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 11

February 4, 2021

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. Among these are the 3-step approach proposed by Vermunt (2010), the approach of Lanza et al. (2013) and the 2-step estimation method proposed in Bakk and Kuha (2018). All of these approaches are available in Mplus. These methods follow the same general pattern. First, the latent class measurement model is estimated. Then, a follow up analysis determines the relationship between the latent class variable and the auxiliary variables.

The details of the Mplus implementation of the 3-step approach and Lanza's approach are discussed in Asparouhov and Muthén (2014). It is also shown 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 must 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 can also fail. We illustrate below that if the distal outcome is conditionally correlated to the latent class indicators (conditional on the latent class), the estimates obtained with Lanza's method can be biased.

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 as shown in Figure 1.

Figure 1: BCH output

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 Simulation study with a categorical distal auxiliary outcome

In this section we compare the BCH method and the Lanza et al. (2013) method for the case when the distal auxiliary outcome is a categorical variable. Lanza's method in this case is obtained in Mplus with the option AUXILIARY = Y(DCAT), where Y is the name of the auxiliary variable. In Section 4 we showed that Lanza's method fails for a continuous distal outcome when the variance of the outcome is not class invariant. For categorical outcomes, however, the variance parameter does not exist and therefore there is no reason to suspect that Lanza's method would fail in that case as well. In addition, Lanza's estimation method for categorical distal outcomes is much more robust than it is for continuous distal outcomes, because it is based on estimating an unconstrained contingency table for the latent class variable and the distal outcome. It turns out, however, that Lanza's method is prone to failures even for categorical distal outcomes. There are two reasons for that. First, the method is based on including the distal outcome as a predictor in the LCA measurement model. It is well understood, however, that including a predictor in the LCA measurement model can yield a distortion of the latent class formation, particularly when there are direct effects from the predictor to the latent class indicators. Such a distortion will inevitably results in a distortion of the contingency table estimates and from there in the distal outcome final results. The second reason for the failure is the assumption of conditional independence. Lanza's method assumes that the LCA indicators and the distal outcome are independent conditional on the latent class variable. Such an assumption, however, is often violated in practical

settings. Furthermore, establishing conditional independence is not an easy task and generally would involve the joint estimation of the LCA measurement model and the distal outcome, which is precisely what we are trying to avoid with the auxiliary modeling.

To illustrate these considerations with a simulation study, we utilize Mplus User's Guide example 7.4, where a 2-class LCA model is measured by 4 binary indicators. We use a large single data set with N=10000 observations so that the results we obtain are indicative of the asymptotic behavior of the estimators. The model parameters are set as in Mplus User's Guide example 7.4.

The auxiliary variable that we use in this study is set to be one of the binary indicators of the LCA, i.e., this binary indicator is used both as an indicator and also as a distal outcome. Thereby, we create the violation of conditional independence. Such a choice for the auxiliary variable allows us to evaluate the performance of the estimators precisely when the conditional assumption underlying Lanza's method is violated. We use 4 different estimations in this illustration. The first method is Lanza's method via the DCAT implementation in Mplus. The second method is the BCH method where the binary auxiliary variable is treated as continuous. Such a method is reasonable for the case when the auxiliary variable is binary because $E(Y) = P(Y = 1)$ when the binary variable is 0/1. This method is estimated in Mplus with the option AUXILIARY = Y(BCH). The third method is the manual BCH method where the BCH weights are saved in the LCA measurement model estimation and are subsequently used to estimate the distal outcome model where the auxiliary variable is treated as a categorical variable. This method can more generally be used when the categorical auxiliary variable has more than two categories. The fourth method is the manual

Table 7: Simulation study for a categorical distal outcome: estimates for $P(Y = 1)$

Class	True value	Lanza (DCAT)	BCH automated (continuous)	BCH manual (categorical)	Manual 3-step (categorical)
1	0.88	1.00	0.93	0.93	0.94
2	0.12	0.00	0.10	0.10	0.9

3-step method, as in Asparouhov and Muthén (2014), where the distal outcome is treated as a categorical variable.

The results of the simulation study are given in Table 7. The two BCH methods yield identical results and outperform Lanza's method. The simulation study confirms that Lanza's method fails when the conditional independence between the latent class indicators and the auxiliary outcome is violated. Note also that the BCH automatic approach can be used only with binary auxiliary variables. For auxiliary variables with more than 2 categories only the manual BCH method applies. The manual 3-step method yields results similar to the BCH method.

7 Using the BCH method for models that require numerical integration

In Mplus Version 8.5, the BCH method has also been implemented for Mixture models that require numerical integration. Examples of such models are growth mixture models with categorical data (Muthén and Asparouhov; 2007), item response mixture models (Muthén and Asparouhov; 2007), and random effect LCA models (Qu et al.; 1996). No new theoretical issues arise because of the numerical integration and the Mplus language and steps are identical to

implementation for models without numerical integration. Both, the automatic and the manual approach are implemented. For the manual BCH, numerical integration can be used in the latent class measurement model. Currently, numerical integration can not be used in the auxiliary model. Thus, in the first step of the manual BCH approach, where the BCH weights are saved, we can include ANALYSIS: ALGORITHM=INTEGRATION if the latent class measurement model requires such a specification. However, in the second step of the BCH approach, where the BCH weights are used as training data, the option ANALYSIS: ALGORITHM=INTEGRATION should not be present, even if the first step required it.

