Observed Response Patterns in Latent Class Analysis

Adam Garber

Norwegian University of Science and Technology - A Course in `MplusAutomation`

June 01, 2021

Lab preparation

Data source: Longitudinal Study of American Youth, Science Attitudes

See documentation about the LSAY here.

Load packages

```
library(tidyverse)
library(glue)
library(MplusAutomation)
library(here)
library(janitor)
library(gt)
library(DT)
library(plotly)
library(gg3D)
library(gganimate)
library(viridis)
library(hrbrthemes)
```

Exploring observed response patterns

Load data

```
lsay_data <- read_csv("https://garberadamc.github.io/project-site/data/lca_lsay_sci.csv",
                     na = c("9999", "9999.00")) %>%
  clean_names() %>%
  dplyr::select(1:5, Enjoy = ab39m, Useful = ab39t,
                    Logical = ab39u, Job = ab39w, Adult = ab39x)
```

Use {`DT::datatable()`} to take a look at the data

```
datatable(lsay_data, rownames = FALSE, filter="top",
          options = list(pageLength = 5, scrollX=T) )
```

Show 5 entries Search:

Enjoy	Useful	Logical	Job	Adult
All	All	All	All	All
1	1	1	1	1
0	0	1	0	0
1	1	0	0	0
0	0	0	1	1
0	1	1	0	0

Showing 1 to 5 of 3,061 entries Previous 1 2 3 4 5 … 613 Next

Figure. Path diagram of science attitude indicators.

Save response frequencies for the 4 class model with `response is _____.dat`.

```
patterns  <- mplusObject(
  TITLE = "C4 LCA - Save response patterns",
  VARIABLE =
  "categorical = Enjoy-Adult;
   usevar = Enjoy-Adult;

   classes = c(4);",

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

  SAVEDATA =
   "File=3step_savedata.dat;
    Save=cprob;
    Missflag= 999;
    !!!!!!!! Code to save response frequency data !!!!!!!!
    response is resp_patterns.dat;
    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!",

  OUTPUT = "sampstat residual patterns tech10 tech11 tech14",
```

```
  PLOT =
    "type = plot3;
    series = Enjoy-Adult(*);",

  usevariables = colnames(lsay_data),
  rdata = lsay_data)

patterns_fit <- mplusModeler(patterns,
                dataout=here("21-response-patterns", "resp_pattrn", "LSAY.dat"),
                modelout=here("21-response-patterns", "resp_pattrn", "patterns.inp") ,
                check=TRUE, run = TRUE, hashfilename = FALSE)
```

Read in observed respnse pattern data

```
patterns <- read_table2(here("21-response-patterns", "resp_pattrn", "resp_patterns.dat"),
                        col_names=FALSE, na = "*")

colnames(patterns) <- c("Frequency", "ENJOY", "USEFUL", "LOGICAL", "JOB", "ADULT",
                      "CPROB1", "CPROB2", "CPROB3", "CPROB4", "C_MODAL")
```

Order responses by highest frequency

```
order_highest <- patterns %>%
  arrange(desc(Frequency))
```

```
loop_cond  <- lapply(1:4, function(k) {
order_cond <- patterns %>%
  filter(C_MODAL == k) %>%
  arrange(desc(Frequency)) %>%
  head(5)
  })

table_data1 <- bind_rows(loop_cond) %>%
  as.data.frame()

table_data2 <-  rbind(order_highest[1:5,], table_data1)
```

Use {gt} to make a nicely formatted table

```
table_data2 %>%
  gt() %>%
    tab_header(
    title = md("**Observed response patterns, estimated frequencies, estimated posterior
                class probabilities, and modal class assignment.**"),
    subtitle = md(" ")) %>%
    tab_source_note(
    source_note = md("Data Source: **Longitudinal Study of American Youth.**")) %>%
    cols_label(
    ENJOY = "Enjoy",
    USEFUL = "Useful",
    LOGICAL = "Logical",
```

```r
    JOB = "Job",
    ADULT = "Adult",
    CPROB1 = html("P<sub>k=1"),
    CPROB2 = html("P<sub>k=2"),
    CPROB3 = html("P<sub>k=3"),
    CPROB4 = html("P<sub>k=4"),
    C_MODAL = md("*k*")) %>%
  tab_row_group(
    group = "Unconditional response patterns ordered by highest frequency",
    rows = 1:5) %>%
  tab_row_group(
    group = "k=1 conditional response pattern ordered by highest frequency",
    rows = 6:10) %>%
  tab_row_group(
    group = "k=2 conditional response pattern ordered by highest frequency",
    rows = 11:15)  %>%
  tab_row_group(
    group = "k=3 conditional response pattern ordered by highest frequency",
    rows = 16:20) %>%
  tab_row_group(
    group = "k=4 conditional response pattern ordered by highest frequency",
    rows = 21:25) %>%
    row_group_order(
      groups = c("Unconditional response patterns ordered by highest frequency",
                 "k=1 conditional response pattern ordered by highest frequency",
                 "k=2 conditional response pattern ordered by highest frequency",
                 "k=3 conditional response pattern ordered by highest frequency",
                 "k=4 conditional response pattern ordered by highest frequency")) %>%
  tab_options(column_labels.font.weight = "bold")
```

