Introduction to Latent Profile Analysis (LPA)

A Course in `MplusAutomation`

Adam Garber

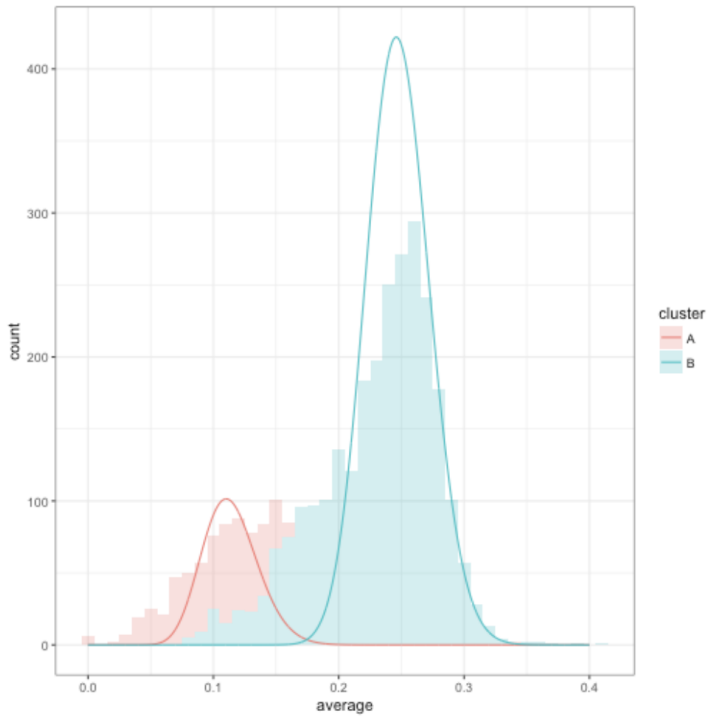


Figure. Gaussian mixture models. Data simulated from a 2-class model.

Preparation

Data source:

1. The first example closely follows the vignette used to demonstrate the `tidyLPA` package (Rosenberg, 2019): See detailed documentation of this model here

- This model utilizes the `PISA` data collected in the U.S. in 2015. To learn more about this data see here.
- To access the 2015 US `PISA` data & documentation in R use the following code:

```
devtools::install_github("jrosen48/pisaUSA15"); library(pisaUSA15); open_codebook()
```

2. The second examples utilizes 4 test score measures from the public-use dataset, *The Longitudinal Survey of American Youth* (**LSAY**): See documentation here

---

Load packages

```
library(naniar)
library(tidyverse)
library(haven)
library(glue)
library(MplusAutomation)
library(here)
library(janitor)
library(gt)
library(tidyLPA)
```

Load data

```
pisa <- pisaUSA15
```

Latent Profile Analysis

Variance/covariance specification decisions in LPA modeling

$$\begin{bmatrix} \sigma_1^2 & \sigma_{21} & \sigma_{31} & \sigma_{41} \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \sigma_{24} \\ \sigma_{13} & \sigma_{12} & \sigma_3^2 & \sigma_{33} \\ \sigma_{14} & \sigma_{12} & \sigma_{12} & \sigma_4^2 \end{bmatrix}$$

Figure. Picture of variance/covariance matrix adapted from tutorial by Rosenberg (2019).

- `model 1` Class-invariant / Diagonal: Equal variances, and covariances fixed to 0
- `model 2` Class-varying / Diagonal: Free variances and covariances fixed to 0
- `model 3` Class-invariant / Non-Diagonal: Equal variances and equal covariances
- `model 4` Free variances, and equal covariances
- `model 5` Equal variances, and free covariances
- `model 6` Class Varying / Non-Diagonal: Free variances and free covariances

---

Example 1: PISA dataset from the `tidyLPA` package

Enumerate using `estimate_profiles()`:

- Estimate models with classes $K = 1:3$
- Model has 4 continuous indicators
- Default variance-covariance specifications (model 1)
- Add line `scale() %>%` to center indicator means

```
lpa_models <- pisa[1:500,] %>%
    select(broad_interest, enjoyment, instrumental_mot, self_efficacy) %>%
    estimate_profiles(1:3,
                      package = "MplusAutomation",
```