We illustrate the BCH method with numerical integration using a model discussed in Qu et al. (1996). The model is an LCA model with 5 binary indicators measuring a latent class variable with 2 classes. In the first class, two of the indicators are not independent, conditional on the latent class variable, and the residual correlation between the two indicators is modeled via a latent factor. The existence of this conditional non-independence between LCA indicators is generally know as a conditional independence violation. One way to resolve the violation is to introduce a continuous latent variables in the LCA model, which requires numerical integration. The model is given by the following equation

$$P(U_j = 0|C = c) = \frac{1}{1 + Exp(-\tau_{cj} + \lambda_{cj}\eta)}, \tag{1}$$

where τ_{cj} are the threshold parameters and λ_{cj} are the loadings parameters for the latent variable η. In this example, $\lambda_{cj} = 0$ except for $\lambda_{11} = \lambda_{12} = 1$. The variance of η is estimated only in class 1 and the mean of η is fixed to 0 in

both classes. A distal outcome Y is regressed on the latent class variable C. In the Mplus language, such a regression is specified via having class specific means for the distal variable Y. To conduct a simulation study, where the distal outcome regression is estimated with the automatic BCH method, we can use the input file specified in Figure 2. In the input file the MODEL POPULATION command gives all the parameters as expected, however, the MODEL command is slightly intricate. To conduct the simulation study, we specify the Y variable as AUXILIARY with the (BCH) specification. In addition, the Y variable must be removed from the measurement model for the latent class variable, so that the latent class variable is measured only by the categorical indicators. To do that, we specify a model for Y that makes the variable independent of the LCA model. This is accomplished by holding the mean and the variance of Y equal across classes. The LCA model is then estimated as if Y is not in the model at all. With this model specification, Mplus automatically estimates the LCA model, essentially ignoring the Y variable, computes the BCH weights, and then performs the regression of Y on C using these weight in a subsequent estimation. This is repeated over all the replications. The results of the simulation study are reported in Figure 3.

The input file necessary to conduct the BCH estimation for a distal outcome with a single data set, i.e., not in a Montecarlo study, is given in Figure 4. The distal outcome variable is listed in the AUXILIARY option using the (BCH) specification. Note that the variable Y is not in the model at all. The results of this analysis are shown in Figure 5.

Figure 2: Montecarlo simulation for BCH with numerical integration: distal outcome regressed on a latent class variable

```
montecarlo:
 names are u1-u5 y;
 genclasses = c(2);
 classes = c(2);
 generate = u1-u5(1);
 categorical = u1-u5;
 nobs = 1500;
 nrep = 100;
 auxiliary=Y(BCH);

analysis:
 type = mixture; algo=int;

model population:
 %OVERALL%
 [c#1*0.4];
 Y*1;
 f by u1-u2@0; [f@0];  f@0;
 %c#1%
 [u1$1-u5$1*-1.2]; [Y*-1];
 f by u1-u2@1; f*0.3; [f@0];
 %c#2%
 [u1$1-u5$1*1.2]; [Y*1];

model:
 %OVERALL%
 [c#1*0.4];
 Y*1;
 f by u1-u2@0; [f@0]; f@0;
 [y] (1); Y (2);
 %c#1%
 [u1$1-u5$1*-1.2];  [f@0];
 f by u1-u2@1; f*0.3; [f@0];
 %c#2%
 [u1$1-u5$1*1.2]; [f@0];
```

Figure 3: Montecarlo output for BCH with numerical integration

```
EQUALITY TESTS OF MEANS ACROSS CLASSES USING THE BCH PROCEDURE
WITH 1 DEGREE(S) OF FREEDOM FOR THE OVERALL TEST

Y

Number of successful replications        100

                          ESTIMATES              S. E.     M. S. E.  95%  % Sig
              Population   Average   Std. Dev.   Average             Cover Coeff

 Class 1          -1.000    -1.0011     0.0486     0.0412     0.0024 0.920 1.000
 Class 2           1.000     1.0012     0.0643     0.0586     0.0041 0.940 1.000

            Average     Average  Rejection
          Chi-Square    P-Value       Rate

 Overall test  692.179     0.0000     1.0000
```

Figure 4: BCH with numerical integration

```
variable:
  names are u1-u5 y;
  classes = c(2);
  categorical = u1-u5;
  auxiliary=Y(BCH);

data: file=1.dat;

 analysis:
  type = mixture; algo=int;

 model:
  %OVERALL%
  [c#1*0.4];
  f by u1-u2@0; [f@0]; f@0;
  %c#1%
  [u1$1-u5$1*-1.2];
  f by u1-u2@1; f*0.3;
  %c#2%
  [u1$1-u5$1*1.2];
```

Figure 5: BCH with numerical integration results

```
EQUALITY TESTS OF MEANS ACROSS CLASSES USING THE BCH PROCEDURE
WITH 1 DEGREE(S) OF FREEDOM FOR THE OVERALL TEST

Y
                           Mean       S.E.

 Class 1                 -1.038      0.034
 Class 2                  0.988      0.056

                     Chi-Square     P-Value

 Overall test         1109.478      0.000
```

8 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

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

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 right-most indices are exhausted first.

Prior to Mplus Version 8.5, the BCH weights had to be computed in separate files and then the joint BCH weights had to be computed either manually or through the DEFINE command in Mplus. This made the process rather cumbersome. In Mplus Version 8.5, the computation of the joint BCH weights is simplified substantially and can now be done with a single Mplus run. The run must contain the measurement model for each latent class and nothing else. That way the measurement models remain independent of each other and only the latent class indicators at the particular time point affect the latent class formation.

Next, we illustrate the BCH-LTA methodology with several examples.

8.1 Example 1: LTA model

In Figures 6-7 we show the input files for estimating an LTA model with 3 binary latent class variables each measured by 4 binary indicators. Figure 6 shows the input file for estimating the joint BCH weights for the three latent class variables. Figure 7 shows the input file for the latent transition model. The **STARTS** option setting to 0, combined with the starting values for the measurement models ensures 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 class specific models in Figure 6 can be constrained to be equal across time. Holding the parameters equal across time will not compromise the independent formation for the latent class variables, i.e., each latent class variable would be measured solely by the indicators at that point. In some particular situations, it may actually be helpful to estimate each latent class separately as a preliminary step. For example, if a large setting is used for the **STARTS** option, in order to find the best latent class solution for each time point, it would be computationally more efficient if this is done for each time point separately (assuming there is no measurement invariance constraint). The option **OUTPUT:SVALUES** can be used in these time specific models to get perfect starting values for the input in Figure 6.