Observed response patterns, estimated frequencies, estimated po probabilities, and modal class assignment.

	Frequency	Enjoy	Useful	Logical	Job	Adult	P<sub>k=1
Unconditional response patterns ordered by highest frequency							
	558	0	0	0	0	0	0.000
	529	1	1	1	1	1	0.957
	313	1	0	0	0	0	0.000
	135	1	0	1	0	0	0.002
	94	1	1	1	0	1	0.687
k=1 conditional response pattern ordered by highest frequency							
	529	1	1	1	1	1	0.957
	94	1	1	1	0	1	0.687
	78	0	1	1	1	1	0.859
	62	1	1	0	1	1	0.580
	55	1	1	1	1	0	0.650
k=2 conditional response pattern ordered by highest frequency							
	135	1	0	1	0	0	0.002
	88	0	0	1	0	0	0.000
	74	1	1	1	0	0	0.063

	47	1	1	0	0	0	0.006
	44	1	0	0	1	0	0.004
k=3 conditional response pattern ordered by highest frequency							
	91	1	0	0	0	1	0.003
	88	1	0	1	1	1	0.337
	76	1	0	1	0	1	0.048
	70	1	0	0	1	1	0.031
	53	0	0	0	0	1	0.001
k=4 conditional response pattern ordered by highest frequency							
	558	0	0	0	0	0	0.000
	313	1	0	0	0	0	0.000
	53	0	0	0	1	0	0.000
	11	0	0	NA	0	0	0.000
	9	0	NA	0	0	0	0.000

Data Source: **Longitudinal Study of American Youth.**

Visualizing observed response patterns

Order rows by modal assignment (K)

```
order_modal <- patterns %>%
  arrange(desc(C_MODAL)) %>%
  rownames_to_column() %>%
  rename('pat_num' = "rowname") %>%
  drop_na(ENJOY:ADULT)
```

