R (LCMM.R)

library(tidyverse)

library(lcmm)

library(ggplot2)

library(lubridate)

# library(gridExtra)

# library(knitr)

# library(corrplot)

install.packages('lcmm', repos='http://rcran.nhisin.or.kr:8080/cran', type='win.binary')

visits\_full <- read\_parquet("hospital\_visits.parquet")

visits <- readRDS("hospital\_visits\_sampled\_5k.rds")

visits <- visits %>%

rename(

patient\_id = INDI\_DSCM\_NO,

year\_month = YEAR\_MONTH,

num\_visits = VISIT\_COUNT

) %>%

mutate(date\_ymd = ymd(paste0(year\_month, "01")))

temp <- visits %>%

group\_by(patient\_id) %>%

arrange(date\_ymd) %>%

slice(1) %>%

select(patient\_id, first\_date = date\_ymd) %>%

distinct()

quarterly\_visits <- visits %>%

left\_join(temp, by = "patient\_id") %>%

arrange(patient\_id, date\_ymd) %>%

mutate(

months\_since\_first = interval(first\_date, date\_ymd) %/% months(1),

quarter = months\_since\_first %/% 3

) %>%

group\_by(patient\_id, quarter) %>%

summarise(

hospital\_visits = sum(num\_visits)

) %>%

group\_by(patient\_id) %>% # adding zeros to missing quarters

complete(

quarter = seq(0, max(quarter)),

fill = list(hospital\_visits = 0)

) %>%

ungroup()

# any missing?

missing\_quarters\_check <- quarterly\_visits %>%

group\_by(patient\_id) %>%

summarise(

min\_q = min(quarter),

max\_q = max(quarter),

actual\_q = n()

) %>%

mutate(

expected\_q = max\_q - min\_q + 1,

missing\_q = expected\_q - actual\_q

) %>%

filter(missing\_q > 0)

check\_patients <- as.numeric(c('257383', '293871','410167','481267','503188','520912'))

check <- quarterly\_visits %>%

semi\_join(tibble(patient\_id = check\_patients), by = "patient\_id")

# TODO: More elaborate consideration for nonrandom censorship

table(quarterly\_visits$quarter)

quarterly\_visits\_filtered <- quarterly\_visits %>%

group\_by(patient\_id) %>%

mutate(max\_q = max(quarter)) %>%

ungroup() %>%

# TODO: More elaborate consideration for nonrandom censorship

filter(max\_q >= 20) %>% # only keeping those with 20+ quarters and only up to 20 quarters

filter(quarter <= 20) %>%

select(-max\_q)

length(unique(quarterly\_visits$patient\_id))

length(unique(quarterly\_visits\_filtered$patient\_id))

sum(quarterly\_visits\_filtered$hospital\_visits == 0)

# tblndb

patient\_demo <- readRDS("tblndb\_sampled\_5k.rds")

patient\_demographics <- patient\_demo %>%

rename(

patient\_id = INDI\_DSCM\_NO

) %>%

select(patient\_id, EPISODE, SEX\_TYPE, AGE\_NHIS, RSLT\_FN, SIGUNGU, SIDO, GAIBJA\_TYPE, INC5, TBCD\_KCDA, xray1, smear1, cul1, xpert1)

names(patient\_demographics)

### baseline data ###

lcmm\_data <- quarterly\_visits\_filtered %>% # should be quarterly\_visits (only change, look at line 65-75)

left\_join(patient\_demographics, by = "patient\_id") %>%

arrange(patient\_id, quarter) %>%

mutate(

patient\_numeric = as.numeric(factor(patient\_id)),

time = quarter

) %>%

filter(!is.na(hospital\_visits))

nrow(lcmm\_data)

length(unique(quarterly\_visits$patient\_id))

length(unique(lcmm\_data$patient\_id))

length(unique(lcmm\_data$time))

table(lcmm\_data$time)

### Visualize the mess ###

smp <- sample(unique(lcmm\_data$patient\_id), 100)

spaghetti\_data <- lcmm\_data %>%

semi\_join(tibble(patient\_id = smp), by = "patient\_id")

ggplot(spaghetti\_data, aes(x = time, y = hospital\_visits, group = patient\_id)) +

geom\_line(alpha = 0.3, color = "steelblue") +

geom\_smooth(aes(group = 1), method = "loess", se = TRUE, color = "red") +

labs(

title = "Individual Trajectory PAtterns, Sampled (n = 100)",

x = "Quarters Since Treatment",

y = "Number of Hospital Visits"

