Educational Measurement
ISSUES AND PRACTICE

NCME national council on measurement in education

Educational Measurement: Issues and Practice
Summer 2013, Vol. 32, No. 2, pp. 37–50

Obtaining Diagnostic Classification Model Estimates Using Mplus

Jonathan Templin, *University of Georgia,* and Lesa Hoffman, *University of Nebraska–Lincoln*

Diagnostic classification models (aka cognitive or skills diagnosis models) have shown great promise for evaluating mastery on a multidimensional profile of skills as assessed through examinee responses, but continued development and application of these models has been hindered by a lack of readily available software. In this article we demonstrate how diagnostic classification models may be estimated as confirmatory latent class models using Mplus, thus bridging the gap between the technical presentation of these models and their practical use for assessment in research and applied settings. Using a sample English test of three grammatical skills, we describe how diagnostic classification models can be phrased as latent class models within Mplus and how to obtain the syntax and output needed for estimation and interpretation of the model parameters. We also have written a freely available SAS program that can be used to automatically generate the Mplus syntax. We hope this work will ultimately result in greater access to diagnostic classification models throughout the testing community, from researchers to practitioners.

Keywords: diagnostic classification models, cognitive diagnosis, skills assessment, Mplus, LCDM

Imagine this scenario: Seeking career advancement through educational or professional opportunities, non-native English speaking examinees take an English test to attain a certificate of proficiency at reading English. Examinees receive their score—and find out whether or not they pass the test and attain their certificate. Those passing have accomplished their goal; however, those not passing are left to try again, likely after spending more time and money on tutors or educational aides. Yet these test results cannot help examinees with what to study, outside of English as a field, such that examinees would have little guidance on how to improve their skills. Had the test provided more skill-specific feedback, examinees may have a better idea as to the nature of what they have yet to learn, aiding them in their future training. This article is about the process of culling such skill-specific information from tests via diagnostic classification models.

In recent years, new psychometric methods have been developed that can provide precise and detailed information about the latent attributes or skills examinees may possess as indicated by their item responses. Although *diagnostic classification models* (DCMs; Rupp & Templin, 2008) have been the focus of numerous articles, books, and conference presentations (also known *cognitive diagnosis models,* e.g., Leighton & Gierl, 2007; Rupp, 2007), most of this work has been heavily technical and limited to the statistical portion of the psychometric community. We argue this is largely due to the lack of available software for researchers and practitioners who wish to use diagnostic classification models in their testing programs and empirical research. Currently, software for these models is limited to stand-alone programs with limited model options (i.e., The Arpeggio Suite; Bolt et al., 2008) or that are available only under a restrictive research license (i.e., MDLTM, von Davier, 2006). Further, methods for estimation of DCMs using Mplus were discussed in Rupp, Templin, and Henson (2010) and de la Torre (2009); however, the discussion focused on a technical treatment of the issue without providing a practical perspective. Therefore, the goal of this article is to make diagnostic models more accessible for researchers and practitioners by showing how they can be estimated using the more flexible and readily available commercial software package Mplus instead (Muthén & Muthén, 2013).

Although diagnostic classification models have shown much promise, the ultimate standard by which they should be judged is by their utility to those who may benefit most from their use: the practitioners and, ultimately, end-users such as classroom teachers and students. Accordingly, this article provides the necessary link between the technical details of diagnostic classification models and their implementation in easy-to-use software so that they will be more accessible to everyone—not just for research, but perhaps more importantly, in practice. We aim to guide readers through each step in the modeling process—we use a single example test measuring three English language skills throughout the article to introduce diagnostic classification models as confirmatory latent class models, to describe the specification of their parameters through Mplus syntax, and to interpret the resulting model output.

Jonathan Templin, Research, Evaluation, Measurement, and Statistics Program, Department of Educational Psychology, University of Georgia, 323 Aderhold Hall, Athens, GA 30602; jtemplin@uga.edu. Lesa Hoffman, Department of Psychology, University of Nebraska–Lincoln.

For consistency throughout, we will refer to the test takers as *examinees*, the questions of the test as *items*, and the latent attributes measured by the test as *skills*. Although the terms *attributes* and *skills* are different concepts—*attribute* referring to a characteristic someone possesses and *skill* referring to a particular ability or task a person does well—we will use the term *skill* to emphasize the educational components of the traits measured (i.e., that after sufficient education or training, one should possess that skill). We first describe the example test data and present the rationale for employing a diagnostic classification model (DCM). We then show how the set of skills measured by a test in DCMs can be viewed as a series of latent classes, allowing for existing software for latent class models to also be used for DCMs. We then describe how to implement a general DCM in Mplus through a series of model constraint syntax statements, followed by a description of the relevant sections of Mplus output that provide the DCM item and examinee parameter estimates. Finally, because the process of specifying all of the necessary syntax can be somewhat tedious, we describe a SAS program we have made freely available with which to automate the Mplus syntax generation process.

Example Data: The Examination for the Certificate of Proficiency in English (ECPE)

The *Examination for the Certificate of Proficiency in English* (ECPE) is a test developed and scored by the English Language Institute of the University of Michigan. The test measures advanced English skills in examinees whose primary language is not English and is administered internationally once a year between November and April, depending on the location. Here we analyze data of 2,922 examinees from a single year's administration. Although the ECPE has four major test sections, we restrict our example to the grammar section, including 28 multiple-choice questions in which a set of words is missing. Examinees are asked to select the appropriate word(s) for the missing section of the statement from four response options, as shown in this example item:

Mary had to lean _____ the counter to open the window.

(a) above
(b) over
(c) after
(d) around

The ECPE is designed to measure three skills: knowledge of (1) morphosyntactic rules, (2) cohesive rules, and (3) lexical rules (see Buck & Tatsuoka, 1998; Henson & Templin, 2007). Although the ECPE commonly provides just a single score for an examinee's overall ability for understanding grammatical rules of English, more specific information about an examinee's ability with respect to each of the three measured skills is often desired. This is where DCMs can be more useful, because rather than provide a single ability score, DCMs specify a profile of multiple skills, typically defined such that examinees can be a master or non-master of each. Our DCM application to the ECPE data will result in a profile for each examinee of whether each of the three grammar skills has been mastered, thus supplying useful information that can guide further instruction towards any missing skills as needed. At the same time, however, DCMs can also help test developers understand the extent to which specific skills are needed to produce correct responses to the test items in the first place. That is, through DCMs we can evaluate empirically whether the items thought to measure each skill really do—as well as the extent to which those skills appear to be compensatory in items that measure more than one skill. In this sense, DCMs can help provide evidence for or against the validity of a test in making inferences about whether each examinee possesses the skills in question (e.g., Dimitrov, 2007; Embretson, 1995; Kane, 2006). Thus, DCMs can be useful at both the test development stage and in evaluating the skills of real-world examinees.

To further describe the ECPE and to provide a baseline understanding of the test from a more classical perspective, test reliability analyses were conducted. For the total score, the Guttman-Cronbach Alpha reliability was .78. If one was to report sum scores representing the three skills thought to underlie the ECPE (using the Q-matrix of the DCM analysis as the indicator for which items were included in each sum score), reliabilities would be .67 for the morphosyntactic skill (13 items), .36 for the cohesive skill (6 items), and .73 for the lexical skill (18 items). We note that reliability is dependent on the statistical model, and that these reliabilities refer to the classical true-score model where sum scores (either across the whole test or for a subscale) provide the estimate of an examinee's ability. Accurate estimates of reliability depend on the correctness of the underlying model for ability, and in no way represent an index of how well a model fits the data. Under DCMs, reliability is calculated for the classification of the skill as a categorical latent variable (see Templin & Bradshaw, in press), which is a different model for ability and would thus result in a different estimate of reliability.

To further describe the structure of the ECPE from a more common measurement model, the dimensionality of the test was investigated by comparing the model fit of a one-dimensional two-parameter item response model with that of a three-dimensional confirmatory item response model (using the Q-matrix of the DCM analysis as the indicator for which items loaded onto each factor). The three dimensional model (AIC of 85,136.848) fit better than the one dimensional model (AIC of 85,205.331), providing evidence for the ECPE having more than one dimension. The correlation between the morphosyntactic and cohesive skills was estimated to be .805 ($p < .001$), the correlation between the morphosyntactic and lexical skills was estimated to be .820 ($p < .001$), and the correlation between the cohesive and lexical skills was estimated to be .791 ($p < .001$). These estimates establish that the skills of the test, though highly related, are differentiable from each other to some degree.

With this background in mind, we now describe how DCMs can be viewed as confirmatory latent class models, thereby allowing for their estimation using Mplus software (or any software that can estimate latent class models with a series of statistical constraints). In our article, we focus on the Mplus package as its set of features, documentation, and widespread use in education and the social sciences makes it a choice familiar to many practitioners.

