Summer 7-2021

INVESTIGATING THE FIT OF THE GENERALIZED GRADED UNFOLDING MODEL (GGUM) WHEN CALIBRATED TO IRT GENERATED DATA FROM DOMINANCE AND IDEAL POINT MODELS

Abdulla Alzarouni
University of Nebraska-Lincoln, aalzarouni@huskers.unl.edu

INVESTIGATING THE FIT OF THE GENERALIZED GRADED UNFOLDING MODEL (GGUM) WHEN CALIBRATED TO IRT GENERATED DATA FROM DOMINANCE AND IDEAL POINT MODELS

by

Abdulla Alzarouni

A THESIS

Presented to the Faculty of
The Graduate College at the University of Nebraska
In Partial Fulfillment of the Requirements
For the Degree of Master of Arts

Major: Educational Psychology

Under the Supervision of Professor Ralph De Ayala

Lincoln, Nebraska

July, 2021

INVESTIGATING THE FIT OF THE GENERALIZED GRADED UNFOLDING MODEL (GGUM) WHEN CALIBRATED TO IRT GENERATED DATA FROM DOMINANCE AND IDEAL POINT MODELS

Abdulla Alzarouni, M. A.

University of Nebraska, 2021

Advisor: Ralph De Ayala

The assessment of model fit in latent trait modelling, better known as item response theory (IRT), is an integral part of model testing if one is to make valid inferences about the estimated parameters and their properties based on the selected IRT model. Though important, the assessment of model fit has been less utilized in IRT research than it should according to research reviews in the organizational literature domain, with dominance IRT models such as the two-parameter logistic model (2PL) and the three-parameter logistic model (3PL) being the most non-Rasch investigated models in terms of fit assessment. However, there have been less research investigating fit for polytomous dominance models such the Graded Response Model (GRM), and to a lesser extent ideal point models such as the Generalized Graded Unfolding Models (GGUM), both in its dichotomous and polytomous forms. For such reasons, examining fit for the GGUM is paramount and should be investigated thoroughly.

The current study tests for different fit indices when calibrating the GGUM model to generated data from different IRT models. For dichotomous items, the GGUM model is fit to GGUM, 2PL, and 3PL generated data. For polytomous data, the GGUM model is fit polytomous GGUM data with four response categories and the GRM. The tested outcomes consist of type I error and power rates across 100 replications for selected number of items and sample sizes with respect to different model fit indices utilized in previous IRT literature. The fit statistics include

both absolute and relative fit statistics such as AIC and BIC. Also, different GGUM data are generated with different delta distribution ranges for dichotomous data when utilizing relative fit indices.

Results from the simulation study show that relative fit indices performed well in identifying the correct dichotomous data model (i.e., GGUM) when the delta ranges are extended beyond the specified distribution ranges for the dominance models. Also, polytomous GGUM data were identified as the best fitting model in almost all the cases, irrespective of the number of items and sample size. On the other hand, the majority of absolute fit indices did not perform well in identifying fit/misfit. Still, there were some fit indices that performed well in detecting fit/misfit for polytomous items only. A possible reason for the shortcomings of absolute fit indices to detect misfit for the GGUM model in general may have to do with utilizing a particular marginal maximum likelihood estimation (MMLE) density form to calibrate the model parameters. Based on the results, it could be said that relative fit indices show some promise in the assessment of model fit for ideal point IRT models such as the GGUM. This applies for both dichotomous and polytomous generated items.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Fit, Data Models, and a Standard for Appraising Latent Variables

The process of estimating fit for data models of choice is important, particularly when the assumptions of selected data models are assumed to be true. For example, fitting data to a desired model and assessing the degree of fit is common practice in latent trait modeling such as the Rasch model (Nye, Joo, Zhang, & Stark, 2019; Wright, 1979; Wright & Masters, 1982). Like many other models in statistics, the Rasch model has its own assumptions such as the choice of dimensionality and the independence between a test's items and its respective examinees' responses (De Ayala, 2009). Nevertheless, the benefits inherent in statistical models will not hold if the fit between the proposed model and respective data is weak. Other latent trait models such as the two-parameter logistic model (2PL), the three-parameter logistic model (3PL), and the generalized graded unfolding model (GGUM) do not involve fitting data to a model per se, but will still require fitting the model of choice to the data in order to utilize the assumptions inherent within each model.

