Obtaining Diagnostic Classification Model Estimates with Mplus

Syntax and Output Handout

NCME Diagnostic Measurement Workshop

To demonstrate how Diagnostic Classification Models (DCMs) can be estimated using Mplus, input syntax and output are included in this document.

Analysis Data Set – Based on ECPE Data from Session 3 (simulated so it is distributable):

- 28 items
- Three attributes
- 2,922 respondents

Analysis/ECPE Q-matrix:

Item	Attribute 1	Attribute 2	Attribute 3
1	1	1	0
2	0	1	0
3	1	0	1
4	0	0	1
5	0	0	1
6	0	0	1
7	1	0	1
8	0	1	0
9	0	0	1
10	1	0	0
11	1	0	1
12	1	0	1
13	1	0	0
14	1	0	0
15	0	0	1
16	1	0	1
17	0	1	1
18	0	0	1
19	0	0	1
20	1	0	1
21	1	0	1
22	0	0	1
23	0	1	0
24	0	1	0
25	1	0	0
26	0	0	1
27	1	0	0
28	0	0	1

To create Mplus syntax, we will follow several steps:

1. 1. Creating a Latent Class-to-Attribute Profile Table
    - Mplus uses generic latent classes in estimation – we must make these into DCM attribute profiles.
2. 2. Creating an Item-to-Profile Table
    - Specifies the form of the LCDM for each combination of item and profile. It will be used to build model syntax.
3. 3. Creating Item Response Function Labels
    - Defines the set of unique item response functions for each item (based on the set of attributes measured by the Q-matrix). Used by Mplus to specify LCDM parameters.
4. 4. Creating the Structural Model
    - For limiting the number of correlational parameters needed; aids in estimation speed and convergence.
5. 5. Specifying Initial Mplus Syntax
    - Commands for input and parsing of data files.
6. 6. Building Mplus MODEL Command Syntax
    - Places labels on all Mplus parameters (thereby enforcing confirmatory model on classes).
7. Building Mplus MODEL CONSTRAINT Command Syntax
    - Specifies LCDM model parameters.

1. Creating a Latent Class-to-Attribute Profile Table

Mplus uses latent classes to model categorical data generally. We must represent each possible attribute profiles as a latent class. For DCMs that measure *A* attributes, a total of 2^A attribute profiles are possible. It is our job to map each profile onto a latent class. To do so, we use the following process. Recall our example data set specifics:

Number of attributes in example: 3

Number of possible *attribute profiles*: $2^3 = 8$

Number of latent classes needed: 8

First, we need a table that has the number of latent classes represented in the rows and the number of attributes represented in the columns.

The following figure shows how to populate the classes with attribute profiles through a sequence of partitions of the table:

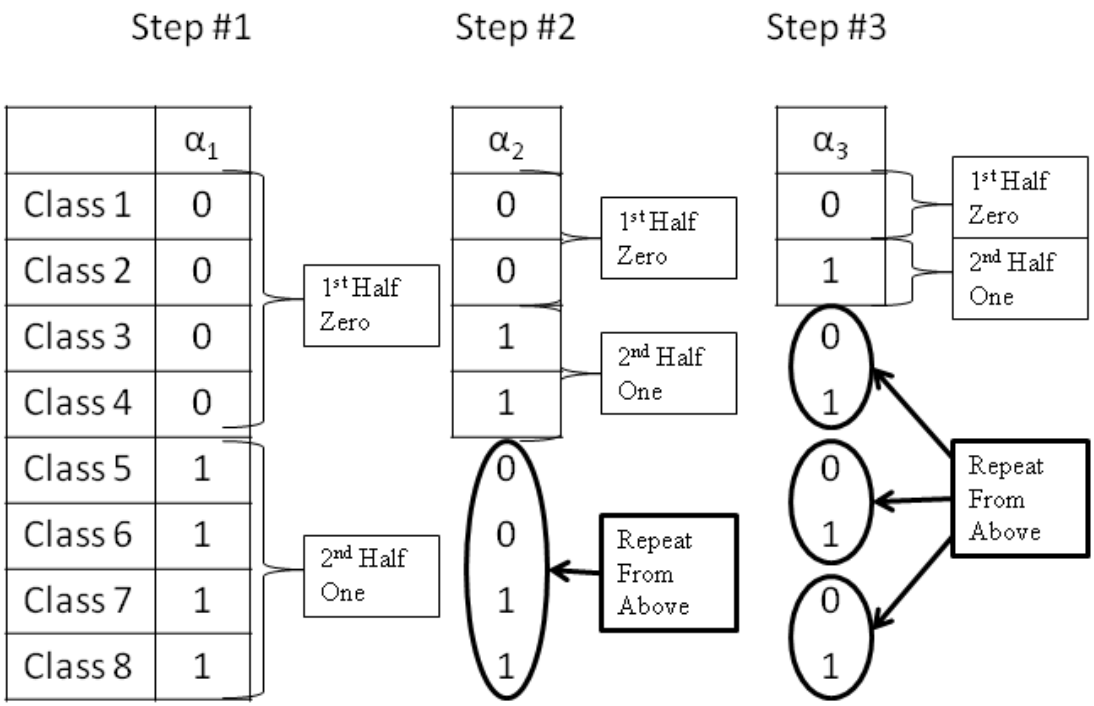

Figure 9.7 from Rupp, Templin, & Henson (2010). Creating a class-to-profile table.

2. Creating an Item-to-Profile Table

The next step is to develop the LCDM model specification for each combination of items and attribute profiles, the item response function for respondents with a given attribute profile. In DCMs, not all attributes are measured by each item. Therefore, there will be repeated item response functions. This table will help to denote which are redundant and therefore able to be omitted. We will use the Q-matrix to help form what model parameters should be present for each combination of item and attribute profile.

