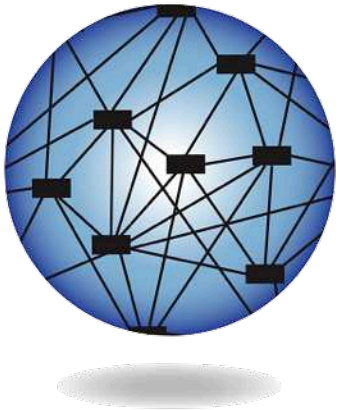

DYNAMIC®
LEARNING MAPS

Bayesian Psychometrics for Diagnostic Assessments: A Proof of Concept

Research Report #19-01

November 2019

Thompson, W. J. (2019). *Bayesian psychometrics for diagnostic assessments: A proof of concept* (Research Report No. 19-01). Lawrence, KS: University of Kansas, Accessible Teaching, Learning, and Assessment Systems.

Contents

List of Tables

List of Figures

Executive Summary

Diagnostic assessments measure the knowledge, skills, and understandings of students at a smaller and more actionable grain size than traditional scale-score assessments. Results of diagnostic assessments are reported as a mastery profile, indicating which knowledge, skills, and understandings the student has mastered and which ones may need more instruction. These mastery decisions are based on probabilities of mastery derived from diagnostic classification models (DCMs).

This report outlines a Bayesian framework for the estimation and evaluation of DCMs. Specifically, this report describes the following:

- a model definition that allows for various parameter equality constraints within a consistent conceptual framework
- the role of prior distributions in the model building process
- an estimation process utilizing the popular *Stan* programming language
- the assessment of estimation diagnostics, such as the $\widehat{R}$ and effective sample size
- the evaluation of model fit using posterior predictive model checks
- model comparison using the cross-validation approximations and model averaging

Findings illustrate the utility of the Bayesian framework for estimating and evaluating DCMs in applied settings. Specifically, the findings demonstrate how a variety of DCMs can be defined within the same conceptual framework. Additionally, using this framework, the evaluation of model fit is more straightforward and easier to interpret with intuitive graphics. Throughout, recommendations are made for specific implementation decisions for the estimation process and the assessment of model fit.

Implications for the Field

DCMs offer many benefits over traditional scale-score reporting methods. For example, DCMs can provide more actionable results through a fine-grained mastery profile (Feldberg & Bradshaw, 2019; Clark & Karvonen, 2019) and more reliable scores with a shorter test length (Templin & Bradshaw, 2013; Wang, Song, Chen, Meng, & Ding, 2015). However, despite a growing field of literature describing the benefits of DCM-based assessments, these models have not seen wide-spread use in applied or operational settings (Sessoms & Henson, 2018). One reason put forward for this gap between the theory and practice of DCMs is a lack a clarity in the applied research community for how these models should be estimated and evaluated (Ravand & Robitzsch, 2015; Ravand & Baghaei, 2019; Rupp & van Rijn, 2018). This report attempts to bridge the gap between theory and practice by describing a Bayesian framework for estimating DCM models using the *Stan* programming language and evaluating model fit using posterior predictive model checks.

This framework, which is used in an applied setting for the Dynamic Learning Maps® (DLM®) alternate assessment, provides a flexible method for defining different types of DCMs. Additionally, the model estimation processes and model fit measures are applicable to the variations in model definition. That is, the same estimation and evaluation procedures can be applied to a wide range of DCMs. Thus, this report provides a practical guide for applied researchers in order to integrate DCMs into their own work.

1. Purpose of the Report

Diagnostic classification models (DCMs) are able to provide fine-grained and actionable scores for a set of assessed skills or attributes (Rupp, Templin, & Henson, 2010; Bradshaw, 2016). However, because this class of models is relatively new to operational use, many psychometric properties require further investigation to support the use of the assessments. One key feature that is not well-defined in the literature is how best to assess the model fit of DCMs (Chen, de la Torre, & Zhang, 2013; Hu, Miller, Huggins-Manley, & Chen, 2016; Rupp et al., 2010). Most evaluations of model fit rely solely on measures of relative fit (Sen & Bradshaw, 2017), which are limited in that these indices are unable to evaluate the fit of the model to the data. Rather, these measures can only make judgments relative to alternative comparison models. The other widely used method for evaluating model fit is limited-information fit indices (e.g., Liu, Tian, & Xin, 2016). In general, these methods consist of univariate, bivariate, and trivariate item tests that rely on χ^2 tests that are known to be asymptotically incorrect (Maydeu-Olivares & Joe, 2006). The M_2 statistic developed by Maydeu-Olivares and Joe (2005) can correct for the distributional assumptions, but that statistic is still only based on limited information (i.e., limited sets of items), and therefore may fail to capture higher-order characteristics of the data.

Due to these concerns, this document investigates a Bayesian framework for the estimation of this class of models. This approach allows for the estimation of alternative methods for the evaluation of model fit through posterior predictive model checking.

2. Defining the Bayesian Model

The general form of DCMs can be seen in equation (1), where the probability of respondent j providing a given item response can be modeled as shown in equation (1).

$$P(\mathrm{X}_j = \mathrm{x}_j) = \sum_{c=1}^{C} \nu_c \prod_{i=1}^{I} \pi_{ic}^{x_{ij}} (1 - \pi_{ic})^{1-x_{ij}} \tag{1}$$

In equation (1), π_{ic} is the probability of a respondent in class c providing a correct response to item i, and x_{ij} is the observed response (i.e., 0, 1) of respondent j to item i. Thus, $\pi_{ic}^{x_{ij}}(1 - \pi_{ic})^{1-x_{ij}}$ represents the probability of a respondent in class c providing the observed response to item i. These probabilities are then multiplied across all items, giving the probability of a respondent in class c providing the observed response pattern. Finally, this probability is multiplied by ν_c, which is the base rate probability that any given respondent belongs to class c. Thus, this product represents the probability that a given respondent is in class c and provides the observed response pattern.

Although DCMs can be estimated with multiple attributes that have more than two latent categories (Bradshaw, 2016), for illustrative purposes, this paper limits the discussion to single-attribute DCMs with a binary latent trait. Thus, for each model, there are two potential mastery profiles for each respondent (e.g., master and non-master). Note, however, that the methods presented in this paper do generalize to models with multiple attributes with nonbinary latent categories.

Where different types of DCMs differ is in how π_{ic} is defined. For example, the log-linear cognitive diagnosis model (LCDM; Henson, Templin, & Willse, 2009) defines π_{ic} similar to the way generalized linear models with a logit link function are defined. Specifically, π_{ic} is defined as seen in equation (2), where α_c is a binary indicator of the mastery status for a respondent in class c.

$$\pi_{ic} = P(\mathrm{X}_{ic} = 1 \mid \alpha_c) = \frac{\exp(\lambda_{i,0} + \lambda_{i,1,1}\alpha_c)}{1 + \exp(\lambda_{i,0} + \lambda_{i,1,1}\alpha_c)} \tag{2}$$

When using this notation introduced by Rupp et al. (2010), the λ subscripts follow the order of item, effect, then attribute. That is, the first subscript identifies the item for the parameter (noted as i). The second subscript denotes the type of effect. Because this discussion is limited to single-attribute models, there are only two types of effects where zero identifies an intercept and one identifies a main effect. In models with multiple attributes, there may be additional effects for two-, three-, or A-way interactions. Finally, the last element of the subscript identifies the attribute or attributes. Again, as these are single-attribute models, this element is either nonexistent (for intercept terms where no attribute is involved) or 1 (for all other effects). It is included here only for consistency with the notation in Rupp et al. (2010).

For additional flexibility, equation (2) can be modified slightly in order to include both attribute- and item-level effects, similar to multilevel models.

$$\pi_{ic} = P(\mathrm{X}_{ic} = 1 \mid \alpha_c) = \frac{\exp[\lambda_0 + b_{i,0} + (\lambda_{1,1} + b_{i,1,1})\alpha_c]}{1 + \exp[\lambda_0 + b_{i,0} + (\lambda_{1,1} + b_{i,1,1})\alpha_c]} \tag{3}$$

Equation (3) shows the similarity to multilevel models. In this model, λ_0 and $\lambda_{1,1}$ represent the attribute-level intercept and main effect, respectively. These are akin to the average intercept and main effect for all items (the fixed effects in the multilevel model literature). In addition to the attribute-level parameters, there are also item-level intercepts ($b_{i,0}$) and main effects ($b_{i,1,1}$). These parameters represent the deviation from the attribute-level effect for each item. Thus, the full intercept for item one would be calculated as $\lambda_0 + b_{1,0}$. This is similar to the estimation of random intercepts and slopes for each item (Stroup, 2012). The difference between the proposed model and multilevel models is the treatment of the variance of these item-level parameters. In multilevel models, the variance of these effects would be estimated. However, the variance of the item-level parameters can also be fixed to pre-specified values.

If the item-level parameters are constrained to be zero, then all items will have parameters equal to the attribute-level parameter (i.e., all of the $b_{i,0}$ and $b_{i,1,1}$ parameters would be zero). This is mathematically equivalent to what is referred to here as the *fungible* model. Alternatively, the item-level parameters can be allowed to vary freely with no constraints (i.e., a *non-fungible* model). Conceptually, these two models can be thought of as using a zero-variance prior (i.e., $\mathcal{N}(0,\ 0)$) or infinite-variance or flat prior (e.g., $\mathcal{N}(0, \infty)$), respectively. Finally, a non-flat prior can be placed on the item-level parameters, such that the parameters are not constrained to be zero but also not allowed to vary completely freely either.

2.1. Prior Specification for Attribute-Level Effects

In equation (3), there are two attribute-level effects that require prior specifications. The first attribute-level effect is λ_0, which represents the average intercept across all items. Thus, this parameter also represents the log-odds (due to the logit link function) of a non-master providing a correct response to an average item. For this parameter, a $\mathcal{N}(\mu = 0,\ \sigma = 2)$ distribution was used as the prior. This prior distribution was chosen because 99% of the distribution encompasses the plausible values for this parameter. Specifically, the middle 99% of the distribution consists of the

log-odds range -5.15 to 5.15, which covers nearly all of the probability scale when other parameters are equal to zero, as seen in Figure 1.