Framing DCMs as Latent Class Models

Latent class models are classification-based techniques used to investigate the number of latent groups of examinees based on similar patterns of item responses. Latent class models have had a long history in education, with ties to DCMs as early as the mastery model of Macready and Dayton (1977). Most relevant here is that DCMs can be viewed as confirmatory latent class models in which the number of classes is determined by the number of skills measured by a test. That is, for a test measuring a total of S dichotomous skills that an examinee has either mastered or not, a total of 2^S distinct mastery profiles exist (as latent classes)—one for each possible combination of mastery and non-mastery of the S total skills. In our ECPE example, the three measured skills (morphosyntactic, cohesive, and lexical knowledge) will result in $2^3 = 8$ possible mastery profiles, only one of which will describe the mastery status of a given examinee. As will be shown, the model for item responses underlying the DCM assumes that examinees with the same mastery profile will provide the same item responses. To show its link to DCMs, we now describe the primary parameters of the general latent class model.

Latent Class Models

The general latent class model is a very restrictive model that only includes a statistical parameter for the difficulty of each item for each latent class (e.g., Lazarsfeld & Henry, 1968). We will denote the response of examinee e to item i with X_{ei}, which can either be correct ($X_{ei} = 1$) or incorrect ($X_{ei} = 0$). In the item response function from the latent class model, the probability examinee e answers item i correctly depends only on examinee e's latent class, or c_e:

$$P\ (X_{ei} = 1|c_e) = \pi_{ic} = \frac{exp\ (\tau_{ic})}{1 + exp\ (\tau_{ic})} \tag{1}$$

The item response function specifies that the probability of a correct response is given by a class-specific item difficulty parameter, π_{ic}, which is the probability that an examinee who is a member of class c answers item i correctly. In order to connect the latent class model to DCMs within Mplus, through the use of a link function, we will replace the class-specific item difficulty parameter π_{ic} that ranges from zero to one with a class-specific item threshold parameter τ_{ic}, an unbounded continuous variable that can be any real number. As in other item response models, an inverse log odds or *logit* function (as seen in the far right of Equation 1) links the threshold τ_{ic} to the probability π_{ic}. For instance, a π_{ic} probability of .50 (50% chance of a correct response) corresponds to a τ_{ic} threshold of 0 (the log-odds for a probability of .50). Because a separate threshold τ_{ic} is required for each item and class, this yields a large number of possible thresholds. For instance, if we were to analyze the 28 ECPE items using an eight-class model (as needed to represent the 2^3 possible skill mastery profiles), we would need $8 \times 28 = 224$ item thresholds. But because the values of these item thresholds within DCMs will depend on the mastery profile of an examinee, far fewer unique item thresholds are be necessary in DCMs than in nonconfirmatory latent class models.

In addition to the class-specific item thresholds that describe the relationship between class membership and item responses, the general latent class model also has a set of structural parameters, which we denote as υ_c, that give the proportion of examinees that are members of each class c. Because these structural parameters are proportions that must sum to one, the number estimated will be one less than the number of latent classes. For our eight-class ECPE example, a total of seven υ_c structural parameters will be estimated within Mplus.

Finally, and perhaps most importantly, once the item threshold parameters (τ_{ic}) and the structural parameters (υ_c) have been calibrated, they can then be used to classify examinees, a process directly analogous to examinee scoring in other psychometric models. That is, examinee classification results in an estimate of the probability $\hat{\alpha}_{ec}$ that examinee e is a member of each latent class c. Examinees can then be classified as members of the class that has the largest $\hat{\alpha}_{ec}$ probability. Thus, in DCMs in which the latent classes represent specific profiles of skill mastery, examinee classification results in a profile of whether each skill measured by a test is likely to have been mastered based on that examinee's responses. Notably, Mplus can also provide classification for new examinees using item and structural model parameters that have already been calibrated in previous samples, a significant advantage in real-world assessment.

Linking DCMs to Latent Class Models

DCMs can be viewed as confirmatory latent class models given that each of the 2^S possible skill mastery profiles in DCMs can be represented by a separate latent class. These skill mastery profiles are then linked to the observed item responses via the τ_{ic} item thresholds that predict the log-odds for the probability of a correct item response according to the skills of an examinee. But because each item only measures certain skills, and because skills are either mastered or not, only a limited set of predicted item responses (created from the possible unique item threshold values) are distinguishable as a result. Thus, rather than following a smooth logistic function as in item response theory models, item responses in DCMs will follow a step-like function with fewer possible probabilities (as calculated by converting the τ_{ic} item thresholds into the π_{ic} item difficulties using the general latent class model in Equation 1.

Further, even though a test may measure multiple skills overall, each item is likely to measure only some of these skills by design. In DCMs, the mapping of which skills are measured by which item is referred to as a *Q-matrix* (Tatsuoka, 1983), with entries q_{is} for each item i and skill s. If item i measures skill s, then $q_{is} = 1$, otherwise $q_{is} = 0$. The Q-matrix mapping of each item to the skill(s) it measures is determined a priori from theory about the item content, just as in multidimensional confirmatory factor models or item response models. The Q-matrix used in our example was the result of psychometric analyses on the ECPE by Buck and Tatsuoka (1998). The analyses showed that items of the test were likely to measure three distinct skills. The left side of Table 1 shows the Q-matrix that maps each item to the three skills in our ECPE example. As shown, eight items measure only one skill, seven items measure two skills, and zero items measure three skills. The morphosyntactic (1), cohesive (2), and lexical (3) skills were each measured by 13, 6, and 18 items, respectively. This Q-matrix will be the basis for our DCM specification, in that the τ_{ic} item thresholds will be held equal for latent classes (skill mastery profiles) with the same mastery status on the skills measured by the item. For instance, item 20 was written to measure knowledge of morphosyntactic and lexical rules only (Skills 1 and 3). As a result, the thresholds for item 20 are the same for any examinee who has mastered both of these rules, regardless of the examinee's mastery on cohesive rules.

Table 1. ECPE Q-Matrix and LCDM Item Parameter Estimates

Item	Skill 1	Skill 2	Skill 3	$\lambda_{i,0}$	$\lambda_{i,1,(1)}$	$\lambda_{i,1,(2)}$	$\lambda_{i,1,(3)}$	$\lambda_{i,2,(1,2)}$	$\lambda_{i,2,(1,3)}$	$\lambda_{i,2,(2,3)}$
1	1	1	0	.835	.000	.600		1.222		
2	0	1	0	1.037		1.247				
3	1	0	1	−.340	.748		.346		.535	
4	0	0	1	−.139			1.691			
5	0	0	1	1.082			2.015			
6	0	0	1	.865			1.692			
7	1	0	1	−.106	2.855		.952		−.952	
8	0	1	0	1.482		1.922				
9	0	0	1	.119			1.195			
10	1	0	0	.055	2.050					
11	1	0	1	−.039	.818		.961		.777	
12	1	0	1	−1.769	.000		1.290		1.515	
13	1	0	0	.660	1.630					
14	1	0	0	.176	1.368					
15	0	0	1	.996			2.114			
16	1	0	1	−.104	2.341		.892		−.864	
17	0	1	1	1.354		.767	.596			.076
18	0	0	1	.926			1.389			
19	0	0	1	−.195			1.848			
20	1	0	1	−1.389	.243		.908		1.410	
21	1	0	1	.164	1.053		1.130		.042	
22	0	0	1	−.872			2.245			
23	0	1	0	.664		2.071				
24	0	1	0	−.673		1.522				
25	1	0	0	.092	1.136					
26	0	0	1	.164			1.119			
27	1	0	0	−.887	1.713					
28	0	0	1	.568			1.745			

Note. Skill 1: Morphosyntactic rules; Skill 2: Cohesive rules; Skill 3: Lexical rules.

Thus, DCMs are confirmatory not only with respect to how many latent classes (corresponding to skill mastery profiles) should be present, but also with respect to how they predict item responses based on the specific combination of the skills measured by a given item and the skills mastered by a given examinee.

Empirical work has resulted in a plethora of specific DCM variants. For instance, the DINA model (Haertel, 1989; Junker & Sijtsma, 2001; Macready & Dayton, 1977) separates people into two classes: those who have mastered all skills measured by an item and those who have not. Other models assume that the probability of a correct response increases for each skill mastered (e.g., the reparameterized unified model, or RUM, and compensatory RUM; Hartz, 2002). In yet other models, only a subset of skills must be mastered to answer an item correctly (e.g., the DINO model; Templin & Henson, 2006). Instead of focusing on any of these specific DCM variants, we use the more general *log-linear cognitive diagnosis model* (LCDM; Henson, Templin, & Willse, 2009) because of its flexibility and extensions to other psychometric models. That is, not only can the LCDM take the form of each of the aforementioned specific DCMs by placing restrictions on its item parameters, but it also allows for parameterizations not possible within the other DCMs, thus providing precise yet flexible information about the structure of items of a test (for additional details, see Henson, Templin, & Willse, 2009).

