Three Step Auxiliary Variable Integration with `MplusAutomation`

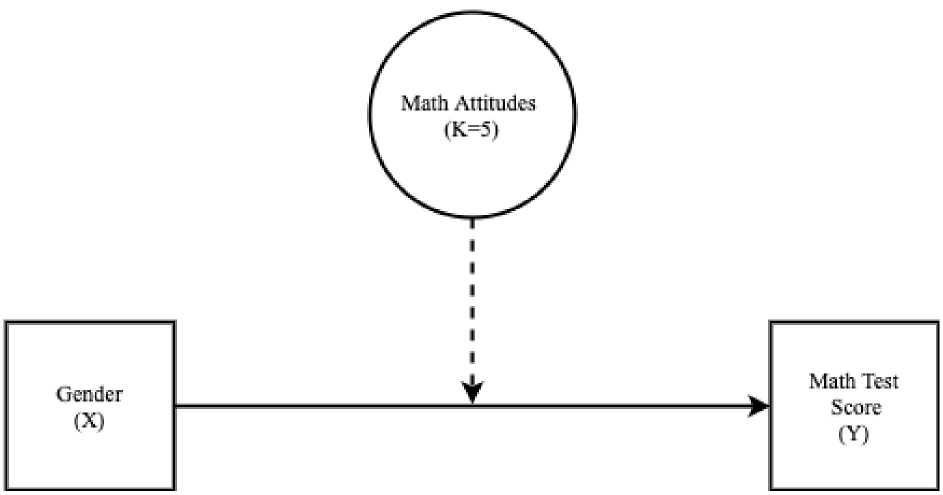
Adding covariates and distal outcome variables to mixture models

Adam Garber

Outline

1. Conduct the "Manual 3-Step ML" procedure
2. Specify a moderation model with the latent class variable as moderator

---



Load packages

```
library(MplusAutomation)  # a conduit between R & Mplus
library(here)             # to locate or send files within the Rproject folder
library(glue)             # to insert R code into strings
library(gt)               # for pretty tables
library(tidyverse)        # for everything else...
```

Data source. This workshop utilizes the public-use data repository named the *Longitudinal Survey of American Youth* (LSAY; Miller et al., 1992).

The "three-step ML approach" was first introduced here:

Vermunt, J. K. (2010). Latent class modeling with covariates: Two improved three-step approaches. Political analysis, 18(4), 450-469.

```
lsay_data <- read_csv(here("14-three-step", "data", "lsay_subset.csv")) %>%
  mutate(female = gender - 1)
```

LCA Indicators & Auxiliary Variables: Math Attitudes Example

Name	Variable Description
enjoy	I enjoy math.
good	I am good at math.
undrstnd	I ussually understand what we are doing in math.
nervous	Doing math often makes me nervous or upset
scared	I often get scared when I open my math book and see a page of problems.
useful	Math is useful in everyday problems.
logical	Math helps a person think logically.
job	It is important to know math to get a good job.
adult	I will use math in many ways as an adult.
Auxiliary Variables	
female	Self-reported student gender (0=Male, 1=Female).
math_irt	Standardized IRT math test score measured two years distal of LCA indicators.

Manual "3-Step" ML Auxiliary Variable Integration Method

Step 1 - Estimate the unconditional model with all covariate & distal outcome variables mentioned in the `auxiliary` statement.

```
m_step1  <- mplusObject(

  TITLE = "Step1_3step_automation - LSAY",

  VARIABLE =
   "categorical = enjoy-adult;

    usevar = enjoy-adult;

    classes = c(5);

    !!! NOTE: All auxiliary variables to be considered in the final model should be listed here !!!

    auxiliary = math_irt female;",

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

  SAVEDATA =
   "!!! NOTE: This saved dataset will contain class probabilities and modal assignment columns !!!
    File=3step_savedata_092021.dat;
    Save=cprob;
    Missflag= 999;",

  MODEL = "",
  OUTPUT = "",

  PLOT =
   "type = plot3;
    series = enjoy-adult(*);",

  usevariables = colnames(lsay_data),
  rdata = lsay_data)

m_step1_fit <- mplusModeler(m_step1,
                 dataout=here("14-three-step", "3step_mplus", "Step1_3step.dat"),
                 modelout=here("14-three-step", "3step_mplus", "Step1_3step.inp"),
                 check=TRUE, run = TRUE, hashfilename = FALSE)
```

Step 2 - Extract logits & saved data from the step 1 unconditional model.