Analysis Q-matrix (rearranged to reflect items measuring same set of attributes):

Items	Attribute 1	Attribute 2	Attribute 3	Set Number
10, 13, 14, 25, 27	1	0	0	1
2, 8, 23, 24	0	1	0	2
4, 5, 6, 9, 15, 18, 19, 22, 26, 28	0	0	1	3
1	1	1	0	4
3, 7, 11, 12, 16, 20, 21	1	0	1	5
17	0	1	1	6

Item-to-Profile Table (first subscript denotes set – change for each item number)

Class	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
α_c	[0, 0, 0]	[0, 0, 1]	[0, 1, 0]	[0, 1, 1]	[1, 0, 0]	[1, 0, 1]	[1, 1, 0]	[1, 1, 1]
Set 1	$\lambda_{1,0}$	$\lambda_{1,0}$	$\lambda_{1,0}$	$\lambda_{1,0}$	$\lambda_{1,0} + \lambda_{1,1,(1)}$	$\lambda_{1,0} + \lambda_{1,1,(1)}$	$\lambda_{1,0} + \lambda_{1,1,(1)}$	$\lambda_{1,0} + \lambda_{1,1,(1)}$
Set 2	$\lambda_{2,0}$	$\lambda_{2,0}$	$\lambda_{2,0} + \lambda_{2,1,(2)}$	$\lambda_{2,0} + \lambda_{2,1,(2)}$	$\lambda_{2,0}$	$\lambda_{2,0}$	$\lambda_{2,0} + \lambda_{2,1,(2)}$	$\lambda_{2,0} + \lambda_{2,1,(2)}$
Set 3	$\lambda_{3,0}$	$\lambda_{3,0} + \lambda_{3,1,(3)}$	$\lambda_{3,0}$	$\lambda_{3,0} + \lambda_{3,1,(3)}$	$\lambda_{3,0}$	$\lambda_{3,} + \lambda_{3,1,(3)}$	$\lambda_{3,0}$	$\lambda_{3,0} + \lambda_{3,1,(3)}$
Set 4	$\lambda_{4,0}$	$\lambda_{4,0}$	$\lambda_{4,0} + \lambda_{4,1,(2)}$	$\lambda_{4,0}+\lambda_{4,1,(2)}$	$\lambda_{4,0}+\lambda_{4,1,(1)}$	$\lambda_{4,0} + \lambda_{4,1,(1)}$	$\lambda_{4,0} + \lambda_{4,1,(1)} + \lambda_{4,1,(2)} + \lambda_{4,2,(1,2)}$	$\lambda_{4,0} + \lambda_{4,1,(1)} + \lambda_{4,1,(2)} + \lambda_{4,2,(1,2)}$
Set 5	$\lambda_{5,0}$	$\lambda_{5,0} + \lambda_{5,1,(3)}$	$\lambda_{5,0}$	$\lambda_{5,0}+\lambda_{5,1,(3)}$	$\lambda_{5,0}+\lambda_{5,1,(1)}$	$\lambda_{5,0} + \lambda_{5,1,(1)} + \lambda_{5,1,(3)} + \lambda_{5,2,(1,3)}$	$\lambda_{5,0} + \lambda_{5,1,(1)}$	$\lambda_{5,0} + \lambda_{5,1,(1)} + \lambda_{5,1,(3)} + \lambda_{5,2,(1,3)}$
Set 6	$\lambda_{6,0}$	$\lambda_{6,0} + \lambda_{6,1,(3)}$	$\lambda_{6,0} + \lambda_{6,1,(2)}$	$\lambda_{6,0} + \lambda_{6,1,(2)} + \lambda_{6,1(3)} + \lambda_{6,2,(2,3)}$	$\lambda_{6,0}$	$\lambda_{6,0} + \lambda_{6,1,(3)}$	$\lambda_{6,0} + \lambda_{6,1,(2)}$	$\lambda_{6,0} + \lambda_{6,1,(2)} + \lambda_{6,1(3)} + \lambda_{6,2,(2,3)}$

3. Creating Item Response Function Labels

The next step in the process is to label each unique item response function for each *item* in the Item-to-Profile table. *In the table below, we use the first number to represent the set of items from the Item-to-Profile table. In code, this will change to be the item number.* The labeling convention we will use will follow the form of t[i]_[#]:

- t represents that label is a threshold (Mplus definition for LCDM item response function)
- [i] is the item used (omit brackets)
- [#] is the index for unique item response function for an item.

The labels are created by reading left-to-right for each row of the Item-to-Profile table. The first entry for the first cell of the first item receives the label t1_1, as do all other entries that are the same. The next unique entry receives the label t1_2, as do all other entries that look the same. The process continues until all labels have been assigned.

Item Response Function Labels

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
α_c	[0,0,0]	[0,0,1]	[0,1,0]	[0,1,1]	[1,0,0]	[1,0,1]	[1,1,0]	[1,1,1]
set 1	T1_1	T1_1	T1_1	T1_1	T1_2	T1_2	T1_2	T1_2
set 2	T2_1	T2_1	T2_2	T2_2	T2_1	T2_1	T2_2	T2_2
set 3	T3_1	T3_2	T3_1	T3_2	T3_1	T3_2	T3_1	T3_2
set 4	T4_1	T4_1	T4_2	T4_2	T4_3	T4_3	T4_4	T4_4
set 5	T5_1	T5_2	T5_1	T5_2	T5_3	T5_4	T5_3	T5_4
set 6	T6_1	T6_2	T6_3	T6_4	T6_1	T6_2	T6_3	T6_4

4. Creating the Structural Model [Optional]

The next step is to define the structural model – the model that produces estimates of the probability any respondent has a given attribute profile. The structural model is how attribute association (i.e., correlation) is modeled in DCMs.