Figure 6: Estimating the joint BCH weights for C1,C2,C3

```
DATA:  FILE IS 0.dat;
VARIABLE:  NAMES ARE u11-u14 u21-u24 u31-u34;
           categorical=all;
           CLASSES = c1(2) c2(2) c3(2);
ANALYSIS:  TYPE = MIXTURE;starts=0;

model c1:
   %c1#1%
   [u11$1-u14$1*-1.3];
   %c1#2%
   [u11$1-u14$1*1.3];

model c2:
   %c2#1%
   [u21$1-u24$1*-1.3];
   %c2#2%
   [u21$1-u24$1*1.3];

model c3:
   %c3#1%
   [u31$1-u34$1*-1.3];
   %c3#2%
   [u31$1-u34$1*1.3];

savedata: file=bch.dat; save=bchweights;
```

Figure 7: Final model estimation for the LTA model: the transition model

```
DATA:  FILE IS bch.dat;
VARIABLE:  NAMES ARE u11-u14 u21-u24 u31-u34 w1-w8;
   usevar=w1-w8;
   CLASSES = c1(2) c2(2) c3(2);
   training=w1-w8(bch);
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;
```

8.2 Example 2: LTA with covariates

Figures 8-9 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. Figure 8 shows the input file for obtaining the joint BCH weights for the two latent class variables. The starting values and the zero setting of the **STARTS** option are needed only if a specific order of the latent classes is desired. Typically, a large number of STARTS is needed. The covariate X is specified in the auxiliary command. This has no estimation implications. The effect of the specification is that the variable is included in the SAVEDATA file for use in the next step. Without it, the variable will have to be manually added to the file. Figure 9 shows the input file for the final LTA model with the covariate used as a predictor for both latent class variables. Note that in Figure 9 the **STARTS** option is set to 0. At this point in the estimation, when the BCH weights are already determined, generally there is no need for generating random starting values. The estimation in Figure 9 is similar to a multiple group estimation, where random starting values are rarely used or needed. In practical applications, random starting values are needed in determining the LCA models in Figure 8, regardless of whether these LCA models are estimated simultaneously or not.

Figure 8: Estimating the joint BCH weights for C1 and C2

```
DATA:  FILE IS 0.dat;
VARIABLE:  NAMES ARE u11-u18 u21-u28 x;
   CLASSES = c1(3) c2(3);
   usevar = u11-u18 u21-u28;
   CATEGORICAL = all;
   auxiliary=x;
ANALYSIS:  TYPE = MIXTURE;starts=0;

MODEL C1:
   %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];

MODEL C2:
   %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 BCH.dat; save=bchweights; format=free;
```

Figure 9: Final model estimation for the LTA model with covariates

```
DATA:  FILE IS BCH.dat;

VARIABLE:  NAMES ARE u11-u18 u21-u28 x w1-w9;
   CLASSES = c1(3) c2(3);
   usevar= x w1-w9;
   training=w1-w9(bch);

ANALYSIS:  TYPE = MIXTURE;starts=0;

MODEL:
   %OVERALL%
   C2 on C1 X; C1 on X;
```

8.3 Example 3: LTA with a distal outcome

In this section we illustrate the BCH-LTA estimation for a distal outcome Y. An LTA model is estimated in the first step. In the second step we estimate the mean of Y in every pattern/combination of latent class values. The first step in this estimation is accomplished as in the previous section Figure 8. The second step of the estimation is illustrated in Figure 10. Note here that the latent transition part of the model is estimated in Figure 10. It is possible, however, to include the C2 on C1 regression in Figure 8 and exclude it from Figure 10. Either approach is valid and should produce the same distal outcome result.

Figure 10: Final model estimation for the LTA model with a distal outcome

```
DATA:  FILE IS BCH.dat;

VARIABLE:  NAMES ARE u11-u18 u21-u28 y w1-w9;
   CLASSES = c1(3) c2(3);
   usevar= y w1-w9;
   training=w1-w9(bch);

ANALYSIS:  TYPE = MIXTURE;starts=0;

MODEL:
   %OVERALL%
   C2 on C1;
   %C1#1.C2#1%
   [y];
   %C1#1.C2#2%
   [y];
   %C1#1.C2#3%
   [y];
   %C1#2.C2#1%
   [y];
   %C1#2.C2#2%
   [y];
   %C1#2.C2#3%
   [y];
   %C1#3.C2#1%
   [y];
   %C1#3.C2#2%
   [y];
   %C1#3.C2#3%
   [y];
```

9 Simplified 3-step estimation for LTA

Two illustrations are provided in Asparouhov and Muthén (2014) for the 3-step estimation with multiple latent class variables. Input files are provided in the online appendices. Appendices G, H and I illustrate the 3-step estimation for a simple LTA model where the auxiliary model is the transition analysis model. Appendices K, L, M and N illustrate the 3-step estimation for an LTA analysis with measurement invariance and a covariate where the auxiliary model is the transition model which also includes the covariate. Starting with Mplus version 8.6, a simplified implementation is available which reduces the estimation for these illustrations down to just two input files. In the first file all of the LCA measurement models are estimated simultaneously and independently of each other. The most likely latent class variables N_i are saved in that estimation, where $i = 1, 2$ is the index for the latent class variables. The logits for each of the nominal indicators N_i are also computed in that step and are printed in the output file. The final step remains unchanged, apart from the NAMES option in the VARIABLE command which needs to identify the correct columns for the N_i variables.

This simplified approach also illustrates that in the Mplus implementation of the 3-step method there are essentially only 2 steps. The middle step in the 3-step procedure is essentially incorporated in the first step because the logits for the nominal variables are computed automatically.

