A Quick Latent Class Analysis (LCA) from Start to Finish in `MplusAutomation`

IMMERSE Video Series Funded by IES

Adam Garber

February 07, 2023

What is included in this video tutorial?

> A demonstration of the speed at which an LCA analysis can be estimated and summarized using the `Tidy MplusAutomation` method.

Tutorial Outline

0. Download scripts & data from Github repository
1. Introduction to data example & LCA indicator variables
2. Load packages
3. Read in data to R
4. Enumeration: Estimate LCA models with 1-6 classes
5. Create model fit table
6. Plot information criteria (elbow plot)
7. Compare conditional item probability plots
8. Plot final model in publication format (e.g., Class-3 model)

---

0. Github repository (everything you need to replicate analysis):

Link: https://github.com/immerse-ucsb/quick-lca-mplusauto

1. Data Source: Civil Rights Data Collection (CRDC)

> The CRDC is a federally mandated school and district level data collection effort that occurs every other year. This public data is currently available for selected variables across 4 years (2011, 2013, 2015, 2017) and all US states. In the following tutorial six focal variables are utilized as indicators of the latent class model; three variables which report on harassment/bullying in schools based on disability, race, or sex, and three variables on full-time equivalent school staff employees (counselor, psychologist, law enforcement). For this example, we utilize a sample of schools from the state of Arizona reported in 2017.

Information about CRCD: https://www2.ed.gov/about/offices/list/ocr/data.html

Data access (`R`): https://github.com/UrbanInstitute/education-data-package-r

Latent Class Indicator Variables

`report_dis` = Number of students harassed or bullied on the basis of disability

`report_race` = Number of students harassed or bullied on the basis of race, color, national origin

`report_sex` = Number of students harassed or bullied on the basis of sex

`counselors_fte` = Number of full time equivalent counselors hired as school staff

`psych_fte` = Number of full time equivalent psychologists hired as school staff

`law_fte` = Number of full time equivalent law enforcement officers hired as school staff

2. Load packages

```r
library(MplusAutomation); library(glue) # estimation
library(tidyverse); library(here); # tidyness
library(gt); library(reshape2); library(cowplot) # tables & figures
```

3. Read in CSV data file from the `data` subfolder

```r
bully_data <- read_csv(here("data", "crdc_lca_data.csv"))
```

---

4. Enumeration

```r
lca_k1_6  <- lapply(1:6, function(k) {

  lca_enum  <- mplusObject(

    TITLE = glue("Class {k}"),

    VARIABLE = glue(
    "categorical = report_dis report_race report_sex counselors_fte psych_fte law_fte;
     usevar = report_dis report_race report_sex counselors_fte psych_fte law_fte;
     classes = c({k}); "),

  ANALYSIS =
   "estimator = mlr;
    type = mixture;
    starts = 500 100;
    processors = 10;",

  OUTPUT = "tech11 tech14;",

  PLOT =
    "type = plot3;
     series = report_dis report_race report_sex counselors_fte psych_fte law_fte(*);",

  usevariables = colnames(bully_data),
  rdata = bully_data)

lca_enum_fit <- mplusModeler(lca_enum,
                            dataout=glue(here("mplus_lca", "lca.dat")),
                            modelout=glue(here("mplus_lca", "c{k}_lca.inp")) ,
                            check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

Always check your model!

- In the RStudio window pane on the `bottom-right`under the `files` tab click on the `mplus_lca` folder
- Click on one of the Mplus output files (`.out`) to check if the model estimated or if there are any error messages

---

5. Generate Model Fit Summary Table

- This syntax can be used to compare model fit from the series of LCA models generated during enumeration
- The code produces a table that is approximately in APA format.

Read in model fit statistics using `readModels()` and `mixtureSummaryTable()` functions

```
output_lca <- readModels(here("mplus_lca"), quiet = TRUE)

enum_summary <- LatexSummaryTable(output_lca,
                keepCols=c("Title", "Parameters", "LL", "BIC", "aBIC",
                           "BLRT_PValue", "T11_VLMR_PValue","Observations"),
                sortBy = "Title")
```

Calculate relevant fit indices for summary table