Prepare plot data

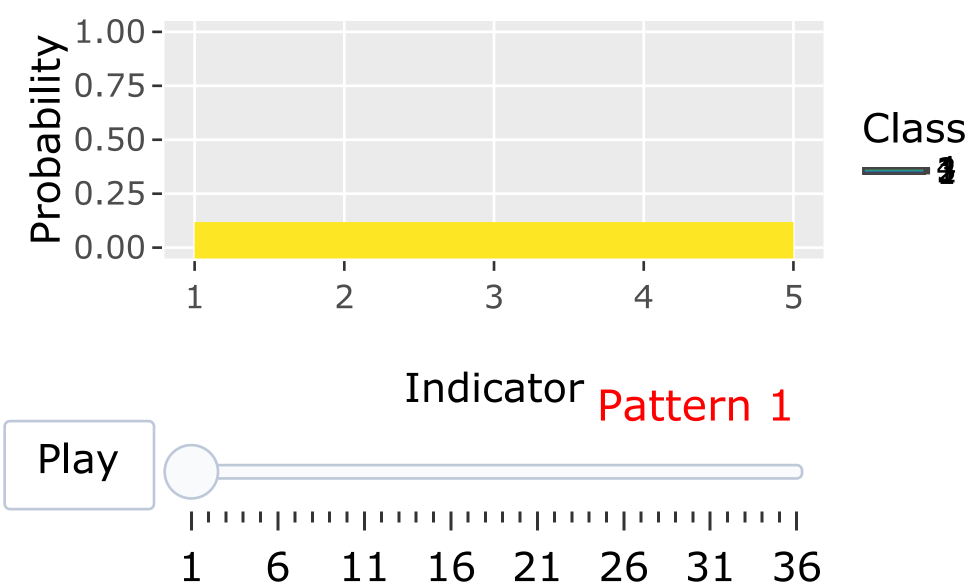
```
p1_long <- order_modal %>%
  dplyr::select(pat_num:ADULT, C_MODAL) %>%
  pivot_longer(`ENJOY`:`ADULT`, # The columns I'm gathering together
               names_to = "var", # new column name for existing names
               values_to = "value") %>%  # new column name to store values
  mutate(obs = rep(1:32, each =5)) %>%
  mutate(Class = factor(C_MODAL)) %>%
  mutate(var = ordered(var,
                      levels = c("ENJOY","USEFUL","LOGICAL","JOB","ADULT"))) %>%
  select(-pat_num, -C_MODAL)

# must first run LCA enumeration (code is out of sequential order)
out_c4 <- readModels(here("21-response-patterns", "resp_pattrn"), filefilter = "patterns", quiet = TRUE
```

```
## <simpleError in startLine:endLine: NA/NaN argument>
```

```r
# extract posterior probabilities
probs_c4 <- as.data.frame(
  out_c4[["gh5"]][["means_and_variances_data"]]
  [["estimated_probs"]][["values"]]
  [seq(2, 10, 2),])

rownames(probs_c4) <- c("ENJOY","USEFUL","LOGICAL","JOB","ADULT")

long_c4 <- probs_c4 %>% rownames_to_column() %>%
  rename('var' = "rowname") %>%
  pivot_longer(`V1`:`V4`, # The columns I'm gathering together
               names_to = "c", # new column name for existing names
               values_to = "value") %>% # new column name to store values
  mutate(Class = rep(1:4,5)) %>%
  arrange(Class) %>%
  mutate(obs = rep(33:36,each=5)) %>%
  mutate(Frequency = rep(c(829,782,619,833),each=5)) %>%
  mutate(var = ordered(var,
                      levels = c("ENJOY","USEFUL","LOGICAL","JOB","ADULT"))) %>%
  select(6,1,3,5,4)

p2_long <- rbind(p1_long, long_c4) %>%
  mutate(Class = as.numeric(Class))
```

Visualize observed response patterns with {plotly}

```r
gg <- ggplot(p2_long, aes(x=var, y=value, color = Class, size=Frequency)) +
  geom_line(aes(as.numeric(var), frame = obs)) +
  scale_color_viridis() + labs(x="Indicator", y= "Probability")

ggplotly(gg) %>%  animation_opts(frame = 1000, transition = 0) %>%
  animation_slider(currentvalue =
                    list(prefix = "Pattern ", font = list(color="red")))
```



Make a 3D plot with packages {ggplot2}, {gg3D}, and {gganimate}.

```r
theta= 170    # change perspective (tilt)
phi=40        # change perspective (rotation)

resp3d <- ggplot(p1_long, aes(x=as.numeric(var),
                              y=as.numeric(value),
                              z = as.numeric(obs)),
                 alpha = .8) +
  axes_3D(theta=theta, phi=phi) +
  stat_3D(theta=theta, phi=phi, geom="path",
          aes(colour = Class, size = Frequency), alpha = .8) +
  scale_color_manual(values=c("#FDE725FF", "#DE7065FF", "#238A8DFF", "#482677FF")) +
  theme_void() +
  annotate("text", x = -.3, y = 0.05, label = "Indicators ") +
  annotate("text", x = .35, y = -.4, label = "Probability") +
  annotate("text", x = .25, y = .42, label = "Pattern") +
  annotate("text", x = .2, y = 0, label = "0.0") +
  annotate("text", x = .34, y = -.33, label = "1.0") +
  annotate("text", x = -.05, y = 0, angle = 6,
           label = "Enjoy - Useful - Logical - Job - Adult") +
  transition_states(obs, transition_length=1, state_length=5) +
  shadow_mark(alpha = .1,) +
  labs(title = "Observed response pattern = {closest_state}")


animate(resp3d, fps = 2)
```

```r
anim_save(here("21-response-patterns", "figures", "responses_3d_anim.gif"), height = 6, width = 8, dpi
```

Comparing model fit

Learning objective: Generate a comprehensive model fit summary table.