) +

theme\_minimal()

ggplot(lcmm\_data, aes(x = factor(time), y = hospital\_visits)) +

geom\_boxplot(alpha = 0.7, fill = "lightblue") +

labs(

title = "Visit Distirbution by Time Point",

x = "Quarters Since Treatment",

y = "Number of Hospital Visits"

) +

theme\_minimal() +

theme(axis.text.x = element\_text(angle = 45, hjust = 1))

### 1 class model for initialization ###

lcmm\_data <- as.data.frame(lcmm\_data) # see numeric\_drama.R

model\_1class <- lcmm(

hospital\_visits ~ time + I(time^2),

subject = 'patient\_numeric',

ng = 1,

data = lcmm\_data,

link = "linear",

verbose = FALSE

)

model\_1class$conv == 1

model\_1class$loglik

model\_1class$AIC

model\_1class$BIC

### 3 class model ###

model\_3class <- lcmm(

hospital\_visits ~ time + I(time^2),

mixture = ~ time + I(time^2),

subject = "patient\_numeric",

ng = 3,

data = lcmm\_data,

link = "linear",

verbose = TRUE,

maxiter = 600,

B = model\_1class

)

model\_3class$conv == 1

model\_3class$loglik

model\_3class$AIC

model\_3class$BIC

round(model\_1class$BIC - model\_3class$BIC, 1)

summary(model\_3class)

### Model examination ###

posterior\_probs <- model\_3class$pprob

class\_assignments <- posterior\_probs[, 1:2]

colnames(class\_assignments) <- c("patient\_numeric", "predicted\_class")

table(class\_assignments$predicted\_class)

round(prop.table(table(class\_assignments$predicted\_class)) \* 100, 1)

posterior\_probs$max\_prb <- apply(posterior\_probs[, c("prob1", "prob2", "prob3")], 1, max, na.rm = TRUE)

round(mean(posterior\_probs$max\_prb, na.rm = TRUE), 3)

round(median(posterior\_probs$max\_prb, na.rm = TRUE), 3)

round(mean(posterior\_probs$max\_prb > 0.7, na.rm = TRUE) \* 100, 1)

round(mean(posterior\_probs$max\_prb > 0.895, na.rm = TRUE) \* 100, 1)

trajectory\_data <- lcmm\_data %>%

left\_join(

data.frame(

patient\_numeric = class\_assignments[, "patient\_numeric"],

predicted\_class = as.factor(class\_assignments[, "predicted\_class"])

),

by = "patient\_numeric"

) %>%

filter(!is.na(predicted\_class))

mean\_trajectories <- trajectory\_data %>%

group\_by(predicted\_class, time) %>%

do(data.frame(n = nrow(.), mean\_visits = mean(.$hospital\_visits, na.rm=TRUE)))

# dont run this

mean\_trajectories <- trajectory\_data %>%

group\_by(predicted\_class, time) %>%

dplyr::summarise(

n = n(),

mean\_visits = mean(hospital\_visits, na.rm = TRUE),

se\_visits = sd(hospital\_visits, na.rm = TRUE) / sqrt(n()),

lower\_ci = mean\_visits - 1.96 \* se\_visits,

upper\_ci = mean\_visits + 1.96 \* se\_visits,

.groups = "drop"

) %>%

ungroup()

sample\_by\_class <- trajectory\_data %>%

group\_by(predicted\_class) %>%

slice(sample(n(), min(30, n()))) %>%

ungroup() %>%

pull(patient\_id)

indi\_patient\_data <- trajectory\_data %>%

filter(patient\_id %in% sample\_by\_class)

# dont run this

sample\_by\_class <- trajectory\_data %>%

group\_by(predicted\_class) %>%

slice\_sample(n = 30) %>%

pull(patient\_id)

ggplot() +

geom\_line(

data = indi\_patient\_data,

aes(x= time, y = hospital\_visits, group = patient\_id, color = predicted\_class),

alpha = 0.3, linewidth = 0.3

) +

# geom\_ribbon(

# data = mean\_trajectories,

# aes(x = time, y = mean\_visits, ymin = lower\_ci, ymax = upper\_ci, fill = predicted\_class),

# alpha = 0.4

# ) +

geom\_line(

data = mean\_trajectories,

aes(x = time, y = mean\_visits, color = predicted\_class),

linewidth = 1.5

) +

scale\_color\_brewer(palette = "Set1", name = "Trajectory Class") +

scale\_fill\_brewer(palette = "Set1", name = "Trajectory Class") +

labs(

title = "Hospital Visit Trajectories by Latent Class (LCMM)",

x = "Quarters since treatment",

y = "Number of Hospital Visits"