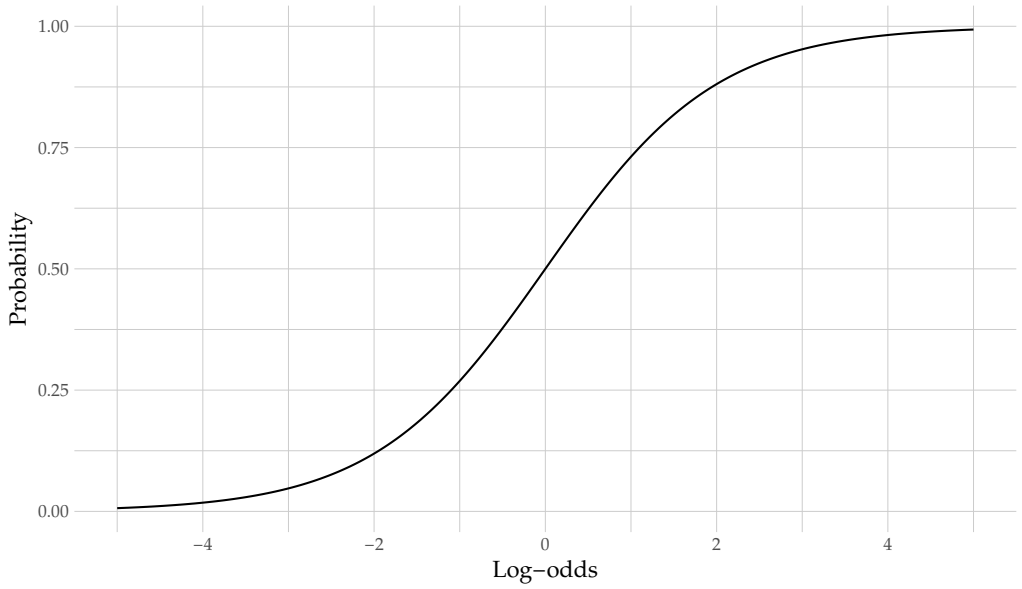

Figure 1. Log-odds to probability conversion.

The main effect parameters in the LCDM are constrained to be positive, thus ensuring monotonicity in the model (e.g., masters always have a higher probability of providing a correct response; Henson et al., 2009). Thus, the attribute-level main effect, $\lambda_{1,1}$, uses a lognormal prior: Lognormal$(\mu = 0, \sigma = 1)$. Similar to the attribute-level intercept, this distribution was chosen because 99% of the distribution covers the range of plausible values. Specifically, the lower 99% of this distribution covers the log-odds range of 0 to 10.24. An upper limit of approximately 10 was desired, as a main effect of 10 would allow for an estimated probability of providing a correct response near 1.0 in the extreme case where the intercept was -5 (the lower tail of the attribute-level intercept prior distribution).

The distributions for these parameters are visualized in Figure 2.

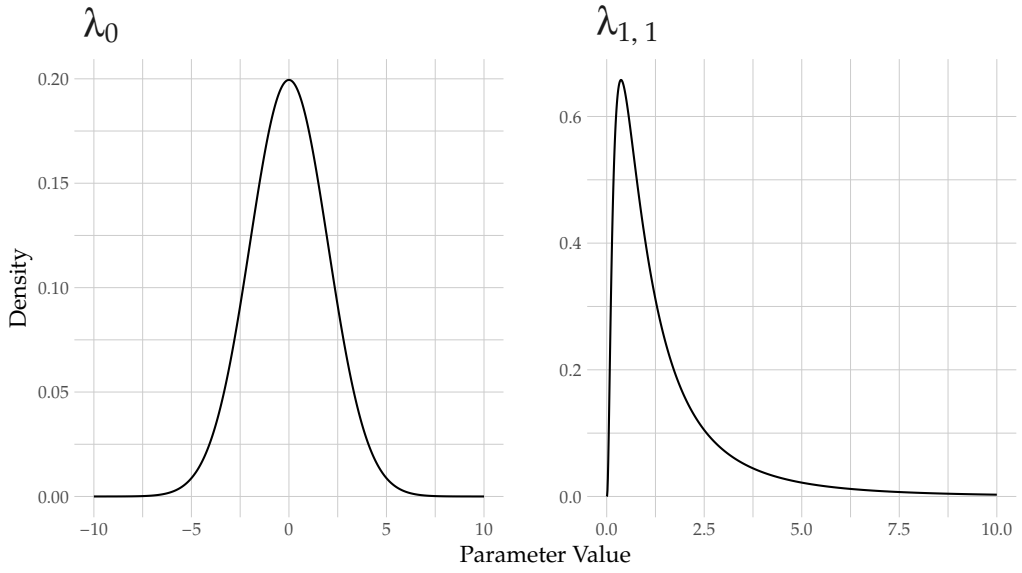

Figure 2. Prior distributions for attribute-level effects.

2.2. Prior Specification for Item-Level Effects

The prior distributions for the item-level effects, $b_{i,0}$ and $b_{i,1,1}$, are determined by the type of model that is being estimated. For this proof of concept, three models are considered: fungible, non-fungible, and partial equivalency.

In the fungible model, it is assumed that all items measuring the attribute have the same item parameters. That is, the item-level effects are equivalent to the attribute-level effect. Thus, the item-level deviations from the attribute-level effects are all equal to 0. Conceptually, this means using a $\mathcal{N}(\mu = 0,\ \sigma = 0)$ prior for all $b_{i,0}$ and $b_{i,1,1}$ terms. In practice, to increase computational efficiency, these terms are left out of the model, and only the attribute-level effects are estimated.

In contrast, the non-fungible model assumes that the item parameters are independent of one another. In other words, the parameters for one item do not dictate the parameters of other items. Conceptually, this means that the item-level deviations from the attribute-level effects are unconstrained, and thus an infinite uniform prior, $\mathcal{U}(-\infty,\ +\infty)$, would be used for all $b_{i,0}$ and $b_{i,1,1}$ terms. In practice, it is more efficient to directly estimate individual parameters for each item rather than attribute-level effects with unconstrained item-level deviations. Therefore, this model more closely resembles a true LCDM in equation (2), with the $\lambda_{i,0}$ and $\lambda_{i,1,1}$ parameters using the prior distributions described for the attribute-level priors.

The partial equivalency model represents a compromise between the fungible and non-fungible models. In this model, item-level parameters are not entirely independent but are also not constrained to be equivalent. Instead, the item-level parameters are assumed to come from some distribution of deviations. The smaller the variance of the distribution, the more fungible the items are. Conversely, a large variance would correspond to less fungibility. Conceptually and in practice,

this model is similar to multilevel models. The item-level deviations use a hierarchical normal prior, $\mathcal{N}(\mu = 0,\ \sigma)$, where σ is an estimated parameter in the model. The σ parameter uses a half-Student's t-distribution with $df = 3$ (Figure 3). This prior ensures that the variance is always positive and also allows for larger variances than a normal distribution would. However, the variances are also constrained to reasonable values (i.e., less than approximately 5).

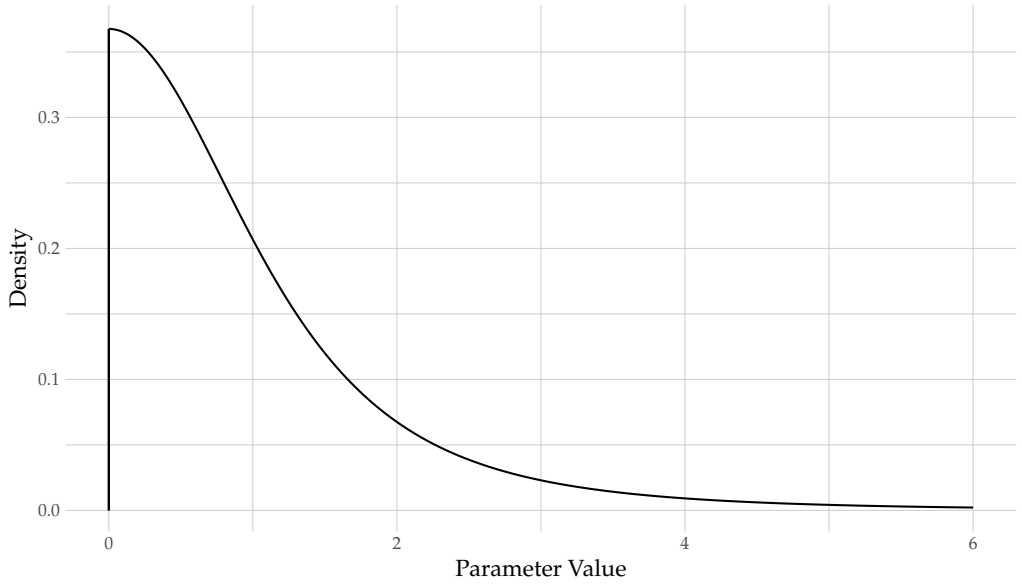

Figure 3. Prior distribution for hierarchical variance prior.

2.3. Prior Specification for Class-Level Parameters

The last parameter that requires a prior is the structural parameter in equation (1), ν_c. This parameter defines the base rate of inclusion for each class. As such, ν is constrained so that all elements sum to one (i.e., there are no non-class respondents). Because this discussion is limited to models with a single binary attribute, there are only two classes and therefore two elements of ν. No assumptions are made about the base rate of mastery for attributes; therefore, a uniform Dirichlet prior, $\text{Dir}(1)$, was used for the prior distribution. As there are only two classes, this is equivalent to using a uniform Beta distribution, $\text{Beta}(\alpha = 1,\ \beta = 1)$, for ν_1 and then calculating ν_2 as $\nu_2 = 1 - \nu_1$.

3. The Bayesian Framework in Practice

In order to demonstrate the utility and benefits of using the Bayesian model definition and estimation process, a single simulated data set was generated. This data set was then used to walk through each step of the Bayesian model fit process, from model estimation to model evaluations and comparisons. All analyses were performed in *R* version 3.6.1 (R Core Team, 2019).

3.1. Measures

To demonstrate the Bayesian framework in practice, the Dynamic Learning Maps® (DLM®) Alternate Assessment System is used as an example diagnostic assessment where this framework is applicable. DLM assessments in English language arts (ELA), mathematics, and science are administered in 19 states to students with the most significant cognitive disabilities. For exemplary purposes, the through-course assessment model, which features instructionally embedded assessments during the year, is used as a template.

In the instructionally embedded model, students cover the entire testing blueprint during each of two testing windows. The first testing window occurs during the fall, from September through December. The second window is open during the spring from February through May. During each window, students take one or more testlets, each consisting of three to nine items, for each alternate content standard (called an Essential Element [EE]) required for blueprint coverage. To ensure that each EE is accessible to all students, each EE is associated with multiple skills that represent the EE at varying levels of depth, breadth, and complexity (called linkage levels). There are five linkage levels for each EE in ELA and mathematics and three linkage levels for each EE in science. Due to the intended flexibility of the instructionally embedded testing model, students may or may not test on the same EE and linkage level multiple times within a testing window or across testing windows. Thus, the number of responses that can be used to estimate student mastery of a linkage level varies by student. For more details on the assignment of testlets, see Chapter 4 of Dynamic Learning Maps Consortium (DLM Consortium, 2016).