This step is optional as Mplus will fit a saturated model if none is specified. Specification of the structural model, however, will improve estimation speed and convergence. Mplus uses a log-linear model for the structural component, meaning the model itself will look like a linear model with intercepts, main effects, and interactions. These terms are all specific to attributes and not to the interaction of attributes to produce item responses.

By definition (and for identification), Mplus sets the value of the last class to zero, causing a few inconveniences in syntax building. The table below shows the label for each structural model parameter along with the actual modeling term used. Note that the first row is a new parameter created to be the intercept (defined because the last term is fixed at zero).

Note: a reasonable approach to a structural model would be to have all main effects and two-way interactions, allowing all attributes to be correlated.

Structural Model Specification Table

Class	Attribute Profile	Mplus Label	Saturated Log-linear Model
-	-	g0 (Intercept)	$\gamma_0 = -(\gamma_{1,(1)} + \gamma_{1,(2)} + \gamma_{2,(1,2)} + \gamma_{1,(1)} + \gamma_{1,(2)} + \gamma_{1,(3)} + \gamma_{2,(1,2)} + \gamma_{2,(1,3)} + \gamma_{2,(2,3)} + \gamma_{3,(1,2,3)})$
1	$\boldsymbol{\alpha}_1$ = [0,0,0]	m1	$\mu_1 = \gamma_0$
2	$\boldsymbol{\alpha}_2$ = [0,0,1]	m2	$\mu_2 = \gamma_0 + \gamma_{1,(3)}$
3	$\boldsymbol{\alpha}_3$ = [0,1,0]	m3	$\mu_3 = \gamma_0 + \gamma_{1,(2)}$
4	$\boldsymbol{\alpha}_4$ = [0,1,1]	m4	$\mu_4 = \gamma_0 + \gamma_{1,(2)} + \gamma_{1,(3)} + \gamma_{2,(2,3)}$
5	$\boldsymbol{\alpha}_5$ = [1,0,0]	m5	$\mu_5 = \gamma_0 + \gamma_{1,(1)}$
6	$\boldsymbol{\alpha}_6$ = [1,0,1]	m6	$\mu_6 = \gamma_0 + \gamma_{1,(1)} + \gamma_{1,(3)} + \gamma_{2,(1,3)}$
7	$\boldsymbol{\alpha}_7$ = [1,1,0]	m7	$\mu_7 = \gamma_0 + \gamma_{1,(1)} + \gamma_{1,(2)} + \gamma_{2,(1,2)}$
8	$\boldsymbol{\alpha}_8$ = [1,1,1]	NONE (Fixed by Mplus)	$\mu_8 = 0$

5. Specifying Initial Mplus Syntax

The initial Mplus syntax contains information about the data set, items, and types of variables.

Note: this comes from file chapter9c.inp

Mplus Syntax	Comments
TITLE: LCDM Estimation simulated ECPE data set.	Provides the title for the analysis that appears in output.
DATA: FILE IS data.dat;	Provides location of input data file. Assumes same folder as input file if no path given.
VARIABLE: NAMES = X1-X28 c; USEVARIABLE = X1-X28; CATEGORICAL = x1-x28; CLASSES = c(8);	The variable section lists details about the data – variables and their types. • NAMES: labels variables in data file (real data will not have c) • USEVARIABLE: defines which variables are used in the analysis • CATEGORICAL: lists which variables are categorical (default is continuous) • CLASSES: provides number of latent classes to be estimated – 2^A for *A* measured attributes
ANALYSIS: TYPE=MIXTURE; STARTS=0; PROCESSORS=8;	The analysis section lists details about the estimation procedure. • TYPE: Mixture indicates latent classes will be used – mandatory for DCMs • STARTS: Turns off default multiple random starts option • PROCESSORS: Uses multithreaded algorithm, if computer has more than one processor
OUTPUT: TECH1 TECH5 TECH8 TECH10;	Requests additional output statistics (convergence history; goodness of fit).
SAVEDATA: FORMAT IS f10.5; FILE IS respondent_lcdm.dat; SAVE = CPROBABILITIES;	Instructs Mplus to save respondent estimates to file named respondent_lcdm.dat. File located in same folder as input syntax file.

The remaining two sections provide annotated Mplus syntax for the rest of the analysis.

6. Building Mplus MODEL Command Syntax

Mplus Syntax	Comments

MODEL:	The MODEL command lists the specifics for the LCDM. It consists of two portions:
	• Class model labels (offset by %class% statements)
	  • Entire item response labels table is entered
	• Model constraints (where LCDM parameters are defined)
	  • Labels are set equal to item response functions