```
allFit <- enum_summary %>%
  mutate(aBIC = -2*LL+Parameters*log((Observations+2)/24)) %>%
  mutate(CIAC = -2*LL+Parameters*(log(Observations)+1)) %>%
  mutate(AWE = -2*LL+2*Parameters*(log(Observations)+1.5)) %>%
  mutate(SIC = -.5*BIC) %>%
  mutate(expSIC = exp(SIC - max(SIC))) %>%
  mutate(BF = exp(SIC-lead(SIC))) %>%
  mutate(cmPk = expSIC/sum(expSIC)) %>%
  dplyr::select(1:5,9:10,6:7,13,14) %>%
  arrange(Parameters)
```

Generate the fit summary table

```
allFit %>%
  mutate(Title = str_remove(Title, " LCA Enumeration ")) %>%
  gt() %>%
  tab_header(
    title = md("**Model Fit Summary Table**"), subtitle = md(" ")) %>%
  cols_label(
    Title = "Classes",
    Parameters = md("Par"),
    LL = md("*LL*"),
    T11_VLMR_PValue = "VLMR",
```

```
    BLRT_PValue = "BLRT",
    BF = md("BF"),
    cmPk = md("*cmPk*")) %>%
  tab_footnote(
    footnote = md(
    "*Note.* Par = parameters; *LL* = log likelihood;
      BIC = bayesian information criterion;
      aBIC = sample size adjusted BIC; CAIC = consistent Akaike information criterion;
      AWE = approximate weight of evidence criterion;
      BLRT = bootstrapped likelihood ratio test p-value;
      VLMR = Vuong-Lo-Mendell-Rubin adjusted likelihood ratio test p-value;
      cmPk = approximate correct model probability."),
    locations = cells_title()) %>%
  tab_options(column_labels.font.weight = "bold") %>%
  fmt_number(10,decimals = 2,
             drop_trailing_zeros=TRUE,
             suffixing = TRUE) %>%
  fmt_number(c(3:9,11),
             decimals = 0) %>%
  fmt_missing(1:11,
              missing_text = "--") %>%
  fmt(c(8:9,11),
    fns = function(x)
    ifelse(x<0.001, "<.001",
           scales::number(x, accuracy = 0.01))) %>%
  fmt(10, fns = function(x)
    ifelse(x>100, ">100",
           scales::number(x, accuracy = .1)))
```

---

6. Plot Information Criteria

```
allFit %>% dplyr::select(2:7) %>%
  rowid_to_column() %>%
  pivot_longer(`BIC`:`AWE`,
    names_to = "Index",
    values_to = "ic_value") %>%
  mutate(Index = factor(Index,
    levels = c("AWE","CIAC","BIC","aBIC"))) %>%
  ggplot(aes(x = rowid, y = ic_value,
    color = Index, shape = Index,
    group = Index, lty = Index)) +
  geom_point(size = 2.0) + geom_line(size = .8) +
  scale_x_continuous(breaks = 1:6) +
  labs(x = "Number of Classes", y = "Information Criteria Value") +
  theme_cowplot() + theme(legend.title = element_blank(), legend.position = "top")
```

```r
ggsave(here("figures","fit_criteria_plot.png"),
       dpi=300, height=4, width=6, units="in")
```

---

7. Compare Conditional Item Probability Plots

```r
model_results <- data.frame()
for (i in 1:length(output_lca)) {
  temp <- data.frame(unclass(output_lca[[i]]$parameters$probability.scale)) %>%
    mutate(model = paste0(i, "-Class Model"))
  model_results <- rbind(model_results, temp) }

pp_plots <- model_results %>% filter(category == 2) %>%
  dplyr::select(est, model, LatentClass, param) %>%
  mutate(param = as.factor(str_to_lower(param)))

pp_plots$param <- fct_inorder(pp_plots$param)

