
## REML criterion at convergence: 2547.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6600 -0.6609 -0.0145  0.6426  3.1122
##
## Random effects:
##  Groups       Name        Variance Std.Dev.
##  school:class (Intercept) 0.4790   0.6921
##  school       (Intercept) 0.6252   0.7907
##  Residual                 0.5009   0.7078
## Number of obs: 1000, groups:  school:class, 200; school, 20
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  49.9576     0.1848   270.3
```

Multilevel Modelling in *R*

```
## ICC: school
performance::icc(fit_lvl3, by_group=TRUE)

## # ICC by Group
##
## Group        |   ICC
## --------------------
## school:class | 0.298
## school       | 0.389

(ICC_school <- 0.6252/(0.6252 + 0.4790 + 0.5009))

## [1] 0.3895084

## ICC: school + class
performance::icc(fit_lvl3)

## # Intraclass Correlation Coefficient
##
##      Adjusted ICC: 0.688
##   Conditional ICC: 0.688

(ICC_school_class <- (0.6252 + 0.4790)/(0.6252 + 0.4790 + 0.5009))

## [1] 0.6879322
```

Multilevel Functional Data

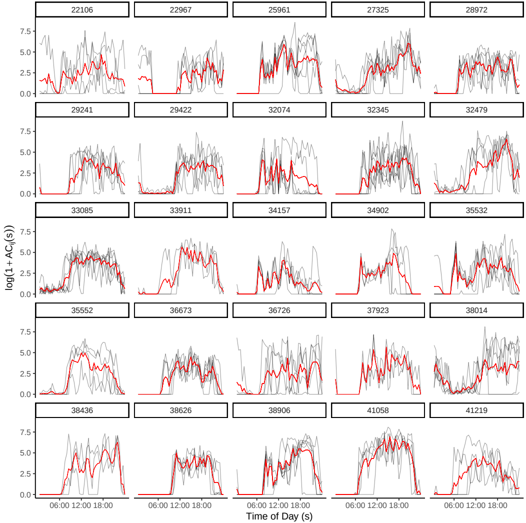
- Thus far in the course
  - We've assumed that each "function" is independent
  - "Two-level" data (repeated observations within a single unit)
- What if there is additional clustering the data? (e.g. multiple functions observed on the same "unit")
- Example: NHANES physical activity data
  - Each "day" (12AM-12AM) is a (realization of) a function
  - Multiple days within observed participants
- Simplest case is when the observations within participants are exchangeable (the case we'll consider here)

Multilevel Functional Data

```r
## change data_path to whereever your NHANES file is located
data_path <- here::here("data","data_processed","NHANES_AC_processed.rds")
df        <- readr::read_rds(data_path)
df_sub    <-
  df %>%
  filter(n_good_days >= 3, good_day == 1,  Age <= 25)
uid <- unique(df_sub$SEQN)