%OVERALL% [C#1] (M1); !profile [000] [C#2] (M2); !profile [001] [C#3] (M3); !profile [010] [C#4] (M4); !profile [011] [C#5] (M5); !profile [100] [C#6] (M6); !profile [101] [C#7] (M7); !profile [110]	The %OVERALL% section is for the structural model. • [C#1] is the Mplus syntax for the value of the first class mean • (m1) is our label (to be used in the model constraints section) • NOTE: [C#8] (m8) is not listed – mean for last class set to zero by Mplus
!column #1 of item response function labels table !for profile [000] %c#1% [X1$1] (T1_1); !item 1 _ threshold 1 [X2$1] (T2_1); !item 2 _ threshold 1 [X3$1] (T3_1); !item 3 _ threshold 1 [X4$1] (T4_1); !item 4 _ threshold 1 ... !items 5-26 omitted for space [X27$1] (T28_1); !item 6 _ threshold 1 [X28$1] (T28_1); !item 7 _ threshold 1	The %c#1% section specifies the labels for item thresholds for profile [000] (class #1). • Comes from column 1 of item response function labels table • Repeated for all columns of table • [X1$1] is the threshold for item X1 • (T1_1) is our label for item 1 – threshold 1
!column #2 of item respond function labels table !for profile [001] %c#2% [X1$1] (T1_1); !item 1 _ threshold 1 [X2$1] (T2_1); !item 2 _ threshold 1 [X3$1] (T3_2); !item 3 _ threshold 2 [X4$1] (T4_2); !item 4 _ threshold 2 ... !items 5-26 omitted for space [X27$1] (T27_1); !item 27 _ threshold 1 [X28$1] (T28_2); !item 28 _ threshold 2	The %c#2% section specifies the labels for item thresholds for profile [001] (class #2). • Comes from column 2 of item response function labels table • Repeated for all columns of table • [X1$1] is the threshold for item X1 • (T1_1) is our label for item 1 – threshold 1
!column #3 of item respond function labels table !for profile [010] %c#3% [X1$1] (T1_2); !item 1 _ threshold 2 [X2$1] (T2_2); !item 2 _ threshold 2 [X3$1] (T3_1); !item 3 _ threshold 1 [X4$1] (T4_1); !item 4 _ threshold 1 ... !items 5-26 omitted for space [X27$1] (T27_1); !item 27 _ threshold 1 [X28$1] (T28_1); !item 28 _ threshold 1	The %c#3% section specifies the labels for item thresholds for profile [010] (class #3). • Comes from column 3 of item response function labels table • Repeated for all columns of table • [X1$1] is the threshold for item X1 • (T1_1) is our label for item 1 – threshold 1

Mplus Syntax	Comments
!column #4 of item respond function labels table !for profile [011] %c#4% [X1$1] (T1_2); !item 1 _ threshold 2 [X2$1] (T2_2); !item 2 _ threshold 2 [X3$1] (T3_2); !item 3 _ threshold 2 [X4$1] (T4_2); !item 4 _ threshold 2 ... !items 5-26 omitted for space [X27$1] (T27_1); !item 27 _ threshold 1 [X28$1] (T28_2); !item 28 _ threshold 2	The %c#4% section specifies the labels for item thresholds for profile [011] (class #4). • Comes from column 4 of item response function labels table • Repeated for all columns of table • [X1$1] is the threshold for item X1 • (T1_1) is our label for item 1 – threshold 1
!column #5 of item respond function labels table !for profile [100] %c#5% [X1$1] (T1_3); !item 1 _ threshold 3 [X2$1] (T2_1); !item 2 _ threshold 1 [X3$1] (T3_3); !item 3 _ threshold 3 [X4$1] (T4_1); !item 4 _ threshold 1 ... !items 5-26 omitted for space [X27$1] (T27_2); !item 27 _ threshold 2 [X28$1] (T28_1); !item 28 _ threshold 1	The %c#5% section specifies the labels for item thresholds for profile [100] (class #5). • Comes from column 5 of item response function labels table • Repeated for all columns of table • [X1$1] is the threshold for item X1 • (T1_2) is our label for item 1 – threshold 2
!column #6 of item respond function labels table !for profile [101] %c#6% [X1$1] (T1_3); !item 1 _ threshold 3 [X2$1] (T2_1); !item 2 _ threshold 1 [X3$1] (T3_4); !item 3 _ threshold 4 [X4$1] (T4_2); !item 4 _ threshold 2 ... !items 5-26 omitted for space [X27$1] (T27_2); !item 27 _ threshold 2 [X28$1] (T28_2); !item 28 _ threshold 2	The %c#6% section specifies the labels for item thresholds for profile [101] (class #6). • Comes from column 6 of item response function labels table • Repeated for all columns of table • [X1$1] is the threshold for item X1 • (T1_2) is our label for item 1 – threshold 2
!column #7 of item respond function labels table !for profile [110] %c#7% [X1$1] (T1_4); !item 1 _ threshold 4 [X2$1] (T2_2); !item 2 _ threshold 2 [X3$1] (T3_3); !item 3 _ threshold 3 [X4$1] (T4_1); !item 4 _ threshold 1 ... !items 5-26 omitted for space [X27$1] (T27_2); !item 27 _ threshold 2 [X28$1] (T28_1); !item 28 _ threshold 1	The %c#7% section specifies the labels for item thresholds for profile [110] (class #7). • Comes from column 7 of item response function labels table • Repeated for all columns of table • [X1$1] is the threshold for item X1 • (T1_2) is our label for item 1 – threshold 2
!column #8 of item respond function labels table !for profile [111] %c#8% [X1$1] (T1_4); !item 1 _ threshold 4 [X2$1] (T2_2); !item 2 _ threshold 2 [X3$1] (T3_4); !item 3 _ threshold 4 [X4$1] (T4_2); !item 4 _ threshold 2 ... !items 5-26 omitted for space [X27$1] (T27_2); !item 27 _ threshold 2 [X28$1] (T28_2); !item 28 _ threshold 2	The %c#8% section specifies the labels for item thresholds for profile [111] (class #8). • Comes from column 8 of item response function labels table • Repeated for all columns of table • [X1$1] is the threshold for item X1 • (T1_2) is our label for item 1 – threshold 2

