# Introduction to Mixture and Trajectory-Based Groups: Mplus Language

## Dr Oliver Perra.

Full resource and contact: https://www.ncrm.ac.uk/resources/online/all/?id=20826

I have opted to use Mplus for the exercises in these resources, mostly because Mplus provides many flexible and powerful options to run this type of analyses.

In what follows, I will provide an introduction to Mplus and its syntax. If you are familiar with Mplus, you can skip this.

I will then provide an introduction concerning the syntax used to specify latent growth models first, then a general introduction to the specification of Mixture models, focusing on Latent Class models, to finally introduce Growth Mixture Models (GMM), and Latent Class Growth Models (LCGA).

# Introduction to Mplus for beginners

Mplus operates by creating INPUT files: these instruct the software on how to read the file, the type of variables, the type of analysis, algorithms and estimators to be used, the model to be estimated. The input file also instructs the software on the results and parameters to be displayed in the OUTPUT file, how to create files with graphics, how to create other files with derived parameters from the models estimated.

Once the analyses have been run, Mplus will produce an OUTPUT file. This is text file that reports the instructions provided in the INPUT and the results of the analyses run.

The principal commands in Mplus input files are:

TITLE:

DATA:

VARIABLE:

ANALYSIS:

MODEL:

OUTPUT:

PLOT:

SAVEDATA:

These are all followed by a colon.

Within each command, it is possible to provide further options. These are separated by semicolon “;” at the end of each option.

Other common rules in Mplus are that the verbs “are/is” and the equal sign “=” are allowed to specify some options. For example, in specifying the file where the data is one can write :

DATA: File is dat.dat;

or

DATA: File = dat.dat;

Hyphens (-) are used to indicate a list of variables or numbers. Items in a list can be separated by space or commas.

Exclamation marks (!) are used to write comments in the input that are not read by the software. What follows the “!” in the line is skipped by the software.

**A SHORT INTRODUCTION TO MPLUS SYNTAX**

Some general rules:

* The main commands are followed by colon “:” . Options within the commands are followed by semi-colon “;”.
* Most Mplus options can be shortened (e.g.: USEVARIABLE=🡪USEVAR=).
* It is possible to use lower- or upper-case characters to name variables, but in the outputs Mplus will report the variable names using upper-case characters.
* It is possible to give names to variables that exceed 8 characters, but in the outputs Mplus will only report the first 8 characters of a variable name.

**TITLE:**

This command allows you to provide a title to the file (free text).

**DATA:**

This command is used to specify where the data are (i.e. the data file to be read), their format and other options (if necessary). It is common to create .dat files (or .txt) that contain data in person-level format (one row for each individual, with variables values in columns). Do not include variable names in the data file !!!

It is possible to specify a path to the data file (e.g.: C:\Desktop\datac1.dat).

NOTE: If there are spaces in the path to the data file, you must enclose the path between parentheses, e.g.:

DATA: FILE=”C:\my files\datac1.dat”

An important option in the DATA: command is **LISTWISE= ON**, which instructs Mplus to conduct analyses with listwise deletion of cases with one or more missing data in the variables of interest. From version 5 of Mplus the default option is to include cases with incomplete information, i.e. with some missing values on some of the variables in the model.

**VARIABLE:**

This command is used to provide names to the variables in the dataset and specify what types of variables they are (their scale) etc.

Important options are:

**NAMES =** 🡪 list the names of the variables in the dataset.

e.g.:

NAMES = cpubid\_xrnd mpubid\_xrnd crace\_xrnd csex\_xrnd cyrb\_xrnd cogna86 emota86 ;

**USEVARIABLE =** 🡪 lists the variables that will be used in the analyses or models. For example, if you will run analyses on the items concerning respondents’ strengths and difficulties, write:

USEVARIABLES = ranti86 ranx86 rhypr86 rdep86 rpeer86;

This will ensure other variables, e.g. IDs, will not included in the analyses.

Note that if a variable is specified after USEVARIABLE= it will be included in the models specified after command **MODEL:** even if not mentioned specifically in the **MODEL:** command.

**CATEGORICAL, NOMINAL, COUNT =** 🡪 are used to define a list of *dependent* variables as ordered categorical (or binary), nominal (unordered), or count variables respectively. Categorical, nominal and count variables that are not treated as dependent should NOT be listed under the CATEGORICAL etc. command.

e.g.:

CATEGORICAL = ranti86 ranx86 rhypr86 rdep86 rpeer86;

**MISSING=** 🡪 specifies the value (or a character such as . or \*) used to identify a missing value for one or more variables. If all the variables use the same missing value indicator (e.g. -999), write:

MISSING = all (-999);

**IDVARIABLE=** is used to indicate the identifier for each observation or case in the dataset. This is necessary if you want to create data files after the analyses (**SAVEDATA:** command) and want to use them for further analyses: in this case, the file saved with **SAVEDATA:**  will contain IDs for the observations. e.g.:

IDVARIABLE = cpubid\_xrnd;

**CLUSTER=** is used to indicate a clustering variable (e.g. school, group). This is necessary for multilevel analyses or for analyses that adjust for clustering (option: **COMPLEX** in **ANALYSIS:** command). For example, you could specify that data are clustered at the county level:

CLUSTER=county;

*!remember that to adjust for clustering, you need to also add the option COMPLEX after ANALYSIS:*

**CLASSES** is used to specify names of latent categorical variables and the number of categories (between parentheses). If we want to estimate a model with a latent variable called “class” with 2 categories we could write:

CLASSES = class (2);

We can add more latent variables after **CLASSES=**, depending on our model. For example, if we wanted to specify a latent class model with latent classes for data collections in the year 1986 and in the year 1988, we could write:

CLASSES = class86 (2) class88(3);

The text above instructs Mplus to estimate a latent class variable with two classes called “class86” and a second latent class variable that includes three classes, “class88”. To specify the associations between these variables and the associations between them and the observed variables (indicators), we will have to use the **MODEL:** command (see below and the section “Introduction to Latent Class syntax in Mplus”).

**ANALYSIS:**

This is the command used to specify the type of analysis and other options in the analysis (e.g. type of estimator).

Option **TYPE=** invokes a specific type of analysis among a range of options (e.g. EFA, i.e. exploratory factor analysis).

**TYPE=BASIC;**

invokes *descriptive statistics* on the variables included by USEVARIABLE.

In order to run Latent Class Analysis, Latent Transition Analysis, or other mixture models (GMM or LCGA, etc.) one has to invoke **TYPE=MIXTURE;**

**MODEL:**

The model command allows to specify a model, constrain parameters, test parameter constraints.

Mplus allows to specify many parameters of a model using syntax. The main operators in the syntax are:

**ON:** short for “regressed on” and defines regression relationships e.g.:

MODEL:

y ON x male;

will estimate the regression of variable “y” on variables “x” and “male”.

**BY:** is short for “measured by”, and defines latent continuous variables (but not latent classes!). For example:

MODEL:

f1 BY y1 y2 y3 y4 y5;

will estimate a latent variable “f1” using indicators “y1”, “y2”, etc.

Note that the same command can be shorted in this way:

f1 BY y1-y5;

where the hyphen indicates all variables from y1 to y5.

**WITH:** indicates correlational relationships. For example:

MODEL:

f1 WITH f2;

indicates a correlation between f1 and f2.

In the **MODEL:** command, the name of variables is used to indicate the variance or residual variance (depending on the model) of variables, while the name of a variable between square brackets refers to means, intercepts, or thresholds.

For example, if “x” and “y” are continuous variables, the command:

MODEL:

[x]; x; [y]; y;

will estimate the mean and variance of x and the mean and variance y, respectively.

However, if we added this line:

MODEL:

**y ON x;**

[x]; x; [y]; y;

the last line will estimate the mean and variance of “x”, and the intercept and residual variance of “y”.

The notation above can be used to constrain parameters using other characters.

The character “at” **@** followed by a number is used to constrain a parameter to a specific value. For example:

MODEL:

y ON x;

[x]; x; [y]; **y@0;**

constrains the residual variance of “y” to be equal to 0.

Adding numbers between parentheses after a parameter can be used to impose equality constraints. For example:

MODEL:

y ON x (1);

w ON z (1)**;**

constrains the linear regression of “y” on “x” to be equal to that of “w” on “z”.

The same equality constrains can be applied by using letters or combination of letters and numbers between parentheses. For example, the equality constraint imposed above could be expressed like this:

MODEL:

y ON x (eq1);

w ON z (eq1)**;**

As long as the string of characters and/or numbers between parentheses is the same, equality constraints are imposed.

**OUTPUT:**

Specifies the information to be reported in the output file.

**PLOT:**

Specifies information necessary to allow MPlus to create plots and graphs.

In Latent Class Analysis, the most useful type of plots are **PLOT3**, which include estimated probabilities for a categorical latent variable as a function of its covariates, and latent variable distribution plots.

When using PLOT3 is also advantageous to list the names of the set of variables that can be used in plots where their values are connected by a line, e.g. the indicators of the latent class model.

To do so use the option SERIES. You can also specify where the values of this variable can be in the x axis of a graph by following the name of the variable with the desired x-axis value between parenthesis.

For example, the following command will ensure you can create PLOT3 types, with the first variable in the list appearing at point 1 of the x-axis, the second appearing at point 2 of the x-axis, etc.:

PLOT: Type = PLOT3;

SERIES= ranx86(1) rdep86(2) ranti86(3) rhypr86(4) rpeer86(5);

If you wanted to represent these variables at 0.5 units of the x-axis, starting from 1, you can write:

PLOT: Type = PLOT3;

SERIES= ranx86(1) rdep86(1.5) ranti86(2) rhypr86(2.5) rpeer86(3);

**SAVEDATA:**

Instructs the programme to create a file with parameters estimated.

**FILE =** used to give a name to new file. For example:

FILE IS analsis1.dat;

**SAVE =** uses to specify what information can be saved in the file. **CPROB** will save posterior latent class probabilities and modal class assignment for each individual included in the analyses. If an IDVARIABLE is specified in **VARIABLE:** command, the file will contain ID variable information, and it will be possible to match-merge the file with other files for further analyses. E.g.:

FILE is analysis1.dat;

SAVE = CPROB;

# Introduction to Latent Growth Models in Mplus

Latent growth models are applied to longitudinal data collected over three or more measurement occasions. These models represent the growth parameters *intercept* (initial status of the variable at the start of the study) and *slopes* (rate of change in the variable per units of time) as continuous latent *factors*. These latent continuous factors explain the change observed over time in the variables collected.