## extract the PA data
lX <- log(1+as.matrix(df_sub[,paste0("MIN",1:1440)]))
lX[is.na(lX)] <- 0
N  <- nrow(lX)
## bin the data into 30 minute intervals
tlen <- 30
nt   <- ceiling(1440/tlen)
inx_cols <- split(1:1440, rep(1:nt, each=tlen)[1:1440])
lX_bin   <- vapply(inx_cols, function(x) rowMeans(lX[,x], na.rm=TRUE), numeric(N))
colnames(lX_bin) <- paste0("epoch_",1:nt)
```

Multilevel Functional Data



Multilevel Functional Data

- Substantial variation in participant-averages and daily deviations within participants
- The "classic" fPCA model would ignore the clustering and model

$$
\begin{aligned}
Y_{ij}(s) &= \mu_0(s) + b_{ij}(s) + \epsilon_{ij}(s) \\
b_{ij} &\overset{\text{iid}}{\sim} GP(0, \boldsymbol{\Sigma}_{\boldsymbol{b}}) \\
\epsilon_{ij}(s) &\overset{\text{iid}}{\sim} N(0, \sigma_\epsilon^2)
\end{aligned}
$$

- May be OK depending on the goal of the analysis, but can we account for the clustering of days within participants?

Multilevel fPCA

- Multilevel fPCA [Di et al., 2009]

$$\begin{aligned} Y_{ij}(s) &= \mu_0(s) + b_i(s) + \nu_{ij}(s) + \epsilon_{ij}(s) \\ b_i &\overset{\text{iid}}{\sim} GP(0, \mathbf{\Sigma_b}) \\ \nu_{ij} &\overset{\text{iid}}{\sim} GP(0, \mathbf{\Sigma_\nu}) \\ \epsilon_{ij}(s) &\overset{\text{iid}}{\sim} N(0, \sigma_\epsilon^2) \end{aligned}$$

- b_i is the participant deviation from the population average
- ν_{ij} is the day deviation from a participants' average $(\mu_0(s) + b_i(s))$

Multilevel fPCA

- mfPCA idea:
  - Expand b_i and ν_{ij} using orthogonal bases
  - Estimate the population mean function μ_0
  - Decompose residual variance into between and within
  - Eigendecomposition on the estimated covariance(s)
  - Estimate scores
- Implemented in *refund::mfpca.sc()*
- Currently fairly slow, but faster version in the works

Multilevel fPCA

]

- Model

$$Y_{ij}(s) = \mu_0(s) + b_i(s) + \nu_{ij}(s) + \epsilon_{ij}(s)$$

- Covariance(s)

$$K_T(s,u) = \mathrm{Cov}(Y_{ij}(s), Y_{ij}(u))$$
$$K_B(s,u) = \mathrm{Cov}(Y_{ij}(s), Y_{ik}(u))$$

Multilevel Functional Data

```
set.seed(10)
## remove unnecessary columns
df_sub <-
  df_sub %>%
  dplyr::select(-one_of(paste0("MIN",1:1440)))
## add in the binned data to our data frame
df_sub[["lX_bin"]] <- lX_bin
## subset to only 100 participants
nsamp <- 100
uid_samp <- sample(uid, size=nsamp, replace=FALSE)
df_mfpca <-
  filter(df_sub, SEQN %in% uid_samp) %>%
  group_by(SEQN) %>%
  mutate(J = 1:n()) %>%
  ungroup() %>%
  arrange(SEQN, J)
```

Multilevel Functional Data

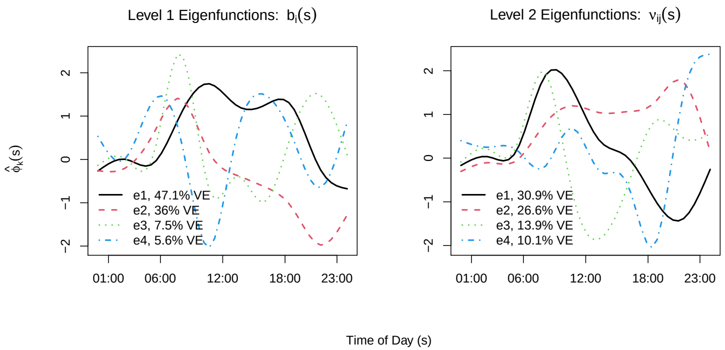
```r
library("refund")
mfpca_fit <- mfpca.sc(Y=df_mfpca$lX_bin,
                      id=df_mfpca$SEQN,
                      visit=df_mfpca$J,
                      pve=0.95)
```

Multilevel Functional Data

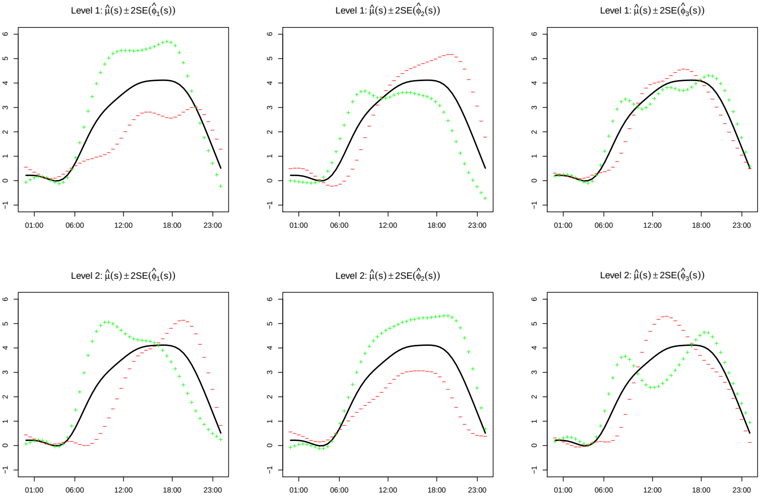
```
str(mfpca_fit)
```