For modeling and scoring the DLM assessments, the linkage level is the unit of analysis. That is, a latent class analysis (LCA; Bartholomew, Knott, & Moustaki, 2011) with two classes is estimated for each linkage level (see Chapter 5 of DLM Consortium, 2017). The latent class model currently employed for operational use represents an unconstrained version of the models defined in Section 2 (Henson et al., 2009; Rupp et al., 2010). Specifically, as discussed in Section 2.2, the main effects of equation (3) are constrained to be positive to ensure monotonicity in the model. When using the unconstrained latent class model, post hoc analysis is needed to ensure the mastery classes are properly defined (i.e., the labels of master and non-master are applied to the correct classes).

Regardless of the choice between the LCA or DCM for estimation, the resulting score is the probability that the student has mastered the linkage level. This probability is often dichotomized into a mastery categorization (Bradshaw & Levy, 2019). For example, the DLM assessments use a mastery threshold of 0.8 (see Chapter 5 of DLM Consortium, 2017). That is, students with a mastery probability of 0.8 or higher are classified as masters, and students with a mastery probability of less than 0.8 are classified as non-masters. Thus, the scores used for reporting are a profile of mastery classification decisions for each linkage level. For further details on the scoring model for DLM assessments, see Chapter 5 of DLM Consortium (2017).

3.2. Simulated Data

To illustrate the Bayesian methods for estimating and evaluating diagnostic models, a single data set was generated. Simulated data was chosen for two reasons. First, because the data is simulated, the expected results of the analysis are known. Thus, the results can be compared to the *a priori* expectations to confirm that the methods work as expected. Second, by using simulated data, it is possible to ensure that some models fit the example data and others do not. This means that when

examining model fit, there will be examples of fitting and non-fitting models that can be compared. Although this is useful for illustrating the methods, it is important to remember that the data was generated to serve this purpose.

When simulating the example data set, the data was structured similarly to the DLM assessments. In this way, the structure of the simulated data matched what could reasonably be expected from an operational assessment scaled with a DCM. Specifically, items were grouped together into testlets, and testlets were assigned to either the fall or spring testing window. By assigning testlets to the testing windows, it was possible to simulate data with students testing on combinations of testlets consistent with observed data. In other words, the amount and structure of missing data (from testlets not assigned to a student) was comparable across the simulated and observed data. Additionally, following the DLM test design, all items were assumed to follow a simple Q-matrix structure, where all items measure a single attribute (DLM Consortium, 2016). Item parameters were simulated according to the partial equivalency model defined in equation (3). Thus, the partial equivalency and non-fungible models are expected to show adequate model fit, as these are the true model and a less-constrained model, respectively. Conversely, the fungible model should show poor fit, as the fungible model is more constrained than the partial equivalency model.[1]

The attribute-level intercept, λ_0, was drawn from a $\mathcal{U}(-2.25, -1.00)$ distribution, and the attribute-level main effect, $\lambda_{1,1}$, from a $\mathcal{U}(1.00, 4.50)$. The item-level deviations $b_{i,0}$ and $b_{i,1,1}$ were drawn from a $\mathcal{N}(\mu = 0,\ \sigma = 1.0)$ distribution. This resulted in total item intercepts and main effects consistent with those reported for other measures that have been scaled with the LCDM (e.g., Bradshaw, Izsák, Templin, & Jacobson, 2014; Templin & Bradshaw, 2014; Templin & Hoffman, 2013). The true parameter values for each testlet and item that were used to simulate the data can be seen in Table 1.

[1]The partial equivalency model was chosen in order to illustrate differences between fitting and non-fitting models and thus should not imply that this model best represents DLM data. See DLM Consortium (2017) for more information on the operational model used for DLM assessments.

Table 1. True Item Parameters

Window	Testlet	Item	λ_0	$\lambda_{1,1}$	$b_{i,0}$	$b_{i,1,1}$	$\lambda_0 + b_{i,0}$	$\lambda_{1,1} + b_{i,1,1}$
Fall	101	1	-1.52	3.19	0.25	1.00	-1.27	4.19
		2	-1.52	3.19	-0.45	0.12	-1.96	3.30
		3	-1.52	3.19	0.09	0.82	-1.43	4.01
		4	-1.52	3.19	-0.48	1.00	-2.00	4.19
		5	-1.52	3.19	0.07	1.00	-1.45	4.19
	102	6	-1.52	3.19	-1.11	-0.28	-2.63	2.91
		7	-1.52	3.19	-0.65	-0.50	-2.17	2.69
		8	-1.52	3.19	-0.07	1.00	-1.59	4.19
Spring	103	9	-1.52	3.19	0.53	-0.42	-0.99	2.76
		10	-1.52	3.19	-0.03	-1.50	-1.55	1.69
		11	-1.52	3.19	1.57	-0.41	0.05	2.77
		12	-1.52	3.19	0.32	-2.32	-1.20	0.87
		13	-1.52	3.19	0.94	-0.43	-0.58	2.75
	104	14	-1.52	3.19	0.20	-0.08	-1.32	3.11
		15	-1.52	3.19	-0.06	-0.43	-1.57	2.76
		16	-1.52	3.19	-1.76	0.92	-3.27	4.11
		17	-1.52	3.19	-0.82	0.94	-2.33	4.12

To mimic the DLM test structure, students were randomly assigned a combination of the simulated testlets. Following the test administration design for the instructionally embedded DLM testing model (for details see Chapter 4 of DLM Consortium, 2016), students were assigned testlets from both the instructionally embedded and spring pools. During spring assessments, students were randomly assigned only one testlet. For instructionally embedded assessments, students had a a 90% chance of taking only one testlet and a 10% chance of taking both testlets. This is consistent with the reported usage of the instructionally embedded assessment window (Clark, Thompson, & Karvonen, 2019). The resulting probabilities for each possible combination of assigned testlets can be seen in Table 2, along with the total number of students actually simulated to have that combination. In total, 1,700 students were simulated, which is consistent with the total number of students that test on a single attribute in a given year from states participating in the instructionally embedded assessment model (see Chapter 7 of DLM Consortium, 2016).

Table 2. Number of Simulated Students Assigned to Each Testlet Combination

Testlet Combination	Probability	*n*
101, 103	0.203	361
101, 104	0.203	356
102, 103	0.203	358
102, 104	0.203	322
101, 102, 103	0.045	68
101, 102, 104	0.045	68
101, 103, 104	0.045	83
102, 103, 104	0.045	61
101, 102, 103, 104	0.010	23

3.3. Model Estimation

The models are estimated in *R* version 3.6.1 (R Core Team, 2019) using the **rstan** package interface (Guo, Gabry, & Goodrich, 2019) to *Stan* (Carpenter et al., 2017), which utilizes Markov chain Monte Carlo (MCMC) and the Hamiltonian Monte Carlo (HMC) algorithm to efficiently transition between draws of the posterior distribution (Betancourt & Girolami, 2013; Neal, 2011). Specifically, *Stan* utilizes the No-U-Turn sampler (NUTS; Hoffman & Gelman, 2014) to dynamically choose a step size and leap trajectory for the HMC algorithm in order to ensure efficient estimation (Betancourt, Byrne, & Girolami, 2015). A complete description HMC with NUTS can be found in Hoffman and Gelman (2014). For a less technical introduction to MCMC and HMC, see McElreath (2015).

The *Stan* code for all models can be found in the online repository for this report[2]. The models were estimated with four chains, each with 2,000 iterations. The first 1,000 iterations of each chain were discarded for warm-up, leaving a total of 4,000 retained iterations that made up the posterior distributions. There were also several settings specific to NUTS (Hoffman & Gelman, 2014) used by *Stan*. First, the adaptive threshold was set to 0.99 to avoid divergent transitions (Betancourt, 2017a). Secondly, the maximum tree depth, which determines how far the algorithm can go before making a U-turn (Betancourt, 2017b), was set to 15. These are both more conservative than the values suggested by the Stan Development Team (2019c). The implications of these setting are discussed in the following sections, along with diagnostics to assess their impact.

After estimating the model but before the parameters can be analyzed and inferences can be made, the model is checked to ensure the estimation process completed in an appropriate manner. This diagnostic information is critical to MCMC estimation, as without proper estimation, no valid inferences can be made. Checks include evaluating convergence, efficiency of the sampler, and parameter recovery. Each check is described in detail below using the estimated partial equivalency model as an example, as this was the true data-generating model.

3.3.1. Convergence

A check of convergence evaluates whether the MCMC chain successfully found the high density area of the posterior distribution and stayed there. When multiple chains are estimated, this can be

[2]https://github.com/atlas-aai/bayes-concept

checked by verifying that each chain is drawing estimates from the same parameter space. For a single chain, this is checked by verifying that the parameter is sampled from roughly the same area at the beginning of the chain (after warm-up) as it is at the end of the chain. This is commonly assessed through trace plots. An example of a trace plot is shown in Figure 4.

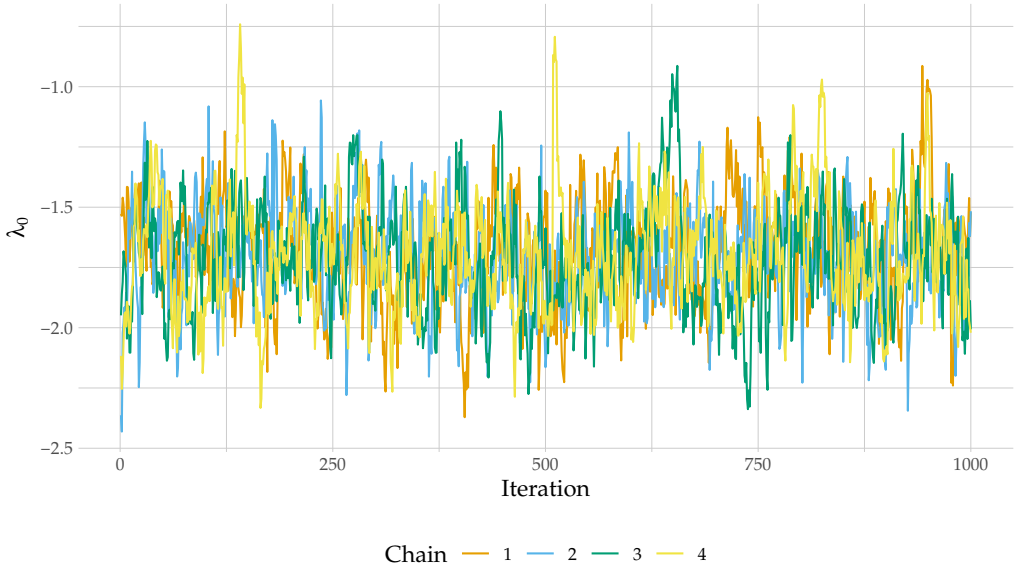

Figure 4. Trace plot for the attribute-level intercept λ_0.