Despite the importance of estimating fit for validating the models' assumptions, there is controversy as to whether relying on pre-specified models is the 'right' way to go about understanding how both manifest and latent variables function. In an article published in 2001 by Leo Breiman, it is mentioned that data scientists should steer away from pre-specifying models, particularly if the objective of inquiry (i.e., criterion) is related to prediction (Breiman, 2001). Breiman argues that his work as a consultant on different projects involving predictions made him realize the limitations of relying on pre-selected data models when making valid predictions.

For example, he draws a dividing line between the conclusions pertaining to the selected model mechanism for making predictions versus that of algorithmic modeling, with the latter negating the necessity for assuming that a pre-selected data model represents truth. He also criticizes the tendency of many statisticians to fit linear models to data, and subsequently use R^2 to estimate goodness of fit; an inflated index contingent on the number of parameters subsumed by the model. He argues for the implementation of algorithmic techniques that calls for data exploration rather than modeling. Data analysis techniques such as decision trees are recommended by Breiman for making predictions, which are substantially utilized in machine learning contexts when dealing with 'large' datasets. Although the definition of a large dataset varies between academic disciplines, hundreds of variables within a single analysis is usually referred to as 'large' in machine learning domains (Raschka & Mirjalili, 2019).

The criticisms of pre-specified models and fitting them to specific data may 'somewhat' be reasonable if the main objective of the analysis is prediction, given the more pronounced methodologies available for such purposes in data science. In the domain of the social sciences however, interpretability is a major concern, and testing pre-specified models with desirable statistical assumptions aids such a process. The availability of large datasets with hundreds of variables are seldom utilized in the social sciences due to the difficulty of obtaining large sample sizes. Also, dealing with a large number of variables can lead to interpreting an endless set of interactions between the variables, which is a practice that social scientists avoid if interpretability is at stake. For example, when conducting linear regression analysis or ANOVA, parsimony is encouraged via utilizing the minimum number of predictive variables that can explain the highest proportion of variance accounted for by the model. The aforementioned process entails pre-specifying a model such as a linear regression model and keeping the number

of predictor variables to a minimum or avoiding higher order polynomial analyses when possible (Keppel & Wickens, 2004). Such processes would in turn aid in the interpretability of the proposed model.

Because many of the cognitive and non-cognitive variables in the social sciences fall within the latent variable category, their intangible nature mandates pre-specified models that should fit the data to increase the possibility of accurate interpretations. Dealing with intangible variables (i.e., constructs) is widely investigated in the social sciences in general and psychology in particular. For example, the seminal works of Lee Cronbach and Paul Meehl in defining construct validity during the 50's was introduced as many psychologists during that time struggled with attaching absolute definitions to latent concepts. Cronbach and Meehl proposed a 'nomological network' that would serve to define a construct based on its relationship with other constructs as proposed by a pre-defined theory (Cronbach & Meehl, 1955; Loevinger, 1957). Though the formulation of construct validity is not directly related to data models and the importance of fit estimation, it demonstrates justifiable concerns about creating a reasonable standard for appraising latent variables. Still, modern critics of construct validity assert that correlational models for inferring validity are problematic and should be replaced by causal ones (Borsboom, 2009). Their arguments stem from the fact that dealing with latent constructs becomes a tricky business as psychologists such as Cronbach and Meehl try to avoid referential meanings, which is a practice that conforms to the school logical positivism (Borsboom, 2009). In short, it is sort of theorizing without getting into the ontological basis of the attribute. Navigating ontology would mandate delving into the metaphysical domain; a philosophical territory that some empirical scientists try to avoid if they can (Janssen, 2001; Kripke, 2008).

