Auxiliary Variables in Mixture Modeling: Using the BCH Method in Mplus to Estimate a Distal Outcome Model and an Arbitrary Secondary Model

Tihomir Asparouhov and Bengt Muthén

Mplus Web Notes: No. 21

Version 8

September 15, 2020

1 Introduction

In mixture modeling, indicator variables are used to identify an underlying latent categorical variable. In many practical applications we are interested in using the latent categorical variable for further analysis and exploring the relationship between that variable and other, auxiliary observed variables. If we use a direct approach where the auxiliary variables are included in the mixture model the latent class variable may have an undesirable shift in the sense that it is no longer measured simply by the original latent class indicator variables but now it is also measured by the auxiliary variables. The shift can be so substantial that the analysis can yield meaningless results because it is no longer based on the original latent class variable.

Different approaches have been proposed recently to remedy this problem and are discussed in detail in Asparouhov and Muthén (2014). Among these are the 3-step approach proposed by Vermunt (2010) and the approach of Lanza et al. (2013). Both of these approaches are implemented in Mplus and the details of that implementation are discussed in Asparouhov and Muthén (2014). It is pointed out in Asparouhov and Muthén (2014) that the 3-step approach does not resolve the problem of shifting classes completely. In some situations when the auxiliary variable is included in the final stage the latent class variable can shift substantially and invalidate the results. Mplus monitors the shift in classes with the 3-stage approach and if this shift is substantial results are not reported. This monitoring is conducted with the automatic Mplus commands DU3STEP and DE3STEP, however, if a manual 3-step approach is conducted the monitoring has to be done manually as well.

Further simulation studies conducted in Bakk and Vermunt (2014) confirm the finding that the 3-step approach fails in certain situations. Bakk and Vermunt (2014) also point out that the approach of Lanza (2013) for distal continuous outcomes, implemented in Mplus with the DCON command, can also fail due to assumptions underlying this method, primarily related to unequal variance across classes. The method yields poor results when the entropy is low and there is a substantial difference between the variances of the distal outcome across classes. If either one of these is not present then Lanza's method works well. With categorical distal outcome Lanza's method has no such drawbacks or any assumptions that can be violated.

A method proposed in Bray et al. (2014) appears to yield results similar to the method in Lanza et al. (2013) for continuous distal outcomes. This method also fails when the distal outcome has unequal variance across classes.

Bakk and Vermunt (2014) also consider in simulation studies the modified BCH method, BCH for short, described in Vermunt (2010) and also in Bakk et al. (2013). For the distal outcome model that evaluates the means across classes for a continuous auxiliary variable these simulations show that the BCH method substantially outperforms Lanza's method and the 3-step method. The BCH method avoids shifts in latent class in the final stage that the 3-step method is susceptible to. In its final stage the BCH method uses a weighted multiple group analysis, where the groups correspond to the latent classes, and thus the class shift is not possible because the classes are known. In addition, the BCH method performs well when the variance of the auxiliary variable differs substantially across classes, i.e., resolving the problems that Lanza's method is susceptible to.

The BCH method uses weights w_{ij} which reflect the measurement error of the

latent class variable. In the estimation of the auxiliary model, the i-th observation in class/group j is assigned a weight of w_{ij} and the auxiliary model is estimated as a multiple group model using these weights. The main drawback of the BCH method is that it is based on weighting the observations with weights that can take negative values. If the entropy is large and the latent class variable is measured without error then the weight w_{ij} is 1 if the i-th observation belongs to class j and zero otherwise. If the entropy is low, however, the weights w_{ij} can become negative and the estimates for the auxiliary model can become inadmissible. For example, it is possible that the variance of the distal outcome is estimated to a negative value or that the frequency table of a categorical auxiliary variable has a negative value. In such cases it would be difficult to utilize the BCH method beyond the basic distal outcome mean comparison model. Bakk and Vermunt (2014) show that the means of a continuous distal outcomes can be estimated correctly even when the sample group specific variances are negative. To obtain an admissible solution the estimated model holds equal the variances across group/class. In this simple model the mean and variance estimates are independent and thus the equal variance restriction has no effect on the mean estimates. However, if one is interested in evaluating the effect of the latent class variable on a more general auxiliary model it is not clear how to resolve the problems with inadmissible solutions due to negative weights.

Note, however, that the negative values in the BCH weights are a normal occurrence and not problematic in general. The negative weights are problematic only when they lead to inadmissible model estimates. In fact, the BCH weights will have negative values for every observation, unless the latent class variable is measured without error. The BCH weights are obtained from the

inverse of a matrix H which can be found in the Mplus output under the heading "Classification Probabilities for the Most Likely Latent Class Membership (Column) by Latent Class (Row)". For each observation the BCH weights are obtained from the j-th row of H^{-1}, where j is the most likely class for that observation. Since H contains only non-negative values, H^{-1} will have negative values, unless H is the identity matrix, representing the case of no classification error. It can also be seen from this computation that the BCH weights for each observation add up to 1.

Two versions of the BCH method are implemented in Mplus. The first version is referred to as the automatic version. This procedure evaluates the mean of a continuous distal outcome variable across classes using the approach of Bakk and Vermunt (2014). In this version one simply specifies the measurement model for the latent class variable and specifies the auxiliary variable as such. The second version is the manual version which allows us to estimate the effect of a latent class variable on an arbitrary auxiliary model. This version requires two separate runs. In the first run we estimate the latent class measurement model and save the BCH weights. In the second run we estimate the general auxiliary model conditional on the latent class variable using the BCH weights. Both BCH versions are illustrated in the next two sections.

2 The automatic BCH approach for estimating the mean of a distal continuous outcome across latent class

This approach is very similar to the DU3STEP and DE3STEP commands in Mplus. With the following input file we estimate a latent class model using the 8 binary indicator variables $U_1, ..., U_8$. We also independently estimate the mean of the auxiliary variable Y across the different classes with the BCH method.

Variable:
Names are U1-U8 Y;
Categorical = U1-U8;
Classes = C(4);
Auxiliary = Y(bch);
Data: file=a1.dat;
Analysis: Type = Mixture;

The model estimates for the latent class model are not affected by the auxiliary variable and the results for the auxiliary variable mean estimates can be located in the output file

EQUALITY TESTS OF MEANS ACROSS CLASSES USING THE BCH PROCEDURE
WITH 3 DEGREE(S) OF FREEDOM FOR THE OVERALL TEST

Y

	Mean	S.E.		Mean	S.E.
Class 1	-1.063	0.079	Class 2	-0.363	0.139
Class 3	1.416	0.096	Class 4	0.295	0.088

	Chi-Square	P-Value		Chi-Square	P-Value
Overall test	420.441	0.000	Class 1 vs. 2	15.023	0.000
Class 1 vs. 3	405.734	0.000	Class 1 vs. 4	99.557	0.000
Class 2 vs. 3	87.209	0.000	Class 2 vs. 4	17.808	0.000
Class 3 vs. 4	60.329	0.000			

3 Using Mplus to conduct the BCH method with an arbitrary secondary model

In many situations it would be of interest to estimate a more advanced secondary model with the BCH method. In the Mplus implementation the secondary model can be an arbitrary model with any number and types of variables. The model is essentially estimated as a multiple group model as if the latent class variable is observed. The BCH method uses group specific weights for each observation that are computed during the latent class model estimation. An outline of the procedure is as follows. First estimate a latent class model using only the latent class indicator variables and save the BCH weights. All variables that will be used in the secondary model should be placed in the auxiliary variable command without any specification. That way the auxiliary variables will be saved in the same file as the BCH weights. This is step 1 of the estimation. In step 2 we simply specify the auxiliary model and we use the BCH weights as training data.