Interpreting the Parameters of the LCDM

The LCDM predicts item responses using the Q-matrix mapping of the skills measured by each item, as illustrated in Equation 2 below using item 20, which measures knowledge of morphosyntactic rules (Skill 1) and knowledge of lexical rules (Skill 3). This results in Q-matrix entries for item 20 of $q_{20,1} = 1$ (because it measures morphosyntactic rules), $q_{20,2} = 0$ (because it does *not* measure cohesive rules), and $q_{20,3} = 0$ (because it measures lexical rules). Conditional on an examinee's mastery profile $\boldsymbol{\alpha}_e = [\alpha_{e1}, \alpha_{e2}, \alpha_{e3}]$, the LCDM then provides the following item response function for item 20 (and for any other item i that measures Skills 1 and 3) as:

$$P(X_{ei} = 1|\boldsymbol{\alpha}_e) = \frac{exp(\lambda_{i,0} + \lambda_{i,1,(1)}\alpha_{e1} + \lambda_{i,1,(3)}\alpha_{e3} + \lambda_{i,2,(1,3)}\alpha_{e1}\alpha_{e3})}{1 + exp(\lambda_{i,0} + \lambda_{i,1,(1)}\alpha_{e1} + \lambda_{i,1,(3)}\alpha_{e3} + \lambda_{i,2,(1,3)}\alpha_{e1}\alpha_{e3})} \tag{2}$$

More generally, the Q-matrix and the psychometric model provide a series of constraints on the general latent class model. The resulting model therefore has a fixed number of classes (set by the number of possible skill patterns) and a fixed item parameter structure (set by the entries of the Q-matrix and the DCM).

The LCDM item parameters are analogous to the differing levels of effects found in an analysis of variance (ANOVA) model. The mastery profile, $\boldsymbol{\alpha}_e = [\alpha_{e1}, \alpha_{e2}, \alpha_{e3}]$, uses dummy-coding to indicate whether examinee e has mastered skill s ($\alpha_{es} = 1$) or has not mastered skill s ($\alpha_{es} = 0$). Mastery status is then mapped onto the predicted item response by four item parameters: an intercept ($\lambda_{i,0}$) for the log-odds of a correct response in an examine who has not mastered either skill, two main effects ($\lambda_{i,1,(1)}$ and $\lambda_{i,1,(3)}$) that increase the log-odds of a correct response given mastery of each skill in the absence of the other, and a two-way interaction between the two skills ($\lambda_{i,2,(1,3)}$) that would further increase the log-odds of a correct response given mastery of both skills. The term within the exponent in Equation 2 becomes the τ_{ic} item threshold as predicted by the parameters of the LCDM and the mastery status of the examinee. For item 20, there are four possible threshold values: $\tau_{ic} = \lambda_{i,0}$ for examinees who have not mastered either skill ($\alpha_{e1} = 0$ and $\alpha_{e3} = 0$), $\tau_{ic} = \lambda_{i,0} + \lambda_{i,1,(1)}$ for examinees who have mastered only the morphosyntactic Skill 1 ($\alpha_{e1} = 1$ and $\alpha_{e3} = 0$), $\tau_{ic} = \lambda_{i,0} + \lambda_{i,1,(3)}$ for examinees who have mastered only the lexical Skill 3 ($\alpha_{e1} = 0$ and $\alpha_{e3} = 1$), and finally $\tau_{ic} = \lambda_{i,0} + \lambda_{i,1,(1)} + \lambda_{i,1,(3)} + \lambda_{i,2,(1,3)}$ for examinees who have mastered both skills ($\alpha_{e1} = 1$ and $\alpha_{e3} = 1$). To ensure that mastery of more skills results in a higher probability of a correct response as intended, the LCDM places order constraints on the item parameters for the main effects and interactions, as will be described (see also Henson, Templin, & Willse, 2009).

To demonstrate how the LCDM predicts an item response, we assign a hypothetical value each item parameter in Equation 2: intercept $\lambda_{i,0} = -2$, main effects $\lambda_{i,1,(1)} = 2$ and $\lambda_{i,1,(3)} = 1$, and the two-way interaction $\lambda_{i,2,(1,3)} = 0$. Table 2 shows the LCDM item response function for each of the four possible combinations of Skills 1 (morphosyntactic) and 3 (lexical: having mastered none, either, or both), through which the log-odds and probability of a correct item response can then be predicted. Each of the possible $2^3 = 8$ mastery profiles maps onto a profile in Table 2 based on the mastery status of Skills 1 and 3 in that profile. That is, the τ_{ic} item thresholds for the predicted log-odds of a correct response (which through Equation 1 can be converted into the π_{ic} latent class model probabilities) can take on one of four possible values given the specific mastery profile for the examinee's class $\alpha_e = \alpha_c$.

Using our example values, when neither Skill 1 (morphosyntactic) nor 3 (lexical) has been mastered by the examinee ($\alpha_{e1} = 0$ and $\alpha_{e3} = 0$), only the intercept ($\lambda_{i,0}$) predicts the item threshold, which is a log-odds of -2 (or a probability of .12) for a correct response here. That is, similar to ANOVA, the intercept is defined as the item threshold for a reference group of examinees that have not mastered either skill measured by the item. The main effects ($\lambda_{i,1,(1)}$ and $\lambda_{i,1,(3)}$) then predict the increase in the item threshold for possessing either Skill 1 or 3, respectively. An examinee who has mastered only the morphosyntactic Skill 1 will have a predicted item threshold of ($-2 + 1 = -1$). An examinee who has mastered only the lexical Skill 3 will have a predicted item threshold of ($-2 + 2 = 0$). Finally, the two-way interaction ($\lambda_{i,2,(1,3)}$) gives the additional change in the item threshold given mastery of both skills. Because the interaction term was 0 in this example, the item threshold for an examinee who has mastered both skills is just the sum of the intercept and main effects for each skill ($-2 + 1 + 2 = 1$). When the interaction term is 0, the item response function is said to be *compensatory*, in which a mastered skill can compensate for non-mastery of other skills. When the interaction term is different from 0 instead, such that the item threshold will be different than the sum of the intercept and main effects, this indicates that the skills are *non-compensatory*, or that having both skills provides an additional unexpected adjustment to the log-odds of a correct response. Similar to ANOVA, the LCDM can be used to test whether each interaction term is different from 0, thus allowing for a compensatory or non-compensatory model for each item as needed.

As demonstrated in Equation 2 for an item measuring two skills, the number of parameters in the LCDM item response function will depend on the number of skills measured by the item. The right-side columns of Table 1 list the parameters of the LCDM specified for each ECPE item given the left-side Q matrix. Items measuring one skill will only have an intercept and a main effect for that skill. All items measuring two skills will have an intercept, two main effects, and a two-way interaction for the pair of skills. Items measuring three skills will have an intercept, three main effects (one for each skill), three two-way interactions (one for each pair of skills), and one three-way interaction (for all three skills). In practice, however, higher-level interactions are frequently omitted if not statistically different from 0 or if they cause difficulty in estimation. The higher order interactions affect the item response probability for classes where multiple skills have been mastered. Because there are a small number of classes where these terms contribute to the response function, the impact of their omission is expected to be minimal. In general, we suggest that at least all two-way interactions be specified for items measuring more two or more skills so as to obtain an optimal balance between model complexity and estimation accuracy.

Mplus Syntax for Estimating the LCDM

We now describe Mplus syntax for estimating the LCDM item, structural, and examinee parameters for our example 28-item test measuring three skills (knowledge of morphosyntactic rules, cohesive rules, and lexical rules). The following section assumes basic knowledge of Mplus syntax, as described in more detail in the Mplus User's Guide (Version 6.11; Muthén & Muthén, 2013). Mplus syntax uses a series of commands, each with a specific purpose and set of options. The initial Mplus syntax includes four commands: (1) TITLE, (2) DATA, (3) VARIABLE, and (4) ANALYSIS, as found in the appendix. These initial syntax commands will remain relatively unchanged across DCMs, and so we provide only limited discussion of these commands below.

First in the syntax file is the optional TITLE command, whose text simply provides a title that will appear on each page of the output. Next is the DATA command, whose options provide the specifics of the data set to be analyzed. In our example, the DATA command only includes the name of our data file: *ecpedata.dat*. The format of our data is space-delimited (a space in between each column), with the data from each examinee located in a single row. The first column holds an identification number while the next 28 columns hold the ECPE item responses.

The VARIABLE command provides the names and characteristics of the variables in the data set and in the analysis, which include four options in our example. First, the NAMES option lists the names for all variables in the data file, including ID for the examinee identification number and x1 through x28 for the item responses. Second, the USEVARIABLE option lists which variables in the data file should actually be used in the analysis. By default, all variables listed in the NAMES command are included. For our analysis, however, we do not wish to analyze the ID variable, so we list variables x1–x28 under the USEVARIABLE option. Third, the CATEGORICAL option indicates that items x1–x28 are categorical response variables, which is necessary because our items are scored as 0/1 (incorrect/correct). For the variables listed under the

Table 2. Example LCDM Item Response Function for Item 20 of the ECPE

			Class-specific	
α_{e1}	α_{e3}	LCDM Item Response Function	Threshold $\tau_{20,c}$	Probability $\pi_{20,c}$
0	0	$\lambda_{20,0} + \lambda_{20,1,(1)}\,(0) + \lambda_{20,1,(3)}(0) + \lambda_{20,2,(1,3)}(0)(0)$	−2	.12
0	1	$\lambda_{20,0} + \lambda_{20,1,(1)}\,(0) + \lambda_{20,1,(3)}(1) + \lambda_{20,2,(1,3)}(0)(1)$	−1	.27
1	0	$\lambda_{20,0} + \lambda_{20,1,(1)}\,(1) + \lambda_{20,1,(3)}(0) + \lambda_{20,2,(1,3)}(1)(0)$	0	.50
1	1	$\lambda_{20,0} + \lambda_{20,1,(1)}\,(1) + \lambda_{20,1,(3)}(1) + \lambda_{20,2,(1,3)}(1)(1)$	1	.73

Note. The Q-matrix entry for Item 2 was [0, 1, 0]. The Q-matrix entry for item 20 was [1, 0, 1].