9.1 Replacement for appendices G, H and I

The two input files that replace these 3 appendices are given in Figures 11 and 13. In Figure 11, the input file simply estimates the two LCA measurement models and saves the most likely latent class variables. The output file from this estimation contains the logits needed for the final estimation, see Figure 12. The input file for the final step is given in Figure 13 and it is almost identical to Appendix I. The only change is in the NAMES option. The results from this simplified approach are identical to those obtained with Appendices G, H and I.

Figure 11: Input file for 3-step LTA analysis, step 1: estimating LCA for C_1 and C_2

```
variable:
Names are u11-u15 u21-u25;
Categorical = u11-u15 u21-u25;
Classes = c1(2) c2(2);
data:file=conc3step.dat;

Analysis: Type = Mixture; Starts=100 20;

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

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

savedata: file is 1.dat; save=cprob;
```

Figure 12: Output file from step 1: locating the nominal indicator logits needed for the final step

```
C-SPECIFIC CLASSIFICATION RESULTS

Classification Quality for C1

Logits for the Classification Probabilities for the Most Likely Latent Class Membership (Column)
by Latent Class (Row)

             1        2

   1      1.864    0.000
   2     -2.138    0.000

Classification Quality for C2

Logits for the Classification Probabilities for the Most Likely Latent Class Membership (Column)
by Latent Class (Row)

             1        2

   1      1.841    0.000
   2     -1.842    0.000
```

Figure 13: Input file for 3-step LTA analysis, final step: estimating the transition model

```
variable:
Names are u11-u15 u21-u25 p1-p4 n1 n2 n;
usevar are n1 n2;
nominal n1 n2;
Classes = c1(2) c2(2);

data: file=1.dat;

Analysis: Type = Mixture; starts=0;

Model:
%Overall%
    c2#1 on c1#1;

MODEL c1:
    %c1#1%
    [n1#1@1.864];
    %c1#2%
    [n1#1@-2.138];

MODEL c2:
    %c2#1%
    [n2#1@ 1.841];
    %c2#2%
    [n2#1@-1.842];
```

9.2 Replacement for appendices K, L, M and N

The two input files that replace these 4 appendices are given in Figures 14 and 15. Figure 14 is essentially identical to Appendix K. The only difference is that we also save the most likely latent class variables with the command SAVEDATA: FILE=1.DAT; SAVE=CPROB. This figure is also nearly identical to Figure 11. The only difference here is that we estimate the LCA measurement models under the assumption of measurement invariance by holding the threshold parameters equal in the two LCA models. Essentially this input file accomplishes the same task that appendices K, L, and M were designed for, i.e., we estimate the LCA measurement models with measurement invariance, save the most likely latent class variables N_i, and obtain the logits for these indicators to be used in the final step. Figure 15 is essentially identical to Appendix N and is the final step in the estimation. The results obtained with the simplified approach described in Figures 14 and 15 are identical to the results obtained with appendices K, L, M and N.

Figure 14: Input file for 3-step LTA estimation with measurement invariance: step 1.

```
variable:
Names are u11-u15 u21-u25 x;
Categorical = u11-u15 u21-u25;
Classes = c1(2) c2(2);
auxiliary=x;

data:file=conc3step.dat;

Analysis: Type = Mixture; Starts=100 20;

MODEL c1:
   %c1#1%
   [u11$1-u15$1*-1] (t1-t5);
   %c1#2%
   [u11$1-u15$1*1] (tt1-tt5);

MODEL c2:
   %c2#1%
   [u21$1-u25$1*-1] (t1-t5);
   %c2#2%
   [u21$1-u25$1*1] (tt1-tt5);

savedata: file is 1.dat; save=cprob;
```

Figure 15: Input file for 3-step LTA estimation with measurement invariance: final step

```
variable:
Names are u11-u15 u21-u25 x p1-p4 n1 n2;
usevar are n1 n2 x;
nominal n1 n2;
Classes = c1(2) c2(2);

data: file=1.dat;

Analysis: Type = Mixture; starts=0;

Model:
%Overall%
c2 on c1 x; c1 on x;

MODEL c1:
   %c1#1%
   [n1#1@1.925];
   %c1#2%
   [n1#1@-2.020];

MODEL c2:
   %c2#1%
   [n2#1@1.787];
   %c2#2%
   [n2#1@-2.084];
```

10 Auxiliary modeling for the RI-LTA model

The RI-LTA model has been discussed in details in Muthén and Asparouhov (2020). The model is an extension of the traditional LTA model, where in addition to the multiple latent class variables, the categorical indicators are correlated through a time invariant continuous latent variable, i.e., the random intercept (RI). The analysis involves both multiple latent class variables and numerical integration. Since both of these features are now supported by the BCH method, it is now possible to use the method with the RI-LTA model to estimate auxiliary models such as distal outcomes and latent class regression on covariates.

A key feature of the RI-LTA model is that the model can separate the correlation that is due to observations nested within subject (RI based correlation) and the correlation that is due to observations taken in proximity of time (the LTA implied correlation). This separation is key for the latent class formation. Therefore, unlike the case of the standard LTA model, for the RI-LTA model, it is not possible to estimate separate time-specific LCA models, for the purpose of obtaining the BCH weights. To obtain the BCH weights, the RI-LTA model must be estimated in its entirety, i.e., including the RI latent variable, all latent class variables and the latent transition model. This observation applies not just to the BCH method but also to the 3-step method (Asparouhov and Muthén, 2014) and the 2-step method (Bakk and Kuha, 2018), where step 2 fixes the measurement parameters to the estimates from step 1. With all multistage estimation methods, in the first step, the entire RI-LTA model must be estimated. If some of the components of the RI-LTA model are not included, the correlation separation will be distorted and the latent class formation will be incorrect.

The RI-LTA model can be viewed as a multilevel model. Thus, auxiliary model estimation for the RI-LTA model provides an illustration for the complexities that can be expected in the auxiliary model estimation for multilevel mixture models, see Asparouhov and Muthén (2008) and Asparouhov, Hamaker and Muthén (2017). However, unlike the multilevel models, the RI-LTA models are typically used with a small number of time points, i.e., when the clusters sizes are small. In a multilevel model, where the cluster sizes are 20 or more, it would be possible in principle to incorporate the multilevel part of the model to reflect the cluster specific latent class measurement error, i.e., with cluster specific BCH weights or with cluster specific logits for the 3-step estimation. In the RI-LTA model, however, due to the small cluster size, such adjustments are unlikely to yield stable estimation and are not pursued here.