3.1 Regression auxiliary model

In the following example we estimate the auxiliary regression model of a dependent variable Y on a covariate X. We measure a 3-class latent variable using an LCA model with 10 binary items and then use that latent variable to estimate class specific regression Y on X. The example and the data are the same as the example presented on page 332 in Asparouhov and Muthén (2014). In the first step we use the following input file to estimate the LCA model and save the BCH weights

Variable:
Names=U1-U10 Y X;
Categorical = U1-U10;
Classes = C(3);
Usevar=U1-U10;
Auxiliary=Y X;
Data: file=manBCH.dat;
Analysis: Type = Mixture;
Savedata: File= manBCH2.dat; Save=bchweights;

Here the key command is **Save=bchweights;** which requests the BCH weight for further analysis. In the second step the following input file can be used to estimate the class specific regression of Y on X.

Variable:
Names = U1-U10 Y X W1-W3 MLC;

Usevar are Y X W1-W3;

Classes = C(3);

Training=W1-W3(bch);

Data: file=manBCH2.dat;

Analysis: Type = Mixture; Starts=0; Estimator=mlr;

Model:

%overall%

Y on X;

%C#1%

Y on X;

%C#2%

Y on X;

%C#3%

Y on X;

Note that the latent class indicator variables U1-U10 are not on the USEVAR list in this step. The key commands here are **Training=W1-W3(bch);** which specifies the BCH weights to be used in this secondary analysis, **Starts=0;** because this is a multiple group analysis and random starting values are not needed, and **Estimator=mlr;** because that estimator leads to better standard errors becasue the analysis utilizes weights, see Bakk and Vermunt (2014). The results of the auxiliary model estimation are found as usual in the output file of the second step run.

3.2 Regression auxiliary model combined with latent class regression

Distal outcomes are often studied in the presence of covariates so that the effect of the latent class variable on the distal is controlled for by those covariates. This is a variation on the modeling just discussed where the covariate X influences not only Y but also the latent class variable. Following is an illustration of the manual BCH estimation for such a model.

The auxiliary model we are interested in estimating with the BCH method is given by the following two equations

$$Y|C = \alpha_c + \beta_c X$$

$$P(C = c|X) = \frac{Exp(\gamma_{0c} + \gamma_{1c} X)}{\sum_c Exp(\gamma_{0c} + \gamma_{1c} X)}$$

We illustrate this BCH manual estimation with a four class model measured by 8 binary indicators U_i where

$$P(U_i = 1|C) = 1/(1 + Exp(s_{ci}\tau))$$

where $s_{1p} = -1$, $s_{4p} = 1$, $s_{2p} = 1$ for $p = 1, ..., 4$, $s_{1p} = -1$ for $p = 5, ..., 8$, $s_{3p} = -1$ for $p = 1, ..., 4$ and $s_{3p} = 1$ for $p = 5, ..., 8$. We set the value of τ to 1 to generate the data. We generate a single data set of size $N = 50000$ according to the above model. The first step model input is as follows.

Variable:

Names are U1-U8 y x;

Usevar=U1-U18;

Categorical = U1-U8;

Classes = C(4);

Auxiliary=Y X;

Data: file=1.dat;

Analysis: Type = Mixture; starts=0;

Savedata: File= 2.dat; Save=bchweights;

Model:

%Overall%

%c#1%

[U1$1-U8$1*-1.0] ;

%c#2%

[U1$1-U4$1*1.0 U5$1-U8$1*-1.0] ;

%c#3%

[U1$1-U4$1*-1.0 U5$1-U8$1*1.0] ;

%c#4%

[U1$1-U8$1*1.0] ;

Starting values are provided so that the class order does not reverse from the generated order. In real data analysis starting values are not needed. Instead, a large number of random starting value should be set using the **starts** command. The second step input is as follows

Variable:

Names = U1-U8 Y X W1-W4 MLC;

Usevar = Y X W1-W4;

Classes = c(4);

Training=W1-W4(bch);

Data: file=2.dat;

Analysis: Type = Mixture; starts=0;

Model:

%Overall%

C on X;

Y on X;

%c#1%

Y on X;

%c#2%

Y on X;

%c#3%

Y on X;

%c#4%

Y on X;

The results of this simulation are presented in Table 1. All estimates are close to the true parameter values and all but one of them are within the implied confidence limits. Thus we conclude that the manual BCH approach can be used for more complex auxiliary models. If we remove the variable Y from the above

Table 1: Manual BCH estimation

Parameter	True Value	Estimated Value	SE
α_1	0	0.013	0.035
α_2	1	0.984	0.030
α_3	0	0.123	0.042
α_4	2	1.979	0.022
β_1	1	0.964	0.037
β_2	2	2.043	0.047
β_3	-1	-0.910	0.046
β_4	0	-0.005	0.027
$\gamma_1 1$	1	1.004	0.027
$\gamma_1 2$	0.5	0.542	0.029
$\gamma_1 3$	-0.3	-0.246	0.030

example we get an example where the auxiliary variable is a latent class predictor. Thus the BCH manual approach can be used as an alternative to the R3STEP auxiliary command which uses a 3-step estimation approach.

3.3 Regression auxiliary model for categorical distal outcome

In the following example we estimate the auxiliary regression model of a dependent categorical variable Y on a covariate X. We measure a 2-class latent variable using an LCA model with 5 binary items and then use that latent variable to estimate class specific regression of a binary variable Y on X. The auxiliary model is given by the following equation

$$P(Y = 1|C) = \frac{1}{1 + Exp(\tau_c - \beta_c X)}$$

We use the following montecarlo setup to generate data for this illustration

MONTECARLO:

names = y u1-u5 x;

nobs =20000;

nrep = 1;

classes=c(2);

genclasses=C(2);

save=1.dat;

generate=y(1) u1-u5(1);

categorical=y u1-u5;

ANALYSIS: type=mixture;

MODEL POPULATION:

%overall%

y on x*1; x*1;

%C#1%

[u1$1-u5$1*-1];

[y$1*0];

y on x*1;

%C#2%

[u1$1-u5$1*1];

[y$1*1];

y on x*0.2;

MODEL:

%overall%

y on x*1;

%C#1%

[u1$1-u5$1*-1];

[y$1*0];

y on x*1;

%C#2%

[u1$1-u5$1*1];

[y$1*1];

y on x*0.2;