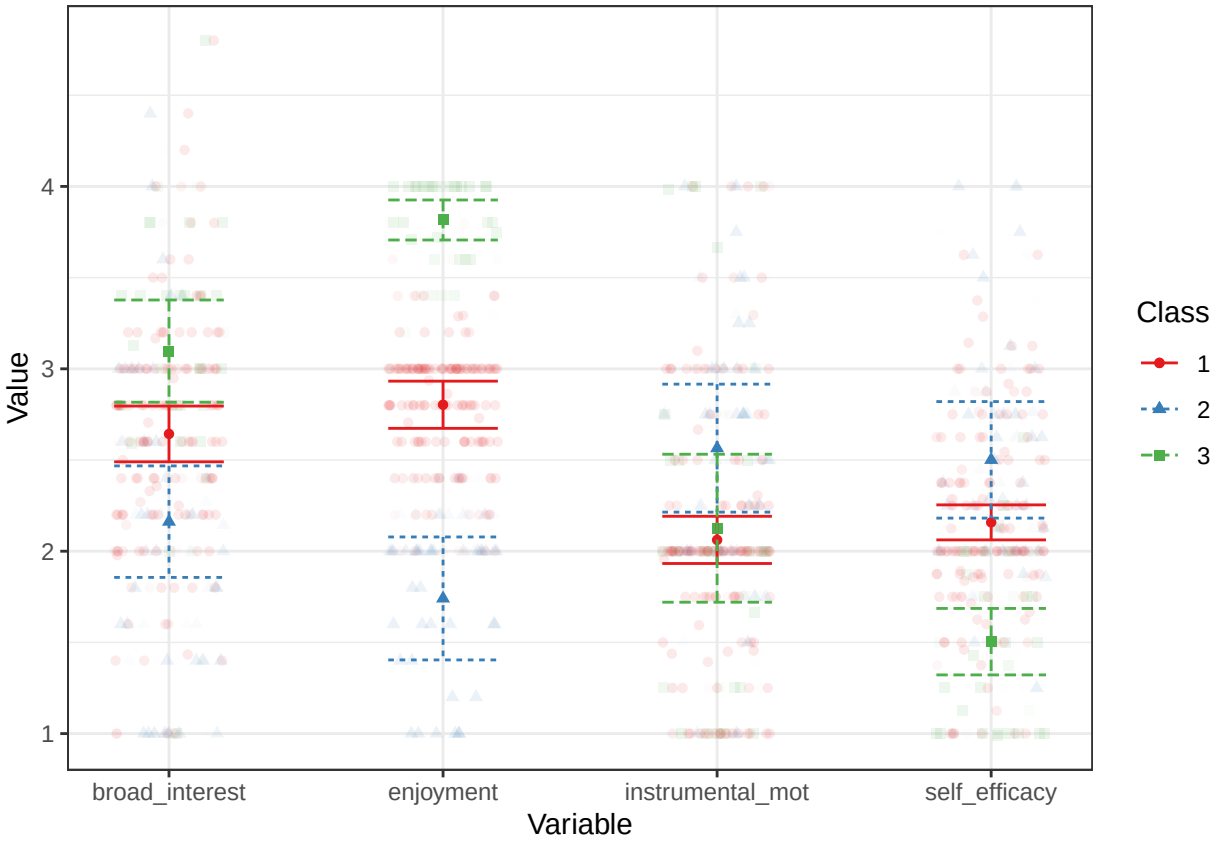
```
                        ANALYSIS = "starts = 100, 20;",
                        variances = c("equal", "varying"),
                        covariances = c("zero", "varying"))

get_fit(lpa_models)
```