Next, we consider the possibility to apply the 3-step method to the RI-LTA model in Mplus 8.5. The most likely latent class variables can be saved in the first step. In the second step, however, the computation of the logits must be performed manually. This computation involves the marginalization of the multivariate latent class distribution table, which can be quite large for larger number of time points. In that case, the manual computation will be prohibitive. On the other hand, the BCH method and the 2-step method are easy to implement.

10.1 RI-LTA with auxiliary covariate

In this section we conduct a simulation study for an auxiliary latent class predictor in the context of the RI-LTA model. We compare the BCH estimation method, the 2-step estimation as well as the 1-step estimation. Simulation studies for

the 2-step method are not automated easily in Mplus at this point because the second step input file changes in every replication. In this simulation, we manually repeated the 2-step estimation for 20 replications and summarized the results outside of Mplus. For the BCH method as well as the 1-step method, we used 100 replications.

Figure 16 shows the input file for generating the data for this RI-LTA simulation study. The model has $T = 3$ time points. A minimum of 3 time points is generally recommended for the RI-LTA model. At each time point t, 8 binary indicators U_{tj} measure a binary latent class variable C_t. The RI factor f is measured by all binary indicators U_{tj}. The covariate X effect on the latent class variables is given by the following 3 equations

$$P(C_1 = 1|X) = \frac{Exp(\alpha_1 + \beta_1 X)}{1 + Exp(\alpha_1 + \beta_1 X)} \tag{3}$$

$$P(C_2 = 1|X, C_1) = \frac{Exp(\alpha_2 + \gamma_{2,C_1} + \beta_2 X)}{1 + Exp(\alpha_2 + \gamma_{2,C_1} + \beta_2 X)} \tag{4}$$

$$P(C_3 = 1|X, C_2) = \frac{Exp(\alpha_3 + \gamma_{3,C_2} + \beta_3 X)}{1 + Exp(\alpha_3 + \gamma_{3,C_2} + \beta_3 X)}. \tag{5}$$

For identification purposes, $\gamma_{2,2}$ and $\gamma_{3,2}$ are fixed to 0. The latent transition model contains 8 parameters: α_t and β_t for $t = 1, 2, 3$ as well as $\gamma_{2,1}$ and $\gamma_{3,1}$. Using the model parameter values given in Figure 16, we obtain a medium size entropy of 0.66. For higher entropy levels, such as 0.8 or above, the different estimation methods are expected to be nearly identical as latent class measurement error becomes negligible. All estimation methods would be similar to the multiple-group (known class) estimation where the latent class measurement error is zero.

To obtain the 1-step estimation results, we augment the Figure 16 input file

with a MODEL statement, which is identical to the MODEL POPULATION statement. The MODEL statement must also include parameter constraints to ensure time-invariance for the threshold and the loading parameters as in Figures 17.

As usual, the BCH estimation is conducted in 2 steps. In the first step, we estimate the same model as in the 1-step method but with β_t fixed to 0. The joint BCH weights are saved and used in the second step, which estimates the latent transition model (3-5).

The input files for the 2-step estimation are given in Figures 17 and 18. The first step, given in Figures 17, is simply the RI-LTA estimation without the covariate. As usual, the starting values of Figure 17 are not needed. This step is identical to first step in the BCH method. In the 2-step estimation, however, instead of saving the BCH weights, we use the OUTPUT: SVALUES option to obtain the model needed for the second step. Replacing the * symbol with the @ symbol in the model statements obtained with the OUTPUT: SVALUES option, we obtain the second step input file given in Figures 18. In this model, all parameters are held fixed to their first step estimates with the exception of the parameters used in equations (3-5) which are estimated as unconstrained parameters. For brevity, Figures 18 includes only MODEL C1. MODEL C2 and MODEL C3 are identical to MODEL C1 but are based on the binary latent class indicators for C2 and C3.

The results of the simulation study are reported in Table 8. All three method perform fairly well, however, some larger bias is noticeable in the results for the BCH method. This bias also results in a drop in the coverage rates. The BCH method appears to perform well in evaluating the effect of the covariate X

Table 8: RI-LTA with covariate: Absolute Bias(Coverage)

Parameter	BCH	2-step	1-step
α_1	.00(.91)	.01(.95)	.01(.93)
α_2	.07(.79)	.02(1.00)	.00(.98)
α_3	.00(.81)	.00(.95)	.01(.93)
β_1	.01(.97)	.02(.90)	.00(.96)
β_2	.04(.94)	.01(1.00)	.01(.99)
β_3	.01(.93)	.02(1.00)	.00(.96)
γ_{21}	.12(.72)	.02(.95)	.00(.96)
γ_{31}	.03(.78)	.05(.80)	.01(.94)

through the parameters β_t. The bias for those parameters is negligible and the coverage is near the nominal levels. Nevertheless, the results in Table 8 indicate that the 2-step method is preferable for the auxiliary estimation in the RI-LTA model with covariates. The 2-step estimation used here does not include the multistage standard error adjustment discussed in Bakk and Kuha (2018). The 2-step coverage rates appear to be near the nominal level. This indicates that the effect of the standard error adjustment must be negligible and that the 2-step method performs well even without the adjustment.