7. Building Mplus MODEL CONSTRAINT Command Syntax (with representative example items shown)

Mplus Syntax	Comments
MODEL CONSTRAINT:	The MODEL CONSTRAINT command is where the LCDM parameters are defined and the item response function is given for each label. • Structural model given [optional] • Syntax needed for all items

!STRUCTURAL MODEL PORTION;
!define structural model parameters:
NEW(G_0 G_11 G_12 G_13 G_212 G_213 G_223 G_3123);
 	
!from structural model specification table:
!intercept:
 	
!profile [000];
M1=-(G_11+G_12+G_13+G_212+G_213+G_223+G_3123);
!profile [001];
M2=G_13-
(G_11+G_12+G_13+G_212+G_213+G_223+G_3123);
!profile [010];
M3=G_12-
(G_11+G_12+G_13+G_212+G_213+G_223+G_3123);
!profile [011];
M4=G_12+G_13+G_223-
(G_11+G_12+G_13+G_212+G_213+G_223+G_3123);
!profile [100];
M5=G_11-
(G_11+G_12+G_13+G_212+G_213+G_223+G_3123);
!profile [101];
M6=G_11+G_13+G_213-
(G_11+G_12+G_13+G_212+G_213+G_223+G_3123);
!profile [110];
M7=G_11+G_12+G_212-
(G_11+G_12+G_13+G_212+G_213+G_223+G_3123);
	The structural model section.
	• Taken from structural model specification table
	• NEW: creates new parameters for Mplus to use in estimation
	• G_[ea1…]: label for gamma parameter; structural model effect parameter
	  • e: effect level (0- intercept; 1- main effect; 2 – two way interaction…)
	  • a1…: attribute(s) to which effect applies; number of attributes is equal to effect level e

! Item 10 [100]
NEW(L10_0 L10_11);
!link labels with LCDM item response function:
T10_1=-(L10_0);
T10_2=-(L10_0+L10_11);
!main effect order constraint:
L10_11>0;
	The LCDM item parameter section for item 10.
	• Links label with item response function
	• NEW: creates new parameters for Mplus to use in estimation
	• L10_[ea1…]: label for lambda parameter; LCDM item parameter
	  • 10: parameter for item 10
	  • e: effect level (0- intercept; 1- main effect; 2 – two way interaction…)
	  • a1…: attribute(s) to which effect applies; number of attributes is equal to effect level e
	• LCDM item response function multiplied by -1 (Mplus modeling difference)

! Item 2 [010]
NEW(L2_0 L2_12);
!link labels with LCDM item response function:
T2_1=-(L2_0);
T2_2=-(L2_0+L2_12);
!main effect order constraint:
L2_12>0;
	The LCDM item parameter section for item 2.
	• Links label with item response function
	• NEW: creates new parameters for Mplus to use in estimation
	• L2_[ea1…]: label for lambda parameter; LCDM item parameter
	  • 2: parameter for item 2
	  • e: effect level (0- intercept; 1- main effect; 2 – two way interaction…)
	  • a1…: attribute(s) to which effect applies; number of attributes is equal to effect level e
	• LCDM item response function multiplied by -1 (Mplus modeling difference)

! Item 4 [001]
NEW(L4_0 L4_13);
!link labels with LCDM item response function:
T4_1=-(L4_0);
T4_2=-(L4_0+L4_13);
!main effect order constraint:
L4_13>0;
	The LCDM item parameter section for item 4.
	• Links label with item response function
	• NEW: creates new parameters for Mplus to use in estimation
	• L4_[ea1…]: label for lambda parameter; LCDM item parameter
	  • 4: parameter for item 4
	  • e: effect level (0- intercept; 1- main effect; 2 – two way interaction…)
	  • a1…: attribute(s) to which effect applies; number of attributes is equal to effect level e
	• LCDM item response function multiplied by -1 (Mplus modeling difference)

! Item 1 [110]
NEW(L1_0 L1_11 L1_12 L1_212);
!link labels with LCDM item response function:
T1_1=-(L1_0);
T1_2=-(L1_0+L1_12);
T1_3=-(L1_0+L1_11);
T1_4=-(L1_0+L1_11+L1_12+L1_212);
!main effect order constraints:
L1_11>0; L1_12>0;
!two-way interaction constraints:
L1_212>-L1_11; L1_212>-L1_12;
	The LCDM item parameter section for item 1.
	• Links label with item response function
	• NEW: creates new parameters for Mplus to use in estimation
	• L1_[ea1…]: label for lambda parameter; LCDM item parameter
	  • 1: parameter for item 1
	  • e: effect level (0- intercept; 1- main effect; 2 – two way interaction…)
	  • a1…: attribute(s) to which effect applies; number of attributes is equal to effect level e
	LCDM item response function multiplied by -1 (Mplus modeling difference)

! Item 7 [101]
NEW(L7_0 L7_11 L7_13 L7_213);
!link labels with LCDM item response function:
T7_1=-(L7_0);
T7_2=-(L7_0+L7_13);
T7_3=-(L7_0+L7_11);
T7_4=-(L7_0+L7_11+L7_13+L7_213);
!main effect order constraints:
L7_11>0; L7_13>0;
!two-way interaction constraints:
L7_213>-L7_11; L7_213>-L7_13;
	The LCDM item parameter section for item 7.
	• Links label with item response function
	• NEW: creates new parameters for Mplus to use in estimation
	• L7_[ea1…]: label for lambda parameter; LCDM item parameter
	  • 7: parameter for item 7
	  • e: effect level (0- intercept; 1- main effect; 2 – two way interaction…)
	  • a1…: attribute(s) to which effect applies; number of attributes is equal to effect level e
	LCDM item response function multiplied by -1 (Mplus modeling difference)