Mplus uses a specific notation to specify latent growth models. This notation uses a vertical bar **|** to separate the latent variables (growth parameters) from the observed variables collected on different occasions. Suppose that we observed variable ***a*** when participants were 15, 16, and 17 years old: thus we have variable ***a15***, ***a16***, and ***a17***. If we want to specify a latent growth model with an intercept ***i*** and a linear slope ***s***, we would write:

VARIABLES:

NAMES=id a15 a16 a17;

USEVAR=a15 a16 a17;

MODEL:

**i s | a15@0 a16@1 a17@2;**

The latter line is specifying the model where latent variables ***i*** and ***s*** are being estimated as predictors of ***a15***, ***a16***, and ***a17*** scores. The first term is an intercept (i.e. initial status of the variable at time 0) and the second one is a slope. They can be called with different names, e.g. “int” and “slo”. The coefficients of the intercept are fixed at one by default.

The statement on the right of the vertical bar | specify the outcomes and the time scores for the growth model. The **@** sign after each outcome fixes the time scores for the slope growth factor. In the example, time scores are fixed at 0, 1, and 2 respectively to define a *linear growth* model where the time points are equidistant. Different time scores can be specified to centre time at different points or specify different types of trajectories.

For example, if we wanted to centre time at the last measurement occasion, we would have written:

**i s | a15@-2 a16@-1 a17@0;**

In the latter example, the intercept represents the variable status at age 17 years.

If we wanted to represent a logarithmic growth, we would have specified:

**i s | a15@0.1 a16@1 a17@10;**

The model will also estimate the variances of the intercept and slope factors, as well as their covariance.

If we wanted to add a quadratic term to represent a quadratic growth trend, we would add a further latent factor on the left side of the horizontal bar |, like this:

**i s q | a15@0 a16@1 a17@2;**

While the first slope ***s*** has time scores as indicated in the right side of the vertical bar |, i.e. 0, 1, and 2, the time scores of the second slope ***q*** are automatically computed as the squared values of the time scores provided, i.e. 0^2=0, 1^2=1, 2^2=4. The second slope can be called any other name, just like the other growth factors. When we include a second slope, the variance of this factor is also estimated, as are the covariances between all the growth factors.

When we estimate a similar model with continuous outcomes, the residual variances of the outcome variables are estimated and allowed to vary at each age, while they are assumed not to be correlated by default.

It is possible to apply the same model to *dichotomous* or *ordered categorical* outcomes:

VARIABLES:

NAMES=id a15 a16 a17 c15 c16 c17;

USEVAR=c15 c16 c17;

**CATEGORICAL=c15 c16 c17;**

MODEL:

**i s | c15@0 c16@1 c17@2;**

In this case, the outcomes are conceptualised as categorisations of a latent continuous response (see the second presentation in these Resources). The latent response is divided into observed categories by thresholds, cut-points that divide the underlying continuous response into observed categories. The assumption of this model is that all the thresholds of the categorical variables ***c15***, ***c16***, and ***c17*** are constant across time. Since the scale of the underlying latent response is arbitrary, the mean of the intercept factor ***i*** is fixed at zero by default. Mplus will also estimate the variances of the intercept and slope factors, as well as their covariances.

The default estimator for a latent growth model with categorical outcomes is a Weighted Least Squares Estimator with Missing Values (WLSMV). This invokes a probit model and a Delta parameterization. Other estimators can be invoked by specifying them in the ANALYSIS: command. For example:

ANALYSIS:

**Estimator=MLR;**

will invoke a Maximum Likelihood Estimator with Robust Standard Errors (MLR).

# Introduction to Mixture Models in Mplus

Mplus estimates latent class models using a combination of the **VARIABLE:** and **ANALYSIS:** commands.

Namely, when you want Mplus to estimate a latent variable model, you should include a line that defines the name of the latent class variables and the number of classes that must be estimated in the **VARIABLE:** command.

If you have a similar syntax to define the variables in the dataset

VARIABLES:

NAMES = […] ;

MISSING= all (-999);

USEVAR = ranti86 ranx86 rhypr86 rdep86 rpeer86;

CATEGORICAL=ranti86 ranx86 rhypr86 rdep86 rpeer86;

and you wanted to estimate a latent class model with two categories, you will add the last line as below:

VARIABLES:

NAMES = […] ;

MISSING= all (-999);

USEVAR = ranti86 ranx86 rhypr86 rdep86 rpeer86;

CATEGORICAL=ranti86 ranx86 rhypr86 rdep86 rpeer86;

**CLASSES=c(2);**

The last line will estimate a latent variable “**c**” with 2 categories (2 classes). The latent class variable can be called whatever you prefer (considering Mplus variables are restricted to 8 characters in the Output file) and you can use the syntax to specify different number of classes. For example, if I wanted to call the latent variable “**disorder**” and estimate 6 classes, I would write:

VARIABLES:

NAMES = […] ;

MISSING= all (-999);

USEVAR = ranti86 ranx86 rhypr86 rdep86 rpeer86;

CATEGORICAL=ranti86 ranx86 rhypr86 rdep86 rpeer86;

**CLASSES=disorder(2);**

Specifying the name and number of latent classes in the **VARIABLE:** command is not sufficient for Mplus to estimate a latent class model. The other necessary step is specifying a mixture model in the option ANALYSIS:, like this:

**ANALYSIS:**

**TYPE=mixture;**

The option “Mixture” invokes estimation of mixture models like latent class analysis, latent transition analysis, etc. This option can be combined with other options, e.g. TWOLEVEL or COMPLEX.