Figure 4 shows the trace plot for the attribute-level intercept, λ_0, and looks the way a trace plot is expected to look. The draws appear to be coming from a stable distribution (i.e., the plot is relatively horizontal with no large upward or downward swings), and all four are mixing well (as evidenced by the overlap of the four colors). However, there is no empirical method that uses visual inspection alone to determine how poor a trace plot must be to conclude convergence was not met. Additionally, when there are many parameters, it is impractical to look at each individual trace plot.

To address these shortcomings of evaluating trace plots directly, the $\widehat{R}$ statistic can be used to evaluate convergence (Brooks & Gelman, 1998; Vehtari, Gelman, Simpson, Carpenter, & Bürkner, 2019). The $\widehat{R}$ statistic is also known as the potential scale reduction (Gelman et al., 2013) and is a measure of how much variance there is between chains relative to the amount of variation within chains. Gelman and Rubin (1992) suggest that in order to conclude that the model has successfully converged, all $\widehat{R}$ values should be less than 1.1. These results can be summarized, as in Figure 5, to demonstrate the $\widehat{R}$ values for the estimated parameters. In the estimated partial equivalency model, all values are below 1.1, indicating that the model converged.

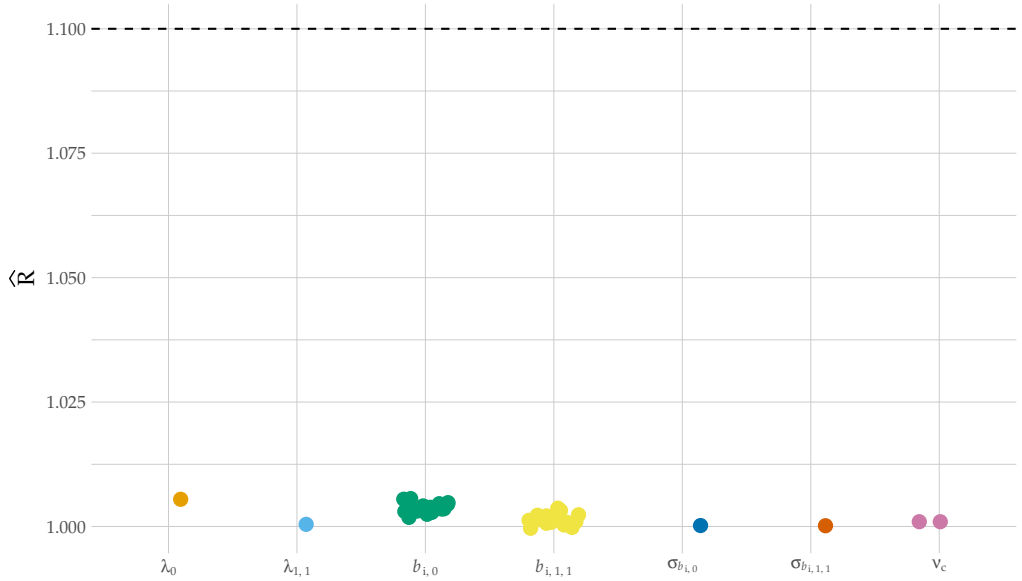

Figure 5. $\widehat{R}$ values for the estimated parameters in the partial equivalency model. Dotted line represents the suggested cutoff by Gelman and Rubin (1992).

3.3.2. Efficiency

A second check of the MCMC estimation is the efficiency of the sampler, which verifies that the algorithm adequately sampled the full posterior distribution. There are several ways this can be examined. The first is by examining the effective sample size. This diagnostic takes into account the autocorrelation (or anticorrelation) within chains to determine the effective number of independent draws from the posterior. If the chain is slow moving, the draws will be highly autocorrelated, and effective sample size will be well below the total number of retained iterations (Geyer, 2011). Conversely, if the chain is moving quickly, it is possible for the draws to be better than independent, or anticorrelated (Kroese, Taimre, & Botev, 2011). In this scenario, the effective sample size is actually larger than the true sample size.

There are two types of effective sample size that can be used to evaluate the efficiency of the location and scale of the posterior distributions. The sampling efficiency of the location (e.g., mean or median) can be assessed with the bulk effective sample size. Similarly, the scale can be assessed through tail effective sample size. This can be useful for diagnosing problems with mixing due to posterior samples having different scales across chains (Vehtari, Gelman, Simpson, et al., 2019). For both measures, the Stan Development Team (2019a) recommend an effective sample size greater than or equal to the number of chains multiplied by 100.

The effective sample size for all parameters in the model can be summarized, as in Figure 6. Because the model was estimated with four chains, the effective sample size should be above 400. Figure 6 shows that all parameters in the estimated partial equivalency model have both a bulk and tail

effective sample size above this threshold.

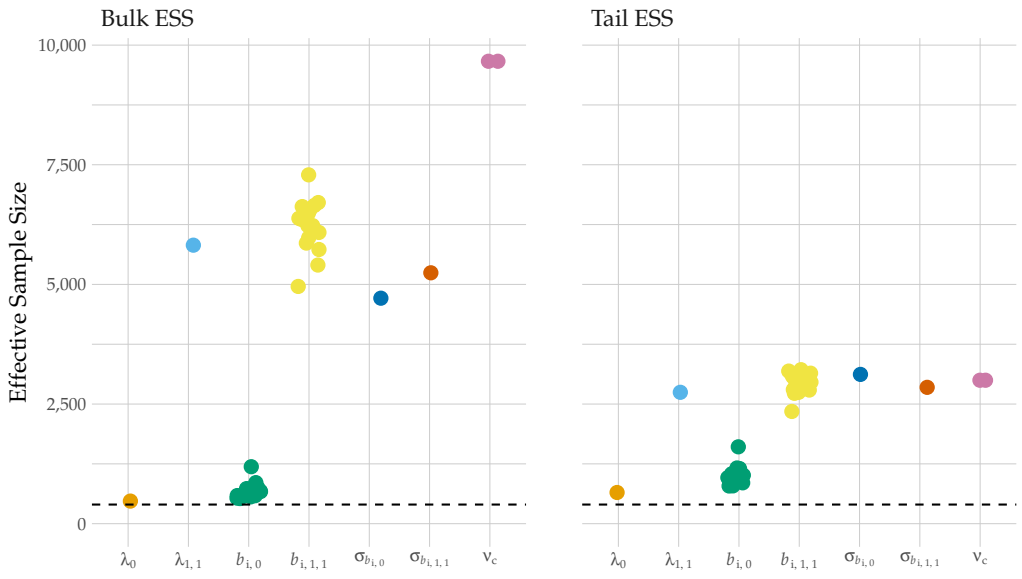

Figure 6. Effective sample size for each estimated parameter. Dotted line represents the suggested cutoff by Stan Development Team (2019a). ESS = effective sample size.

There are also measures of efficiency that are exclusive to NUTS (Hoffman & Gelman, 2014). For example, the Bayesian factor of missing information gives an estimate of how well the sampler adapted and explored the posterior distribution. The Bayesian factor of missing information generally ranges from zero to one, with zero and one representing poor and excellent estimation, respectively. This is calculated for each chain overall, rather than each individual parameter (Betancourt, 2016).

The Bayesian factor of missing information values for this example are shown in Table 3 and indicate that the sample was able to adequately visit the posterior distributions. Additionally, Table 3 shows the mean acceptance rate for each chain. As expected, these values are very close to the 0.99 adaptive threshold that was specified during the model estimation. As mentioned previously, a target acceptance rate this high is needed to prevent divergent transitions.

The concern with setting the target acceptance rate this high is that for parameters with wider posteriors, the sampler will not be able to move fast enough. When using NUTS, at each iteration, the sampler looks for a place to "U-Turn" in a series of possible branches. If the sample is terminating before the maximum possible tree depth (which was specified to be 15), then the algorithm is able to adequately find good values for the next iteration of the chain, despite the small steps being enforced by the high target acceptance rate. Bumping up against the maximum allowed tree depth, or going beyond it, indicates that the step size is too small (Stan Development Team, 2019c; Stan Development Team, 2019b). Because the maximum tree depth values in Table 3 are all below the maximum

specified, and the Bayesian factor of missing information values are all close to one, there is strong evidence that in this model, the sampler was able to adequately sample the posteriors.

Table 3. Diagnostic Statistics for the No-U-Turn Sampler

Chain	BFMI	Mean Acceptance Rate	Max Tree Depth
1	0.993	0.991	7
2	1.008	0.991	8
3	1.048	0.994	8
4	1.018	0.993	8

Note: BFMI = Bayesian factor of missing information.

3.3.3. Parameter Recovery

In addition to having diagnostics to ensure that the model is estimated properly, it is also important to establish that the model as defined in the *Stan* code is able to accurately recover the true parameter values. Otherwise, a model may estimate well but be miss-specified, leading to incorrect parameter estimates. Figure 7 shows the true (from Table 1) versus estimated item parameter values, indicating successful parameter recovery for the partial equivalency model with the simulated data.