```
## List of 10
##  $ Yhat        : num [1:558, 1:48] 0.533 0.184 -0.225 0.11 -0.31 ...
##  $ Yhat.subject: num [1:558, 1:48] 0.0687 0.0687 0.0687 0.0687 0.0687 ...
##  $ Y.df        :'data.frame': 558 obs. of  3 variables:
##   ..$ id   : int [1:558] 21130 21130 21130 21130 21130 21130 21130 21529 21529 21529 ...
##   ..$ visit: int [1:558] 1 2 3 4 5 6 7 1 2 3 ...
##   ..$ Y    : num [1:558, 1:48] 0 0 0.346 0 0 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr [1:48] "epoch_1" "epoch_2" "epoch_3" "epoch_4" ...
##  $ scores      :List of 2
##   ..$ level1: num [1:100, 1:5] 0.4834 -0.4831 0.6288 -0.0264 -0.4184 ...
##   ..$ level2: num [1:558, 1:7] -0.69 0.663 -0.245 0.562 0.322 ...
##  $ mu          : num [1:48(1d)] 0.218 0.219 0.211 0.187 0.143 ...
##   ..- attr(*, "dimnames")=List of 1
##   .. ..$ : chr [1:48] "1" "2" "3" "4" ...
##  $ efunctions  :List of 2
##   ..$ level1: num [1:48, 1:5] -0.267228 -0.180634 -0.102018 -0.03936 -0.000639 ...
##   ..$ level2: num [1:48, 1:7] -0.16728 -0.0999 -0.03928 0.00781 0.03463 ...
##  $ evalues     :List of 2
##   ..$ level1: num [1:5] 0.313 0.2393 0.0496 0.0375 0.0251
##   ..$ level2: num [1:7] 0.3014 0.2587 0.1357 0.0986 0.0911 ...
##  $ npc         :List of 2
##   ..$ level1: int 5
##   ..$ level2: int 7
##  $ sigma2      : num 1.68
##  $ eta         : num [1:48, 1:7] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "class")= chr "mfpca"
```

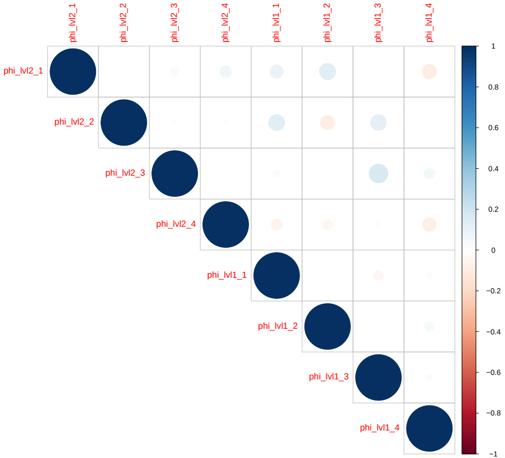
Multilevel Functional Data



Multilevel Functional Data



Multilevel Functional Data



Next Class

- Details on estimation
- Obtaining subject- and day-level predictions (estimating scores)
- Incorporation in functional regression models

References I

Di, C., Crainiceanu, C. M., Caffo, B. S., and Punjabi, N. M. (2009).
Multilevel functional principial component analysis.
Annals of Applied Statistics, 3(1):458–488.