Figure 16: Data generation for RI-LTA model with a covariate

```
montecarlo:
   names are u11-u18 u21-u28 u31-u38 x;
   genclasses = c1(2) c2(2) c3(2);
   classes = c1(2) c2(2) c3(2);
   generate = u11-u38(1);
   categorical = u11-u38;
   nobs = 2000;
   nrep = 100;
   repsave=all;
   save = r*.dat;

analysis: type = mixture; algo=int;

model population:
   %overall%

   [c1#1*0.4 c2#1*-0.7 c3#1*0] ;
   C2#1 on C1#1*0.5;
   C3#1 on C2#1*0.5;
   C1#1 on x*0.5;
   C2#1 on x*-0.5;
   C3#1 on x*0.3;
   x*1;
   f by u11-u38*0.6; f@1; [f@0];

model population-c1:
   %c1#1%
   [u11$1-u18$1*-1.1];
   %c1#2%
   [u11$1-u18$1*1.1];

model population-c2:
   %c2#1%
   [u21$1-u28$1*-1.1];
   %c2#2%
   [u21$1-u28$1*1.1];

model population-c3:
   %c3#1%
   [u31$1-u38$1*-1.1];
   %c3#2%
   [u31$1-u38$1*1.1];
```

Figure 17: Step 1 in 2-step estimation for RI-LTA model with a covariate

```
DATA:  FILE IS a1.dat;
VARIABLE:  NAMES ARE u11-u18 u21-u28 u31-u38 x;
   CLASSES = c1(2) c2(2) c3(2);
   CATEGORICAL = u11-u18 u21-u28 u31-u38;
   auxiliary=x;
ANALYSIS:  TYPE = MIXTURE;starts=0;algo=int;
MODEL:

   %overall%

   f by u11-u18*0.6 (L1-L8);
   f by u21-u28*0.6 (L1-L8);
   f by u31-u38*0.6 (L1-L8);
   f@1; [f@0];
   c2 on c1;
   c3 on c2;

model c1:
   %c1#1%
   [u11$1-u18$1*-1.1] (t11-t18);
   %c1#2%
   [u11$1-u18$1*1.1] (t21-t28);

model c2:
   %c2#1%
   [u21$1-u28$1*-1.1] (t11-t18);
   %c2#2%
   [u21$1-u28$1*1.1] (t21-t28);

model c3:
   %c3#1%
   [u31$1-u38$1*-1.1] (t11-t18);
   %c3#2%
   [u31$1-u38$1*1.1] (t21-t28);

output:svalues;
```

Figure 18: Step 2 in 2-step estimation for RI-LTA model with a covariate

```
DATA:  FILE IS a1.dat;
VARIABLE:  NAMES ARE u11-u18 u21-u28 u31-u38 x;
   CLASSES = c1(2) c2(2) c3(2);
   CATEGORICAL = u11-u18 u21-u28 u31-u38;
ANALYSIS:  TYPE = MIXTURE;starts=0;algo=int;
MODEL:

   %overall%

F  BY
U11@0.686 U12@0.575 U13@0.67 U14@0.588
U15@0.669 U16@0.689 U17@0.681 U18@0.63
U21@0.686 U22@0.575 U23@0.67 U24@0.588
U25@0.669 U26@0.689 U27@0.681 U28@0.63
U31@0.686 U32@0.575 U33@0.67 U34@0.588
U35@0.669 U36@0.689 U37@0.681 U38@0.63;

   f@1; [f@0];
   c1 on x;
   c2 on c1 x;
   c3 on c2 x;

 MODEL C1:
    %C1#1%
    [ u11$1@-1.14625 ] (t11);
    [ u12$1@-1.08893 ] (t12);
    [ u13$1@-1.08516 ] (t13);
    [ u14$1@-1.19674 ] (t14);
    [ u15$1@-1.25197 ] (t15);
    [ u16$1@-1.18936 ] (t16);
    [ u17$1@-1.13704 ] (t17);
    [ u18$1@-1.18284 ] (t18);
    %C1#2%
    [ u11$1@1.13309 ] (t21);
    [ u12$1@1.06080 ] (t22);
    [ u13$1@1.02622 ] (t23);
    [ u14$1@1.09370 ] (t24);
    [ u15$1@1.05472 ] (t25);
    [ u16$1@1.05815 ] (t26);
    [ u17$1@1.02328 ] (t27);
    [ u18$1@1.03855 ] (t28);
```

10.2 RI-LTA with a distal outcome

As shown earlier, see Table 5, the BCH method is the only reliable method for distal outcomes estimation even for simple mixture models. In this section, we study the performance of the BCH method for a distal outcome in the context of the RI-LTA model. We also describe the steps necessary to conduct a Monte Carlo simulation study for the BCH method and an arbitrary auxiliary model.

We use a setup similar to the one used in the previous section: the RI-LTA model has 3 time points and at each time point a binary latent class variable is measured by 8 binary indicators. All of the binary indicators also measure a time invariant latent factor. The latent class predictor in this setup is replaced by a concinnous distal outcome Y. In Section 8.3, the distal outcome is predicted by all latent class patterns. Here we use a simpler setup where only the last latent class variable affects the distal outcome. The input file needed for the data generation is given in Figure 19. The simulation study utilizes the framework of Mplus external montecarlo, where the data is generated separately from the analysis of the data. Figure 19 input file generates 100 data sets with the names r1.dat, r2.dat, etc; as well as a rlist.dat file which contains the names of all generated data sets. The next step in the montecarlo simulation is to estimate the RI-LTA model without the distal outcome and save the BCH weights for each of the 100 data sets. The input file for estimating the RI-LTA model and saving the BCH weights is given in Figure 20. Because this step must be performed 100 times, it should be automated. On the windows operating system, one way to do that is with a batch file. A batch file is simply a text file with DOS commands, saved with the extension .BAT and shown in Figure 21. The input file Step1.inp, referred to

in Figure 21, is the input file given in Figure 20. The BAT file runs a DOS loop, where the Mplus program is run 100 times, using the generated data files r1.dat, r2.dat etc. and saves the files with the BCH weights in the files b1.dat, b2.dat etc.