! Item 17 [011]
NEW(L17_0 L17_12 L17_13 L17_223);
!link labels with LCDM item response function:
T17_1=-(L17_0);
T17_2=-(L17_0+L17_13);
T17_3=-(L17_0+L17_12);
T17_4=-(L17_0+L17_12+L17_13+L17_223);
!main effect order constraints:
L17_12>0; L17_13>0;
!two-way interaction constraints:
L17_223>-L17_12; L17_223>-L17_13;
	The LCDM item parameter section for item 17.
	• Links label with item response function
	• NEW: creates new parameters for Mplus to use in estimation
	• L17_[ea1…]: label for lambda parameter; LCDM item parameter
	  • 17: parameter for item 7
	  • e: effect level (0- intercept; 1- main effect; 2 – two way interaction…)
	  • a1…: attribute(s) to which effect applies; number of attributes is equal to effect level e
	LCDM item response function multiplied by -1 (Mplus modeling difference)

Understanding Mplus Output

Once the syntax has been built and Mplus finishes estimating the model, the Mplus output contains all the information needed to evaluate the model run. Mplus syntax is voluminous in its size, with only a few sections of interest (and many that are redundant). Each section is discussed in the following pages, in order of appearance in the output file.

1. Tests of Model Fit Output Section
2. Final class counts and proportions
3. New/Additional parameters
4. Technical 10

Tests of Model Fit Output Section

```
MODEL FIT INFORMATION

Number of Free Parameters                       81

Loglikelihood

          H0 Value                      -42610.450
          H0 Scaling Correction Factor       0.961
            for MLR

Information Criteria

          Akaike (AIC)                   85382.900
          Bayesian (BIC)                 85867.282
          Sample-Size Adjusted BIC       85609.915
            (n* = (n + 2) / 24)

Chi-Square Test of Model Fit for the Binary and Ordered Categorical
(Ordinal) Outcomes**

          Pearson Chi-Square

          Value                          22539.264
          Degrees of Freedom            268433375
          P-Value                           1.0000

          Likelihood Ratio Chi-Square

          Value                           4289.813
          Degrees of Freedom            268433375
          P-Value                           1.0000
```

- Provides basic model fit information
- Information Criteria: Used to compare model fit for non-nested models (smaller is better)
- Chi-square Test: Absolute measure of fit (for small numbers of items only – not this example)

Final Class Counts and Estimated Proportions Output Section

```
FINAL CLASS COUNTS AND PROPORTIONS FOR THE LATENT CLASSES
BASED ON THE ESTIMATED MODEL

    Latent
   Classes

       1        861.71242          0.29491
       2        450.16073          0.15406
       3         43.93011          0.01503
       4        563.85001          0.19297
       5          8.27932          0.00283
       6          9.79413          0.00335
       7          0.00000          0.00000
       8        984.27328          0.33685
```

- Provides the probability an random respondent has a given attribute profile
  - 29.49% of sample has profile [0,0,0]
  - 15.41% of sample has profile [0,0,1]

New/Additional Parameters Output Section

New/Additional Parameters

	Estimate	S.E.	Est./S.E.	Two-Tailed P-Value	
G_11	-4.645	0.908	-5.114	0.000	
G_12	-2.976	0.963	-3.092	0.002	
G_13	-0.649	0.112	-5.775	0.000	Structural Model Parameters
G_212	-20.273	2.174	-9.327	0.000	
G_213	0.817	2.408	0.339	0.734	
G_223	3.202	0.923	3.470	0.001	
G_3123	24.658	0.000	0.000	1.000	
L1_0	0.931	0.073	12.671	0.000	
L1_11	0.000	0.000	100.000	0.000	LCDM Item Parameters
L1_12	0.480	0.220	2.179	0.029	
L1_212	1.290	0.270	4.772	0.000	
L2_0	1.029	0.079	12.961	0.000	
L2_12	1.327	0.145	9.157	0.000	
L3_0	-0.407	0.075	-5.461	0.000	
L3_11	0.000	0.000	100.000	0.000	
L3_13	0.430	0.116	3.701	0.000	
L3_213	1.380	0.132	10.472	0.000	
L4_0	-0.091	0.073	-1.249	0.212	
L4_13	1.670	0.102	16.411	0.000	
L5_0	1.079	0.081	13.299	0.000	
L5_13	2.139	0.159	13.473	0.000	
L6_0	0.894	0.078	11.437	0.000	
L6_13	1.707	0.130	13.101	0.000	
L7_0	-0.116	0.073	-1.582	0.114	
L7_11	0.000	0.000	100.000	0.000	
L7_13	1.040	0.120	8.634	0.000	
L7_213	2.049	0.221	9.265	0.000	
L8_0	1.532	0.088	17.398	0.000	
L8_12	1.732	0.210	8.248	0.000	
L9_0	0.203	0.073	2.791	0.005	
L9_13	1.153	0.098	11.800	0.000	
L10_0	0.067	0.050	1.344	0.179	
L10_11	2.062	0.144	14.346	0.000	

- Provides each parameter estimate and standard error
- Hypothesis test provides way of testing whether each parameter is equal to zero
  - Works for all structural model parameters
  - Works for LCDM intercept item parameters (although test isn't informative)
  - Will not work for LCDM main effect parameters
  - Is only approximate for interaction parameters

Technical 10 Output Section

		Estimated Probabilities		
Variable	Variable	H1	H0	Standardized Residual (z-score)
X1	X2			
Category 1	Category 1	0.039	0.039	-0.158
Category 1	Category 2	0.152	0.151	0.085
Category 2	Category 1	0.128	0.128	0.092
Category 2	Category 2	0.681	0.682	-0.066
Bivariate Pearson Chi-Square				0.039
Bivariate Log-Likelihood Chi-Square				0.039

- Provides quick goodness of fit check for pairs of items – similar to raw residuals in SEM

Obtaining Diagnostic Classification Model Estimates with Mplus

Adaptation to Hierarchical Diagnostic Classification Model

AERA Diagnostic Measurement Workshop

To demonstrate how Diagnostic Classification Models (DCMs) can be estimated using Mplus, input syntax and output are included in this document.