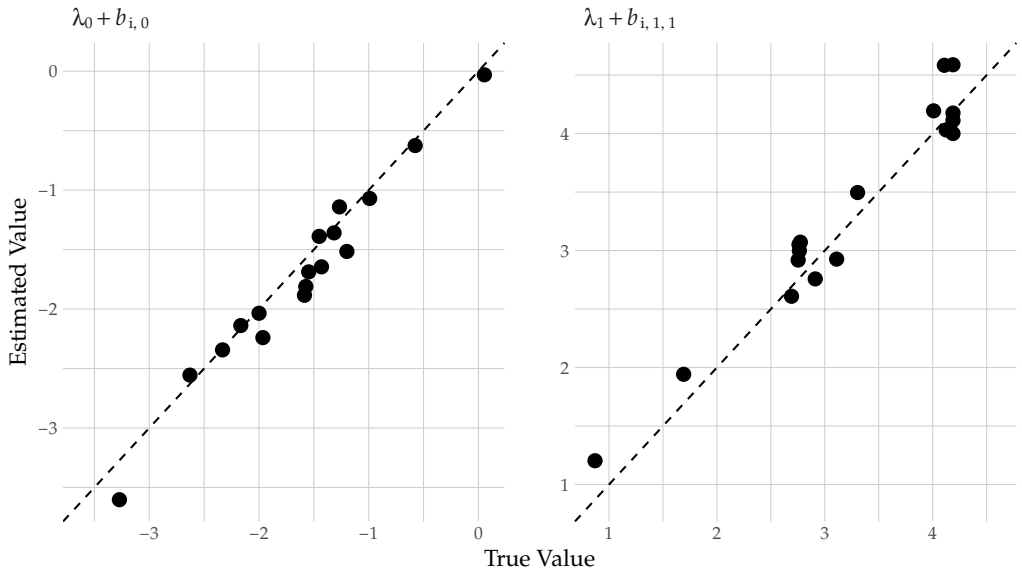

Figure 7. Parameter recovery from the example partial equivalency model with simulated data.

It is also possible to examine the accuracy of the respondent classifications as a master or non-master. For this analysis, respondents were classified as masters if the median of the posterior distribution for the probability of mastery was greater than or equal to 0.5. This threshold places respondents in their

most likely class; however, any threshold can be used in practice to facilitate stakeholder understanding of scores (Bradshaw & Levy, 2019). Respondent classification results for the partial equivalency model are summarized in Table 4. In total, 99% of the simulated students were correctly classified as masters or non-masters. This is not surprising, given that the data was simulated to fit the model, and all of the items are fairly discriminating (Table 1).

Table 4. Respondent Classification Accuracy

True Mastery	Estimated Mastery	*n*
0	0	676
0	1	10
1	0	8
1	1	1,006

3.4. Evaluating Model Fit

Model fit can be assessed in both an absolute and relative sense. Absolute fit is used to evaluate whether or not the estimated model adequately reflects the observed data and is a prerequisite for the evaluation of relative fit. Relative fit compares the fit of two more models that all show adequate absolute model fit (Chen et al., 2013; Sen & Bradshaw, 2017). In this document, methods are presented for assessing absolute and relative fit through posterior predictive model checking and information criteria, respectively. In order to demonstrate how these methods work in practice, both the fungible and non-fungible models were estimated on the same data used to estimate the partial equivalency model in the previous section. Thus, there are a total of three models to compare and calculate posterior predictive checks for. As described previously, the partial equivalency model was the true data generation model.

3.4.1. Absolute Fit

Posterior predictive model checks are used to assess the absolute fit of a specific model to the observed data. Posterior predictive checks involve simulating replications of the data using the values of the posterior distributions and then comparing the replicated data sets back to the observed data (Gelman et al., 2013). As explained in the model estimation section, a total of 4,000 iterations were retained from the MCMC estimation. Thus, 4,000 replicated data sets can be simulated, one for each iteration, using the current values of the parameters at each iteration. The process for simulating a replicated data set for a single iteration is as follows:

1. Randomly assign the first respondent to the master or non-master class, with probability equal to the current value of the respondent's probability of attribute mastery.
2. For the first item the respondent takes, simulate a response using the current values of the item parameters and the mastery status that was simulated in step 1.
3. Repeat step 2 for all items the respondent tested on.
4. Repeat steps 1–3 for all respondents.

This process is repeated for each iteration in the chain. Because the replicated data sets are simulated from the current values of the parameters, these data sets represent what the data would be expected

to look like *if the specified model were true*. Therefore, summaries of these data sets can then be used to look for systematic differences in the characteristics of the observed data and the replicated data sets, often through visualizations (Gelman & Hill, 2006).

3.4.1.1. Model-Level Fit

At the model level, posterior predictive checks can be calculated for the raw score distribution. This is accomplished by counting the number of respondents at each raw score point in each of the 4,000 replicated data sets. Thus, a distribution is derived from the number of respondents expected to be present at each raw score point in the observed data. Figure 8 shows the distribution of expected raw scores along with the number of observed students in the simulated data.

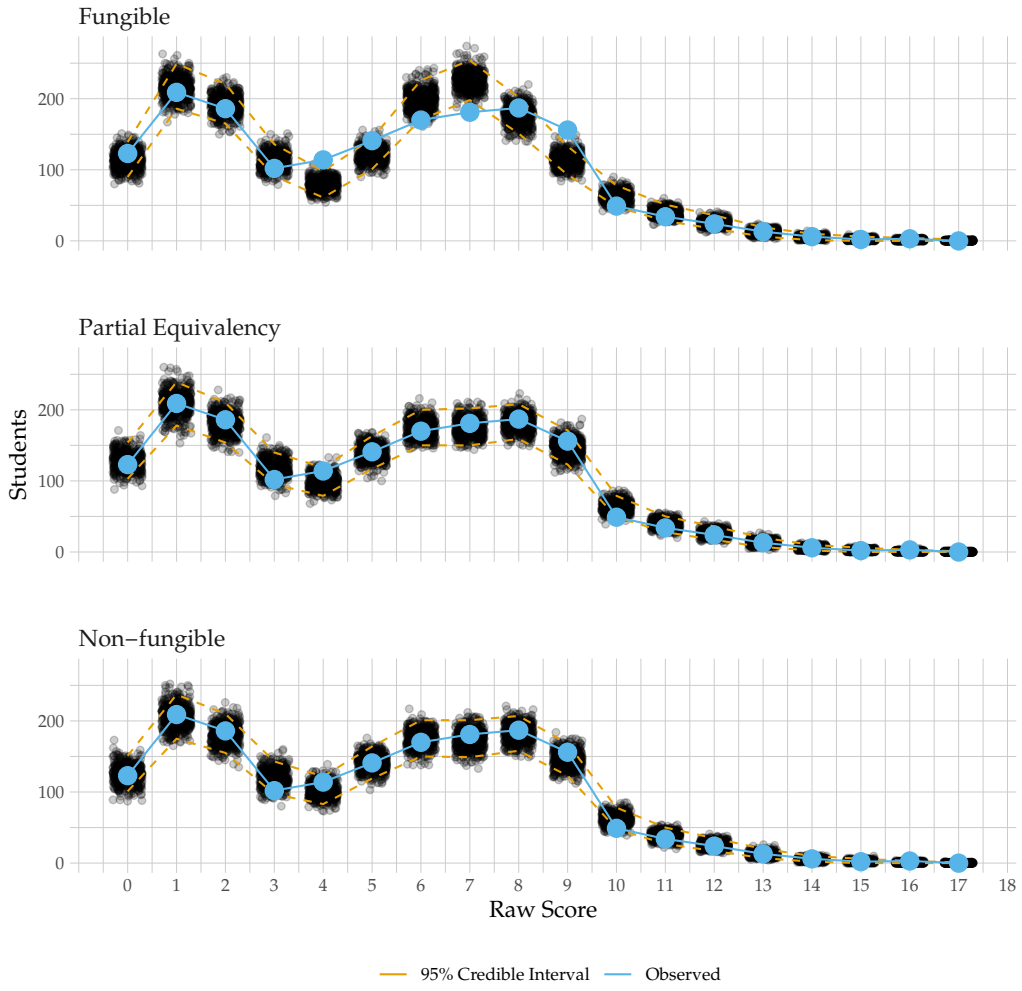

Figure 8. Posterior predictive model check for the raw score distribution.

Figure 8 shows a bimodal distribution, which is a result of the mixture of raw score distributions for

masters and non-masters. Additionally, very few students are expected to have a high raw score due to the fact that relatively few students test on more than two testlets (Table 2). Finally, the expected distribution for fungible model shows some deviations from the observed scores. Specifically, the fungible model overestimates the number of students with a raw score of six and seven and underestimates the number of students with a raw score of four and nine.

Similar to the examination of trace plots above (Figure 4), this visualization alone is insufficient for determining if the amount of misfit in the distribution is significant. Rather, Béguin and Glas (2001) suggest a χ^2 discrepancy measure can be calculated according to equation (4).

$$\chi^2_{obs} = \sum_{s=0}^{S} \frac{[n_s - E(n_s)]^2}{E(n_s)} \tag{4}$$

In equation (4), s represents the score point, n_s is the number of respondents at score point s, and $E(n_s)$ is the expected number of respondents at score point s, calculated as the average over all of the replicated data sets. Like the χ^2 tests that are used to assess model fit when the expectation-maximization algorithm is used, the χ^2_{obs} statistic does not follow a true χ^2 distribution. However, when using posterior predictive model checks, none of the distributional assumptions are required. This is because the reference distribution can be generated directly from the replicated data sets, similar to a parametric bootstrap. Using the same definition of $E(n_s)$ as above, a χ^2_{rep} can be computed for each of the replicated data sets. The 4,000 χ^2_{rep} values then make up the reference distribution to compare back to χ^2_{obs}. A posterior predictive *p*-value (*ppp*) can then be calculated as shown in equation (5).

$$ppp = P(\chi^2_{rep} \geq \chi^2_{obs} \mid n_s) \tag{5}$$

Equation (5) says that the posterior predictive *p*-value is the proportion of replicated data sets whose χ^2_{rep} value is greater than the χ^2_{obs} value from the observed data. Posterior predictive *p*-values close to zero indicate poor model fit (a cutoff of .05 could be used, for example), whereas values very close to one may indicate possible over-fitting. The χ^2_{obs} distributions and posterior predictive *p*-values for the fungible, partial equivalency, and non-fungible model are shown in Table 5 and visualized in Figure 9. As expected, given the distributions in Figure 8, the fungible model shows poor model fit, with a posterior predictive *p*-value of less than .05. In contrast, both the partial equivalency and non-fungible models show acceptable fit to the simulated data.

Table 5. χ^2_{obs} Values and Summaries of χ^2_{rep} Distributions

Model	χ^2_{obs}	χ^2_{rep} Mean	χ^2_{rep} 5%	χ^2_{rep} 95%	*ppp*
Fungible	57.38	17.98	8.86	29.96	0.001
Partial Equivalency	14.30	17.97	8.64	30.28	0.669
Non-fungible	14.04	17.74	8.50	29.99	0.681

Note: *ppp* = posterior predictive *p*-value.