For example, if the data are clustered (e.g. adolescents clustered across schools), you can control for this clustering by specifying the clustering variable in the **VARIABLE:** command, and including the option **TYPE**=**COMPLEX** in the **ANALYSIS:** command.

The default estimator for MIXTURE models is MLR (maximum likelihood with robust standard errors and chi-square). However, one can invoke another estimator by writing the name of the estimator after **ESTIMATOR =**....;

The default algorithm with MLR is the Expectation-Maximisation (EM) algorithm (but other algorithms can be invoked).

It is possible (and indeed, advised) to change the number of initial stage starts and final stage optimizations of the EM algorithm by using option **STARTS=** in the **ANALYSIS:** command, for example:

STARTS = 100 20;

It is also possible to change the number of initial stage iterations in the EM algorithm using option STITERATIONS, for example specifying:

STITERATIONS = 20;

**PROCESSORS**= can also be used to devote more computer memory resources (processors) to the estimation process (default is PROCESSORS = 1).

meters to be the same across latent classes.

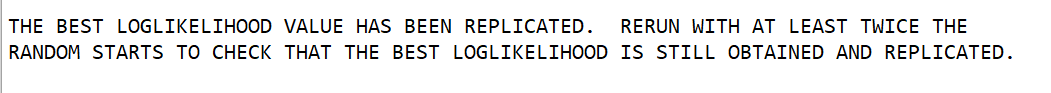
Increasing the number of starts in the EM algorithm will reduce the chances that the latent class solution is based on a “local” maximum, rather than a “global” one. You can look at the NCRM resources that I have prepared on Latent Class Analysis for more details, see <https://www.ncrm.ac.uk/resources/online/all/?id=20806>

To ensure the latent class solution is reliable, you will have to check the Mplus output that details the replication of the model log-likelihood:

A screenshot of a computer

Description automatically generated with medium confidence

It is important to check that the model best log-likelihood is replicated a number of times (some authors suggest at least 20 times). In many instances, Mplus will provide this warning:



It is good practice to follow this advice and ensure the best log-likelihood is replicated when you increase the number of random starts.

The specification of a latent class model does not require statements in the **MODEL:** command if the latent class model is only invoked to explain the associations between indicators listed in the **VARIABLE: USEVAR=**…option.

It is however possible to specify model parameters of the latent classes by using the **MODEL:** command.

Whenever we estimate latent classes, the **MODEL:** command is divided in two sections. The first section is preceded by a line **%OVERALL%** that indicates that all the options following it concern model parameters that apply across all latent classes being estimated. The second section of the **MODEL:** command is preceded by a line where the name of the latent variable is between **%** signs, and the number of the latent class is indicated by **#**. For example, if we estimate a latent categorical variable ***c*** with 2 classes, the two sections of the **MODEL:** command will read:

VARIABLES:

NAMES = […] ;

USEVAR = a b c d;

**CLASSES=c(2);**

ANALYSIS:

TYPE=MIXTURE;

STARTS= 500 50;

**MODEL:**

**%OVERALL%**

**%c#1%**

**%c#2%**

For example, if we wanted to regress our latent categorical variable **c** on a covariate ***sex***, we would write this statement in the **%OVERALL%** section of the model:

**MODEL:**

**%OVERALL%**

**c on sex;**

%c#1%

%c#2%

If we wanted to ensure that the means of continuous variables ***a***, ***b***, and ***c*** are fixed to zero for individuals in latent class 1, we would specify these constraints within the **%c#1%** section:

**MODEL:**

%OVERALL%

c on sex;

**%c#1%**

**[a@0];**

**[b@0];**

**[c@0];**

%c#2%

Overall, using these options we can specify parameters of the model for all the latent classes being estimated as well as class-specific parameters.

Note that if we specify parameters for all the latent classes in **%OVERALL%** but do not intend to specify class-specific parameters, we can avoid writing the lines that invoke class-specific parameters altogether . For example, if the only parameter we want to specify is the regression of latent classes on covariate ***sex***, we would just write:

**MODEL:**

**%OVERALL%**

**c on sex;**

# Introduction to Mixture and Trajectory-Based Groups in Mplus: Continuous Outcomes

We can specify these models by combining the commands that invoke Mixture models and those that invoke growth models.