The concerns of creating a standard to appraise and measure latent variables also resonates with methodologists in the social sciences. For example, the premise of the Rasch model is about creating a standard unit for measuring variables. In other words, the log odds (logit) of a desired response can be considered a standard when measuring an attribute, given its property to remain constant across the metric of interest (De Ayala, 2009). Another attempt to create a standard when investigating latent variables came in the form of a unidimensional unfolding model, pioneered by the seminal works of mathematical psychologist Clyde Coombs (Coombs, 1964). The premise of this model is the possible existence of a common latent attribute that is unidimensional and can be conceptualized on a single scale (i.e., referred to as the *J* scale). The proposed scale allows one to gauge the different preference orderings of subjects being tested on a particular attribute. The unfolding model allows both the respondents' preferences and the attribute of interest to be compared in the same dimensional space such that the distances between the respondents' standings on the scale and the stimuli points of the attribute represent the actual psychological proximity of the stimuli to the individual (McIver & Carmines, 1981). There is also a multidimensional unfolding model variant that is an extension of the Coombs unidimensional unfolding model to multivariate response data (Bennett & Hays, 1960; Coombs, 1964; Coombs, Dawes, & Tversky, 1970). The premise of Coombs unfolding is integrated into the derivation of ideal point models in item response theory (IRT) (Roberts & Laughlin, 1996). The ideal point models are the focus of this paper and its different fit indices. However, before delving into ideal point models and fit estimation comparisons, a brief introduction to IRT and the importance of model fit is warranted.

Item Response Theory (IRT) & Model Fit

In the field of psychometrics, item response theory (IRT) data models allow the estimation of an item's response probability given the level of the measured attribute (Bandalos, 2018). Since its inception around 70 years ago by people such as Frederic Lord and Georg Rasch (Lord, 1952; Wright, 1979), IRT or *latent trait modelling* has gained popularity among researchers due to its methodological advantages over other psychometric models such as classical test theory (CTT) and generalizability theory (G theory) (Cronbach, Rajaratnam, & Gleser, 1963; Traub, 2005). IRT models provide both item location (delta δ) and theta (θ) (i.e., person ability) invariant parameters. Invariance is a desired feature in modern testing applications such as computer adaptive testing (CAT; Linden & Glas, 2000), test equating (Cook & Eignor, 1989), and differential item functioning (DIF) (Tay, Meade, & Cao, 2014). The property of invariance would also allow reliability and error indices to be independent from specific items or people utilized for model calibration. For example, researchers can design test items for a criterion-referenced assessment inventory that calls for a specific ability level. This can be achieved by pre-selecting a discrimination parameter (α) that is sample invariant and use its value as an index to retain items that will be on the assessment inventory. Researchers can also make use of the invariance feature in IRT to create parallel test forms, which is possible given the independence of the difficulty index (δ) from the respondents' respective scores (Bandalos, 2018). The invariance features in IRT would allow ability scores (θ) to be compared on a single metric, irrespective of the items (i.e., test forms) or respondents (i.e., test group) used for the calibration process when estimating the model's parameters; a lacking feature in other psychometric models such as CTT in which item difficulties and discriminations are sample dependent.

As mentioned at the outset of this paper, the advantageous features of IRT will not be realized without confirming that a chosen IRT model actually fits the data of interest. Given that, the estimation of model fit for IRT should be a necessary step for such latent data models. Surprisingly, the estimation of model fit in IRT literature is not as common as it should be given its aforementioned advantages when compared to other domains such as structural equation modeling (SEM) (Nye et al., 2019). For example, in organizational research literature, it has been estimated that more than 40% of published articles utilizing IRT models do not include any fit estimations (Foster, Min, & Zickar, 2017). The choice of fitting an incorrect IRT model to a selected data is detrimental to the many features and applications that define the usefulness of such latent models. Such features include but not limited to the construction of assessment scales (Roberts, Laughlin, & Wedell, 1999), the estimation of IRT parameters (DeMars, 2010), CAT (De Ayala, Dodd & Koch, 1992), DIF (Bolt, 2002), and test equating (Kaskowitz & De Ayala, 2001). Also, as cited in Nye et al. (2019), fitting an incorrect IRT model to the data can "*affect the rank order of individuals in a sample*" (p. 460), as well as validity via altering the magnitude of correlations with external variables.