Next we estimate the 2-class LCA model using the 5 binary indicators U_1,..., U_5 and save the BCH weights from this analysis. The Mplus model input is as follows

VARIABLE:

names=y u1-u5 x;

classes=c(2);

usevar=u1-u5;

categorical=u1-u5;

auxiliary=y x;

DATA: file=1.dat;

ANALYSIS: type=mixture;

MODEL:

%overall%

%C#1%

[u1$1-u5$1*-1];

%C#2%

[u1$1-u5$1*1];

savedata: file=2.dat; save=bch;

In the final step we estimate the auxiliary model only using the BCH weights as training data with the following input file

VARIABLE:

names=u1-u5 y x bch1-bch2;

classes=c(2);

usevar=y x bch1-bch2;

categorical=y;

training=bch1-bch2(bch);

DATA: file=2.dat;

ANALYSIS: type=mixture; starts=0;

MODEL:

%overall%

y on x;

%C#1%

y on x;

%C#2%

y on x;

The results of this simulation are presented in Table 2. All estimates are close to the true parameter values and all but one of them are within the implied confidence limits. Thus we conclude that the manual BCH approach can be used for estimating auxiliary models with categorical distal outcomes.

Table 2: Manual BCH estimation for categorical distal regression

Parameter	True Value	Estimated Value	SE
τ_1	0	0.064	0.026
τ_2	1	0.994	0.027
β_1	1	0.986	0.033
β_2	.2	0.186	0.027

4 Simulation study with a continuous distal auxiliary outcome

In this section we extend the simulation studies presented in Section 6.1 of Asparouhov and Muthén (2014) to include the BCH method and the Lanza et al. (2013) method referred to as DCON. For completeness we describe the simulation and include the results already presented in that article.

We estimate a 2-class model with 5 binary indicator variables. The distribution for each binary indicator variable U is determined by the usual logit relationship

$$P(U = 1|C) = 1/(1 + Exp(\tau_c))$$

where C is the latent class variable which takes values 1 or 2 and the threshold value τ_c is the same for all 5 binary indicators. In addition we set $\tau_2 = -\tau_1$ for all five indicators. We choose three values for τ_1 to obtain different level of class separation/entropy. Using the value of $\tau_1 = 1.25$ we obtain an entropy of 0.7, with value $\tau_1 = 1$ we obtain an entropy of 0.6, and with value $\tau_1 = 0.75$ we obtain an entropy of 0.5. The latent class variable is generated with proportions 43% and 57%. In addition to the above latent class model we also generate a normally

Table 3: Distal outcome simulation study: Bias/Mean Squared Error/Coverage

N	Entropy	PC (E)	3-step (DU3STEP)	Lanza (DCON)	1-step	BCH
500	0.7	.10/.015/.76	.00/.007/.95	.00/.006/.92	.00/.006/.94	.00/.007/.94
500	0.6	.16/.029/.50	.01/.008/.94	.00/.007/.89	.00/.007/.94	.01/.008/.94
500	0.5	.22/.056/.24	.03/.017/.86	.00/.012/.80	.01/.012/.96	.03/.017/.86
2000	0.7	.10/.011/.23	.00/.002/.93	.00/.002/.89	.00/.002/.93	.00/.002/.93
2000	0.6	.15/.025/.03	.00/.002/.93	.00/.002/.87	.00/.002/.94	.00/.002/.94
2000	0.5	.22/.051/.00	.00/.004/.91	.00/.003/.80	.00/.003/.94	.00/.004/.91

distributed distal auxiliary variable with mean 0 in class one and mean 0.7 in class 2 and variance 1 in both classes. We apply the pseudo-class method, the 3-step method, Lanza's method, the 1-step method, and the BCH method to estimate the mean of the auxiliary variable in the two classes.

Table 3 presents the results for the mean of the auxiliary variable in class 2. We generate 500 samples of size 500 and 2000 and analyze the data with the five methods. The results in Table 3 show that the BCH procedure and the 3-step procedure have almost identical performance in terms of bias, MSE and coverage. In this simulation the BCH method shows no bias and the coverage is near the nominal level with the exception of the case of low entropy of 0.5 and sample size of 500 where a small bias is observed which also leads to decrease of coverage.

Next we conduct a simulation study to compare the performance of the four different methods DU3STEP, DE3STEP, Lanza's method and the BCH method in the situation when the distal variable variances are different across class. The two 3-step approaches DU3STEP and DE3STEP differ in the third step. The DU3STEP approach estimates different means and variances for the distal variable in the different classes while the DE3STEP approach estimates different means but

Table 4: Distal outcome with unequal variance simulation study: Bias/Mean Squared Error/Coverage

N	Entropy	DE3STEP	DU3STEP	Lanza(DCON)	BCH
500	0.7	.05/.147/.95	.00/.099/.94	.03/.129/.77	.00/.114/.93
500	0.6	.06/.174/.96	.00/.099/.95	.15/.397/.70	.00/.121/.94
500	0.5	.12/.822/.93	.01/.101/.95	1.20/5.755/.46	.04/.160/.94
2000	0.7	.05/.040/.92	.00/.027/.92	.03/.035/.76	.00/.029/.94
2000	0.6	.09/.056/.92	.00/.027/.93	.07/.056/.70	.00/.031/.93
2000	0.5	.11/.094/.95	.00/.029/.92	1.18/4.613/.44	.00/.041/.94

equal variances. The second approach is more robust and more likely to converge but may suffer from the mis-specification that the variances are held equal in the different classes. We use the same simulation as above except that we generate a distal outcome in the second class with variance 20 instead of 1. The results for the mean in the second class are presented in Table 4.

It is clear from these results that the unequal variance 3-step approach (DU3STEP) is superior particularly when the class separation is poor (entropy level of 0.6 or less). The equal variance approach (DE3STEP) can lead to severely biased estimates when the class separation is poor and the variances are different across classes. Lanza's method appears to have completely failed particularly when the class separation is poor. The BCH method appears to be slightly worse than the DU3STEP approach in terms of bias and MSE but the coverage remains good near the nominal level. Thus for the continuous distal variable estimation if the distal variable variances are unequal across class we can recommend only the DU3STEP and the BCH methods.

5 Simulation study with a non-normal distal auxiliary outcome

In Section 7.1 of Asparouhov and Muthén (2014) it was shown that when the distal outcome is not normally distributed the 3-step estimation can fail due to switching of the classes and the parameter estimates maybe severely biased. Further simulations illustrating this point were conducted in Bakk and Vermunt (2014). In this section we conduct a simulation study similar to the those in Bakk and Vermunt (2014).