Suppose that we observed a variable ***a*** when participants were 15, 16, and 17 years old: thus we have variable ***a15***, ***a16***, and ***a17***. A latent growth model is specified by this command:

VARIABLES:

NAMES=id a15 a16 a17;

USEVAR=a15 a16 a17;

MODEL:

**i s | a15@0 a16@1 a17@2;**

However, if we wanted to estimate a latent categorical variable ***k***, with two categories that represent individuals with different growth parameters, we would add these statements in bold:

VARIABLES:

NAMES=id a15 a16 a17;

USEVAR=a15 a16 a17;

**CLASSES=k(2);**

**ANALYSIS:**

**TYPE=MIXTURE:**

MODEL:

**%OVERALL%**

i s | a15@0 a16@1 a17@2;

In this case, the model will be a Mixture model, as specified in **ANALYSIS:**, and we are estimating a latent categorical variable with two classes.

The growth factors are specified after the **%OVERALL%** line, which indexes the model parameters that concerns all the classes. This means that the model is asking to estimate growth factors across all classes. In other words, the growth factors ***i*** and ***s*** will be different for individuals in latent class 1 from those of individuals in latent class 2. The model will also estimate the residual variance of the growth factors, as well as their residual covariance. Since the variable in this example is continuous, the residual variances of the outcome variables are estimated at each time point and can differ. However, the residuals are assumed not correlated by default. Effectively, this model is a Growth Mixture Model (GMM).

If we wanted to specify a Latent Class Growth Analysis (LCGA) model, the main difference lies in the fact we assume that there is no residual variance in the growth factors. We can thus specify the model by constraining the growth factor residual variances to zero:

VARIABLES:

NAMES=id a15 a16 a17;

USEVAR=a15 a16 a17;

CLASSES=k(2);

ANALYSIS:

TYPE=MIXTURE:

MODEL:

%OVERALL%

i s | a15@0 a16@1 a17@2;

**i@0; s@0;**

Since there is no variance in the growth factors, all individuals within one class will have the same growth parameters as other individuals within the latent class.

# Introduction to Mixture and Trajectory-Based Groups in Mplus: Dichotomous and Ordered Categorical Outcomes

Suppose that we observed an ordered categorical variable ***c*** when participants were 15, 16, and 17 years old: thus we have variable ***c15***, ***c16***, and ***c17***. A GMM models is specified by this command:

VARIABLES:

NAMES=id a15 a16 a17;

USEVAR=a15 a16 a17;

CATEGORICAL=c15 c16 c17;

CLASSES=k(2);

ANALYSIS:

TYPE=MIXTURE;

**ALGORITHM=INTEGRATION;**

MODEL:

%OVERALL%

i s | c15@0 c16@1 c17@2;

Mplus will estimate a latent categorical variable ***k***, with two categories that represent individuals with different growth parameters. The addition of **ALGORITHM=INTEGRATION** in the **ANALYSIS:** command ensures that the residual variances of the growth factors, as well as their covariance, will be estimated.

The model assumes that the thresholds of the categorical outcomes are the same across age and across latent classes. Furthermore, Mplus fixes the average of the intercept factor to zero in the last class estimated by default. This is done in order to anchor the latent response distribution underlying the categorical variable to a fixed point.

If we wanted to estimate a LCGA model, we would just have to delete the option **ALGORITHM=INTEGRATION** from the **ANALYSIS:** command:

VARIABLES:

NAMES=id a15 a16 a17;

USEVAR=a15 a16 a17;

CATEGORICAL=c15 c16 c17;

CLASSES=k(2);

**ANALYSIS:**

**TYPE=MIXTURE;**

MODEL:

%OVERALL%

i s | c15@0 c16@1 c17@2;

# The Three-Step Approach

Mixture models are often used to identify groups that show different propensities, e.g. different expected trajectories. Once researchers have a satisfactory and parsimonious classification, they may explore how this classification relates to predictors, other processes, other outcomes, and so on.

A problem that has marred the use of Mixture Models in practice lies in the process of estimating the latent class models *concurrently* with covariates or distal outcomes.

In fact, when we run latent class models simultaneously considering indicator variables and distal outcome or covariates, **the latent classes will represent individual variation (heterogeneity) in the indicators *and* the other variables**.

This causes **practical problems**: while we had a satisfactory latent class model when we only included indicators, our latent class measurement model may be different when we introduce the distal outcome. The problems are **not just practical**, as we also must interpret the model considering it represents heterogeneity across different types of variables.

**Different solutions have been tried to solve this problem**. The more naïve solution would be to estimate a latent class model, for example a LCGA, and then consider to which latent class participants are most likely to belong. The latent classes assignment is then used as a nominal variable in further analyses: We can investigate the association between covariates and participants’ class affiliation, or that between latent class affiliation and distal outcomes. For example, do individuals in different latent classes of depression retire at significantly different ages?

The main problem with this naïve approach is that it does not take into account the measurement error in latent class membership. The latent class models are probabilistic, and participants’ assignment to estimated latent classes (their latent class membership) is uncertain. If we fail to account for this uncertainty and use latent class membership as a variable in a model, we will obtain biased results. See <http://statmodel.com/download/relatinglca.pdf> for a more in-depth discussion.

There are different approaches that could deal with this issue (e.g. Pseudo-Class Draws), but the **Three-Step Approach** is the one that I find particularly useful.