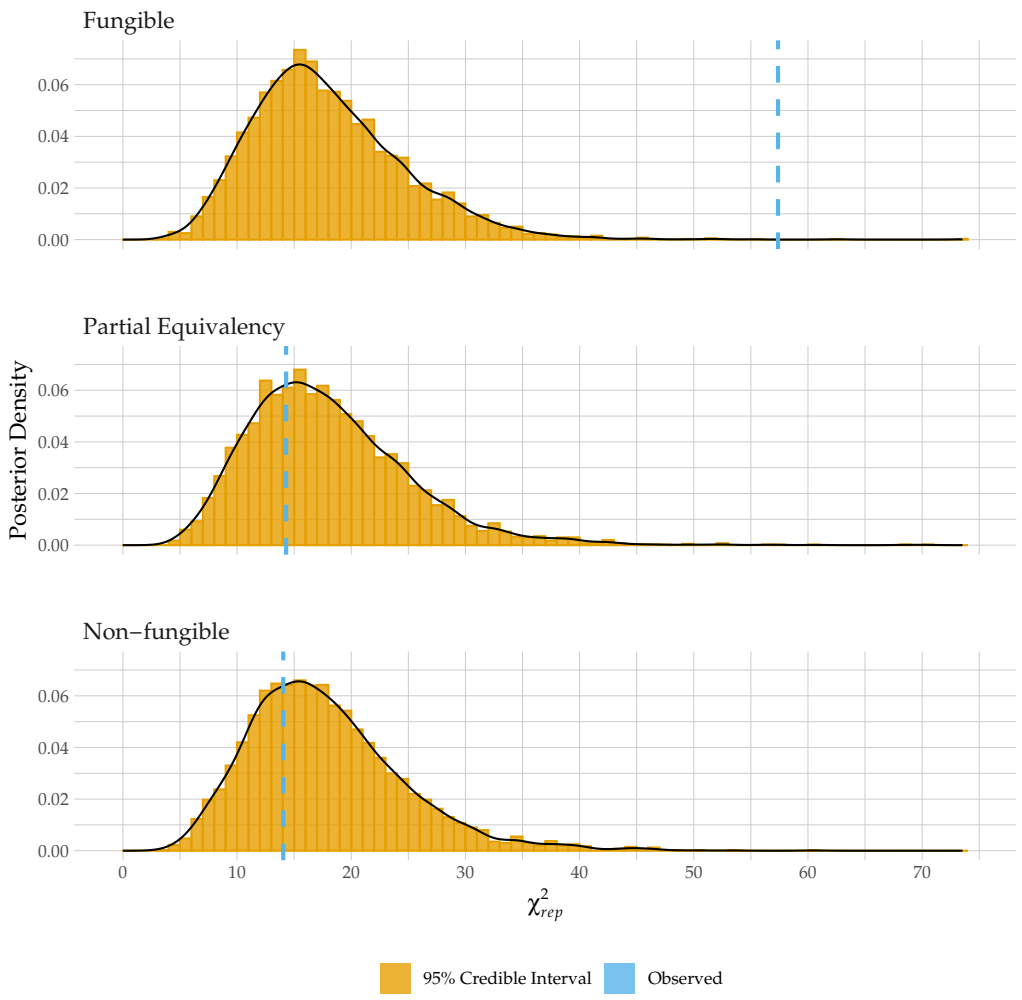

Figure 9. Posterior predictive model check for χ^2_{rep} distributions and χ^2_{obs} values. Dashed lines represent the observed χ^2_{obs} value.

3.4.1.2. Item-Level Fit

Posterior predictive checks can also be used to assess item-level fit by creating posterior summaries of item *p*-values. This is accomplished by calculating a *p*-value for each item in each of the 4,000 replicated data sets. This provides a distribution for plausible item *p*-values if the model fits. The item *p*-values from the observed data can then be compared to these distributions. Figure 10 shows an example of these comparisons for the fungible, partial equivalency, and non-fungible models. This shows the assumptions of the fungible model. That is, each item has the same expected *p*-value in the fungible model, but different expectations for the partial equivalency and non-fungible models. Because the simulated data was not fungible (i.e., was generated from the partial equivalency model), Figure 10 correctly shows that the fungible model was unable to successfully recover the observed

item *p*-values.

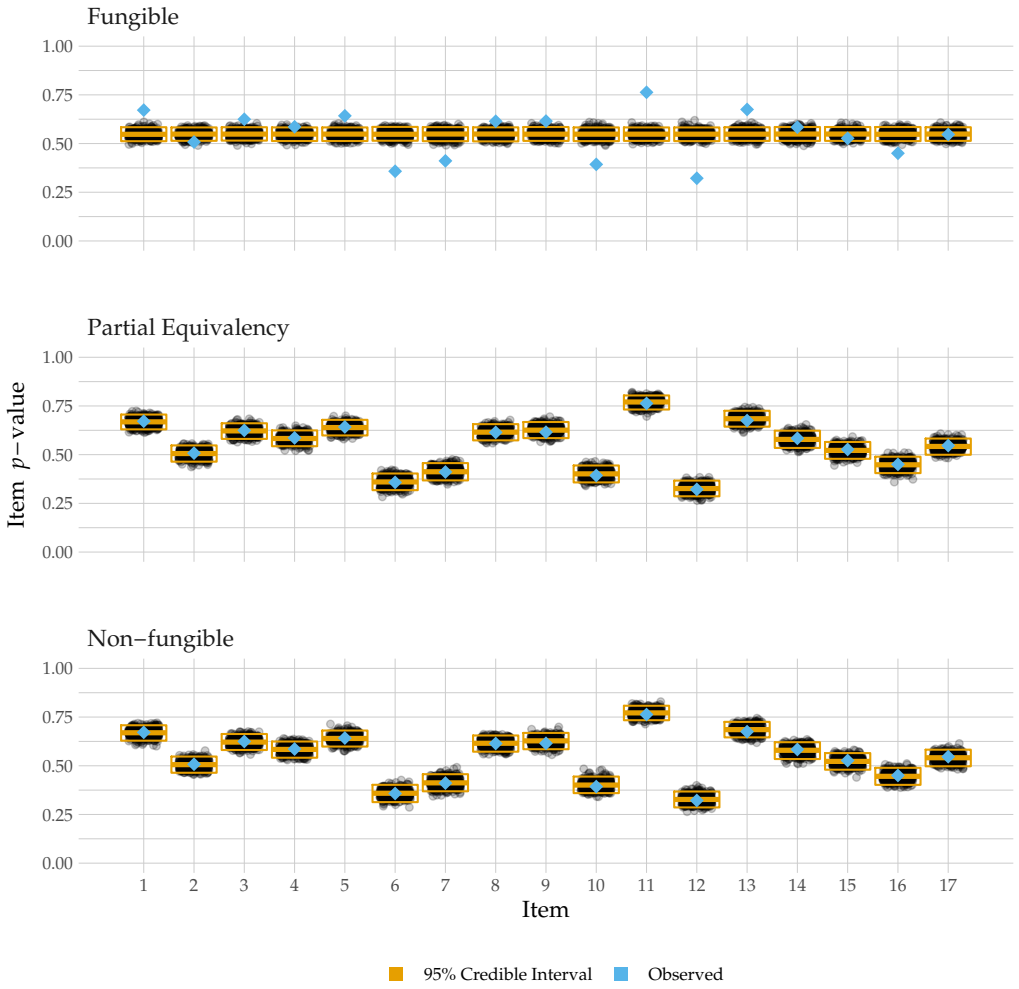

Figure 10. Posterior predictive model check for overall item *p*-values.

One limitation of using the overall item *p*-values is that this method overlooks an important aspect of the model. As shown in equation (3), the model actually defines two *conditional* probabilities, not one unconditional probability. In other words, the model defines a probability of non-masters providing a correct response, and a separate probability of masters providing a correct response. Thus, using a single *p*-value may miss important characteristics of the data. To examine this, posterior distributions for *p*-values conditional on mastery status can be estimated, similar to the overall *p*-value method.

This process is slightly complicated by that fact that mastery status is unobserved. Practically, this means there is not an observed *p*-value for each mastery class to compare back to the posterior distributions readily available in the data. To calculate the *p*-value of an item for each mastery class, a procedure similar to that described by Sinharay and Almond (2007) is followed. When using a

Bayesian MCMC estimation, a mastery classification can be made at each iteration of the Markov chain, as described above. Specifically, the probability of mastery is dichotomized at .5, although other thresholds can also be used. Using these classifications, a class-level *p*-value can be calculated for each iteration using the observed item responses and the respondents who were assigned to each class in that iteration. This results in a series of class-level *p*-values (one per item per iteration). The observed class-level *p*-values are then defined as the median for each class and item across all iterations (see Sinharay & Almond, 2007, for full details).

Similarly, an expected class-level *p*-value can be calculated by following the same procedure, but with item responses generated in the replicated data sets rather than those in the observed data. In this way, a distribution of expected conditional *p*-values can be estimated to compare the observed conditional *p*-values to. This comparison is shown in Figure 11.

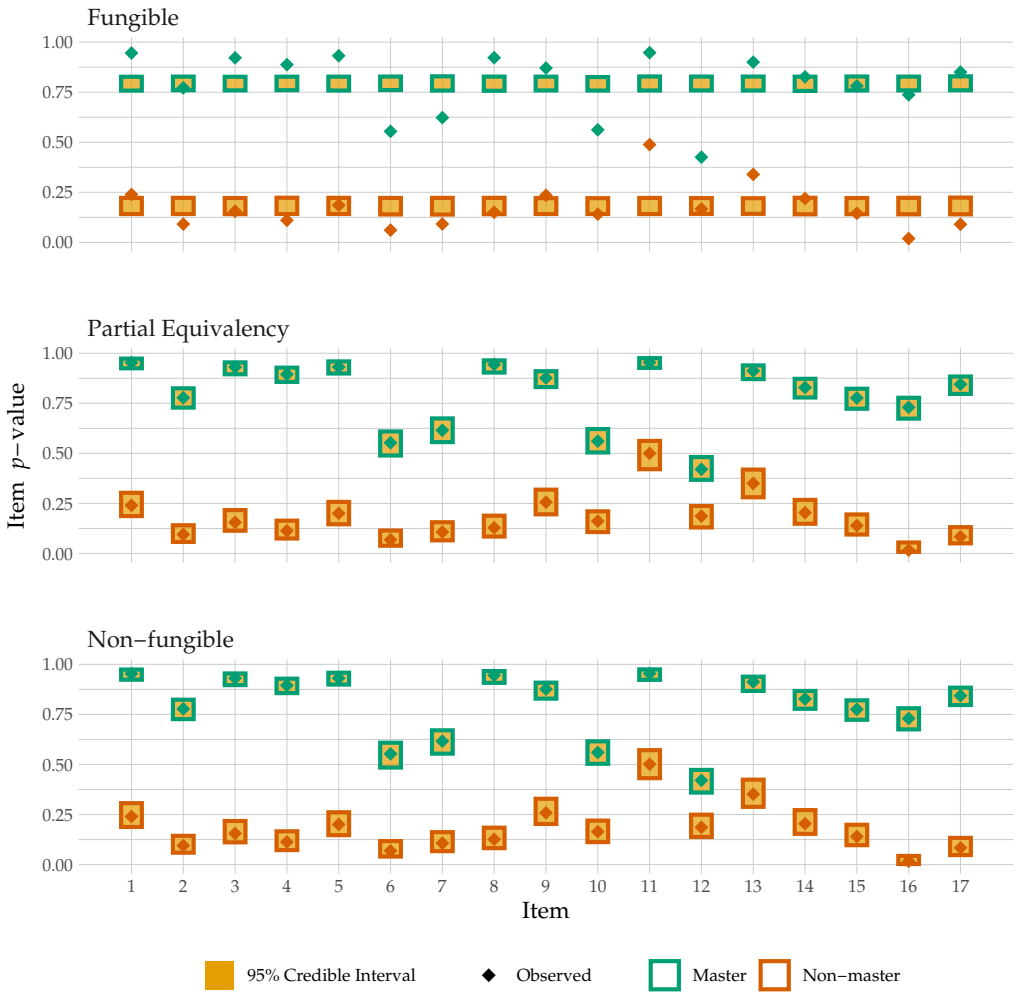

Figure 11. Posterior predictive model check for conditional item *p*-values.