We estimate and generate data according to a 4 class LCA model with 8 binary indicators. The class proportions are as follows: 0.375, 0.25, 0.1875 and 0.1875. The measurement model is described as follows

$$P(U_p = 1|C) = 1/(1 + Exp(s_{cp}\tau))$$

where $s_{2p} = 1$, $s_{4p} = -1$, $s_{1p} = -1$ for $p = 1, ..., 5$, $s_{1p} = 1$ for $p = 6, ..., 8$, $s_{3p} = 1$ for $p = 1, ..., 5$ and $s_{3p} = -1$ for $p = 6, ..., 8$. We vary the value of τ to obtain different entropy value and class separation. If $\tau = 1.5$ the entropy is 0.7. If $\tau = 1.25$ the entropy is 0.6. If $\tau = 1$ the entropy is 0.5. The distal outcome in class 1 has the following bimodal distribution $0.5N(0, 0.1)+0.5N(-2, 0.1)$, in class two it is also bimodal $0.75N(-2/3, 0.1) + 0.25N(2, 0.1)$, in class 3 it is the normal distribution $N(2, 0.1)$ and in class 4 it is the normal distribution $N(0.5, 0.1)$. We use three different sample sizes N=2000, 5000 and 10000 and generate and analyze 500 replications for each size. In this simulation we can expect that the DU3STEP, DE3STEP, 1-step and PC method to fail due to non-normality and we can expect

Table 5: Non-normal distal outcome simulation study

Method	Bias	MSE	Coverage
DE3STEP	-	-	-
DU3STEP	-	-	-
Lanza	0.663	0.440	0.00
BCH	0.004	0.001	0.89
1-Step	0.647	0.419	0.00
2-Step	0.181	0.036	0.00
PC	0.151	0.024	0.00

Lanza's method to fail due to varying variances across class. We also include in this simulation study the 2-step estimation method proposed in Bakk and Kuha (2018).

In Table 5 we present the results for the distal mean in class 2 for the most favorable case where Entropy=0.7 and $N = 10000$ for all of the estimation methods. No results are presented for the DE3STEP and DU3STEP because in almost all replications there was no convergence due to large differences between the step 1 class allocation and step 3 class allocation. Mplus will not report any results if substantial shift in the classes occur in step 3. The remaining methods fail dramatically as well with the exception of the BCH method. This simple simulation suggest that BCH may indeed be much more robust than any other method.

Next we evaluate the performance of the BCH method for different sample sizes and entropy levels. The results are presented in Table 6. The estimates are unbiased in all cases with small bias being visible for smaller sample sizes and entropy levels. On the other hand the coverage drops substantially particularly when the entropy is low. Also the ratio of the standard errors to the standard

Table 6: Non-normal distal outcome simulation study for the BCH method

N	Entropy	Bias	MSE	Coverage	Std. Err/Std. Dev.
2000	0.7	0.00	0.007	0.89	0.82
5000	0.7	0.00	0.003	0.89	0.83
10000	0.7	0.00	0.001	0.89	0.81
2000	0.6	0.00	0.016	0.80	0.62
5000	0.6	0.00	0.005	0.82	0.66
10000	0.6	0.00	0.003	0.82	0.67
2000	0.5	0.05	0.057	0.58	0.42
5000	0.5	0.01	0.021	0.59	0.43
10000	0.5	0.00	0.010	0.67	0.43

deviation, which should be near 1 for large sample sizes is consistently smaller and it does not improve with increasing the sample size. For example in the last row of Table 6 we see that even when the sample size is 10000 and entropy is 0.5 the ratio is 0.43, i.e., the standard errors are underestimated by 57% and should be nearly twice to what the method currently computes. This has been noted also in Bakk and Vermunt (2014) and has been suggested there that the underestimation occurs due to unaccounted variability of the posterior probabilities that are used as weights in step 3. The BCH method heavily depends on these posterior probabilities and one can expect that this effect is substantial. When the class separation is large the underestimation disappears which also reflects the diminished variability in the posterior probabilities. At this point no reasonable method is available to resolve this shortcoming although bootstrapping would resolve this problem and it can be run in Mplus as external montecarlo where the bootstrap samples are obtained separately.

6 Using the BCH method with multiple latent class variables

In latent transition analysis (LTA), several latent class variables are measured at different time points and the relationship between these variables is estimated through a logistic regression. A multi-step estimation procedure, based on the BCH method, can be conducted for the LTA model where the latent class variables are estimated independently of each other and are formed purely based on the latent class indicators at the particular point in time. Although the BCH method was originally developed for distal outcomes, distal outcomes are not needed in the LTA application. This estimation approach is desirable in the LTA context because the 1-step approach has the drawback that an observed measurement at one point in time affects the definition of the latent class variable at another point in time. We illustrate this estimation with two different examples. The first example is a simple LTA model with three latent class variables. The second example is an LTA model with covariates.

The estimation process is an extension of the manual BCH method. The first step is to save the BCH weights for every latent class variable as discussed in Section 3. The measurement model for each latent class variable is estimated separately and the BCH weights are saved. The final auxiliary model is then estimated where the BCH weights are multiplied together to obtain the joint BCH weights. For example, if there are three latent class variables with 2 classes each, the final model will use $8 = 2 \times 2 \times 2$ BCH weights computed as follows. Let the BCH weights for the first latent class variable be b_{11} and b_{12}, for the second latent class variable be b_{21} and b_{22}, and for the third variable be b_{31} and b_{32}. The

joint BCH weights are computed as the following product

$$d_{ijk} = b_{1i} b_{2j} b_{3k}, \tag{1}$$

where the values of i, j and k are 1 and 2, representing the two classes for each latent class variable. It is important to list the joint BCH weights in the **TRAINING** option in the correct order. In the above example the weights should be listed as d_{111}, d_{112}, d_{121}, d_{122}, d_{211}, d_{212}, d_{221}, d_{222}, i.e., the first index stays constant as the values of the other indices are exhausted and in general the most right indices are exhausted first.

Next, we illustrate this multistage estimation process with two LTA examples. All input files as well as the data generation models can be downloaded at statmodel.com.

6.1 Example 1: LTA model

In the Figures 1-4 we show the input files for estimating an LTA model with 3 binary latent class variables each measured by 4 binary indicators. Figures 1-3 show the input file for estimating the BCH weights for the three latent class variables. Figure 4 shows the input file for the latent transition model. Using the **Auxiliary** option of the **Variable** command we can carry over variables from one file to another so that all variables needed for the final LTA analysis are in the same file. The **STARTS** option setting to 0, combined with the starting values for the measurement model can ensure that the classes do not reverse and appear in the desired order. If the class ordering is not important then the **STARTS** option and the starting values are not needed. If measurement invariance is desired, the

Figure 1: Estimating the BCH weights for C1

```
DATA:  FILE IS 0.dat;
VARIABLE:  NAMES ARE u11-u14 u21-u24 u31-u34;
   CLASSES = c1(2);
   CATEGORICAL = u11-u14;
   auxiliary=u21-u24 u31-u34;
ANALYSIS:  TYPE = MIXTURE;starts=0;
MODEL:
   %OVERALL%
   %c1#1%
   [u11$1-u14$1*-1];
   %c1#2%
   [u11$1-u14$1*1];
OUTPUT:    TECH1 TECH8;
savedata: file is 1.dat; save=bchweights;
```

models in the first 3 figures can be estimated together, specifying independence between the latent class variables. The measurement parameters obtained from the simultaneous estimation will then be used in Figures 1-3 as fixed parameters. Figure 4 shows how the joint BCH weights given in formula (1) can be computed using the **DEFINE** command. The Figure 4 input also shows that we use a variable $U11$, which has its mean and variance held equal cross classes, i.e., it is independent from the LTA model. That is needed in Mplus 8.4 because the analysis unnecessarily requires at least one variable to be observed. In Mplus 8.5 this will not be required when BCH weights are used.