CATEGORICAL option, Mplus treats each numeric value as its own category level. The lowest numeric value, 0 in our data, will be labeled as CATEGORY 1 by Mplus, and all data values of 1 will be labeled as CATEGORY 2. Fourth, the CLASSES option specifies the label we will use to name the latent classes and the number of latent classes present. Here we are measuring three skills, resulting in $2^3 = 8$ latent classes for the possible skill profiles. Thus, the syntax c(8) denotes that c will be the label for our eight latent classes.

The Mplus ANALYSIS command is used to specify numerous options for estimation, of which we will include TYPE and STARTS. The TYPE option tells Mplus the type of analysis. Because DCMs are latent class models that are in turn subsumed into a larger group of finite mixture models (e.g., McLachlan & Peel, 2000), the option TYPE = MIXTURE will be needed. The STARTS option indicates the number of random starts to be used for the analysis, which is set to 10 by default in a mixture model without constrained parameters. Because of the constraints of the LCDM, we turn off this unnecessary option via STARTS = 0.

So far we have described the preliminary syntax for Mplus and how categorical outcomes in latent class models (and thus DCMs) are modeled in Mplus. We now describe three non-syntax steps needed before we can translate the LCDM predictions into model constraints in the Mplus syntax.

Creating a Class-to-Profile Table Relating Each Latent Class to a Skill Profile

Because each latent class represents a unique profile of three skills that are either mastered or not mastered, we must first map the skill profiles onto the eight latent classes. One way to do so is to create a *class-to-profile table*, as shown in Figure 1, in which all possible classes are listed down the rows and each skill is listed across the columns. The assignment of nonmastery (0) or mastery (1) of each skill in the profile proceeds by successively partitioning the latent classes into two sets (0 or 1) for each skill. Because our example measures three skills, three successive partitions are needed, as depicted in Figure 1 and described below. Further, this process will work for any number of measured skills, as the total number of skill profiles, 2^S, will always be divisible by two.

In the first step, mastery values are assigned to Skill 1 such that Classes 1 through 4 (the first half of the eight total classes) get a 0 for non-mastery of Skill 1. The remaining Classes 5 through 8 get a 1 for mastery of Skill 1. In the second step, we assign values for Skill 2. We partition only the classes where Skill 1 is set to 0 (Classes 1 through 4) into two equal sets: Classes 1 and 2 (the first half of the four classes) get a 0, and Classes 3 and 4 get a 1. For the remaining Classes 5 through 8, we repeat the pattern from Classes 1 through 4 (so Classes 5 and 6 get a 0, and Classes 7 and 8 get a 1). In the third step we assign values for Skill 3. We partition only the first classes of Skill 2 that were assigned a 0 (Classes 1 and 2): Class 1 gets a 0, and Class 2 gets a 1. We replicate this partitioning pattern for Classes 3 and 4 (3 gets a 0, 4 gets a 1), Classes 5 and 6 (5 gets a 0, 6 gets a 1), and Classes 7 and 8 (7 gets a 0, 8 gets a 1). Thus, through this process a unique profile of skills is assigned to each latent class, as shown in the second row of Table 3. For example, the skill profile for Class 4 is $\boldsymbol{\alpha}_4 = [0, 1, 1]$, indicating that although examinees in Class 4 have not mastered morphosyntactic rules (Skill 1), they have mastered cohesive rules (Skill 2) and lexical rules (Skill 3). Given $2^S = 2^3 = 8$ possible classes and 28 items, we can now use this class-to-profile table to specify the $8 \times 28 = 224$ possible τ_{ic} item thresholds.

Creating an LCDM Specification Table

Next, we use the LCDM item parameters and the Q-matrix entries to provide the item response function for each of the 224 class-specific item thresholds. For sake of illustration, only the LCDM predictions for items 2 and 20 are shown in the middle of Table 3. First, because item 2 only measures Skill 2 (cohesive rules) according to our Q-matrix (as seen in the left columns of Table 1), it has just two possible LCDM parameters: an intercept, $\lambda_{2,0}$, and a main effect for Skill 2, $\lambda_{2,1,(2)}$. Accordingly, each of the eight possible skill profiles will have one of two possible LCDM item response functions based on mastery of Skill 2. For skill profiles in which Skill 2 is *not* mastered (skill profiles $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \boldsymbol{\alpha}_5$, and $\boldsymbol{\alpha}_6$), the LCDM item response function is:

$$\tau_{2,c} = \lambda_{2,0} + \lambda_{2,1,(2)}\alpha_{c2} = \lambda_{2,0} + \lambda_{2,1,(2)}\,(0) = \lambda_{2,0}. \quad (3)$$

For skill profiles in which Skill 2 is mastered (profiles $\boldsymbol{\alpha}_3$, $\boldsymbol{\alpha}_4$, $\boldsymbol{\alpha}_7$, and $\boldsymbol{\alpha}_8$), the LCDM item response function is:

$$\begin{aligned}\tau_{2,c} &= \lambda_{2,0} + \lambda_{2,1,(2)}\alpha_{c2} = \lambda_{2,0} + \lambda_{2,1,(2)} \\ &= \lambda_{2,0} + \lambda_{2,1,(2)}. \end{aligned} \quad (4)$$

As such, each of the τ_{2c} class-specific item thresholds for item 2 must be defined by either Equation 3 for the classes in which Skill 2 is not mastered ($\tau_{2,1}$, $\tau_{2,2}$, $\tau_{2,5}$, and, $\tau_{2,6}$) or Equation 4 for the classes in which Skill 2 is mastered instead ($\tau_{2,3}$, $\tau_{2,4}$, $\tau_{2,7}$, and, $\tau_{2,8}$).

Second, because item 20 measures Skill 1 (morphosyntactic rules) and Skill 3 (lexical rules) according to our Q-matrix, it has four possible LCDM parameters: an intercept, $\lambda_{20,0}$, a main effect for Skill 1, $\lambda_{20,1,(1)}$, a main effect for Skill 3, $\lambda_{20,1,(3)}$, and a two-way interaction between Skills 1 and 3, $\lambda_{20,1,(3)}$. As seen in Table 2, each skill profile will have one of four possible item response functions based on mastery of Skills 1 and 3, resulting in one of four possible thresholds.

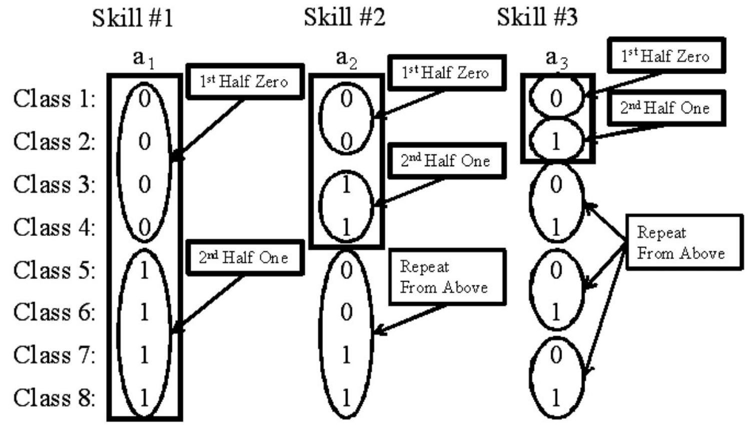

FIGURE 1. Schematic diagram of class-to-profile labels. Adaptation from Rupp, Templin, and Henson (2010).

Converting the LCDM Specification Table to an Item Threshold List

The last step before writing Mplus syntax is to build a *threshold list*: to determine which τ_{ic} item thresholds will be the same across classes and which LCDM parameters will predict each threshold. Each unique item threshold value (as taken from the LCDM specification table) will receive a separate label used by Mplus to equate these item threshold values across classes. For instance, item 2 had two possible threshold values resulting from either $\lambda_{2,0}$ (for classes in which the cohesive Skill 2 was not mastered) or $\lambda_{2,0} + \lambda_{2,1,(2)}$ (for classes in which the cohesive Skill 2 was mastered). Although arbitrary for Mplus, to help us keep them organized we will label the item thresholds using a format of: *T*[*i*]_[#]. The first character, *T*, will be used in each to indicate we are labeling a Threshold. The second character (in brackets to denote that the value will change) will index which item the threshold is for. The third character, an underscore, separates the item index from the final character, which holds the number of the threshold used for the item (which will also change).