Information criteria: model is endorsed by lowest value:

- `BIC`:
$$= -2 * LL + Npar * LN(N)$$
- `aBIC`:
$$-2 * LL + Npar * LN((N + 2)/24)$$
- `CIAC`:
$$-2 * LL + Npar * (LN(N) + 1))$$
- `AWE`:
$$-2 * LL + 2 * Npar * (LN(N) + 1.5)$$

Run a quick enumeration

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

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

    VARIABLE = glue(
    "categorical = Enjoy-Adult;
     usevar = Enjoy-Adult;
     classes = c({k}); "),

  ANALYSIS =
   "estimator = mlr;
    type = mixture;
    starts = 200 50;
    processors = 10;",

  OUTPUT = "sampstat residual tech11 tech14;",

  PLOT =
    "type = plot3;
     series = Enjoy-Adult(*);",

  usevariables = colnames(lsay_data),
  rdata = lsay_data)

lca_enum_fit <- mplusModeler(lca_enum,
    dataout=glue(here("21-response-patterns", "enum_mplus", "lsay.dat")),
    modelout=glue(here("21-response-patterns", "enum_mplus", "c{k}_lca.inp")) ,
    check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

Create model fit summary table

Extract data and calculate indices derived from the Log Likelihood

```
all_output <- readModels(here("21-response-patterns", "enum_mplus"), quiet = TRUE)
```

```
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
```

```
n_size <- all_output[["c1_lca.out"]][["summaries"]][["Observations"]]

enum_extract <- LatexSummaryTable(all_output,
                keepCols=c("Title","Parameters", "LL", "BIC",
```

```r
                           "aBIC", "BLRT_PValue", "T11_VLMR_PValue"),
                sortBy = "Title")

all_fit <- enum_extract %>%
  mutate(aBIC = -2*LL+Parameters*log((n_size+2)/24)) %>%
  mutate(CIAC = -2*LL+Parameters*(log(n_size)+1)) %>%
  mutate(AWE = -2*LL+2*Parameters*(log(n_size)+1.5)) %>%
  mutate(SIC = -.5*BIC) %>%
  mutate(expSIC = exp(SIC - max(SIC))) %>%
  mutate(expSUM = sum(expSIC)) %>%
  mutate(BF = exp(SIC-lead(SIC))) %>%
  mutate(cmPk = expSIC/expSUM) %>%
  select(1:5,8:9,7,6,13,14)
```

Format table with package {gt}

```r
all_fit %>%
  gt() %>%
    tab_header(
    title = md("**Model Fit Summary Table**"), subtitle = md(" ")) %>%
    tab_source_note(
    source_note = md("Data Source: **Longitudinal Study of American Youth.**")) %>%
    cols_label(
    Title = "Classes",
    Parameters = md("*NPar*"),
    LL = md("*LL*"),
    T11_VLMR_PValue = html("VLMR"),
    BLRT_PValue = html("BLRT"),
    BF = html("Bayes<br>Factor"),
    cmPk = html("cmP<sub>k</sub>")) %>%
  tab_options(column_labels.font.weight = "bold") %>%
  fmt_number(10:11,decimals = 2,
             drop_trailing_zeros=TRUE,
             suffixing = TRUE) %>%
  fmt_number(2:9,decimals = 2)
```

Model Fit Summary Table

Classes	*NPar*	*LL*	BIC	aBIC	CIAC	AWE	VLMR	BLRT	Bayes Factor
Class 1	5.00	−10,250.60	20,541.34	20,525.45	20,546.34	20,596.47	NA	NA	0
Class 2	11.00	−8,785.32	17,658.92	17,623.97	17,669.93	17,780.22	0.00	0.00	0
Class 3	17.00	−8,693.57	17,523.59	17,469.57	17,540.59	17,711.04	0.00	0.00	0
Class 4	23.00	−8,664.09	17,512.79	17,439.71	17,535.79	17,766.40	0.00	0.00	5.22*B*
Class 5	29.00	−8,662.39	17,557.54	17,465.40	17,586.54	17,877.31	0.67	1.00	12.32*B*
Class 6	35.00	−8,661.54	17,604.01	17,492.80	17,639.01	17,989.94	0.75	1.00	NA

Data Source: **Longitudinal Study of American Youth.**

References

Drew A. Linzer, Jeffrey B. Lewis (2011). poLCA: An R Package for Polytomous Variable Latent Class Analysis. Journal of Statistical Software, 42(10), 1-29. URL http://www.jstatsoft.org/v42/i10/.

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.

Miller, J. D., Hoffer, T., Suchner, R., Brown, K., & Nelson, C. (1992). LSAY codebook. Northern Illinois University.

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

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