The final step of the Montecarlo study is to estimate the auxiliary model where the distal outcome variable is regressed on the last latent class variable. The input file for this step is given in Figure 22. Here all 100 data sets b1.dat, b2.dat etc are analyzed and are summarized automatically by Mplus. The names of the data sets must be listed in a file with the name blist.dat. The easiest way to construct this file is to simply use a copy of the existing rlist.dat file and replace the letter "r" with the letter "b". The results of the simulation study are given in Figure 23 and show that the BCH method performs well for the distal outcome estimation for the RI-LTA model.

To conduct this analysis with a single data set, only steps 1 and 2 are needed, i.e., Figures 20 and 22. Typically, the STARTS option will be used in Step 1 to ensure that the best solution is found for the RI-LTA model. For example, STARTS=200 40 can be used in Step 1. In Step 2, random starting values are generally not needed, i.e., the STARTS=0 setting remains unchanged. In Step 2, the DATA command must be changed to DATA: FILE IS B.DAT and TYPE=MONTECARLO must be deleted, so that a single data set is analyzed. Starting values for the parameters can be omitted in both steps.

Figure 19: Data generation for RI-LTA model with a distal outcome

```
montecarlo:
   names are u11-u18 u21-u28 u31-u38 y;
   genclasses = c1(2) c2(2) c3(2);
   classes = c1(2) c2(2) c3(2);
   generate = u11-u38(1);
   categorical = u11-u38;
   nobs = 2000;
   nrep = 100;
   repsave=all;
   save = r*.dat;

analysis: type = mixture; algo=int;

model population:
   %overall%

   [c1#1*0.4 c2#1*-0.7 c3#1*0] ;
   C2#1 on C1#1*0.5;
   C3#1 on C2#1*0.5;
   y*1;
   f by u11-u38*0.6; f@1; [f@0];

model population-c1:
   %c1#1%
   [u11$1-u18$1*-1.1];
   %c1#2%
   [u11$1-u18$1*1.1];

model population-c2:
   %c2#1%
   [u21$1-u28$1*-1.1];
   %c2#2%
   [u21$1-u28$1*1.1];

model population-c3:
   %c3#1%
   [u31$1-u38$1*-1.1];
   [y*0];
   %c3#2%
   [u31$1-u38$1*1.1];
   [y*1];
```

Figure 20: Step 1: Estimating RI-LTA model and saving the BCH weights

```
DATA:  FILE IS r.dat;
VARIABLE:  NAMES ARE u11-u18 u21-u28 u31-u38 y;
   CLASSES = c1(2) c2(2) c3(2);
   CATEGORICAL = u11-u18 u21-u28 u31-u38;
   auxiliary=y;
ANALYSIS:  TYPE = MIXTURE;starts=0;algo=int;
MODEL:

   %overall%

   f by u11-u18*0.6 (L1-L8);
   f by u21-u28*0.6 (L1-L8);
   f by u31-u38*0.6 (L1-L8);
   f@1; [f@0];
   c2#1 on c1#1*0.5;
   c3#1 on c2#1*0.5;

model c1:
   %c1#1%
   [u11$1-u18$1*-1.1] (t11-t18);
   %c1#2%
   [u11$1-u18$1*1.1] (t21-t28);

model c2:
   %c2#1%
   [u21$1-u28$1*-1.1] (t11-t18);
   %c2#2%
   [u21$1-u28$1*1.1] (t21-t28);

model c3:
   %c3#1%
   [u31$1-u38$1*-1.1] (t11-t18);
   %c3#2%
   [u31$1-u38$1*1.1] (t21-t28);

savedata: file is b.dat; save=bch;
```

Figure 21: Estimating RI-LTA model and saving the BCH weights for multiple files using a batch file

```
for /l %%i in (1, 1, 100) do (
  copy r%%i.dat r.dat
  Mplus Step1.inp
  copy b.dat b%%i.dat
)
```

Figure 22: Step 2: Estimating the distal outcome variable in the RI-LTA model

```
DATA:  FILE IS blist.dat; type=montecarlo;
VARIABLE:  NAMES ARE u11-u18 u21-u28 u31-u38 y w1-w8;
   CLASSES = c1(2) c2(2) c3(2);
   usevar= y w1-w8;
   training=w1-w8(bch);
ANALYSIS:  TYPE = MIXTURE;starts=0;
MODEL:
   %OVERALL%
   C2 on C1;
   C3 on C2;
model C3:
   %C3#1%
   [y*0];
   %C3#2%
   [y*1];
```

Figure 23: Montecarlo results for the distal outcome variable in the RI-LTA model

		ESTIMATES		S. E.	M. S. E.	95%	% Sig
	Population	Average	Std. Dev.	Average		Cover	Coeff
Parameters for Class-specific Model Parts of C3							
Latent Class C3#1							
Means							
Y	0.000	0.0029	0.0400	0.0361	0.0016	0.930	0.070
Latent Class C3#2							
Means							
Y	1.000	0.9941	0.0400	0.0410	0.0016	0.950	1.000

11 Missing Data

In this section we discuss 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. First, we consider

the case when the latent class regression on the predictors in estimated in the last stage of the estimation. One possible approach to deal with the missing predictors is to impute the missing values with the Bayesian methodology. We illustrate this approach using the BCH method. The same approach can be applied also in the 3-step estimation.

In Mplus Version 8.5, the Bayesian estimator has been expanded and now includes nominal variables as well as the possibility to regress a latent class variable on covariates that have missing values, see Asparouhov and Muthén (2020). These new features create the possibility to resolve the missing data problem, simply by estimating the last step in the 3-step estimation using the Bayesian estimator. Such an approach is simpler than the multiple imputation approach and we illustrate that method as well. Note, however, that this simplified approach is not available for the BCH method because the Bayesian estimation is not available with weights. The method applies only to the 3-step estimation.

A somewhat different situation arises when the missing latent class covariates are indented to be used as a part of the latent class measurement model, i.e., in the first step in the estimation. This situation can also be resolved with the multiple imputation methodology but the approach is slightly more intricate than when the covariates are used in the last step. The approach is illustrated below using the BCH methodology.