For example, the threshold labels for item 2 (measuring only the cohesive Skill 2) appear at the bottom of Table 3, as constructed from the LCDM specification table as follows. The first possible threshold for item 2 is labeled *T2_1* and is used when only item 2's intercept predicts the log-odds of a correct response—that is, when the cohesive Skill 2 has not been mastered, which occurs in both Class 1 (skill profile $\boldsymbol{\alpha}_1 = [0, 0, 0]$) and Class 2 (skill profile $\boldsymbol{\alpha}_2 = [0, 0, 1]$). But Class 3 ($\boldsymbol{\alpha}_3 = [0, 1, 0]$), does have mastery of the cohesive Skill 2, so it gets a different threshold label, *T2_2*, which predicts the log-odds of a correct response using item 2's intercept plus the main effect for Skill 2. The *_2* part of the label reflects that this is the second threshold that appears for item 2. Class 4 ($\boldsymbol{\alpha}_4 = [0, 1, 1]$) also has mastery of the cohesive Skill 2, and so it also receives the *T2_2* threshold label. The remaining classes follow a similar pattern, with Class 5 ($\boldsymbol{\alpha}_5 = [1, 0, 0]$) and Class 6 ($\boldsymbol{\alpha}_6 = [1, 0, 1]$) receiving label *T2_1* because they do not have mastery of Skill 2, and Class 7 ($\boldsymbol{\alpha}_7 = [1, 1, 0]$) and Class 8 ($\boldsymbol{\alpha}_8 = [1, 1, 1]$) receiving label *T2_2* because they do have mastery of the cohesive Skill 2. Thus, rather than eight unique thresholds for item 2 across the eight latent classes, the LCDM instead estimates only two distinct thresholds, resulting in a much more parsimonious model. Table 3 also lists the threshold labels for item 20 measuring Skills 1 (morphosyntactic rules) and 3 (lexical rules), and thus for which four labels are needed to represent the four possible item thresholds (rather than eight unique thresholds) corresponding to non-mastery or mastery of Skills 1 and 3 (none, either, or both). The other item thresholds would be assigned to each class based on the relevance to that item of each mastered skill in their profile.

Mplus MODEL Command Syntax for the LCDM

Using the item threshold list just created, we can now write the Mplus MODEL command syntax that specifies the model within each latent class. The appendix provides syntax for items 2 and 20 within Classes 1, 2, and 8 (taken from Table 3). The *%c#1%* heading indicates that the text that follows defines the item thresholds specifically for Class 1 (skill profile $\boldsymbol{\alpha}_1 = [0, 0, 0]$). The first term [x2$1], refers to the threshold ($1) for item x2 (our name for item 2). Because they are dichotomous, each item requires only one model threshold (i.e., for the difference between an incorrect and correct response). The second term (T2_1), is the label we provide to place equality constraints on the threshold for item 2. Therefore, the line as a whole instructs Mplus that the first (and only) threshold of item 2 ([x2$1]) will be labeled T2_1. Critically, any threshold labeled *T2_1* will receive that same value across classes. The rest of the *%c#1%* section would list the thresholds and accompanying labels for *all* items within the first class. A similar syntax follows for the second class, beginning with *%c#2%*, after which threshold labels would again be specified for all items. In total, the threshold value for each item in every latent class will need to be constrained by the label assigned by the LCDM. Due to its size, the full MODEL syntax for all classes and all items is provided in an electronic appendix instead.

Mplus MODEL CONSTRAINT Syntax for the LCDM

After specifying the syntax labels for the item thresholds, next comes the MODEL CONSTRAINT command, in which each item threshold (represented by the label created in the MODEL section) will be predicted from the LCDM parameters (as shown for items 2 and 20 in Table 3). Further, given that the LCDM specifies that mastery of more skills must increase the log-odds of the probability of a correct response

Table 3. LCDM Specification Table for ECPE Items 2 and 20

Class:	Class 1 c#1	Class 2 c#2	Class 3 c#3	Class 4 c#4	Class 5 c#5	Class 6 c#6	Class 7 c#7	Class 8 c#8
Pattern:	$\alpha_1 = [0,0,0]$	α_2 [0,0,1]	α_3 [0,1,0]	$\alpha_4 = [0,1,1]$	$\alpha_5 =[1,0,0]$	$\alpha_6 = [1,0,1]$	$\alpha_7 = [1,1,0]$	$\alpha_8 = [1,1,1]$
Latent class model thresholds								
Item 2	$\tau_{2,1}$	$\tau_{2,2}$	$\tau_{2,3}$	$\tau_{2,4}$	$\tau_{2,5}$	$\tau_{2,6}$	$\tau_{2,7}$	$\tau_{2,8}$
Item 20	$\tau_{20,1}$	$\tau_{20,2}$	$\tau_{20,3}$	$\tau_{20,4}$	$\tau_{20,5}$	$\tau_{20,6}$	$\tau_{20,7}$	$\tau_{20,8}$
LCDM model formula								
Item 2	$\lambda_{2,0}$	$\lambda_{2,0}$	$\lambda_{2,0} + \lambda_{2,1,(2)}$	$\lambda_{2,0} + \lambda_{2,1,(2)}$	$\lambda_{2,0}$	$\lambda_{2,0}$	$\lambda_{2,0} + \lambda_{2,1,(2)}$	$\lambda_{2,0} + \lambda_{2,1,(2)}$
Item 20	$\lambda_{20,0}$	$\lambda_{20,0} + \lambda_{2,1,(3)}$	$\lambda_{20,0}$	$\lambda_{20,0} + \lambda_{2,1,(3)}$	$\lambda_{20,0} + \lambda_{2,1,(1)}$	$\lambda_{20,0} + \lambda_{2,1,(1)} + \lambda_{20,1,(3)} + \lambda_{2,2,(1,3)}$	$\lambda_{20,0} + \lambda_{2,1,(1)}$	$\lambda_{20,0} + \lambda_{2,1,(1)} + \lambda_{20,1,(3)} + \lambda_{2,2,(1,3)}$
Mplus threshold label								
Item 2	T2_1	T2_1	T2_2	T2_2	T2_1	T2_1	T2_2	T2_2
Item 20	T20_1	T20_2	T20_1	T20_2	T20_3	T20_4	T20_3	T20_4

Note. The Q-matrix entry for Item 2 was [0, 1, 0]. The Q-matrix entry for item 20 was [1, 0, 1].

(i.e., monotonicity), each item will also have a set of ordering constraint statements for all LCDM main effect and interaction parameters.

The appendix lists MODEL CONSTRAINT syntax for items 2 and 20. The first line for item 2, *NEW(L2_0 L2_12)*, instructs Mplus to create two new variables named *L2_0* and *L2_12* for the LCDM model parameters for item 2. We must use another naming convention given that Mplus only allows eight characters in these names. The first letter *L* refers to being an LCDM parameter (Lambda: λ). The second character gives the item (item 2 here). The character after the underscore represents the level of the effect: 0 for an intercept, 1 for a main effect, and 2 for a two-way interaction. The characters following the level of the effect refer to which skills(s) are involved in the effect (literally which α multiply the parameter in the LCDM from Equation 2). Here, the label for the intercept ($\lambda_{2,0}$) is *L2_0* and for the main effect of the cohesive Skill 2 ($\lambda_{2,1,(2)}$) is *L2_12*.

The next line of syntax predicts the first threshold value for item 2 (T2_1) from the LCDM parameters. But because Mplus models the log-odds for the probability of an incorrect response rather than a correct response, each item threshold equation must be multiplied by -1. From Table 2, we see that *T2_1* is used when only the item 2's intercept predicts the log-odds of a correct response, so only the label for the intercept (L2_0) appears in parentheses. Similarly, the next line provides the LCDM equation to predict the second threshold value for item 2 (T2_2). Table 2 lists the second threshold for item 2 as predicted by item 2's intercept (L2_0) plus the main effect for the cohesive Skill 2 (L2_12), so the resulting equation is T2_2 $= -1($L2_0 $+$ L2_12$)$. The final portion of the syntax for item 2 provides the ordering constraints for the LCDM parameters. Specifically, all main effects in the LCDM must be positive to ensure that masters of a skill have a higher probability of answering the item correctly than non-masters. The syntax indicating this constraint is *L2_12>0*, which states that the label for the main effect *L2_12* must be greater than 0. All other items measuring one skill will have similar syntax, with changes for the notation for the main effect (i.e., *L3_11* refers to the main effect of the morphosyntactic Skill 1 for item 3).

The syntax for item 20 demonstrates the ordering constraints when two skills are measured by an item. Because the LCDM specifies all possible main effects and interactions among skills, as the number of skills measured per item increases, so does the number of LCDM parameters. As such, item 20 has four LCDM parameters: its intercept (L20_0), two main effects (L20_11 for morphosyntactic Skill 1 and L20_13 for lexical Skill 3), and a two-way interaction between Skills 1 and 3 (L20_213). The syntax for the first three thresholds (T20_1, T20_2, and T20_3) is a direct extension of that for item 2, including only the intercept or the intercept plus a main effect. The fourth threshold (T20_4) represents the sum of all possible LCDM parameters for item 20, which includes its intercept, both main effects, and a two-way interaction (L20_212). As such, item 20 now has two additional ordering constraints for the interaction so that the log-odds of a correct response will be greater when both skills are mastered than when only one skill is mastered. Specifically, the fourth threshold (for classes in which Skills 1 and 3 are mastered) must be greater that the third threshold (for classes in which only Skill 1 is mastered) and greater than the second threshold (for classes in which only

Skill 3 is mastered). These constraints are expressed by two inequalities:

$$\tau_{20,4} > \tau_{20,3} \rightarrow \lambda_{20,0} + \lambda_{20,1,(1)} + \lambda_{20,1,(3)} + \lambda_{20,2,(1,3)} > \lambda_{20,0} + \lambda_{20,1,(1)} \rightarrow \lambda_{20,2,(1,3)} > -\lambda_{20,1,(3)}, \tag{5}$$

and

$$\tau_{20,4} > \tau_{20,2} \rightarrow \lambda_{20,0} + \lambda_{20,1,(1)} + \lambda_{20,1,(3)} + \lambda_{20,2,(1,3)} > \lambda_{20,0} + \lambda_{20,1,(3)} \rightarrow \lambda_{20,2,(1,3)} > -\lambda_{20,1,(1)}. \tag{6}$$

These inequalities are expressed by the lines *L20_213>-L20_11* and *L20_213>-L20_13*. All items measuring two skills will have similar constraints to ensure monotonicity across all skill profiles. For higher level interactions, the inequalities are always formed by comparing the interaction term to each of the thresholds representing the level of the effect immediately below the interaction term. For instance, a threshold including a three-way interaction would be compared to the three thresholds including a two-way interaction, and so forth.