) +

theme\_minimal() +

theme(

legend.position = "bottom",

plot.title = element\_text(size = 14, face = "bold"),

axis.title = element\_text(size = 12)

)

### 2:6 class verification ###

# fit multiple models

fit\_lcmm\_models <- function(data, k\_range = 2:6) {

models <- list()

model\_fit <- tibble()

for(k in k\_range) {

cat("Fitting LCMM", k, "group model... \n")

tryCatch({

fit <- lcmm(

hospital\_visits ~ poly(time, 2),

mixture = ~ poly(time, 2),

subject = "patient\_numeric",

ng = k,

data = data,

link = "linear",

verbose = FALSE,

maxiter = 2000,

B = model\_1class

)

models[[paste0("k", k)]] <- fit

model\_fit <- bind\_rows(model\_fit, tibble(

k = k,

loglik = fit$loglik,

aic = fit$AIC,

bic = fit$BIC,

converged = fit$conv

))

}, error = function(e) {

cat("Error fitting", k, '"group model:', e$message, "\n")

})

}

return(list(models, fit\_stats = model\_fit))

}

lcmm\_results <- fit\_lcmm\_models(lcmm\_data, k\_range = 2:6) # even k = 5 might be too much...

# saveRDS(lcmm\_results, file = "my\_result.rds")

lcmm\_results <- readRDS("my\_result.rds")

lcmm\_results[[1]]$k3

lcmm\_results[[2]]

print(lcmm\_results)

if(nrow(lcmm\_results$fit\_stats) > 0){

best\_k\_lcmm <- lcmm\_results$fit\_stats$k[which.min(lcmm\_results$fit\_stats$bic)]

cat("\nBest model based on BIC: ", best\_k\_lcmm, "groups\n")

best\_lcmm <- lcmm\_results$models[[paste0("k", best\_k\_lcmm)]]

lcmm\_groups <- best\_lcmm$pprob[, 1:2]

colnames(lcmm\_groups) <- c("patient\_numeric", "predicted\_group")

cat("Group sizes (LCMM):\n")

print(table(lcmm\_groups[, 2]))

}

lcmm\_results[[1]]$k6$BIC

lcmm\_results[[1]]$k5$BIC

SAS (data extraction)

LIBNAME COHORT '/userdata06/room241/data\_source/COHORT';

/\* LIBNAME CLAIM '/userdata06/room241/data\_source/user\_data/20250502';\*/

LIBNAME HOME '/userdata06/room241/data\_source/youb';

/\*

proc print data=COHORT.tblndb (obs=10); title "structure of tblndb"; run;

\*/

/\* lookup table with treatment end dates \*/

**data** HOME.patient\_treatment\_lookup;

SET COHORT.tblndb (keep=INDI\_DSCM\_NO TRTFN\_DATE);

/\* extract year and month from TRTFN\_DATE \*/

trt\_year = int(TRTFN\_DATE / **10000**); /\* input(substr(TRTFN\_DATE, 1, 4), 4.); \*/

trt\_month = int((TRTFN\_DATE - trt\_year \* **10000**) / **100**); /\* input(substr(TRTFN\_DATE, 6, 2), 2.); \*/

put "looking at" trt\_month trt\_year ;

if trt\_month = **12** then do;

start\_count\_year = trt\_year + **1**;

start\_count\_month = **1**;

end;

else do;

start\_count\_year = trt\_year;

start\_count\_month = trt\_month + **1**;

end;

length TRT\_END\_YEARMONTH $**6**;

TRT\_END\_YEARMONTH = put(start\_count\_year, **4.**) || put(start\_count\_month, z2.);

keep INDI\_DSCM\_NO TRT\_END\_YEARMONTH;

**run**;

**proc** **freq** data=home.t20\_201001;

tables FORM\_CD / missing;

**run**;

**data** HOME.hospital\_visits\_combined /view=HOME.hospital\_visits\_combined;

if \_n\_ = **1** then do;

declare hash trt\_lookup(dataset:'HOME.patient\_treatment\_lookup', hashexp:**18**);

trt\_lookup.definekey('INDI\_DSCM\_NO');

trt\_lookup.definedata('TRT\_END\_YEARMONTH');

trt\_lookup.definedone();

end;

set

**%macro** create\_table\_list;

%local year month formatted\_month year\_month table\_name;

%do year = **2010** %to **2017**;

%do month = **1** %to **12**;

%let formatted\_month = %sysfunc(putn(&month, z2.));

%let year\_month = &year.&formatted\_month;

%let table\_name = HOME.t20\_&year\_month;