Figure 2: Estimating the BCH weights for C2

```
DATA:  FILE IS 1.dat;
VARIABLE:  NAMES ARE u11-u14 u21-u24 u31-u34 bc11 bc12;
   CLASSES = c2(2);
   CATEGORICAL = u21-u24;
   auxiliary=u11-u14 u31-u34 bc11 bc12;
   usevar=u21-u24;
ANALYSIS:  TYPE = MIXTURE;starts=0;
MODEL:
   %OVERALL%
   %c2#1%
   [u21$1-u24$1*-1];
   %c2#2%
   [u21$1-u24$1*1];
OUTPUT:    TECH1 TECH8;
savedata: file is 2.dat; save=bchweights;
```

Figure 3: Estimating the BCH weights for C3

```
DATA:  FILE IS 2.dat;
VARIABLE:  NAMES ARE u11-u14 u21-u24 u31-u34 bc11 bc12 bc21 bc22;
   CLASSES = c3(2);
   CATEGORICAL = u31-u34;
   auxiliary=u11-u14 u21-u24 bc11 bc12 bc21 bc22;
   usevar=u31-u34;
ANALYSIS:  TYPE = MIXTURE;starts=0;
MODEL:
   %OVERALL%
   %c3#1%
   [u31$1-u34$1*-1];
   %c3#2%
   [u31$1-u34$1*1];
OUTPUT:    TECH1 TECH8;
savedata: file is 3.dat; save=bchweights;
```

Figure 4: Final model estimation for the LTA model

```
DATA:  FILE IS 3.dat;
VARIABLE:  NAMES ARE u31-u34 u11-u14 u21-u24 bc11 bc12 bc21 bc22 bc31 bc32;
   CLASSES = c1(2) c2(2) c3(2);
   usevar=u11 dc111 dc112 dc121 dc122 dc211 dc212 dc221 dc222;
   training=dc111 dc112 dc121 dc122 dc211 dc212 dc221 dc222 (bch);
define:
dc111=bc11*bc21*bc31;
dc112=bc11*bc21*bc32;
dc121=bc11*bc22*bc31;
dc122=bc11*bc22*bc32;
dc211=bc12*bc21*bc31;
dc212=bc12*bc21*bc32;
dc221=bc12*bc22*bc31;
dc222=bc12*bc22*bc32;
ANALYSIS:  TYPE = MIXTURE;starts=0;
MODEL:
   %OVERALL%
   [c1#1*0.4 c2#1*-0.7 c3#1*0.8];
   C2#1 on C1#1*0.5;
   C3#1 on C2#1*-0.3;
   u11 (1); [u11] (2);
OUTPUT:    TECH1 TECH8 tech15;
```

6.2 Example 2: LTA with covariates

Figures 5-7 show the input files for estimating an LTA model with 2 latent class variables with 3 classes each. The latent class variables are also regressed on a covariate. Both latent class variable are measured by 8 binary indicators. Figures 5-6 show the input files for obtaining the BCH weights for the two latent class variables. Figure 7 shows the input file for the final LTA model with a covariate. The joint BCH weights are computed in the **DEFINE** command. Here again the starting values and the **STARTS** option are needed only if a specific order of the latent classes is desired.

Figure 5: Estimating the BCH weights for C1

```
DATA:  FILE IS 0.dat;
VARIABLE:  NAMES ARE u11-u18 u21-u28 x;
   CLASSES = c1(3);
   CATEGORICAL = u11-u18;
   auxiliary=u21-u28 x;
ANALYSIS:  TYPE = MIXTURE;starts=0;
MODEL:
   %OVERALL%
   %c1#1%
   [u11$1-u14$1*-1.3 u15$1-u18$1*-1.3];
   %c1#2%
   [u11$1-u14$1*1.3 u15$1-u18$1*-1.3];
   %c1#3%
   [u11$1-u14$1*1.3 u15$1-u18$1*1.3];
savedata: file is 1.dat; save=bchweights;
```

Figure 6: Estimating the BCH weights for C2

```
DATA:  FILE IS 1.dat;
VARIABLE:  NAMES ARE u11-u18 u21-u28 x bc11 bc12 bc13;
   CLASSES = c2(3);
   CATEGORICAL = u21-u28;
   auxiliary=u11-u18 x bc11-bc13;
   usevar=u21-u28;
ANALYSIS:  TYPE = MIXTURE;starts=0;
MODEL:
   %OVERALL%
   %c2#1%
   [u21$1-u24$1*-1.3 u25$1-u28$1*-1.3];
   %c2#2%
   [u21$1-u24$1*1.3 u25$1-u28$1*-1.3];
   %c2#3%
   [u21$1-u24$1*1.3 u25$1-u28$1*1.3];
savedata: file is 2.dat; save=bchweights;
```

Figure 7: Final model estimation for the LTA model with covariates

```
DATA:  FILE IS 2.dat;
VARIABLE:  NAMES ARE u11-u18 u21-u28 x bc11 bc12 bc13 bc21 bc22 bc23;
   CLASSES = c1(3) c2(3);
   usevar= x dc11 dc12 dc13 dc21 dc22 dc23 dc31 dc32 dc33;
   training=dc11 dc12 dc13 dc21 dc22 dc23 dc31 dc32 dc33 (bch);
define:
dc11=bc11*bc21;
dc21=bc12*bc21;
dc31=bc13*bc21;
dc12=bc11*bc22;
dc22=bc12*bc22;
dc32=bc13*bc22;
dc13=bc11*bc23;
dc23=bc12*bc23;
dc33=bc13*bc23;
ANALYSIS:  TYPE = MIXTURE;starts=0;
MODEL:
   %OVERALL%
   [c1#1*0.4 c1#2*1 c2#1*-0.7 c2#2*0];
   C2#1 on C1#1*0.5;
   C2#1 on C1#2*-0.3;
   C2#2 on C1#1*1.3;
   C2#2 on C1#2*.3;
   C1#1 on x*0.5;
   C1#2 on x*0.1;
   C2#1 on x*-0.5;
   C2#2 on x*-0.1;
```

7 Missing Data

In this section we discuss two practical issues that arise in the application of the BCH and the 3-step estimation methods due to missing data. The first issue we address is how to deal with missing values for latent class predictors. We provide this illustration for the BCH method but the method can also be used with the 3-step estimation. The second issue that we will address is the situation when the measurement model is missing at some of the time points in an LTA analysis. By missing measurement model at a given time point, we mean that all of the latent class indicators at that time point have missing data. We will use the 3-step estimation method for this case.