In the next section I will introduce the principles underlying the Three-Step Approach.

# The Three-Step Approach: Introduction

This approach has been more recently developed. The solution to the problem of including covariates and distal outcomes lies in conducting the measurement model and the modelling of structural relationships (e.g. regressing latent classes on covariates) in separate steps (respectively the first and the third steps of this procedure). An intermediate step links the other two steps by estimating measurement error in class assignment, thus allowing to control for this error when imposing structural relationships between other variables and the latent classes estimated.

I will illustrate these steps with a practical example.

* **Step 1: Estimate the Optimal Model and Assign Individuals to the Most Likely Class (Modal Class)**

Let’s assume we have estimated two latent classes representing different trajectories of antisocial behaviour. The output of the model will provide posterior probabilities of being in each of these two classes, with the “most likely” latent class membership for each individual, see Figure 3.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Antisocial Beh 13yrs | Antisocial Beh 14yrs | Antisocial Beh 15yrs | Antisocial Beh 16yrs | p Class1 | p Class2 | Most likely class |
| 101 | 1 | 1 | 2 | 1 | .043 | .957 | 2 |
| 102 | 3 | 3 | 2 | 3 | .969 | .031 | 1 |
| 103 | 1 | 2 | 1 | 1 | .099 | .901 | 2 |
| 104 | 2 | 1 | 3 | 2 | .424 | .576 | 2 |
| … |  |  |  |  |  |  |  |

*Table 1: Fictional example of data representing frequency of antisocial behaviour (higher value=more frequent), probability of membership in two latent classes, and the most likely class (Latent Class Modal Assignment).*

The most likely class to which each individual is assigned will be used in Step 3 as a nominal variable to estimate class membership while controlling for uncertainty in this membership, as I will illustrate in Step 3. Before that, I will explain the necessary steps to obtain estimates of uncertainty in latent class estimation.

* **Step 2: Estimate measurement error (i.e. uncertainty in class allocation)**

The posterior probabilities indicate the level of uncertainty in class membership. For example, while membership into Class 1 appears more certain for ID=102, membership into Class 2 for ID=104 appears quite uncertain.

We can use these probabilities to calculate the average probability of being in each class if the most likely class is 1 or 2. Considering the example in Figure 3, **the average probability of being in latent Class 2 if the most likely class=2** will be given by:

That is, the probability of being in latent Class 2 for IDs 101, 103, and 104, who are most likely in latent Class 2.

In the same way, we can calculate all the others average probabilities of being in class 1 or 2 if the most likely class is 1 or 2. These average probabilities can then be reported in Table like the one in Table 2.

Table

Description automatically generated

*Table 2: Average Latent Class Probabilities for Most Likely Latent Class Membership (Rows) by Latent Class (Columns)*

For example, the first cell in the table in Table 2 represents the average probability of being in latent Class 1 if the most likely latent Class = 1 (0.924). The *Ns* in the last columns represent the number of participants who, in this fictional example, have been assigned to latent Class 1 and latent Class 2, respectively.

Taking the Table 2 as a reference, we can then calculate the **classification probabilities** for the most likely latent class membership by latent class. For example, the classification probability when the most likely class membership is Class 1 and individuals are classified in latent class 1 will be equal to:

Namely, this classification probability is equal to the product of the average probability of being in Class 1 when the most likely class=1 by the number of individuals whose most likely class=1, divided by the sum of the latter product and the product of the average probability of being in Class 1 when the most likely class=2 by the number of individuals whose most likely class=2.

In the same way, we can calculate the other classification probabilities, which we can then report in another table, see Table 3.

Graphical user interface, application

Description automatically generated with medium confidence

*Table 3: Classification Probabilities for the Most Likely Latent Class Membership (Rows)by Latent Class (Columns).*

Now we can use these classification probabilities to calculate the logit ratios of being in Class 1 rather than Class 2 when the most likely class=1:

Similarly, we can calculate the logit odds of being in Class 1 rather than in Class 2 when the most likely class=2:

These log-odds represent in a single number the uncertainty about classification of being assigned into latent class 1 and latent class 2. By feeding this information into the model in Step 3, we *inform* the model that the nominal assignment into one class (provided in Step 1) is an imperfect assignment, with the level of uncertainty equal to the log-odds value we have estimated in Step 2. By feeding this information in Step 3, we do not need to re-estimate the measurement models, since these are fixed on the parameters estimated in Step 2. We can thus investigate the associations between latent classes and between these and covariates or distal outcomes, provided we have a fixed optimal measurement model.

* **Step 3: Impose structural relationships between classes and covariates/distal outcomes, while controlling for measurement error in class assignment**

In this final step we use the information from Step 1 (i.e. the most likely class membership of each participant) and from Step 2 (i.e. the measurement error expressed by the logits for classification probabilities) to create a latent class model that is defined by these estimated values. In other words, the latent class model is fixed to the values that reflect the uncertainty in latent class membership, and we can therefore add covariates and distal outcomes without re-estimating the latent class model. In Figure 3 the 3rd step in this approach is represented schematically.