%if %sysfunc(exist(&table\_name)) %then %do;

&table\_name (in=in\_&year\_month keep=INDI\_DSCM\_NO FORM\_CD CMN\_KEY where=(FORM\_CD in ('03', '08')))

%end;

%end;

%end;

**%mend**;

%***create\_table\_list***;

;

length YEAR\_MONTH $**6** TRT\_END\_YEARMONTH $**6**;

**%macro** ***assign\_year\_month***;

%do year = **2010** %to **2017**;

%do month = **1** %to **12**;

%let formatted\_month = %sysfunc(putn(&month, z2.));

%let year\_month = &year.&formatted\_month;

if in\_&year\_month then YEAR\_MONTH = "&year\_month";

%end;

%end;

**%mend**;

%***assign\_year\_month***;

**run**;

/\* lookup treatment start count date \*/

if trt\_lookup.find() = **0** then do;

if YEAR\_MONTH >= TRT\_END\_YEARMONTH then output;

endrsubmit;

keep INDI\_DSCM\_NO YEAR\_MONTH CMN\_KEY;

**run**;

**proc** **sort** data=HOME.hospital\_visits\_combined

out=HOME.hospital\_visits\_sorted

nodupkey;

by CMN\_KEY;

**run**;

/\*

proc summary data=HOME.hospital\_visits\_sorted nway;

class INDI\_DSCM\_NO YEAR\_MONTH;

output out=HOME.hospital\_visits (drop=\_type\_) n=VISIT\_COUNT;

run;

\*/

**proc** **sql**;

create table HOME.hospital\_visits AS

select

INDI\_DSCM\_NO,

YEAR\_MONTH,

count(\*) AS VISIT\_COUNT

FROM HOME.hospital\_visits\_sorted

GROUP BY INDI\_DSCM\_NO, YEAR\_MONTH

ORDER BY INDI\_DSCM\_NO, YEAR\_MONTH;

**QUIT**;

**proc** **datasets** library=HOME nolist;

delete hospital\_visits\_sorted;

**quit**;

**proc** **datasets** library=HOME nolist;

%do yeaer=**2010** %to **2017**;

%do month = **1** %to **12**;

%let formatted\_month = %sysfunc(putn(&month, z2.));

%let year\_month = &year.&formatted\_month;

%let table\_name = T20\_&year\_month;

%if %sysfunc(exist(HOME.&table\_name)) %then %do;

delete &table\_name;

%put NOTE: Deleted Home.&tablee\_name;

%end;

%end;

%end;

**quit**;

**proc** **sql**;

create table HOME.hospital\_visits\_summary as

select

count(distinct INDI\_DSCM\_NO) as unqiue\_patients format=comma12.,

count(distinct YEAR\_MONTH) as months\_with\_data,

count(\*) as patient\_month\_records format=comma12.,

sum(VISIT\_COUNT) as total\_valid\_visits format=comma12.,

min(VISIT\_COUNT) as min\_monthly\_visits,

max(VISIT\_COUNT) as max\_monthly\_visits,

mean(VISIT\_COUNT) as avg\_monthly\_visits format=**8.2**

from HOME.hospital\_visits;

**QUIT**;

Basic agg

LIBNAME COHORT '/userdata06/room241/data\_source/COHORT';

LIBNAME CLAIM '/userdata06/room241/data\_source/user\_data/20250502';

LIBNAME HOME '/userdata06/room241/data\_source/youb';

**%macro** generate\_table\_list();

%local i year month table\_list;

%let table\_list = ;

%do year = **2010** %to **2011**;

%do month = **1** %to **2**;

%let formatted\_month = %sysfunc(putn(&month, z2.));

%let table\_name = CLAIM.T20\_&year.&formatted\_month;

%if %sysfunc(exist(&table\_name)) %then %do;

%if &table\_list = %then %let table\_list = &table\_name;

%else %let table\_list = &table\_list &table\_name;

%end;

%else %do;

%put WARNING: table dne;

%end;

%end;

%end;

&table\_list

**%mend** generate\_table\_list;

**data** HOME.hospital\_visits;

set %***generate\_table\_list***();

where FORM\_CD in ('03', '08');

keep INDI\_DSCM\_NO FORM\_CD CMN\_KEY;

**run**;

**proc** **sort** DATA=HOME.hospital\_visits;

by CMN\_KEY;

**run**;

**data** HOME.hospital\_visits;

SET HOME.hospital\_visits;

by CMN\_KEY;

**run**;

**proc** **SQL**;

select count(\*) as final\_record\_count

from HOME.hospital\_visits;

**quit**;