Extract logits to estimate classification error in the step 2 model

```
logit_cprobs <- as.data.frame(m_step1_fit[["results"]]
                                        [["class_counts"]]
                                        [["logitProbs.mostLikely"]])
```

Extract saved data from the step 1 model `mplusObject` named “m_step1_fit”

```
savedata <- as.data.frame(m_step1_fit[["results"]]
                                    [["savedata"]])
```

Rename the column in savedata for “C” and change to “N”

```
colnames(savedata)[colnames(savedata)=="C"] <- "N"
```

Step 3 (part 1) - Estimate the unconditional model with logits from step 2.

This model is estimated to check that the class proportions are approximately the same as in step 1.

```
m_step2  <- mplusObject(

  TITLE = "Step2_3step_automation LSAY",

  VARIABLE =
 "nominal=N;
  USEVAR = N;
  missing are all (999);
  classes = c(5); ",

  ANALYSIS =
 "estimator = mlr;
  type = mixture;
  starts = 0;",

  MODEL =
    glue(
 "%C#1%
  [n#1@{logit_cprobs[1,1]}];
  [n#2@{logit_cprobs[1,2]}];
  [n#3@{logit_cprobs[1,3]}];
  [n#4@{logit_cprobs[1,4]}];

  %C#2%
  [n#1@{logit_cprobs[2,1]}];
  [n#2@{logit_cprobs[2,2]}];
  [n#3@{logit_cprobs[2,3]}];
  [n#4@{logit_cprobs[2,4]}];

  %C#3%
  [n#1@{logit_cprobs[3,1]}];
  [n#2@{logit_cprobs[3,2]}];
  [n#3@{logit_cprobs[3,3]}];
  [n#4@{logit_cprobs[3,4]}];

  %C#4%
  [n#1@{logit_cprobs[4,1]}];
  [n#2@{logit_cprobs[4,2]}];
  [n#3@{logit_cprobs[4,3]}];
  [n#4@{logit_cprobs[4,4]}];

  %C#5%
  [n#1@{logit_cprobs[5,1]}];
  [n#2@{logit_cprobs[5,2]}];
  [n#3@{logit_cprobs[5,3]}];
```

```
  [n#4@{logit_cprobs[5,4]}]; "),

  OUTPUT = "",

  PLOT = "",

  usevariables = colnames(savedata),
  rdata = savedata)

m_step2_fit <- mplusModeler(m_step2,
                 dataout=here("14-three-step", "3step_mplus", "Step2_3step.dat"),
                 modelout=here("14-three-step", "3step_mplus", "Step2_3step.inp"),
                 check=TRUE, run = TRUE, hashfilename = FALSE)
```

Step 3 (part 2) - Estimate the moderation model

Add covariates & distal outcomes to the model.

Specification details:

- This example contains one distal outcome variable (`math_irt`) and one binary covariate (`female`).
- Under each class-specific statement (e.g., `%C#1%`) the distal outcome is mentioned to estimate the intercept mean (in square brackets) & variance parameters.
- Moderation is specified by mentioning the `"outcome ON covariate;"` syntax under each of the class-specific statements.
- Note that the binary covariate is centered so that reported distal means (intercepts) are estimated at the weighted average of `female`.

```
m_step3  <- mplusObject(
  TITLE = "Step3_3step_automation LSAY",

  VARIABLE =
 "nominal = n;
  usevar = n math_irt female;
  missing are all (999);
  classes = c(5); ",

  DEFINE =
 "center female (grandmean);",

  ANALYSIS =
 "estimator = mlr;
  type = mixture;
  starts = 0;",

  MODEL =
  glue(
 "!!! DISTAL = math_irt  !!!
  !!! COVARIATE = female !!!

  %OVERALL%

  c on female;
  math_irt on female;
  math_irt;

  %C#1%
  [n#1@{logit_cprobs[1,1]}];
  [n#2@{logit_cprobs[1,2]}];
  [n#3@{logit_cprobs[1,3]}];
  [n#4@{logit_cprobs[1,4]}];
```

```
 [math_irt] (m1);
 math_irt;
 math_irt on female (s1);

 %C#2%
 [n#1@{logit_cprobs[2,1]}];
 [n#2@{logit_cprobs[2,2]}];
 [n#3@{logit_cprobs[2,3]}];
 [n#4@{logit_cprobs[2,4]}];

 [math_irt] (m2);
 math_irt;
 math_irt on female (s2);

 %C#3%
 [n#1@{logit_cprobs[3,1]}];
 [n#2@{logit_cprobs[3,2]}];
 [n#3@{logit_cprobs[3,3]}];
 [n#4@{logit_cprobs[3,4]}];

 [math_irt] (m3);
 math_irt;
 math_irt on female (s3);

 %C#4%
 [n#1@{logit_cprobs[4,1]}];
 [n#2@{logit_cprobs[4,2]}];
 [n#3@{logit_cprobs[4,3]}];
 [n#4@{logit_cprobs[4,4]}];

 [math_irt] (m4);
 math_irt;
 math_irt on female (s4);

 %C#5%
 [n#1@{logit_cprobs[5,1]}];
 [n#2@{logit_cprobs[5,2]}];
 [n#3@{logit_cprobs[5,3]}];
 [n#4@{logit_cprobs[5,4]}];

 [math_irt] (m5);
 math_irt;
 math_irt on female (s5); "),

 MODELCONSTRAINT =
"New (diff12 diff13 diff14 diff15
     diff23 diff24 diff25
     diff34 diff35 diff45
     slope12 slope13 slope14 slope15
     slope23 slope24 slope25
     slope34 slope35 slope45);

 diff12 = m1-m2;   diff24 = m2-m4;
```

```
  diff13 = m1-m3;   diff25 = m2-m5;
  diff14 = m1-m4;   diff34 = m3-m4;
  diff15 = m1-m5;   diff35 = m3-m5;
  diff23 = m2-m3;   diff45 = m4-m5;

  slope12 = s1-s2;   slope24 = s2-s4;
  slope13 = s1-s3;   slope25 = s2-s5;
  slope14 = s1-s4;   slope34 = s3-s4;
  slope15 = s1-s5;   slope35 = s3-s5;
  slope23 = s2-s3;   slope45 = s4-s5;",

  MODELTEST =
  ## NOTE: Only a single Wald test can be conducted per model run. Therefore,
  ## this example requires running separate models for each omnibus test (e.g.,
  ## 2 models; 1 outcome and 1 slope coefficient). This can be done by
  ## commenting out all but one test and then making multiple input/output files.

 "m1=m2;       !!! Distal outcome omnibus Wald test for `math_irt` !!!
  m2=m3;
  m3=m4;
  m4=m5;

  !s1=s2;      !!! Slope difference omnibus Wald test `math_irt on female` !!!
  !s2=s3;
  !s3=s4;
  !s4=s5; ",

  usevariables = colnames(savedata),
  rdata = savedata)

m_step3_fit <- mplusModeler(m_step3,
                 dataout=here("14-three-step", "3step_mplus", "Step3_3step.dat"),
                 modelout=here("14-three-step", "3step_mplus", "Step3_3step.inp"),
                 check=TRUE, run = TRUE, hashfilename = FALSE)
```

References

Garber, A. C. (2021). 3-Step ML Auxiliary Variable Integration Using MplusAutomation. Retrieved from psyarxiv.com/phtxa

Hallquist, Michael N., and Joshua F. Wiley. 2018. "MplusAutomation: An R Package for Facilitating Large-Scale Latent Variable Analyses in Mplus." Structural Equation Modeling, 1–18. https://doi.org/10.1080/10705511.2017.1402334.

Nylund, K. L., Asparouhov, T., & Muthén, B. O. (2007). Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study. Structural equation modeling: A multidisciplinary Journal, 14(4), 535-569.

R Core Team. 2019.R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Vermunt, J. K. (2010). Latent class modeling with covariates: Two improved three-step approaches. Political analysis, 18(4), 450-469.

Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R, Grolemund G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V, Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to the tidyverse." Journal of Open Source Software, 4(43), 1686. doi: 10.21105/joss.01686.

For more examples using `MplusAutomation`:

https://garberadamc.github.io/project-site/