Figure 3 highlights that the association between the most likely class and the latent class is fixed at the measurement error parameters estimated in Step 2: therefore the latent class model is given and will not change.

A picture containing text, screenshot, font, diagram

Description automatically generated

Educational Attainment

*Figure 3: Schematic representation of Step 3 in the Three-Step Approach.*

# The Three-Step Approach in Mplus

Mplus provides some facilities to include covariates and distal outcomes without carrying out the calculations I illustrated above. In particular, Mplus uses the **AUXILIARY** option in the **VARIABLE:** command to specify variables that can be used as predictors of latent classes and as distal outcomes. By following the auxiliary variable with (**R3STEP**), the variable will be used as latent class predictor using the three-step method. If the auxiliary variable is followed by (**DU3STEP**), the three-step method will be used considering the variable as a distal variable with unequal means and variances. By following the variable with (**DE3STEP**), the three-step method will be used considering the variable as a distal variable with unequal means and equal variances. Mplus also allows to use other similar approaches to include covariates and distal outcome without the need to re-estimate the latent class model simultaneously: See Asparouhov and Muthén (2014) for more details.

For example, if we estimated a LCGA with 2 classes, using antisocial behaviour indicators as outcomes, and we wanted to estimate the regression of latent class variable ***c*** on covariate ***male*** using the three-step method, we will write:

VARIABLES:

NAMES= id antisc13 antisc14 antisc15 antisc16 male ses1 ses2 ses3 ses4 ses5 schatt;

USEVAR = antisc13 antisc14 antisc15 antisc16;

CATEGORICAL = antisc13 antisc14 antisc15 antisc16;

**AUXILIARY=male(R3STEP)**

CLASSES= k(2);

ANALYSIS:

TYPE=MIXTURE;

MODEL:

%OVERALL%

i s q | antisc13@0 antisc14@1 antisc15@2 antisc16@3 ;

However, in some occasions we may need to have more control over the models and thus run the Three-Step method in a different way. The Three-Step method is facilitated in Mplus by the fact that the logit odds (logits) that are used to fix the measurement parameters in the 3rd step are readily available in the Mplus Output when running latent class models.

* **Step 1**

The first step is to estimate the latent class model. If, for example, after initial analyses the optimal model for the data appears to be a LCGA model with 2 latent classes, estimate this model ensuring that a data file is saved that includes the posterior latent class probabilities and the most likely class membership for each participant.

To this aim, add **SAVEDATA:** command in the INPUT file. For example:

**SAVEDATA:**

**FILE= twoclasses.dat;**

**SAVE=cprob;**

**MISSFLAG=-999;**

The options above indicate the name of the datafile that will be created when Mplus runs the model (“twoclasses.dat”). Note that you can also specify the path where you want to save this file, e.g.: **FILE= “C:\DESKTOP\ twoclasses.dat”**; You can also save the datafile in other text-based formats (e.g. .txt).

The option **SAVE=cprob**; ensures that the datafile created will include the posterior probabilities of latent class membership, as well as the most likely class of each participant (as long as the participant has valid data for at least one of the indicators).

The option **MISSFLAG= -999**; instructs Mplus to assign value -999 to cells with missing data.

To ensure this datafile can also be match-merged with other datafiles for checks and other uses, make sure you also save the participants IDs in the datafile created by Mplus. To this end, include the ID variable in the **VARIABLE:** command using the option **IDVAR=** , as in the example below:

VARIABLES:

NAMES= id antisc13 antisc14 antisc15 antisc16 male ses1 ses2 ses3 ses4 ses5 schatt;

USEVAR = antisc13 antisc14 antisc15 antisc16;

CATEGORICAL = antisc13 antisc14 antisc15 antisc16;

MISSING = all (-999);

CLASSES= k(2);

**IDVAR=id;**

ANALYSIS:

TYPE=MIXTURE;

MODEL:

%OVERALL%

i s q | antisc13@0 antisc14@1 antisc15@2 antisc16@3 ;

Since the datafile that Mplus will produce after this model estimation will include the most likely class membership, which will be used in the 3rd step of the analysis, it would be useful to also ensure that covariates and distal outcomes are saved in the datafile. We can do this by adding option **AUXILIARY=** and the name of the variables we want to transfer in the datafile that Mplus will create:

VARIABLES:

NAMES= id antisc13 antisc14 antisc15 antisc16 male ses1 ses2 ses3 ses4 ses5 schatt;

USEVAR = antisc13 antisc14 antisc15 antisc16;

CATEGORICAL = antisc13 antisc14 antisc15 antisc16 ;

MISSING = all (-999);

CLASSES= k(2);

IDVAR=id;

**AUXILIARY= male ses1 ses2 ses3 ses4 ses5 ageretir;**

The last line in the box above ensures that variables listed after **AUXILIARY=** will not be included in model estimation, but will be saved in the datafile we will create using command **SAVEDATA:**

After estimating the model, the Mplus OUTPUT will provide information about the datafile it created, where the variable **CPROB1** and **CPROB2** are the probabilities of membership in latent class 1 and in latent class 2 respectively, and the variable ***k*** represents the most likely class membership for each participant.