The Remaining Mplus Syntax

Finally, just two other Mplus commands remain. As shown in the appendix, the TECH10 option on the OUTPUT command requests additional model fit statistics not given by default. The SAVE option on the SAVEDATA command instructs Mplus to save to an external file the examinee estimates of the probability of membership in each class (i.e., the probability of each mastery profile for each examinee). The FORMAT option specifies the format in which the data are to be saved as *f10.5*—a total of 10 digits per estimate, with 5-decimal precision. Finally, the FILE = option specifies the name and path of the external file of examinee estimates. Here, we call the file *ecpe_examinee.dat*, which will be saved to the folder with our syntax file by default.

Understanding Mplus Output for LCDM Parameters

Once the full syntax file has been constructed, it is then submitted to Mplus by using either the RUN button found along the top of the user interface or through calling Mplus in batch mode from the command line syntax in Windows. Assuming there are no syntax errors, once Mplus terminates successfully the output will appear in a text file with the same name as the input file, ending with the extension *.out*. We now describe how to interpret the sections of Mplus output relevant to the LCDM. Although Mplus output is voluminous, most key LCDM output is contained in just a few sections. Using the Mplus headings and going in order of appearance in the output file, we will describe two important sections: *final class counts and proportions* and *new/additional parameters*. Both of these output sections appear in abbreviated form following the syntax in the appendix. After describing these model output sections, we then discuss and interpret the estimates in the saved external file of examinee posterior probabilities.

Final Class Counts and Estimated Proportions Output

We begin with the output section for the *final class counts and estimated proportions*. For each skill profile (latent class c), the estimated number of examinees with that profile is shown along with the DCM structural parameters, υ_c, for the proportion of the sample that is a member of class c. Mplus reports this information in several ways: *based on the estimated model*, *based on estimated posterior probabilities*, and *based on the most likely class*. We focus on the results *based on the estimated model* that provides the most likely estimate of each structural model parameter. These results are technically marginal maximum likelihood estimates, which provide several beneficial statistical properties.

The output section has three columns, the first beginning with the label *latent classes*. Results are given for each latent class simply numbered 1 through 8, such that we must provide each latent class with our assigned skill profiles (as in Figure 1). The second column provides the expected count of respondents with each class. This count is found by multiplying the value from the third column, the estimated υ_c parameter from the latent class model, by the sample size. For our example data, we see that Class 1 (representing skill profile $\boldsymbol{\alpha}_1 = [0, 0, 0]$) has an estimate of $\upsilon_1 = 0.30074$. This means that approximately 30% of our sample is expected to have mastered none of the three skills measured by the ECPE (or .30074*2922 = 878.8 examinees). Skill profile $\boldsymbol{\alpha}_8 = [1, 1, 1]$ has an estimate of $\upsilon_8 = .34561$, meaning that approximately 34.6% of our sample (or 1009.9 examinees) is expected to have mastered every skill measured by the ECPE. The other parameter estimates in this section can be interpreted similarly.

New/Additional Parameters Output—LCDM Item Parameter Estimates

The next section of output is perhaps the most relevant to the process of estimating and calibrating the LCDM from our data. The *New/Additional Parameters* section contains the parameter estimates for all LCDM model parameters. The parameters appearing in this section were created by the use of the NEW option under the MODEL CONSTRAINT command. The values of the LCDM parameters were set by the syntax equating the LCDM model to each item threshold. Therefore, this section will contain all the information that is needed to assess how well each item actually measures the skills it is supposed to be based on the Q-matrix.

The *New/Additional Parameters* output contains five columns. The first column contains the names of the parameters. These are the labels created for each of the LCDM parameters under the MODEL CONSTRAINT command. The values under the *Estimate* column are the estimated values of the LCDM item parameters. The third column, *S.E.*, contains the standard errors of the estimated LCDM item parameters. The fourth column is a Wald test statistic for the parameter (Est./S.E.), and the fifth column is a two-tailed p-value providing an approximate test of the null hypothesis that the parameter is equal to 0. If the test is non-significant, we can omit an LCDM parameter from a model without significantly affecting measurement precision or model fit.

To illustrate, we examine the output for items 2 and 20. Item 2 measured the cohesive rules skill, giving it two item parameters, an intercept $\lambda_{2,0} = 1.037$ and the main effect of mastering the cohesive skill $\lambda_{2,1(2)} = 1.247$. The intercept is the log-odds of a correct item response for the reference group, examinees who have not mastered the cohesive rules skill. When converted into a probability, the intercept shows that examinees not mastering this skill have a .74 probability of answering the item correctly. The main effect of cohesive rules is the increase in the log-odds of a correct for examinees mastering the cohesive rules skill, giving these examinees a log-odds of a correct response of 1.037+1.247 = 2.284. When converted to a probability, this indicates that examinees who were masters of the cohesive rules skill had a .91 probability of answering the item correctly. The size of the intercept indicates that item 2 may not be well-measured by the cohesive rules skill, given that non-masters of the skill still have a high chance of answering the item correctly. Results such as this can indicate a lack of evidence for validity for the item (i.e., it does not measure the cohesive rules skill especially well), item misfit (i.e., the Q-matrix entry is lacking one or more skills that are actually being measured by the item), or simply that the item is very easy for examinees.

Likewise, the estimated intercept for item 20, $\lambda_{20,0} = -1.389$ (label *L20_0*), is the log-odds of a correct response to item 20 for examinees mastering neither of the skills it measures (morphosyntactic and lexical rules). When converted into a probability, this means that examinees without these two skills have a .20 probability of getting item 20 correct. We are generally unconcerned whether the intercept is significantly different from 0 (i.e., in which 0 indicates that non-masters have a 50% chance of a correct response). The estimated main effect for Skill 1 of morphosyntactic rules, $\lambda_{20,1,(1)} = .243$ (label *L20_11*), is the increase in the log-odds of a correct response to item 20 for mastering Skill 1 (in examinees who have not mastered Skill 3). The larger its main effect of Skill 1, the more item 20 discriminates between masters and non-masters of Skill 1 (morphosyntactic rules). Similarly, the estimated main effect for Skill 3 of lexical rules, $\lambda_{20,1,(3)} = .908$ (label *L20_13*), is the increase in the log-odds of a correct response to item 20 for mastering Skill 3 (in examinees who have not mastered morphosyntactic Skill 1). Finally, the two-way interaction parameter, $\lambda_{20,2,(1,3)} = 1.410$ (label *L20_213*), is the additional increase in log-odds of a correct response to item 20 when both Skills 1 and 3 are mastered (often called the over-additive effect). Because its p-value = .239 indicates that the two-way interaction between morphosyntactic rules and lexical rules is *not* significantly different from 0, the two-way interaction for item 20 can be omitted from the model, indicating that Skills 1 and 3 have a compensatory relationship for item 20. Table 1 lists the estimated values for all LCDM parameters for the ECPE items, which can be interpreted in a similar fashion.

Examinee Estimates: Probabilities of Class Membership

Our SAVEDATA command instructed Mplus to save examinee estimates into an external file called *ecpe_examinee.dat*. Table 4 lists entries in this file for five selected examinees. The first 28 columns of the file (omitted from the table) contain the examinee's responses to the 28 ECPE items, followed by the examinee's original ID variable so that the external file can be merged back into the original data file. Next are a series of eight probabilities that the examinee is a member of each latent class, one for each possible skill profile. We label the probability that examinee e has profile c as $\hat{\boldsymbol{\alpha}}_{ec}$ (in boldface to indicate the probability is for the entire profile of skills). The next column in the table displays the skill profile estimate for the examinee, as based on the latent class for which the examinee had the highest probability of membership. Although not in the output directly (but resulting from computations of the output to be described), the final three columns of the table, labeled $\hat{\alpha}_{es}$ (without boldface to indicate the probability is for a single skill) give the marginal probability that examinee e is a master of skill s.