Plot 3-class model

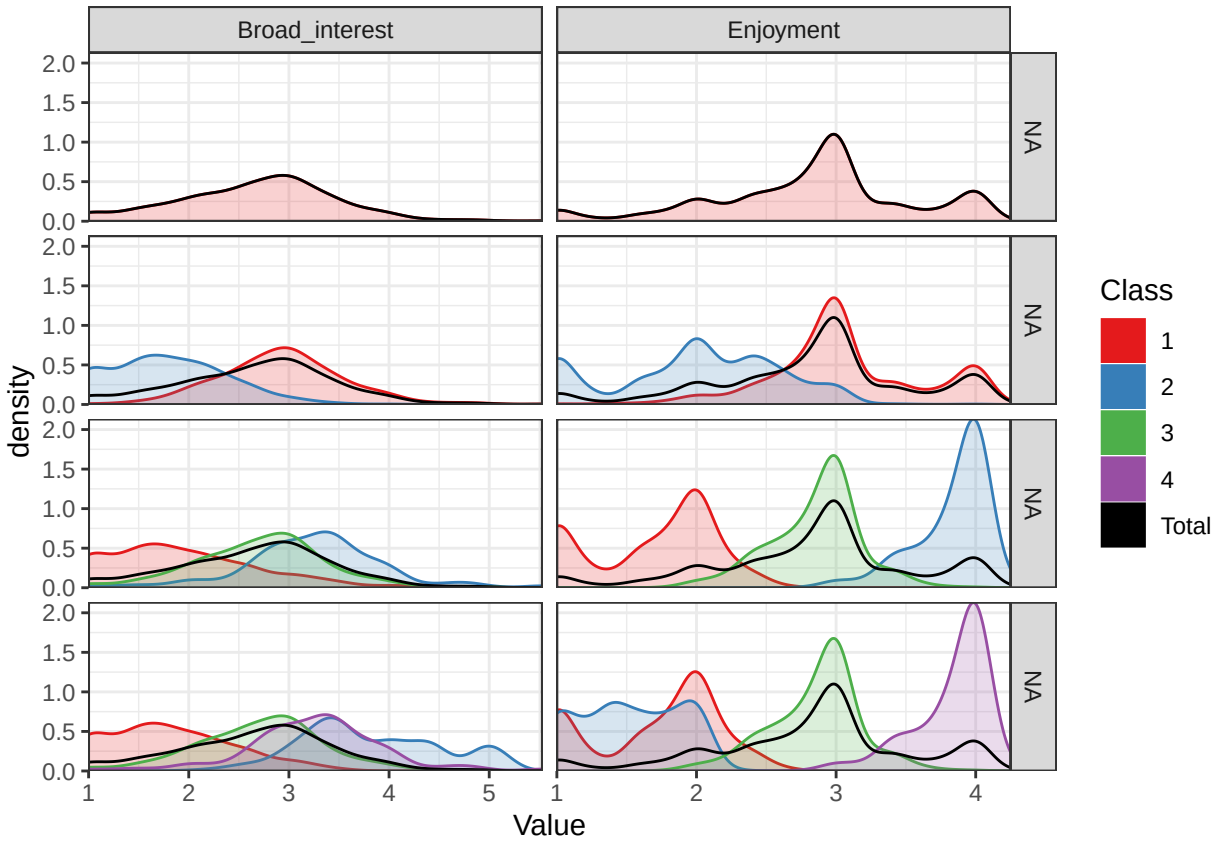
Note: single imputation is used in this example as `plot_profiles()` requires complete cases

```
pisa[1:200,] %>%
    select(broad_interest, enjoyment, instrumental_mot, self_efficacy) %>%
    single_imputation() %>%
    estimate_profiles(3, package = "MplusAutomation") %>%
    plot_profiles(sd=FALSE)
```



Plot densities for classes `k = 1:4`

```
pisa[1:500, c("broad_interest","enjoyment")] %>%
  single_imputation() %>%
  estimate_profiles(1:4, package = "MplusAutomation") %>%
  plot_density()
```



Example 2: Math, Science, Physics, and Biology measures (LSAY).

Read in data

```
lsay_data <- read_csv("https://garberadamc.github.io/project-site/data/lsay_lab10.2_lpa.csv")
```

Run a quick enumeration

```
lpa_k14  <- lapply(1:4, function(k) {
```

```
  lpa_enum  <- mplusObject(

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

    VARIABLE = glue(
    "usevar = mth_scor-bio_scor;
     classes = c({k}); "),

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

  OUTPUT = "sampstat residual tech11 tech14;",

  PLOT =
    "type = plot3;
     series = mth_scor-bio_scor(*);",

  usevariables = colnames(lsay_data),
  rdata = lsay_data)

lpa_enum_fit <- mplusModeler(lpa_enum,
                dataout=here("15-LPA", "enum_lpa", "lpa_lsay.dat"),
                modelout=glue(here("15-LPA", "enum_lpa", "c{k}_lpa_lsay.inp")) ,
                check=TRUE, run = TRUE, hashfilename = FALSE)
})
```

Plot 3-class profile

```
lsay_data[1:500,5:8] %>%
    single_imputation() %>%
    estimate_profiles(1:4, package = "MplusAutomation") %>%
    plot_profiles(sd=FALSE)
```