* **Step 2**

In this step we estimate measurement errors in latent class membership for the model we estimated in Step 1. Mplus facilitates this task by providing in the OUTPUT file tables with the average latent class probabilities, classification probabilities and, crucially, the logits for the classification probabilities.

In the example of the 2-class model estimated in Step 1, we obtain a table such as this in the OUTPUT file:

Text

Description automatically generated

We can use these logits as measurement errors in latent class affiliation in Step 3.

* **Step 3**

In this step we will use the datafile we obtained in Step 1

The variable file name and variable names are those that Mplus indicated, so we will write a similar INPUT file:

**DATA**:

FILE= twoclasses.dat ;

**VARIABLES**:

NAMES= antisc13 antisc14 antisc15 antisc16 id male ses1 ses2 ses3 ses4 ses5

schatt cprob2 cprob2 k;

USEVAR = k male ses2 ses3 ses4 ses5;

NOMINAL = k;

MISSING = all (-999);

CLASSES= newcl(2);

Note that the order of the variables must follow exactly the order in which Mplus put these variables in the datafile.

In the **VARIABLE:** command, we will define the variable **k** as a nominal variable. This is the variable that represents the most likely class membership for each participant. This variable is then used to estimate latent class membership in a new latent class variable with 2 classes, **newcl**, specified in **CLASSES= newcl(2);**

To ensure the association between the most likely class (variable **k**) and the **newcl** variable is fixed according to the measurement error estimated in Step 2, we will fix the association between the indicator **k** and **newcl** in the **MODEL:** command in this way:

**MODEL**:

%OVERALL%

newcl ON male ses2 ses3 ses4 ses5;

%newcl#1%

[k#1 @ 2.295];

%newcl#2%

[k#1 @ -2.775];

Remember that the **%OVERALL%** statement in **MODEL:** specifies the part of the model that concerns all latent classes. In the box above, we are instructing Mplus to estimate the multinomial regression of latent classes **newcl** on covariates **male** and SES (through the use of dummy variables ses2, etc.).

The **%newcl#1%** statement concerns just class 1 of the latent variable **newcl**. The statement **[k#1 @ 2.295];** is fixing the measurement relationship between the nominal most likely class variable **k** and latent class **newcl** to the level of uncertainty determined in Step 2. This is effectively fixing the estimation of latent class to the measurement error determined in Step 2, therefore avoiding a new estimation of the latent class measurement model.

Because of that, when running Step 3, the **STARTS=** option in command **ANALYSIS:** should be set to 0. This avoids re-estimating the measurement model, since the model has been fixed to the level of uncertainty determined in Step 2. Thus, the **ANALYSIS:** command should state:

**ANALYSIS**:

TYPE=MIXTURE;

**STARTS=0;**

Putting all this together, Mplus will run multinomial regression models where latent class affiliation into Class 1 or Class 2 is regressed on the covariates, and the latent class affiliation is represented with the uncertainty.

In Step 3 it is also possible to estimate the association between latent classes and distal outcomes such as **schatt** (School Attainment). Since this variable is continuous, we can estimate the average value of this variable across the two latent classes estimated:

**DATA**:

FILE= twoclasses.dat ;

**VARIABLES**:

NAMES= antisc13 antisc14 antisc15 antisc16 id male ses1 ses2 ses3 ses4 ses5 schatt cprob2 cprob2 k;

USEVAR = k male ses2 ses3 ses4 ses5 **schatt**;

NOMINAL = k;

MISSING = all (-999);

CLASSES= newcl(2);

**ANALYSIS**:

TYPE=MIXTURE;

STARTS=0;

**MODEL**:

%OVERALL%

newcl ON male ses2 ses3 ses4 ses5;

%newcl#1%

[k#1 @ 2.295];

**[schatt] (p1);**

%newcl#2%

[k#1 @ -2.775];

**[schatt] (p2);**

**MODEL TEST:**

**p1=p2;**

The statements **[schatt]** in **%newcl#1%** and **%newcl#2%** ask Mplus to estimate the average value of **schatt** for latent class 1 and latent class 2 respectively. By adding a name (p1) and (p2) for these two estimated means, we can use the **MODEL TEST:** command to invoke a Wald test testing the null hypothesis that the mean of **schatt** for latent class 1 (which we labelled p1) is equal to the mean of **schatt** for latent class 2. If the *p* value of the test is <.05, we can reject the null hypothesis and accept p1 ≠ p2. In a similar way, we can also free the variances of the distal outcome to differ across classes, and test hypotheses concerning them.

**References:**

Asparouhov, T., & Muthén, B. (2014). Auxiliary variables in mixture modeling: Three-step approaches using Mplus. *Structural Equation Modeling, 21*, 329–341. doi:10.1080/10705511.2014.915181

Clark, S. L., & Muthén, B. (2009). *Relating latent class analysis results to variables not included in the analysis*. Retrieved from <https://www.statmodel.com/download/relatinglca.pdf>

Nylund-Gibson, K., Grimm, R., Quirk, M., & Furlong, M. (2014). A latent transition mixture model using the three-step specification. *Structural Equation Modeling, 21*, 329–341. doi:10.1080/10705511.2014.915181

Perra, O. (2020). Latent transition analysis. *SAGE Research Methods: Foundations*.