Much like the overall *p*-values (Figure 10), the fungible model shows a consistent expectation across items for each mastery class, which often misses the observed value. Additionally, the partial equivalency and non-fungible models successfully recover the observed conditional *p*-values for both classes. However, Figure 11 shows that the fungible model tends to miss *p*-values for the master class more often and by a greater magnitude than for the non-master class. Thus, it appears that there is more non-fungibility in item parameters among the master than non-master class for the data that was simulated for this document. This can also be seen when examining the true parameter values that were used to simulate the data in Table 1, which show wider variability for $\lambda_{1,1} + b_{i,1,1}$ than for $\lambda_0 + b_{i,0}$.

3.4.2. Relative Fit

Relative fit is assessed through the comparison of models to determine if one model has relatively better fit than another. Because there is no direct comparison to the data with these methods, it is important that models show adequate absolute model fit before they are compared to each other to determine the best fitting model (Sen & Bradshaw, 2017). Therefore, the fungible model is excluded from these analyses, as adequate absolute model fit was not demonstrated in this example data.

In this document, three methods of comparison are demonstrated: Pareto smoothed importance sampling leave-one-out cross validation (PSIS-LOO; Vehtari, Gelman, & Gabry, 2017; Vehtari, Gelman, & Gabry, 2019), the widely applicable information criterion (WAIC; Watanabe, 2010), and Bayesian stacking (Yao, Vehtari, Simpson, & Gelman, 2018). Both the PSIS-LOO and WAIC provide point estimates for the out-of-sample prediction accuracy using the log-likelihood posterior distribution. However, although the PSIS-LOO and WAIC are asymptotically equivalent, Vehtari et al. (2017) found that the PSIS-LOO is more robust than the WAIC when weak priors are used and when there are influential observations (e.g., a student providing an incorrect response despite a high probability of success). When comparing these indices in practice, a difference between models is considered significant if the absolute difference in the indices is greater than 2.5 times the standard error of the difference (Bengio & Grandvalet, 2004).

Additionally, model comparisons can be made using model stacking or averaging. These methods work by assigning weights to each model in the comparison that correspond to the weight that should be given to predictions from each model. Thus, the more weight assigned to a model, the more preferred it would be in isolation. This method has the benefit of being less prone to over-fitting (Piironen & Vehtari, 2017) and also allowing for more refined inferences (Vehtari & Ojanen, 2012). For example, because the weights are on a probability scale, they are much easier to interpret and compare across models than the PSIS-LOO or WAIC. The Bayesian stacking method described by Yao et al. (2018) is implemented.

Table 6 shows that the PSIS-LOO and WAIC fit indices are very similar for both the partial equivalency and non-fungible models. This is expected, as the PSIS-LOO and WAIC are asymptotically equivalent (Vehtari et al., 2017). In the model comparisons, both the PSIS-LOO and WAIC prefer the partial equivalency model, as indicated by the zero in the model comparison (the difference between that model and the preferred model). However, the difference in PSIS-LOO and WAIC of -0.3 between the partial equivalency and non-fungible model is less than 2.5 times the standard error of the difference (1.5). Thus, using the criteria outlined by Bengio and Grandvalet (2004), these models fit equally well. In contrast, Bayesian stacking shows a moderate preference for the partial equivalency. In this comparison, nearly two-thirds of the weight is given to the partial

equivalency model. Given the totality of these analyses, the partial equivalency model appears to be the model best suited to this simulated data. This data was generated from the partial equivalency model, so this finding is consistent with what is expected.

Table 6. Relative Fit Indices and Model Comparisons

	Fit Indices		Model Comparisons		
Model	**PSIS-LOO**	**WAIC**	**PSIS-LOO**	**WAIC**	**Bayesian Stacking**
Partial Equivalency	-8,005.0 (78.5)	-8,004.9 (78.5)	0.0 (0.0)	0.0 (0.0)	0.6315
Non-fungible	-8,005.3 (78.2)	-8,005.2 (78.2)	-0.3 (1.5)	-0.3 (1.5)	0.3685

Note: PSIS-LOO = Pareto smoothed importance sampling leave-one-out cross validation; WAIC = widely applicable information criterion. Parentheses represent the standard error.

4. Discussion

This document presents a Bayesian approach to the estimation and evaluation of DCMs and latent class analyses. As a proof of concept, a simulated data set mimicking the test design of the DLM assessment was used to demonstrate these methods in practice. The Bayesian framework offers several advantages for model estimation and evaluation. First, the ability to place priors on item-level parameters provides a conceptual framework that allows for a straightforward definition of various equality constraints. This allows for a clear delineation of how various models differ from each other.

Following the model definition, there are several existing and well-tested software programs for implementing these models. Specifically, the *Stan* ecosystem provides software for estimating and evaluating Bayesian models in a variety of interfaces. This proof of concept utilizes the *R* interface (Guo et al., 2019), but other interfaces exist for *Python* (PyStan; Stan Development Team, 2018b), *MATLAB* (MatlabStan; Stan Development Team, 2017a), *Julia* (Stan.jl; Stan Development Team, 2018c), *Stata* (StataStan; Stan Development Team, 2017b), and the command-line terminal (CmdStan; Stan Development Team, 2018a). Thus, practitioners have the flexibility to work in the environment they are most comfortable with. In addition to the flexibility in interface, the *Stan* ecosystem also has built-in measures for evaluating the estimation process. This includes measures of model convergence, efficiency, and checks on sampler performance. However, access to this ecosystem currently requires users to be familiar with the *Stan* language in order to write the *Stan* code for each model. Future work should focus on developing a high-level interface to *Stan* for estimating DCMs, similar to the **rstanarm** (Gabry & Goodrich, 2019) and **brms** (Bürkner, 2017; Bürkner, 2018) *R* packages that estimate nonlinear and multilevel regression models.

Additionally, the Bayesian framework facilitates the evaluation of absolute model fit through posterior predictive model checks. This offers a significant improvement over existing model fit measures that rely on unmet assumptions of an asymptotic χ^2 distribution or limited information indices that don't fully capture the complexity of the data. With posterior predictive model checks, replicated data sets from the posterior distributions can be generated and compared to the observed data. This gives a direct and flexible scheme for evaluating model fit. In this document, the raw score distribution, item p-values, and conditional item p-values were examined. However, other characteristics of the data could be calculated and compared. Thus, researchers and practitioners are able to evaluate model based on characteristics they identify as important.

When there are competing models that show adequate absolute model fit, direct comparisons can be made using information criteria. Although using information criteria to compare models is not exclusive to a Bayesian estimation process (e.g., Sen & Bradshaw, 2017), some indices, such as the PSIS-LOO, are. In this document, the PSIS-LOO and WAIC were used, as they are integrated into the *Stan* ecosystem via the **loo** *R* package (Vehtari, Gelman, Gabry, & Yao, 2019) and perform well under a variety of conditions (Vehtari et al., 2017). However, as discussed for posterior predictive model checks, practitioners could use other information criteria for their comparisons, such as the Akaike information criterion (AIC; Akaike, 1973), Bayesian information criterion (BIC; Schwarz, 1978), or deviance information criterion (DIC; Spiegelhalter, Best, Carlin, & van der Linde, 2002).

In summary, this document provides a practical guide to estimating and evaluating DCMs in applied settings. Although the models described represent a simplified one-attribute use case, they are generalizable to additional attributes. For example, Thompson and Nash (2019) used an expanded version of this framework for evaluating map structures and attribute hierarchies in multivariate extensions of these models. Future work will focus on further developing and improving the Bayesian framework for estimating and evaluating DCMs, including extending the methods for evaluating item-level misfit, as well as illustrating how the framework can be implemented in applied and operational settings.

References

Akaike, H. (1973, September). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Csáki (Eds.), *Proceedings of the Second International Symposium on Information Theory* (pp. 267–281). Budapest: Akademiai Kiado.