7.1 Missing values for latent class predictors

In this section we illustrate how to utilize the multiple imputation methodology implemented in Mplus to deal with missing data for latent class predictors. This method is preferable to other alternatives such as montecarlo integration because not only does it avoid heavy numerical integration computations but it can seamlessly deal with different type of covariates that have missing values, i.e., the method can incorporate categorical and continuous predictors in the most optimal way. The multiple imputation method will take advantage of existing correlations in the data to more accurately impute the missing values.

Figure 8 shows how we generate the data for this illustration. We generate data from a two-class model with 5 binary indicators. The latent class variable has 3 predictors: U_0 which is a binary variable, X_1 and X_2 which are continuous variables. Both X_1 and U_0 have missing values and are correlated with X_2 which

does not have any missing values. The MODEL MISSING specified in the input file shows that the probability of missing for X_1 and U_0 depends on the value of X_2, i.e., the generated missing data is MAR, i.e., not MCAR. This is the more challenging type of missing data but likelihood-based methods are generally able to deal with it accurately. The Mplus input here does not have an actual MODEL statement (it only has MODEL POPULATION) because in this case it is not needed to generate the data. Note, however, that in general, before any simulation study is undertaken, we recommend that both MODEL and MODEL POPULATION are used as a preliminary step, using identical models. Such a preliminary step can ensure that latent variables are sufficiently measured, i.e., entropy is in a desired range and can also prevent accidental errors and typos in the model construction. Using identical MODEL and MODEL POPULATION acts as a benchmark for how well a model can be estimated under perfect conditions.

Figure 9 shows the input file that is needed as a first step in this analysis. We estimate the LCA model and we save the BCH weights. The AUXILIARY option here is used to store the predictors in the same file as the BCH weights.

Figure 10 shows the input file for the imputation of the missing values for the covariates. In this stage of the estimation any number of variables can be used to aid the imputation process. Variables that could be connected to the covariates that have missing values should be included. We have included here the BCH weights as well. Since the covariates are related to the latent class variable, that connecting information can be utilized in the imputation process by including the BCH weights in the model. This is essential and if the BCH weights are not included, the final results could be biased. Other variables not related to the LCA model can be included in the imputation as well. The

more variables are included in the imputation process the more accurate the imputation will be. Note, however, that including variables that are not correlated with the covariates will not be helpful and could cause convergence problems in the imputation process. Thus, the choice of which variables to include in the imputation process should be carefully considered. Some general practical guidelines on the imputation methodology are given is Section 4 of Asparouhov and Muthén (2010b). In our example, the BCH weights and the variable X_2 provide essential information on the missing values and are therefore included. The imputation model could potentially use the latent class indicators as well, but this would be useful only if there are direct effects from the covariates to the latent class indicator because otherwise the BCH weights carry all the information of the latent class indicators.

In the IMPUTE option of the DATA IMPUTATION command the categorical variable U_0 is listed with the (c) specification. This tells Mplus to impute this variable as a categorical variable rather than as continuous. As a result of that, all imputed values for U_0 will be categorical, i.e., 0 or 1 in our example. We used 100 imputations in this example, specified in the NDATASETS option. Limited simulation studies indicate that there is a small but important benefit in using a larger number of imputations, rather than the typical choice of 5 imputations. All 100 imputed data sets are saved and ready to be used in the final estimation.

Figure 11 shows the input file that can be used to perform the BCH analysis with multiple imputations. In this model we simply regress the latent class variable on the imputed covariates. All 100 data sets are analyzed and the results are combined according to the multiple imputation rules. Further information on the Multiple imputation methodology can be found in Asparouhov and Muthén

(2010a) and Asparouhov and Muthén (2010b). This input file can include additional MODEL TEST and MODEL CONSTRAINT commands to obtain any particular tests that are needed.

Figure 8: Data generation for LCA with missing values for the latent class predictors

```
MONTECARLO: NAMES ARE u0 u1-u5 x1 x2;
            NOBS =2000;
            NREP = 1;
            save=1.dat;
            classes=c(1);
            genclasses=C(2);
            generate=u0(1) u1-u5(1);
            categorical=u0-u5;
            missing=u0 x1;

model missing:
%overall%
[u0*-1.5 x1*-1.5]; u0 x1 on x2*0.4;

ANALYSIS:   TYPE IS mixture; algo=int; integration=montecarlo;

MODEL POPULATION:
%overall%
[u0$1*0]; u0 on x2*1; x2*1;
[x1*0]; x1 on x2*1; x1*1;
C#1 on u0*0.5 x1*-0.3 x2*0.2;
%C#1%
[u1$1-u5$1*-1.5];
%C#2%
[u1$1-u5$1*1.5];
```

7.1.1 Missing values for latent class predictors used in the first step, i.e., in the LCA measurement model

Suppose that the missing covariate is intended to be used in the first step of the BCH estimation (Figure 9), while in the final step of the BCH estimation

Figure 9: Estimating the LCA model and saving the BCH weights

```
variable:   NAMES ARE u0 u1-u5 x1 x2;
            classes=c(2);
            usevar=u1-u5;
            categorical=u1-u5;
            auxiliary=u0 x1 x2;
            missing=all(999);

data:file=1.dat;

ANALYSIS:   TYPE IS mixture;

MODEL:
%overall%
%C#1%
[u1$1-u5$1*-1.5];
%C#2%
[u1$1-u5$1*1.5];

savedata: file is 2.dat; save=BCHweights;
```

Figure 10: Imputing the missing latent class predictors

```
variable:  NAMES ARE u1-u5 u0 x1 x2 bch1 bch2;
           usevar=u0 x1 x2 bch1 bch2;
           missing=*;

data:file=2.dat;

ANALYSIS:   TYPE=basic; biter=(1000);

DATA IMPUTATION:
   IMPUTE = x1 u0(c);
   NDATASETS = 100;
   SAVE = 2imp*.dat;
   THIN=100;
```

Figure 11: BCH analysis with multiple imputations

```
variable:  NAMES ARE u0 x1 x2 bch1 bch2;
           classes=c(2);
           training=bch1-bch2(bch);

data: file=2implist.dat; type=imputation;

ANALYSIS:   TYPE IS mixture;

MODEL:
%overall%
C on u0 x1 x2;
```

(Figure 11) we have a distal outcome variable Y that is regressed on the latent class variable C, i.e., the means of Y are estimated across the different classes. This situation must be addressed differently. The multiple imputations must be performed prior to step 1, i.e., Figure 10 analysis must be conducted prior to Figure 9.

The multiple imputation process in this case (Figure 10) should include all latent class indicators and the BCH variables will not be used as they are not available yet. The multiple imputation can be done again as an H1 type imputation, i.e., using TYPE=BASIC. Next, the first step estimation, i.e., the Figure 9 analysis where the LCA measurement model is estimated, must be completed for all imputed data sets and the BCH weights must be saved for all imputed data sets. If the number of imputed data sets is M, this would require manually creating M input files which would result in M saved data files that include the BCH weights. The structure of the final step, i.e., Figure 11 analysis, should remain as is (with a different model where the latent class predictor is no longer included as a latent class predictor). Note that for this final step, the 2implist.dat file should be manually created and it should include

all of the M saved data files that include the BCH weights (and not the original multiple imputation files produced from the Mplus multiple imputation). Because this process requires some manual manipulation, the number of imputations M should be set to a lower value, for example 10 or 20.