Analysis Data Set – Based on ECPE Data from Session 3 (simulated so it is distributable):

- 28 items
- Three attributes
    - Attribute 1 (morphosyntactic rules) nested within Attribute 2 (cohesive rules) nested within Attribute 3 (lexical rules)
- 2,922 respondents

To create Mplus syntax, we will follow the same steps (although we will not build syntax here as it follows from the example given previously).

1. 1. Creating a Latent Class-to-Attribute Profile Table
    - Mplus uses generic latent classes in estimation – we must make these into DCM attribute profiles.
2. 2. Creating an Item-to-Profile Table
    - Specifies the form of the LCDM for each combination of item and profile. It will be used to build model syntax.
3. 3. Creating Item Response Function Labels
    - Defines the set of unique item response functions for each item (based on the set of attributes measured by the Q-matrix). Used by Mplus to specify LCDM parameters.
4. 4. Creating the Structural Model
    - For limiting the number of correlational parameters needed; aids in estimation speed and convergence.

1. Creating a Latent Class-to-Attribute Profile Table

Mplus uses latent classes to model categorical data generally. We must represent each possible attribute profiles as a latent class. For DCMs that measure *A* attributes, a total of 2^A attribute profiles are possible. **Because we have an attribute hierarchy, we must first create this table, then eliminate the attribute profiles not possible due to the hierarchical structure of attributes.** It is our job to map each profile onto a latent class. To do so, we use the following process. Recall our example data set specifics:

Number of attributes in example: 3

Number of possible *attribute profiles*: $2^3 = 8$

Number of latent classes needed: 8

First, we need a table that has the number of latent classes represented in the rows and the number of attributes represented in the columns.

The following figure shows how to populate the classes with attribute profiles through a sequence of partitions of the table:

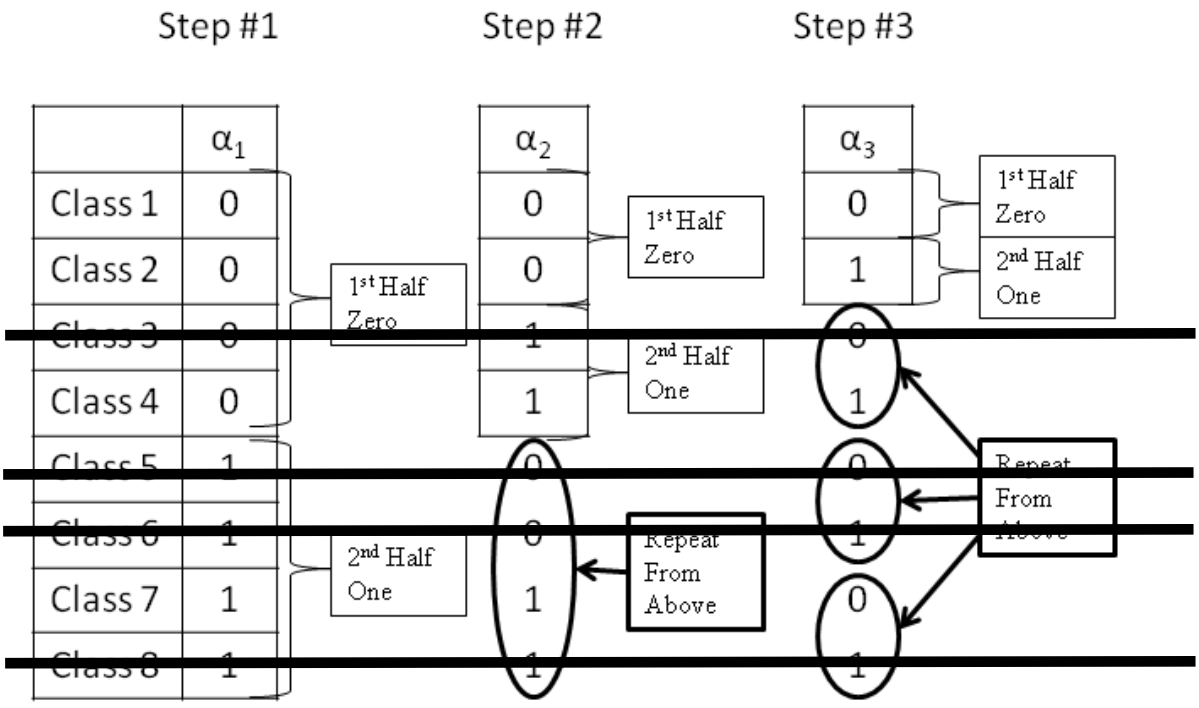

Figure 9.7 from Rupp, Templin, & Henson (2010). Creating a class-to-profile table.