Examining IRT model data fit involves a comparison between the observed responses on test items and those predicted by the fitted IRT model. Such a comparison usually involves examining the squared residuals (r^2_{ni}) between the observed (x_{ni}) and predicted scores (P_{nix}), and summing them up to determine the degree of misfit between the data and fitted model. Although the aforementioned method is generic and would involve additional mathematical manipulations for performing such the needed computations, the majority of IRT model fit methods would follow such a premise. The process of examining and comparing residuals in IRT for model fit estimation usually involves chi-square or likelihood-ratio tests (Ames & Penfield, 2015). These

tests share the basic premise of examining residuals to determine misfit. They differ in terms of setting a criterion for grouping respondents based on either their ability levels or observed test scores, which will be discussed in the next chapter. Still, there are alternative methods to model fit estimation that are prevalent in the SEM literature, such as those involving the estimation of approximate fit (Maydeu-Olivares & Joe, 2014), or posterior predictive checks that involves a Bayesian approach of model evaluation (Rubin, 1984; Sinharay, Johnson, & Stern, 2006). Although chi-square, likelihood-ratio, and approximate fit methods will be explained in further detail in the next chapter, the paper will not be covering Bayesian methods of fit. Interested readers are referred to Ames and Penfield (2015) and Sinharay et al. (2006).

When it comes to applying model fit estimation to IRT models, the majority of the research literature focuses on applying fit estimation to *dominance* IRT models (Nye et al., 2019). These models use a monotonically increasing function that allows the desired response probability to increase relative to the level of the latent trait (De Ayala, 2009; Roberts & Laughlin, 1996). In other words, respondents with higher ability levels theta (θ) will have higher probabilities of responding correctly on an item. Dominance or *cumulative* models can include both dichotomous (i.e., binary) and polytomous (i.e., graded) data, and can accommodate both unidimensional and multidimensional models. The 1, 2, 3 parameter-logistic models, and the graded response model (GRM; Samejima, 1969) are examples of dominance based models.

Another class of IRT models is referred to as *ideal point models* (Coombs, 1964). As mentioned above, ideal point models are influenced by Coombs unfolding in terms of measuring the distance between an item and a response as an indicator of preference/agreement. These models assume that a person's response to an item located on the latent trait continuum (i.e., analogous to the *J* scale in Coombs unfolding) will be close in proximity, contingent on whether

the item's content matches the person's actual standing on the latent trait. In simple terms, individuals are more likely to endorse an item that matches their location on the latent trait. Conversely, extreme items are less likely to be endorsed given the greater distance between their respective locations to that of the respondent. The graded unfolding model (GUM) and the generalized graded unfolding model (GGUM) are examples of ideal point models (Roberts & Laughlin, 1996; Roberts, Donoghue, & Laughlin, 2000). Ideal point models can also accommodate both dichotomous and polytomous data, as well as unidimensional and multidimensional models (Wang & Wu, 2015). Although there have been attempts by researchers to examine item and model data fit for ideal point models such as the GGUM (Roberts, 2008; Nye et al., 2019), there is a shortage of analyses pertaining to such an objective. As mentioned earlier, the majority of publications covering model fit estimation for IRT examined dominance based IRT models. For such reasons, it is incumbent to investigate which method(s) of fit works best with ideal point IRT models. This paper will be comparing different fit indices for the GGUM under different conditions pertaining to the number of items, sample size, and item response type (i.e., dichotomous and polytomous). This study considers only unidimensional data as well as IRT generated models that assume a continuous latent trait. Based on the results, suggestions will be made as to which fit statistics are the most useful for the IRT unfolding model.

CHAPTER 2

LITERATURE REVIEW

IRT Dominance Models

For dichotomous unidimensional data, IRT logistic models are usually utilized to estimate the desired parameters for dominance models. For example, the three-parameter logistic model (3PL) is represented by the following equation:

$$p(x_i = 1|\theta, \alpha_i, \delta_i, \chi_i) = \chi_i + (1 - \chi_i) \frac{e^{\alpha_i(\theta - \delta_i)}}{1 + e^{\alpha_i(\theta - \delta_i)}} \qquad (1)$$