7.2 Missing measurement model in LTA analysis using the 3-step estimation

In this section, we will consider missing data on all of the latent class indicators at a certain time point in Latent Transition Analysis. We will refer to this as a missing measurement model. We build upon the 3-step LTA estimation described in Section 4 of Asparouhov and Muthén (2014), see also Appendices F-I. We will essentially repeat the 3-step estimation with the added complexity that the measurement model is entirely missing for certain observations at certain time points. In this illustration we use 3 time points instead of 2 time points as it was done in Appendices F-I.

We begin with Figure 12 which describes the input file we use to generate the data. There are 3 latent class variables measured by 5 binary indicators in this LTA analysis. We use this data set and insert missing values for the measurement model as follows. The total sample size in this illustration is 2000. We insert missing values for the measurement model at time point 1 for the first 500 observations. We also insert missing values for the measurement model at time point 2 for the next 500 observations. Finally, we insert missing values for the measurement model at time point 3 for the next 500 observations. The last 500 observations have no missing values at any of the three measurement models.

In the next 7 figures we illustrate the proper and most optimal way to use the 3-step estimation in this context.

Figures 13-15 amounts to simply estimating the LCA at each of the three time points. We will only be using these runs to obtain the error structure for the most likely class variable for each of the three LCA analyses, i.e., we only need the results in the tables "Logits for the Classification Probabilities for the Most Likely Latent Class Membership (Column) by Latent Class (Row)". These values will be used as the nominal variable parameters in the final stage. Note also that in these three runs we are actually not saving the most likely class variables. This will be done separately as a part of the more complex data management that is needed.

Figures 16-18 describe the same models as Figures 13-15 but with an added level of data management that aligns the data in the most suitable way for the final estimation. The models in Figures 13-15 are not suitable for saving the most likely class variables because in these run the missing measurement model observations at a particular time point will be removed and the data sets will be misaligned and will present a challenge to combine. Figures 16-18 use one additional observed variable P. This variable can be any/arbitrary variable which has no missing values. This can for instance be the ID variable. In each LCA model, the parameters for this new variable are held equal across class so that the new variables does not change the measurement model for the latent class variable. The measurement model parameter estimates of Figure 16-18 should match exactly the results obtained with the input files of Figures 13-15. The primary purpose of the new variable P is to prevent Mplus from removing entire observations from the data when the measurement model is missing at the particular time point.

Essentially Figures 16-18 are the same as Figures 13-15 but they are based on the full data set. In the full data set estimation, the data sets are linked across time, meaning that after we run the LCA at time point one, we save the data and proceed to the next time point with the new data set that contains all the variables, including the most likely class variables from the previous time points.

Figure 19 shows the final model where the LTA transition model is estimated and the most likely class variables at each time point are as usual used as measurements for the latent class variables with certain misclassification errors specified in the nominal variables. There are two things to note here. First, the nominal parameters means are obtained from the results of the inputs of Figures 13-15, and not those in Figures 16-18. The second thing to note is that we use the DEFINE statements to specify missing values for the nominal variables for those cases where the measurement model is missing. This uses the _MISSING option described in the Mplus User's Guide, see Muthén and Muthén (1998-2017) page 643. In this example, missing value on the first latent class indicator implies missing values on all the latent class indicators, which implies a missing measurement model and therefore missing value for the most likely latent class variable. If the missing data is more complex and some latent class indicators at a particular time point are missing while other are not, these DEFINE statements must be modified. The most likely latent class variable should be set to missing only if all of the latent class indicators are missing.

Figure 12: Data generation for LTA with 3 time points

```
Montecarlo:
Names are u11-u15 u21-u25 u31-u35;
Generate = u11-u35(1);
Categorical = u11-u35;
Genclasses = c1(2) c2(2) c3(2);
Classes = c1(2) c2(2) c3(2);
Nobservations = 2000;
Nrep = 1;
save=conc3step.dat;

Analysis: Type = Mixture; starts=0;

Model Population:
%Overall%
[c1#1*0.3];
[c2#1*0.3];
[c3#1*0.3];
c2#1 on c1#1*0.5;
c3#1 on c2#1*0.5;

MODEL population-c1:
           %c1#1%
           [u11$1-u15$1*-1];
           %c1#2%
           [u11$1-u15$1*1];

MODEL population-c2:
           %c2#1%
           [u21$1-u25$1*-1];
           %c2#2%
           [u21$1-u25$1*1];

MODEL population-c3:
           %c3#1%
           [u31$1-u35$1*-1];
           %c3#2%
           [u31$1-u35$1*1];
```

Figure 13: Estimating the LCA at time point 1

```
variable:  Names are u11-u15 u21-u25 u31-u35 p;
usevar are u11-u15;
Categorical = all;
Classes = c1(2);
missing=all(999);

data: file=conc3stepM.dat;

Analysis: Type = Mixture; starts=0;

Model:
%Overall%
[c1#1*0.3];
%c1#1%
[u11$1-u15$1*-1];
%c1#2%
[u11$1-u15$1*1];
```

Figure 14: Estimating the LCA at time point 2

```
variable: Names are u11-u15 u21-u25 u31-u35 p;
usevar are u21-u25;
Categorical = all;
Classes = c2(2);
missing=all(999);

data: file=conc3stepM.dat;

Analysis: Type = Mixture; starts=0;

Model:
%Overall%
[c2#1*0.3];
%c2#1%
[u21$1-u25$1*-1];
%c2#2%
[u21$1-u25$1*1];
```

Figure 15: Estimating the LCA at time point 3

```
variable: Names are u11-u15 u21-u25 u31-u35 p;
usevar are u31-u35;
Categorical = all;
Classes = c3(2);
missing=all(999);

data: file=conc3stepM.dat;

Analysis: Type = Mixture; starts=0;

Model:
%Overall%
[c3#1*0.3];
%c3#1%
[u31$1-u35$1*-1];
%c3#2%
[u31$1-u35$1*1];
```

Figure 16: Estimating the LCA at time point 1 on the full data set

```
variable: Names are u11-u15 u21-u25 u31-u35 p;
usevar are u11-u15 p;
Categorical = all;
Classes = c1(2);
auxiliary=u21-u35;
missing=all(999);

data: file=conc3stepM.dat;

Analysis: Type = Mixture; starts=0;

Model:
%Overall%
[c1#1*0.3]; [p$1] (1);
%c1#1%
[u11$1-u15$1*-1];
%c1#2%
[u11$1-u15$1*1];

savedata: file=c1.dat; save=cprob;
```

Figure 17: Estimating the LCA at time point 2 on the full data set

```
variable: Names are u11-u15 p u21-u25 u31-u35 p1 p2 n1;
usevar are u21-u25 p;
Categorical = all;
Classes = c2(2);
auxiliary=u11-u15 u31-u35 n1;
missing=*;

data: file=c1.dat;

Analysis: Type = Mixture; starts=0;

Model:
%Overall%
[c2#1*0.3]; [p$1] (1);
%c2#1%
[u21$1-u25$1*-1];
%c2#2%
[u21$1-u25$1*1];

savedata: file=c2.dat; save=cprob;
```