Next, we must eliminate the not-possible profiles. Here, these are:

1. **Any time Attribute 1 is mastered when Attribute 2 is not (Classes 5 and 6).**
2. **Any time Attribute 2 is mastered when Attribute 3 is not (Classes 3 and 7).**

2. Creating an Item-to-Profile Table

The next step is to develop the HDCM model specification for each combination of items and attribute profiles, the item response function for respondents with a given attribute profile. In DCMs, not all attributes are measured by each item. Therefore, there will be repeated item response functions. This table will help to denote which are redundant and therefore able to be omitted. We will use the Q-matrix to help form what model parameters should be present for each combination of item and attribute profile. **This table will contain only the classes left after the hierarchy (renumbered from 1-4).**

Analysis Q-matrix (rearranged to reflect items measuring same set of attributes):

Items	Attribute 1	Attribute 2	Attribute 3	Set Number
10, 13, 14, 25, 27	1	0	0	1
2, 8, 23, 24	0	1	0	2
4, 5, 6, 9, 15, 18, 19, 22, 26, 28	0	0	1	3
1	1	1	0	4
3, 7, 11, 12, 16, 20, 21	1	0	1	5
17	0	1	1	6

Item-to-Profile Table (first subscript denotes set – change for each item number)

Class	c_1	c_2	c_3	c_4
α_c	[0, 0, 0]	[0, 0, 1]	[0, 1, 1]	[1, 1, 1]
Set 1	$\lambda_{1,0}$	$\lambda_{1,0}$	$\lambda_{1,0}$	$\lambda_{1,0} + \lambda_{1,1,(1)}$
Set 2	$\lambda_{2,0}$	$\lambda_{2,0}$	$\lambda_{2,0} + \lambda_{2,1,(2)}$	$\lambda_{2,0} + \lambda_{2,1,(2)}$
Set 3	$\lambda_{3,0}$	$\lambda_{3,0} + \lambda_{3,1,(3)}$	$\lambda_{3,0} + \lambda_{3,1,(3)}$	$\lambda_{3,0} + \lambda_{3,1,(3)}$
Set 4	$\lambda_{4,0}$	$\lambda_{4,0}$	$\lambda_{4,0} + \lambda_{4,1,(2)}$	$\lambda_{4,0} + \lambda_{4,1,(2)} + \lambda_{4,2,(1(2))}$
Set 5	$\lambda_{5,0}$	$\lambda_{5,0} + \lambda_{5,1,(3)}$	$\lambda_{5,0} + \lambda_{5,1,(3)}$	$\lambda_{5,0} + \lambda_{5,1,(3)} + \lambda_{5,2,(1(3))}$
Set 6	$\lambda_{6,0}$	$\lambda_{6,0} + \lambda_{6,1,(3)}$	$\lambda_{6,0} + \lambda_{6,1,(3)} + \lambda_{6,2,(2(3))}$	$\lambda_{6,0} + \lambda_{6,1,(3)} + \lambda_{6,2,(2(3))}$

3. Creating Item Response Function Labels

The next step in the process is to label each unique item response function for each *item* in the Item-to-Profile table. *In the table below, we use the first number to represent the set of items from the Item-to-Profile table. In code, this will change to be the item number.* The labeling convention we will use will follow the form of t[i]_[#]:

- t represents that label is a threshold (Mplus definition for LCDM item response function)
- [i] is the item used (omit brackets)
- [#] is the index for unique item response function for an item.

The labels are created by reading left-to-right for each row of the Item-to-Profile table. The first entry for the first cell of the first item receives the label t1_1, as do all other entries that are the same. The next unique entry receives the label t1_2, as do all other entries that look the same. The process continues until all labels have been assigned.

Item Response Function Labels

Class	c_1	c_2	c_3	c_4
α_c	[0, 0, 0]	[0, 0, 1]	[0, 1, 1]	[1, 1, 1]
Set 1	T1_1	T1_1	T1_1	T1_2
Set 2	T2_1	T2_1	T2_2	T2_2
Set 3	T3_1	T3_2	T3_2	T3_2
Set 4	T4_1	T4_1	T4_2	T4_3
Set 5	T5_1	T5_2	T5_2	T5_3
Set 6	T6_1	T6_2	T6_3	T6_3

4. Creating the Structural Model [Optional]

The next step is to define the structural model – the model that produces estimates of the probability any respondent has a given attribute profile. The structural model is how attribute association (i.e., correlation) is modeled in DCMs.

This step is optional as Mplus will fit a saturated model if none is specified. Specification of the structural model, however, will improve estimation speed and convergence. Mplus uses a log-linear model for the structural component, meaning the model itself will look like a linear model with intercepts, main effects, and interactions. These terms are all specific to attributes and not to the interaction of attributes to produce item responses.

By definition (and for identification), Mplus sets the value of the last class to zero, causing a few inconveniences in syntax building. The table below shows the label for each structural model parameter along with the actual modeling term used. Note that the first row is a new parameter created to be the intercept (defined because the last term is fixed at zero).

Note: a reasonable approach to a structural model would be to have all main effects and two-way interactions, allowing all attributes to be correlated.

Structural Model Specification Table

Class	Attribute Profile	Mplus Label	Saturated Log-linear Model
-	-	g0 (Intercept)	$\gamma_0 = -\left(\gamma_{1,(1)} + \gamma_{1,(3)} + \gamma_{2,(2(3))} + \gamma_{3,(1(2,3))}\right)$
1	$\boldsymbol{\alpha}_1$ = [0,0,0]	m1	$\mu_1 = \gamma_0$
2	$\boldsymbol{\alpha}_2$ = [0,0,1]	m2	$\mu_2 = \gamma_0 + \gamma_{1,(3)}$
3	$\boldsymbol{\alpha}_3$ = [0,1,1]	m3	$\mu_3 = \gamma_0 + \gamma_{1,(3)} + \gamma_{2,(2(3))}$
4	$\boldsymbol{\alpha}_4$ = [1,1,1]	NONE (Fixed by Mplus)	$\mu_4 = 0$

The remainder of the handout will follow the Mplus syntax described previously.