Another issue that we discuss here 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 use the 3-step estimation for this illustration.

11.1 Missing values for latent class predictors: BCH with Multiple Imputations

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 24 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 25 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 26 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 27 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 24: 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];
```

Figure 25: 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 26: 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 27: 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;
```

11.2 Missing values for latent class predictors: 3-step estimation with Bayesian third step

In this section we describe a completely different method for resolving the problem with the missing latent class predictors. The method can be used with the 3-step estimation and is described as follows. The manual 3-step procedure is implemented, where the third step is estimated with the Bayesian estimator instead of the ML estimator. We illustrate this method using the example discussed in the previous section. The first step in the 3-step estimation is as in Figure 25, where instead of saving the BCH weights, we save the most likely latent class variable using the option SAVEDATA: FILE=2.DAT; SAVE=CPROB. The input file for the third stage is given in Figure 28. This input file is identical to the input file that would be used if there were no missing values for the covariates. There is one change: all the covariates are correlated with each other. The impact of this statement is two-fold. First, the covariates are now treated as dependent variables that are modeled. A model for the covariates is needed, so that the missing values can be modeled and imputed internally as part of the Bayesian

Figure 28: Using the Bayesian estimator to deal with missing covariates in 3-step estimation

```
variable:   NAMES ARE u1-u5  u0 x1 x2 p1 p2 N;
            classes=c(2);
            usevar=N  u0 x1 x2;
            nominal=N;
            missing=*;

data:file=2.dat;

ANALYSIS:   TYPE IS mixture; estimator=bayes;

MODEL:
%overall%
C on u0 x1 x2;
u0 x1 x2 with u0 x1 x2;
%C#1%
[N#1@3.134];
%C#2%
[N#1@-2.864];
```

estimation. Second, correlating all the covariates gives an unrestricted model that will be used for internally imputing the missing values. This is important in those situations when the covariates are correlated with each other and some observed covariates can be used to imply/impute more accurately the missing values. Using the Figure 28 input file in the above example, we obtain results nearly identical to the results obtained with the multiple imputation method described in the previous section.

It is important to note here that the method used in Figure 28 treats all covariates as continuous variables. The method assumes a multivariate normal distribution for all covariates. This is not ideal when some of the covariates are categorical. The multiple imputation methodology described in the previous section can most properly treat categorical covariates. However, we expect that in most practical

situations, the results obtained by the two methods would be quite close. Unless the amount of missing data is substantial, it would be difficult for distributional misspecifications to manifest into bias for the parameter estimates.

11.3 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 25), while in the final step of the BCH estimation (Figure 27) 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 26 analysis must be conducted prior to Figure 25.

The multiple imputation process in this case (Figure 26) 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 25 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 27 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.

11.4 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 29 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 30-32 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 33-35 describe the same models as Figures 30-32 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 30-32 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 33-35 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 33-35 should match exactly the results obtained with the input files of Figures 30-32. 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 33-35 are the same as Figures 30-32 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 36 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 30-32, and not those in Figures 33-35. 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 29: 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 30: 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 31: 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 32: 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 33: 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 34: 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 35: 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 36: 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];
```

12 Summary

Many methods have been proposed in recent years for mixture modeling with auxiliary variables. To clarify the choice of method, Tables 9 and 10 list the Mplus options, give their intended use, and give recommendations on which method should be used for which purpose.

Table 9: Alternative auxiliary settings for mixture modeling

	BCH
Useage:	Continuous and categorical 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 and binary distal outcomes. Preferred method for non-binary categorical distal outcomes via the manual BCH
	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:	Good method for continuous distal outcomes Use when Mplus reports results, i.e., there are no class formation changes, otherwise use BCH.
	R3STEP
Useage:	Covariates
Description; reference:	Classification-error corrected; Vermunt (2010)
Pros and cons:	Works well
Recommendation:	Recommended method with covariates
	2-STEP
Useage:	Covariates
Description; reference:	Bakk, Z. and Kuha, J. (2018)
Pros and cons:	Works well
Recommendation:	Recommended method with covariates, particularly with multiple latent class variables

Table 10: 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.
	DCAT
Useage:	Categorical distal outcomes
Description; reference:	Distal treated as covariate; Lanza et al. (2013)
Pros and cons:	Avoids class changes. Automated Mplus analysis, as compared to the manual BCH.
Recommendation:	Use only if conditional independence can be verified
	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 or for methods research purposes. 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.
	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. (2008). Multilevel mixture models. In G. R. Hancock & K. M. Samuelsen, K. M. (Eds.). Advances in Latent Variable Mixture Models. Charlotte, NC: Information Age Publishing, Inc.

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

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

[4] 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

[5] Asparouhov, T., Hamaker, E. L., & Muthén, B. (2017). Dynamic latent class analysis. Structural Equation Modeling, 24, 257–269. doi: 10.1080/10705511.2016.1253479

[6] Asparouhov T. & Muthén B. (2020). Expanding the Bayesian Structural Equation, Multilevel and Mixture Models to Logit, Negative-Binomial and Nominal Variables. http://statmodel.com/download/PGpaper.pdf

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] Muthén, B., & Asparouhov, T. (2006). Item response mixture modeling: Application to tobacco dependence criteria. Addictive Behaviors, 31, 1050–1066.

[13] Muthén, B., & Asparouhov, T. (2007). Growth mixture analysis: Models with non-Gaussian random effects. Forthcoming in Fitzmaurice, G., Davidian, M., Verbeke, G., & Molenberghs, G. (eds.), Advances in Longitudinal Data Analysis. Chapman & Hall/CRC Press.

[14] Muthén, B., & Asparouhov, T. (2020). Latent transition analysis with random intercepts (RI-LTA). Psychological Methods. Advance online publication. https://doi.org/10.1037/met0000370

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

[16] Qu T., Tan M., & Kutner M.H. (1996). Random-effects models in latent class analysis for evaluating accuracy of diagnostic tests. Biometrics, 52, 797–810.

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

[18] 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.