Figure 18: Estimating the LCA at time point 3 on the full data set

```
variable: Names are u21-u25 p u11-u15 u31-u35 n1 p1 p2 n2;
usevar are u31-u35 p;
Categorical = all;
Classes = c3(2);
auxiliary=u11-u15 u21-u25 n1 n2;
missing=*;

data: file=c2.dat;

Analysis: Type = Mixture; starts=0;

Model:
%Overall%
[c3#1*0.3]; [p$1] (1);
%c3#1%
[u31$1-u35$1*-1];
%c3#2%
[u31$1-u35$1*1];

savedata: file=c3.dat; save=cprob;
```

Figure 19: Estimating the final LTA model

```
variable: Names are u31-u35 p u11-u15 u21-u25 n1 n2 p1 p2 n3;
usevar are n1 n2 n3;
nominal n1 n2 n3;
Classes = c1(2) c2(2) c3(2);
missing=*;

data: file=c3.dat;

Analysis: Type = Mixture; starts=0;

define: if (u11==_MISSING) then N1=_MISSING;
        if (u21==_MISSING) then N2=_MISSING;
        if (u31==_MISSING) then N3=_MISSING;

Model:
%Overall%
[c1#1*0.3 c2#1*0.3 c3#1*0.3];
c2#1 on c1#1*0.5;
c3#1 on c2#1*0.5;

MODEL c1:
%c1#1%
[n1#1@1.975];
%c1#2%
[n1#1@-2.169];

MODEL c2:
%c2#1%
[n2#1@2.083];
%c2#2%
[n2#1@-1.641];

MODEL c3:
%c3#1%
[n3#1@2.241];
%c3#2%
[n3#1@-1.712];
```

8 Summary

Many methods have been proposed in recent years for mixture modeling with auxiliary variables. To clarify the choice of method, Table 6 and 7 list the Mplus options, give their intended use, and give recommendations on which method should be used for which purpose.

Table 7: Alternative auxiliary settings for mixture modeling

	DU3STEP
Useage:	Continuous distal outcomes
Description; reference:	Classification-error corrected; Vermunt (2010) and Asparouhov-Muthén (2014)
Pros and cons:	Susceptible to class changes. Mplus will not report results if the class formation changes. Manual version also available for an arbitrary auxiliary model, including controlling for covariates. Estimates unequal distal variances across classes.
Recommendation:	Preferred method for continuous distal outcomes Use when Mplus reports results, i.e., there are no class formation changes, otherwise use BCH.
	BCH
Useage:	Continuous distal outcomes
Description; reference:	Measurement-error weighted; Bakk and Vermunt (2014)
Pros and cons:	Avoids class changes. Avoids the DCON shortcomings with class-varying variances for distals. Manual version also available for an arbitrary auxiliary model, including controlling for covariates. Possible SE underestimation with low entropy.
Recommendation:	Preferred method for continuous distal outcomes
	DCAT
Useage:	Categorical distal outcomes
Description; reference:	Distal treated as covariate; Lanza et al. (2013)
Pros and cons:	Avoids class changes
Recommendation:	Preferred method for categorical distal outcomes
	R3STEP
Useage:	Covariates
Description; reference:	Classification-error corrected; Vermunt (2010)
Pros and cons:	Works well
Recommendation:	Recommended method with covariates

Table 8: Alternative auxiliary settings for mixture modeling, continued

	DE3STEP
Useage:	Continuous distal outcomes. Equal distal variances across classes
Description; reference:	Classification-error corrected; Vermunt (2010) and Asparouhov-Muthén (2014)
Pros and cons:	Susceptible to class changes and class-varying variances. Mplus will not report results if the class formation changes.
Recommendation:	Inferior to BCH and DU3STEP. Use only when DU3STEP does not converge.
	DCON
Useage:	Continuous distal outcomes
Description; reference:	Distal treated as covariate; Lanza et al. (2013) and Asparouhov-Muthén (2014)
Pros and cons:	Avoids class changes. Sensitive to class-varying variances for distals when entropy is low
Recommendation:	Inferior to BCH and DU3STEP when DU3STEP does not change the class formation. Use only when entropy is higher than 0.6. If variance appears to be varying across class more than a factor of 2 do not use this method. This check can be done using most likely class assignment - it is not done automatically by Mplus. Use only for methods research purposes
	E
Useage:	Continuous distal outcomes
Description; reference:	Pseudo-class (PC) method; Wang et al. (2005)
Pros and cons:	Gives biased results
Recommendation:	Superseded by BCH and DU3STEP. Use only for methods research purposes
	R
Useage:	Covariates
Description; reference:	Pseudo-class (PC) method; Wang et al. (2005)
Pros and cons:	Gives biased results
Recommendation:	Superseded by R3STEP. Use only for methods research purposes

References

[1] Asparouhov T. & Muthén B. (2010a). Chi-square statistics with multiple imputation. https://www.statmodel.com/download/MI7.pdf

[2] Asparouhov T. & Muthén B. (2010b). Multiple imputation with Mplus. http://www.statmodel.com/download/Imputations7.pdf

[3] Asparouhov T. & Muthén B. (2014). Auxiliary variables in mixture modeling: Three-step approaches using Mplus. Structural Equation Modeling: A Multidisciplinary Journal, 21, 329-341. Online Appendicies: http://statmodel.com/download/AppendicesOct28.pdf

[4] Bakk, Z. and Kuha, J. (2018). Two-step estimation of models between latent classes and external variables. Psychometrika, 83, 871–892.

[5] Bakk, Z. and Vermunt, J.K. (2015). Robustness of stepwise latent class modeling with continuous distal outcomes. Structural Equation Modeling: A Multidisciplinary Journal, 23, 20-31.

[6] Bakk, Z., Tekle, F.B., & Vermunt, J.K. (2013). Estimating the association betwen latent class membership and external variables using bias adjusted three-step approaches. In T.F. Liao (ed.), Sociological Methodology. Thousand Oake, CA: SAGE publications.

[7] Bray, B.C., Lanza, S. T. & Tan, X. (2014) Eliminating Bias in Classify-Analyze Approaches for Latent Class Analysis. Structural Equation Modeling: A Multidisciplinary Journal. 22, 1-11.

[8] Lanza S. T., Tan X., & Bray B. C. (2013). Latent Class Analysis With Distal Outcomes: A Flexible Model-Based Approach. Structural Equation Modeling, 20, 1-26.

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

[10] Vermunt, J. K. (2010). Latent Class Modeling with Covariates: Two Improved Three-Step Approaches. Political Analysis, 18, 450-469.

[11] Wang C.P., Brown, C.H., Bandeen-Roche, K. (2005). Residual diagnostics for growth mixture models: Examining the impact of preventive intervention on multiple trajectories of aggressive behavior. Journal of the American Statistical Association, 100 (3), 1054-1076.