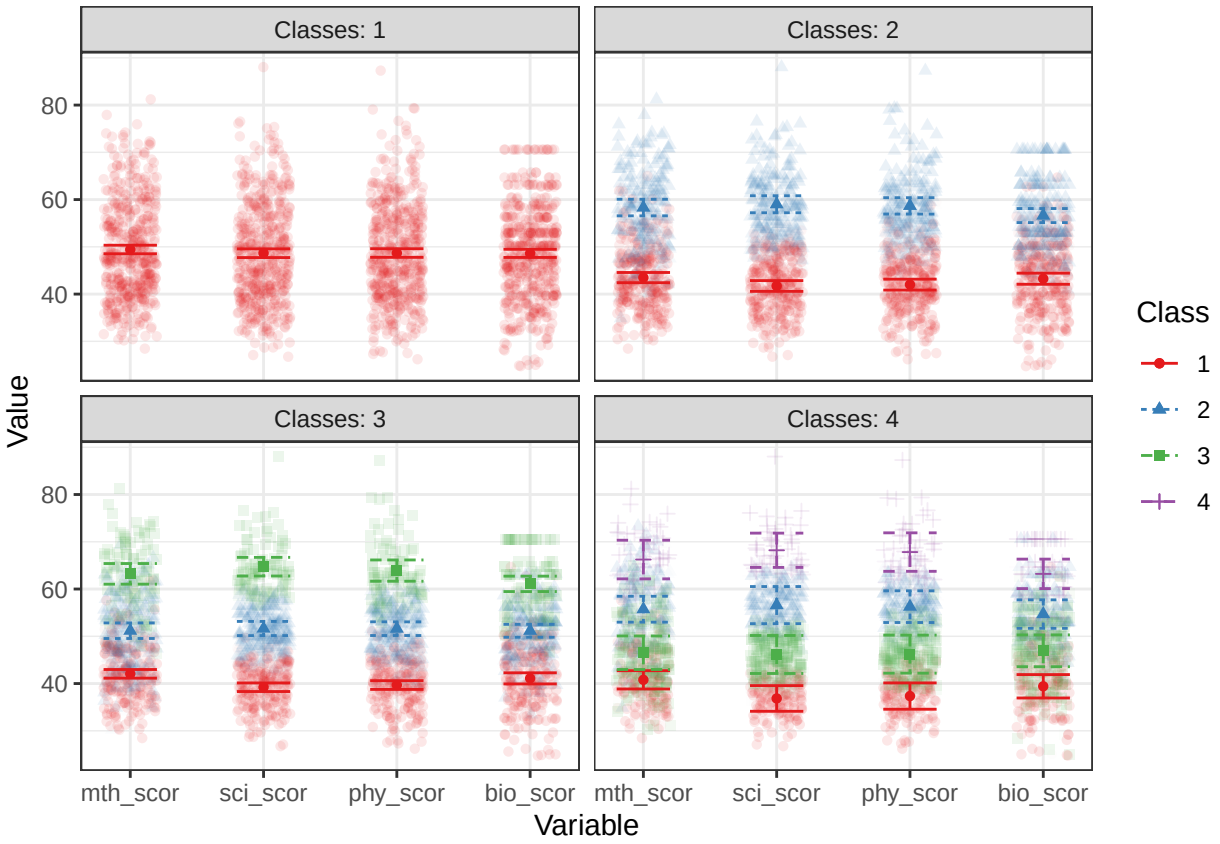


Figure. Here we see ordered solutions.

Compare model fit.

```
all_output <- readModels(here("15-LPA", "enum_lpa"), quiet = TRUE)

enum_extract <- LatexSummaryTable(all_output,
                keepCols=c("Title","Parameters", "LL", "BIC",
                           "aBIC", "BLRT_PValue", "T11_VLMR_PValue"),
                sortBy = "Title")

enum_extract %>% gt()
```

Title	Parameters	LL	BIC	aBIC	BLRT_PValue	T11_VLMR_PValue
Class 1	8	-46288.29	92640.89	92615.47	NA	NA
Class 2	13	-43352.36	86809.23	86767.93	0	0
Class 3	18	-42126.11	84396.93	84339.74	0	0
Class 4	23	-41433.72	83052.37	82979.29	0	0

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.

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

Rosenberg, J. M., van Lissa, C. J., Beymer, P. N., Anderson, D. J., Schell, M. J. & Schmidt, J. A. (2019). tidyLPA: Easily carry out Latent Profile Analysis (LPA) using open-source or commercial software [R package]. https://data-edu.github.io/tidyLPA/

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

Further resources & examples here:

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

https://www.adam-garber.com/