To illustrate, we consider the entries for examinee 1 in the first row in Table 4. Examinee 1 answered 26 of the 28 items correctly, suggesting he or she is likely to have mastered all three skills of the test. Consequently, the skill profile with the highest probability for examinee 1 is profile 8 ($\boldsymbol{\alpha}_8 = [1, 1, 1]$) with $\hat{\boldsymbol{\alpha}}_{1,8} = .96$. This means that examinee 1 has a 96% chance of having skill profile $\boldsymbol{\alpha}_8$ that specifies mastery on all three skills. The estimated probability for profile 6, $\hat{\boldsymbol{\alpha}}_{1,6} = .04$, (for $\boldsymbol{\alpha}_6 = [1, 0, 1]$), indicates that although examinee 1 answered nearly every item correctly, the combination of the two items answered incorrectly (item 4 measuring Skill 3 and item 24 measuring Skill 2) made it slightly possible (a 4% chance) that the examinee had profile 6 in which Skill 2 was not mastered ($\boldsymbol{\alpha}_6 = [1, 0, 1]$) instead of profile 8 in which Skill 2 was mastered ($\boldsymbol{\alpha}_8 = [1, 1, 1]$). Each response pattern yields a specific set of posterior probabilities, so examinees with the same response patterns will have the same estimates of profile membership.

These profile probability estimates are a straightforward way to provide diagnoses about the total skills profile for examinees, but another form of an examinee estimate, the *skill mastery probability*, provides additional useful information about each individual skill rather than the overall profile of skills. The formula for the probability of mastery for skill s, denoted by $\hat{\alpha}_{es}$ (without boldfaced α, given a single skill and not a profile), comes from the expected value for each skill: summing the product of the skill profile probability times the value for skill s (0 for non-masters, 1 for masters) across each possible skill mastery profile (latent class) c, or $\boldsymbol{\alpha}_{cs}$:

$$\hat{\alpha}_{es} = \sum_{c=1}^{2^S} \hat{\boldsymbol{\alpha}}_{ec} \boldsymbol{\alpha}_{cs} \tag{7}$$

Using the formula, we find that examinee 1 has a skill mastery probability for Skill 1 of $\hat{\alpha}_{1,1} = 1.00$, for Skill 2 of $\hat{\alpha}_{1,2} = .96$, and for Skill 3 of $\hat{\alpha}_{1,3} = 1.00$. These values summarize the strength of the evidence obtained from the item responses through the LCDM that a given examinee is a master of each skill. In the case of examinee 1, we are virtually certain he or she is a master of all three skills measured by the test: morphosyntactic rules, lexical rules, and cohesive rules. Thus, even though the model is specified such that each skill is either mastered or not, the probabilities provided by the LDCM for the overall profile and the mastery of each skill can still be used to more accurately convey the shades of gray that exist in assessing the abilities of real people. This type of examinee feedback can be used to provide tailored instruction or tutoring plans for examinees that have yet to master the skills measured by the test. For instance, examinees who are not a master of a given skill (i.e., that have a predicted probability less than .50), can be given extra instruction on that skill specifically.

Discussion

In this article we have shown how diagnostic classification models (DCMs) can provide informative results to analysts, evaluators, and examinees. Unfortunately, development and application of DCMs have primarily been conducted using *ad hoc* software written to estimate only specific model variants. Further, the software currently available for fitting DCMs is not easily obtainable for most users, either due to the use of advanced estimation techniques that require a high level of technical proficiency (such as the Markov Chain Monte

Table 4. Estimated Examinee Posterior Probabilities for Four Examinees

ID	Skill Profile Probability Estimates $\hat{\alpha}_{e1}$	$\hat{\alpha}_{e2}$	$\hat{\alpha}_{e3}$	$\hat{\alpha}_{e4}$	$\hat{\alpha}_{e5}$	$\hat{\alpha}_{e6}$	$\hat{\alpha}_{e7}$	$\hat{\alpha}_{e8}$	Max $\hat{\alpha}_{ec}$	Skill Mastery Probability $\hat{\alpha}_{e1}$	$\hat{\alpha}_{e2}$	$\hat{\alpha}_{e3}$
1	.00	.00	.00	.00	.00	.04	.00	**.96**	[1,1,1]	**1.00**	**.96**	**1.00**
10	.34	.03	.00	.01	**.46**	.02	.10	.04	[1,0,0]	**.62**	.15	.10
14	.46	**.49**	.00	.05	.00	.00	.00	.00	[0,0,1]	.00	.05	**.54**
29	.02	.41	.00	.01	.03	**.46**	.00	.07	[1,0,1]	**.56**	.08	**.95**
33	**.75**	.00	.04	.00	.07	.00	.12	.02	[0,0,0]	.21	.18	.02

Note. Boldface entries represent highest probability across all profiles, $\hat{\alpha}_{ec}$, and for each skill separately, $\hat{\alpha}_{es}$. Max $\hat{\alpha}_{ec}$ is frequently given to the examinee as the reported attribute profile.

Carlo methods found in Arpeggio; Bolt et al., 2008) or due to restrictive research license requirements (i.e., as found with programs such as MDLTM; von Davier, 2006). The field of diagnostic modeling is still in its infancy, and its potential contributions to educational assessment will only be realized through continued research on and with these models. We hope that our demonstration of how DCMs can be implemented within Mplus will be a helpful step towards this end goal.

The analysis of the ECPE served to demonstrate how to obtain DCM item parameter and examinee probability estimates. In practice, the process of evaluating DCM information begins with an investigation of the item parameters estimates. First, items that have relatively high intercepts indicate a large number of examinees who are *not* masters of their measured skills are still answering the item correctly, which could indicate a misfit in the Q-matrix. In our example, the intercept for item 2 was large (i.e., non-masters still had 74% probability of answering item 2 correctly). Depending on the item, one could change Q-matrix entries for the item (i.e., as warranted by the empirical work or theory surrounding the measured skills), or one may consider specifying a new skill altogether (i.e., a new column in the Q-matrix). Second, for items measuring more than one skill, the interaction terms should be inspected to determine if they are needed. In our analysis, from Table 1 we can see that item 21 (measuring the morphosyntactic rules and lexical rules skills) appears to have a very low estimate for the interaction (.042). We can remove this interaction from the item by deleting the entry for it in the MODEL CONSTRAINT section of the Mplus syntax. Finally, for items without interactions, the main effects for each skill should be inspected for statistical significance to determine if they indicate the skill is being measured by the item. Each of our items without interactions had main effects that were somewhat large (the smallest was the main effect for the lexical rules skill on item 26, which was 1.119), indicating that each item measured its skill at an acceptable level. We note that item main effects can only be removed from a model in the absence of any interaction terms involving their attributes. Upon inspection, the Q-matrix can be refined, and the model can be re-estimated in Mplus.

The process of examining the results of the analysis leads to a discussion of the validity of the tests and the skills themselves. At the test level, the skill information can help indicate that the test is measuring the relevant content. For example, small parameters for every item measuring a given skill would cast doubt as to whether or not the skill is actually present in the test. Validity is also relevant at the skill level. When providing skill-level feedback to an examinee, it is important to ascertain that the skill measured is what it is purported to be. This information, however, cannot be obtained solely from the results of an analysis but must be evaluated using evidence external to a test. Methods for investigating the validity of skills measured by a test can follow methods for investigating any latent trait, and can include references to external criteria (e.g., do masters of a skill perform better on related tasks). Although different in form, the skills measured by a DCM are no different in function from the traits measured in any classical psychometric procedure. Therefore, their properties and the validity of each must be continually evaluated to justify their use in practice.

Once the item parameters appear to be acceptable and the traits being measured have evidence for their validity, then examinee probabilities can be used. In this sense, examinee probabilities provide a roadmap for what examinees should focus on during instruction and remediation. Examinees with very low probabilities of mastery for skills should focus on learning those skills, rather than focus on skills that already have high probabilities of mastery. In practice, though, teaching is not as easily segmented into separate skills as it may appear. Thus, it may be difficult to find methods for instruction for fine-grain skills that are embedded within large skills or for content areas that are not easily disentangled. Therefore, skill-level information may be useful in specific situations but potentially less useful in others. As such, there is a need to build tests for use with DCMs rather than to adapt existing tests for DCM use in order to ensure that useful skills are incorporated into the test. Knowledge of the precise type of skill level information needed to augment instruction is critically important, as then test items can be constructed to measure such information and Q-matrices can be built to model such items.

Practically, although our choice of Mplus was motivated by its widespread availability, the use of Mplus for DCM analyses is not without limitations. Because Mplus is a general program that estimates a host of psychometric and statistical models, estimation time can be lengthy. For Q-matrices with six or fewer skills, estimation time can take several hours. Each additional skill brings about an exponential increase in estimation time, making Mplus ill-suited for Q-matrices with more than six skills. Furthermore, although Mplus is limited only by the size of the memory in the computer on which it is installed, in our experience Mplus has difficulty in estimating more than 80 items. In tests with more than six skills or more than 80 items, we recommend the use of software created by psychometricians in the field of DCMs (such software can be obtained by contacting the first author).

In addition, as the reader might have guessed by now, specification of the syntax to estimate diagnostic classification models can be very labor-intensive and error-prone, especially for models with large numbers of items measuring many skills. To facilitate this process, we have developed a SAS program (freely available from the authors) to do most of this work for us. The SAS program requires the user to input a Q-matrix mapping items to the skills they measure and to provide information about the data file (e.g., file names, number of items, and number of skills). The SAS program then writes the Mplus syntax based on data and its Q-matrix, executes Mplus, and parses the output into SAS data sets. It is our hope that this SAS program can further facilitate research with and using diagnostic classification models as estimated in Mplus.