Bartholomew, D., Knott, M., & Moustaki, I. (2011). Latent class models. In *Latent Variable Models and Factor Analysis: A Unified Approach* (3rd, pp. 157–189). doi:10.1002/9781119970583.ch6

Béguin, A. A., & Glas, C. A. W. (2001). MCMC estimation and some model-fit analysis of multidimensional irt models. *Psychometrika*, *66*, 541–561. doi:10.1007/BF02296195

Bengio, Y., & Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning*, *5*, 1089–1105. Retrieved from http://www.jmlr.org/papers/v5/grandvalet04a.html

Betancourt, M. (2016). Diagnosing suboptimal cotangent disintegrations in Hamiltonian Monte Carlo. arXiv:1604.00695

Betancourt, M. (2017a). *Diagosing biased inference with divergences*. New York, NY: Stan Governing Body. Retrieved from https://mc-stan.org/users/documentation/case-studies/divergences_and_bias.html

Betancourt, M. (2017b). *Robust statistical workflow with RStan*. New York, NY: Stan Governing Body. Retrieved from https://mc-stan.org/users/documentation/case-studies/rstan_workflow.html

Betancourt, M., Byrne, S., & Girolami, M. (2015). Optimizing the integrator step size for Hamiltonian Monte Carlo. arXiv:1411.6669

Betancourt, M., & Girolami, M. (2013). Hamiltonian Monte Carlo for hierarchical models. arXiv:1312.0906

Bradshaw, L. (2016). Diagnostic classification models. In A. A. Rupp & J. P. Leighton (Eds.), *Handbook of Cognition and Assessment* (pp. 297–327). doi:10.1002/9781118956588.ch13

Bradshaw, L., Izsák, A., Templin, J., & Jacobson, E. (2014). Diagnosing teachers' understandings of rational numbers: Building a multidimensional test within the diagnostic classification framework. *Educational Measurement: Issues and Practice*, *33*, 2–14. doi:10.1111/emip.12020

Bradshaw, L., & Levy, R. (2019). Interpreting probabilistic classifications from diagnostic psychometric models. *Educational Measurement: Issues and Practice*, *38*, 79–88. doi:10.1111/emip.12247

Brooks, S. P., & Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, *7*, 434–455. doi:10.2307/1390675

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*, 1–28. doi:10.18637/jss.v080.i01

Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, *10*, 395–411. doi:10.32614/RJ-2018-017

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., … Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*. doi:10.18637/jss.v076.i01

Chen, J., de la Torre, J., & Zhang, Z. (2013). Relative and absolute fit evaluation in cognitive diagnosis modeling. *Journal of Educational Measurement*, *50*, 123–140. doi:10.1111/j.1745-3984.2012.00185.x

Clark, A. K., & Karvonen, M. (2019, April). Teacher assessment literacy: Implications for diagnostic assessment systems. In *Assessment as Feedback for Teachers and Students*. Paper presented at the National Council on Measurement in Education annual meeting, Toronto, Canada.

Clark, A. K., Thompson, W. J., & Karvonen, M. (2019). *Instructionally embedded assessment: Patterns of use and outcomes* (Technical Report No. 19-01). Lawrence, KS: University of Kansas, Accessible Teaching, Learning, and Assessment Systems (ATLAS).

Dynamic Learning Maps Consortium. (2016). *2014–2015 Technical Manual—Integrated Model*. Lawrence, KS: University of Kansas, Center for Educational Testing and Evaluation.

Dynamic Learning Maps Consortium. (2017). *2015–2016 Technical Manual Update—Integrated Model*. Lawrence, KS: University of Kansas, Center for Educational Testing and Evaluation.

Feldberg, Z., & Bradshaw, L. (2019, April). Reporting results from diagnostic classification models for teachers. In *Assessment as Feedback for Teachers and Students*. Paper presented at the National Council on Measurement in Education annual meeting, Toronto, Canada.

Gabry, J., & Goodrich, B. (2019). *RStanArm: Bayesian applied regression modeling via Stan*. R package version 2.19.2. Retrieved from https://CRAN.R-project.org/package=rstanarm

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis* (3rd). Boca Raton, FL: Chapman and Hall/CRC.

Gelman, A., & Hill, J. (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. doi:10.1017/CBO9780511790942

Gelman, A., & Rubin, D. B. (1992). Inference form iterative simulation using multiple sequences. *Statistical Science*, *7*, 457–511. doi:10.1214/ss/1177011136

Geyer, C. J. (2011). Introduction to Markov Chain Monte Carlo. In S. Brooks, A. Gelman, G. L. Jones, & X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo* (pp. 3–48). Boca Raton, FL: Chapman and Hall/CRC.

Guo, J., Gabry, J., & Goodrich, B. (2019). *RStan: R interface to Stan*. R package version 2.19.2. Retrieved from https://CRAN.R-project.org/package=rstan

Henson, R. A., Templin, J. L., & Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, *74*, 191–210. doi:10.1007/S11336-008-9089-5

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively setting path lengths in Hamilttonian Monte Carlo. *Journal of Machine Learning Research*, *15*, 1593–1623. Retrieved from http://jmlr.org/papers/v15/hoffman14a.html

Hu, J., Miller, M. D., Huggins-Manley, A. C., & Chen, Y.-H. (2016). Evaluation of model fit in cognitive diagnosis models. *International Journal of Testing*, *16*, 119–141. doi:10.1080/15305058.2015.1133627

Kroese, D. P., Taimre, T., & Botev, Z. I. (2011). Variance reduction. In *Handbook of Monte Carlo Methods* (pp. 347–380). doi:10.1002/9781118014967.ch9

Liu, Y., Tian, W., & Xin, T. (2016). An application of M_2 statistic to evaluate the fite of cognitive diagnostic models. *Journal of Educational and Behavioral Statistics*, *41*, 3–26. doi:10.3102/1076998615621293

Maydeu-Olivares, A., & Joe, H. (2005). Limited- and full-information estimation and goodness-of-fit testing in 2^n contingency tables: A unified framework. *Journal of the American Statistical Association*, *100*, 1009–1020. doi:10.1198/016214504000002069

Maydeu-Olivares, A., & Joe, H. (2006). Limited information goodness-of-fit testing in multidimensional contingency tables. *Psychometrika*, *71*, 713–732. doi:10.1007/s11336-005-1295-9

McElreath, R. (2015). Markov chain Monte Carlo. In *Statistical Rethinking: A Bayesian Course with Examples in R and Stan* (1st, pp. 241–265). doi:10.1201/9781315372495

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. L. Jones, & X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo* (pp. 113–162). Boca Raton, FL: Chapman and Hall/CRC.

Piironen, J., & Vehtari, A. (2017). Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, *27*, 711–735. doi:10.1007/s11222-016-9649-y

R Core Team. (2019). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. Retrieved from https://www.R-project.org/

Ravand, H., & Baghaei, P. (2019). Diagnostic classification models: Recent developments, pracical issues, and prospects. *International Journal of Testing*. doi:10.1080/15305058.2019.1588278

Ravand, H., & Robitzsch, A. (2015). Cognitive diagnostic modeling using R. *Pracical Assessment, Research & Evaluation*, *20*, 1–12. Retrieved from https://pareonline.net/getvn.asp?v=20&n=11

Rupp, A. A., Templin, J., & Henson, R. A. (2010). *Diagnostic Measurement: Theory, Methods, and Applications*. New York, NY: Guilford Press.

Rupp, A. A., & van Rijn, P. W. (2018). GDINA and CDM packages in R. *Measurement: Interdisciplinary Research and Perspectives*, *16*, 71–77. doi:10.1080/15366367.2018.1437243

Schwarz, G. E. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464. doi:10.1214/aos/1176344136

Sen, S., & Bradshaw, L. (2017). Comparison of relative fit indices for diagnostic model selection. *Applied Psychological Measurement*, *41*, 422–438. doi:10.1177/0146621617695521

Sessoms, J., & Henson, R. A. (2018). Applications of diagnosic classification models: A literature review and critical commentary. *Measurement: Interdisciplinary Research and Perspectives*, *16*, 1–17. doi:10.1080/15366367.2018.1435104

Sinharay, S., & Almond, R. G. (2007). Assessing fit of cognitive diagnostic models: A case study. *Educational and Psychological Measurement*, *67*, 239–257. doi:10.1177/0013164406292025

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & van der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society, Series B*, *64*, 583–639. doi:10.1111/1467-9868.00353

Stan Development Team. (2017a). *MatlabStan: The Matlab interface to Stan*. Retrieved from https://mc-stan.org

Stan Development Team. (2017b). *StataStan: The Stata interface to Stan*. Retrieved from https://mc-stan.org

Stan Development Team. (2018a). *CmdStan: The command-line interface to Stan*. Version 2.18.0. Retrieved from https://mc-stan.org

Stan Development Team. (2018b). *PyStan: The Python interface to Stan*. Version 2.17.1.0. Retrieved from https://mc-stan.org

Stan Development Team. (2018c). *Stan.jl: The Julia interface to Stan*. Version 3.5.0. Retrieved from https://mc-stan.org

Stan Development Team. (2019a). Brief guide to Stan's warnings. Retrieved from https://mc-stan.org/misc/warnings.html

Stan Development Team. (2019b). Stan best practices. Retrieved from https://github.com/stan-dev/stan/wiki/Stan-Best-Practices

Stan Development Team. (2019c). *Stan user's guide*. New York, NY: Stan Governing Body. Retrieved from https://mc-stan.org/docs/2_19/stan-users-guide/index.html

Stroup, W. W. (2012). *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*. Boca Raton, FL: Chapman and Hall/CRC.

Templin, J., & Bradshaw, L. (2013). Measuring the reliablity of diagnostic classification model examinee estimates. *Journal of Classification*, *30*, 251–275. doi:10.1007/s00357-013-9129-4

Templin, J., & Bradshaw, L. (2014). Hierarchical diagnosic classification models: A family of models for estimaing and testing attribute hierarchies. *Psychometrika*, *79*, 317–339. doi:10.1007/s11336-013-9362-0

Templin, J., & Hoffman, L. (2013). Obtaining diagnostic classification model estimates using Mplus. *Educational Measurement: Issues and Practice*, *32*, 37–50. doi:10.1111/emip.12010

Thompson, W. J., & Nash, B. (2019, April). Empirical methods for evaluating maps: Illustrations and results. In M. Karvonen (Chair), *Beyond Learning Progressions: Maps as Assessment Architecture*. Paper presented at the National Council on Measurement in Education annual meeting, Toronto, Canada.

Vehtari, A., Gelman, A., & Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, *27*, 1413–1432. doi:10.1007/s11222-016-9696-4

Vehtari, A., Gelman, A., & Gabry, J. (2019). Pareto smoothed importance sampling. arXiv:1507.02646

Vehtari, A., Gelman, A., Gabry, J., & Yao, Y. (2019). *loo: Efficient leave-one-out cross-validation and WAIC for Bayesian models*. R package version 2.1.0. Retrieved from https://CRAN.R-project.org/package=loo

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., & Bürkner, P.-C. (2019). Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. arXiv:1903.08008

Vehtari, A., & Ojanen, J. (2012). A survey of Bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys*, *6*, 142–228. doi:10.1214/12-SS102

Wang, W., Song, L., Chen, P., Meng, Y., & Ding, S. (2015). Attribute-level and pattern-level classification consistency and accuracy indices for cognitive diagnostic assessment. *Journal of Educational Measurement*, *52*, 457–476. doi:10.1111/jedm.12096

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, *11*, 3571–3594. Retrieved from http://www.jmlr.org/papers/v11/watanabe10a.html

Yao, Y., Vehtari, A., Simpson, D., & Gelman, A. (2018). Using stacking to average Bayesian predictive distributions (with discussion). *Bayesian Analysis*, *13*, 917–1007. doi:10.1214/17-BA1091