ggplot(pp_plots,
       aes(x = param, y = est, color = LatentClass, shape = LatentClass, group = LatentClass)) +
  geom_point() + geom_line() + facet_wrap(~ model, ncol = 2) + labs(x= "", y = "Probability") +
  theme_minimal() + theme(legend.position = "none", axis.text.x = element_text(size = 6))
```

---

8. Plot Final Model - Conditional Item Probability Plot

This syntax creates a function called `plot_lca_function` that requires 7 arguments (inputs):

- `model_name`: name of Mplus model object (e.g., `model_step1`)
- `item_num`: the number of items in LCA measurement model (e.g., `5`)
- `class_num`: the number of classes (k) in LCA model (e.g., `3`)
- `item_labels`: the item labels for x-axis (e.g., `c("Enjoy","Useful","Logical","Job","Adult")`)
- `class_labels`: the class label names (e.g., `c("Adaptive Coping","Externalizing Behavior","No Coping")`)
- `class_legend_order` = change the order that class names are listed in the plot legend (e.g., `c(2,1,3)`)
- `plot_title`: include the title of the plot here (e.g., `"LCA Posterior Probability Plot"`)

Read in plot data from Mplus output file `c3_lca.out`

```r
model_c3 <- readModels(here("mplus_lca", "c3_lca.out"), quiet = TRUE)
```

Load plot_lca_function into R environment

```r
plot_lca_function <- function(model_name,item_num,class_num,item_labels,
                              class_labels,class_legend_order,plot_title){

mplus_model <- as.data.frame(model_name$gh5$means_and_variances_data$estimated_probs$values)
plot_data <- mplus_model[seq(2, 2*item_num, 2),]

c_size <- as.data.frame(model_name$class_counts$modelEstimated$proportion)
colnames(c_size) <- paste0("cs")
c_size <- c_size %>% mutate(cs = round(cs*100, 2))
colnames(plot_data) <- paste0(class_labels, glue(" ({c_size[1:class_num,]}%)"))
plot_data <- plot_data %>% relocate(class_legend_order)

plot_data <- cbind(Var = paste0("U", 1:item_num), plot_data)
plot_data$Var <- factor(plot_data$Var,
               labels = item_labels)
plot_data$Var <- fct_inorder(plot_data$Var)

pd_long_data <- melt(plot_data, id.vars = "Var")

p <- pd_long_data %>%
  ggplot(aes(x = as.integer(Var), y = value,
  shape = variable, colour = variable, lty = variable)) +
  geom_point(size = 4) + geom_line() +
  scale_x_continuous("", breaks = 1:item_num,
                     labels = function(x) str_wrap(plot_data$Var, width = 13)) +
  labs(title = plot_title, y = "Probability") +
  theme_cowplot() +
  theme(legend.title = element_blank(),
        legend.position = "top",
        axis.text.x = element_text(size=8))

p
return(p)
}
```

Run C3 Plot

```r
plot_lca_function(
  model_name = model_c3,
  item_num = 6,
  class_num = 3,
  item_labels = c("harassment: disability","harassment: race","harassment: sex",
```

```
                   "school staff: counselor","school staff: psychologist",
                   "school staff: law enforcement"),
  class_labels = c("C1","C2","C3"),
  class_legend_order = c(1,3,2),
  plot_title = "Harrasment & School Staff (K = 3)"
  )
```

```
ggsave(here("figures","c3_lca_plot.png"),
       dpi=300, height=4, width=6, units="in")
```

References

Hallquist, M. N., & Wiley, J. F. (2018). MplusAutomation: An R Package for Facilitating Large-Scale Latent Variable Analyses in Mplus. Structural equation modeling: a multidisciplinary journal, 25(4), 621-638.

Muthén, B. O., Muthén, L. K., & Asparouhov, T. (2017). Regression and mediation analysis using Mplus. Los Angeles, CA: Muthén & Muthén.

Muthén, L.K. and Muthén, B.O. (1998-2017). Mplus User's Guide. Eighth Edition. Los Angeles, CA: Muthén & Muthén

US Department of Education Office for Civil Rights. (2014). Civil rights data collection data snapshot: School discipline. Issue brief no. 1.

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686