As Mplus syntax can be difficult to construct for estimation of DCM model parameters, we expect educational measurement practitioners are most likely to build the necessary syntax, rather than classroom teachers. However, if model parameters have already been obtained though precalibration with existing data, Mplus can also be used to provide examinee skill probabilities with new test data. In fact, once a calibration syntax file containing item parameters exists, Mplus can be run with a new data file using the same syntax file. In these situations, we expect that anyone with access to Mplus can readily obtain examinee skill mastery probabilities. Because item parameter estimation takes the bulk of the time, such examinee estimation only takes a fraction of a second. As such, DCM-scored tests could be built with Mplus scoring files, and teachers or practitioners can obtain examinee estimates from data. For more information about how to build such files, we refer the reader to Rupp, Templin, and Henson (2010).

In summary, the steps outlined in this article show how to estimate the parameters of the LCDM with a readily available and user-friendly software package, Mplus. We showed how the LCDM can be specified as a confirmatory latent class model so that its estimation becomes possible using more general software programs. Furthermore, because many commonly used DCMs can be fitted through the LCDM parameterization, the methods explicated in this article will be useful for many different model variants. Additionally, the Mplus package is flexible enough to be used in many real-world testing situations, handling missing data and differing response variable types with ease. In addition, the MONTECARLO command in Mplus can be used to conduct simulation studies for psychometric research, further expanding the software's capacity to provide information about the statistical properties of DCMs. It is our hope that these gains in accessibility and practicality through more widely available software like Mplus will help spur further development of and applications with models for diagnostic assessment.

Appendix

Abbreviated Mplus Syntax and Relevant Output

```
SYNTAX:
!---------------------------------------------------------------------------
TITLE:                      ! Section that appears in header of output file
    DCM for ecpedata.csv with 3 skills,
    28 items, and maximum 2-order item model.
    Saturated structural model (Mplus default).

DATA:
    FILE = ecpedata.dat;    ! Location of free format data file

VARIABLE:
    NAMES = ID x1-x28;      ! List of variables in data file
    USEVARIABLE = x1-x28;   ! Variables to be analyzed
    CATEGORICAL = x1-x28;   ! Binary outcomes
    CLASSES = c(8);         ! Number of possible skill patterns (2^S)
    IDvariable = ID;        ! Person ID variable to save respondent data

ANALYSIS:
    TYPE = MIXTURE;         ! Estimates latent classes
    STARTS = 0;             ! Turn off multiple random start feature

MODEL:

%c#1%  ! Model for Class 1 – Profile [0,0,0]
!----{code for other items omitted}----!
    [x2$1]  (T2_1);         ! Threshold 1 for item 2
    [x20$1] (T20_1);        ! Threshold 1 for item 20

%c#2%  ! Model for Class 2 – Profile [0,0,1]
!----{code for other items omitted}----!
    [x2$1]  (T2_1);         ! Threshold 1 for item 2
    [x20$1] (T20_2);        ! Threshold 2 for item 20
```

```
!----{code for other classes omitted}----!

%c#8%  ! Model for Class 8 - Profile [1,1,1]
    [x2$1]  (T2_2);         ! Threshold 2 for item 2
    [x20$1] (T20_4);        ! Threshold 4 for item 20

MODEL CONSTRAINT:           ! Used to define LCDM parameters
                            ! Mplus uses P(X=0) so multiply by -1
! Item 2: Define LCDM parameters present for item 2
! Q-matrix entry: [0,1,0]
NEW(L2_0 L2_12);            ! Define two LCDM parameters for the item
T2_1=-(L2_0);               ! Item 2 Threshold 1
T2_2=-(L2_0+L2_12);         ! Item 2 Threshold 2

L2_12>0;                    ! Main effect order constraints

! Item 20: Define LCDM parameters present for item 20
! Q-matrix entry: [1,0,1]
NEW(L20_0 L20_11 L20_13 L20_213); ! Define four LCDM parameters for the item
T20_1=-(L20_0);                   ! Item 20 Threshold 1
T20_2=-(L20_0+L20_13);            ! Item 20 Threshold 2
T20_3=-(L20_0+L20_11);            ! Item 20 Threshold 3
T20_4=-(L20_0+L20_11+L20_13+L20_213); ! Item 20 Threshold 4

! Main effect order constraints
L20_11>0;
L20_13>0;

! Two-way interaction order constraints
L20_213>-L20_11;
L20_213>-L20_13;

OUTPUT:
    TECH10;                       ! Request additional model fit statistics

SAVEDATA:
    FORMAT = F10.5;
    FILE = ecpe_examinee.dat;     ! Skill estimates per respondent
    SAVE = CPROBABILITIES;        ! Save skill profile probabilities

!-------------------------------------------------------------------------
!OUTPUT
!-------------------------------------------------------------------------
FINAL CLASS COUNTS AND PROPORTIONS FOR THE LATENT CLASSES
BASED ON THE ESTIMATED MODEL

    Latent
   Classes
       1       878.75529         0.30074
       2       376.88999         0.12898
       3        34.83030         0.01192
       4       511.57020         0.17508
       5        25.53837         0.00874
       6        53.02421         0.01815
       7        31.52607         0.01079
       8      1009.86558         0.34561
```



```
!------------------------------------------------------------------------

New/Additional Parameters
                                                            Two-Tailed
                    Estimate       S.E.  Est./S.E.    P-Value

    L2_0               1.037      0.089     11.619      0.000
    L2_12              1.247      0.155      8.059      0.000

    L20_0             -1.389      0.114    -12.150      0.000
    L20_11             0.243      1.168      0.208      0.835
    L20_13             0.908      0.159      5.712      0.000
    L20_213            1.410      1.197      1.178      0.239
!------------------------------------------------------------------------
```


Acknowledgments

This research was funded by National Science Foundation Grants DRL-0822064, SES-0750859, and SES-1030337. We would like to thank the editor and three anonymous reviewers for their helpful comments and insights that improved the quality of this manuscript. All example syntax and the SAS macro can be found on the first author's website: http://www.jonathantemplin.com

References

Bolt, D., Chen, H., DiBello, L., Hartz, S., Henson, R., Roussos, L., Stout, W., & Templin, J. (2008). *The Arpeggio Suite: Software for cognitive skills diagnostic assessment* [Computer software]. Saint Paul, MN: Assessment Systems.

Buck, G., & Tatsuoka, K. K. (1998). Application of the rule-space procedure to language testing: Examining attributes of a free response listening test. *Language Testing*, *15*, 119–157.

de la Torre, J. (2009). DINA model and parameter estimation: A didactic. *Journal of Educational and Behavioral Statistics*, *34*, 115–130.

Dimitrov, D. M. (2007). Least squares distance method of cognitive validation and analysis for binary items using their IRT parameters. *Applied Psychological Measurement*, *31*, 367–387.

Embretson, S. E. (1995). A measurement model for linking individual learning to process and knowledge: Application to mathematical reasoning. *Journal of Educational Measurement*, *20*, 277–294.

Haertel, E. (1989). Using restricted latent class models to map the skill structure of achievement items. *Journal of Educational Measurement*, *26*, 333–352.

Hartz, S. M. (2002). *A Bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practicality* (Unpublished doctoral dissertation). University of Illinois at Urbana-Champaign, Urbana, IL.

Henson, R., & Templin, J. (2007, April). *Large-scale language assessment using cognitive diagnosis models*. Paper presented at the Annual meeting of the National Council for Measurement in Education, Chicago, IL.

Henson, R., Templin, J., & Willse, J. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, *74*, 191–210.

Junker, B. W., & Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, *25*, 258–272.

Kane, M. T. (2006). Validation. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 17–64). Portsmouth, NH: Greenwood.

Lazarsfeld, P. F., & Henry, N. W. (1968). *Latent structure analysis*. Boston, MA: Houghton Mifflin.

Leighton, J. P., & Gierl, M. J. (Eds.) (2007), *Cognitive diagnostic assessment for education: Theory and applications*. Cambridge, UK: Cambridge University Press.

Macready, G. B., & Dayton, C. M. (1977). The use of probabilistic models in the assessment of mastery. *Journal of Educational Statistics*, *2*, 99–120.

McLachlan, G., & Peel, D. (2000). *Finite mixture models*. New York, NY: Wiley.

Muthén, L. K., & Muthén, B. O. (2013). *Mplus user's guide (Version 6.1)* [Computer software and manual]. Los Angeles, CA: Muthén & Muthén.

Rupp, A. A. (2007). The answer is in the question: A guide for investigating the theoretical potentials and practical limitations of cognitive psychometric models. *International Journal of Testing*, *7*, 95–125.

Rupp, A., & Templin, J. (2008). Unique characteristics of diagnostic models: A review of the current state-of-the-art. *Measurement*, *6*, 219–262.

Rupp, A., Templin, J., & Henson, R. (2010). *Diagnostic measurement: Theory, methods, and applications*. New York, NY: Guilford Press.

Tatsuoka, K. K. (1983). Rule-space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, *20*, 345–354.

Templin, J., & Bradshaw, L. (in press). Measuring the reliability of diagnostic classification model examinee estimates. *Journal of Classification*.

Templin, J. L., & Henson, R. A. (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological Methods*, *11*, 287–305.

von Davier, M. (2006). *Multidimensional latent trait modeling (MDLTM)* [Computer software]. Princeton, NJ: Educational Testing Service.