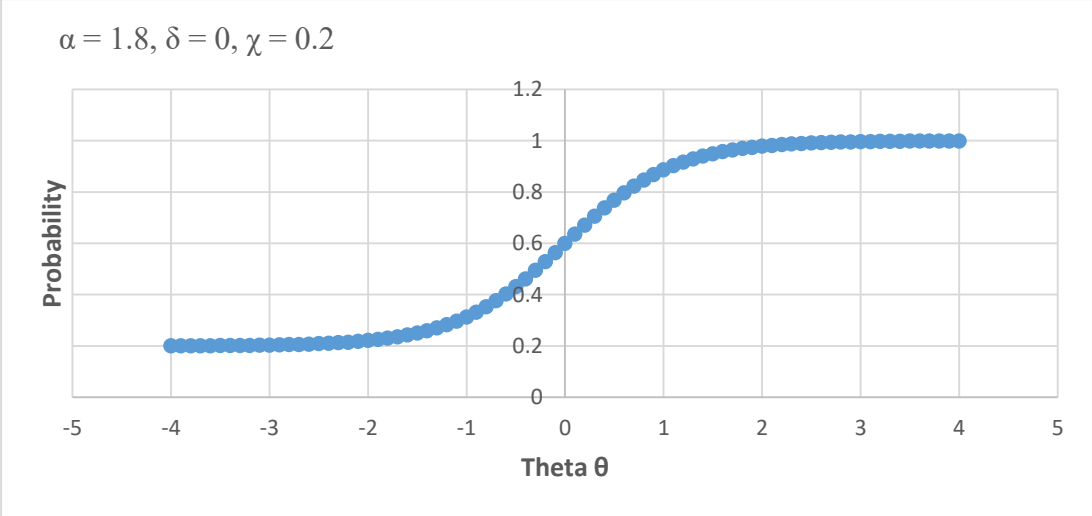
where p is the probability of a response (e.g., correct response/endorsement) given the latent trait of interest (θ), α_i is the discrimination parameter for item *i*, in which items with higher positive values of α will discriminate better between respondents given their expected locations on the latent trait, δ_i is the difficulty parameter for item *i*, in which items located towards the higher end of the ability continuum θ (i.e., higher positive values of δ) will usually be more difficult to answer correctly or endorse, χ_i is the guessing parameter, with higher values indicating a higher probability of a correct response for respondents on item *i*, particularly those with lower θ values (Birnbaum, 1968). As mentioned in (De Ayala, 2009), a scaling factor *D* is sometimes presented in equation 1 due to the existence of a normal ogive model for the 3PL, which functions to minimize the "*difference between the normal and the logistic* distribution functions" (Camilli, 1994). By adding the scaling factor *D*, which is about 1.702 and re-adjusting the formula for efficiency, the 3PL equation will be in the following form:

$$p(\theta) = \chi_i + (1 - \chi_i) \frac{1}{1 + e^{-D\alpha_i(\theta - \delta_i)}} \qquad (2)$$

Other logistic IRT models such the two- and one-parameter (i.e., 2PL and 1PL) models are nested versions of the 3PL. The 2PL model will exclude the guessing parameter, while the 1PL

will also set $\chi_i = 0$ but in addition will constrain the discrimination parameter α_i to be equal across items. Figure 1 displays the item response function (IRF) (i.e., item characteristic curve) for a hypothetical item calibrated using the 3PL model with $\alpha = 1.8$, $\delta = 0$, and $\chi = 0.2$:

Figure 1. *IRF for a 3PL Model*



The process of fitting IRT models involves the estimation of item and person parameters, which include δ, α, χ, and θ for the 3PL model. Marginal maximum likelihood estimation (MMLE) is performed for recovering the item parameters (Bock & Aitkin, 1981; Bock & Lieberman, 1970). MMLE resolves some of the inherent problems with other MLE approaches such as the joint maximum likelihood estimation (JMLE), which involves estimating the item parameters from a fixed set of person parameters. MMLE resolves this problem via estimating the item parameters from the larger population distribution. Conditioning the item parameters on the population distribution resolves the issue of re-calibrating the instrument multiple times due to the possible removal of misfitting items, which would require re-calibrating the person locations all over again (De Ayala, 2009). Statistical packages such R and M*plu*s (Muthén & Muthén, 1998-2017) can be used to calibrate IRT models' parameters. Person location estimates

(θ) may be obtained via *expected a posteriori* (EAP) (Bock & Mislevy, 1982), in which the estimated ability parameter ($\hat{\theta}$) corresponds to the mean of the posterior distribution. By default, many R packages calculating EAP will assume a normal distribution for the *prior probability distribution* (e.g., mirt; Chalmers, 2012; GGUM; Tendeiro & Castro-Alvarez, 2020).

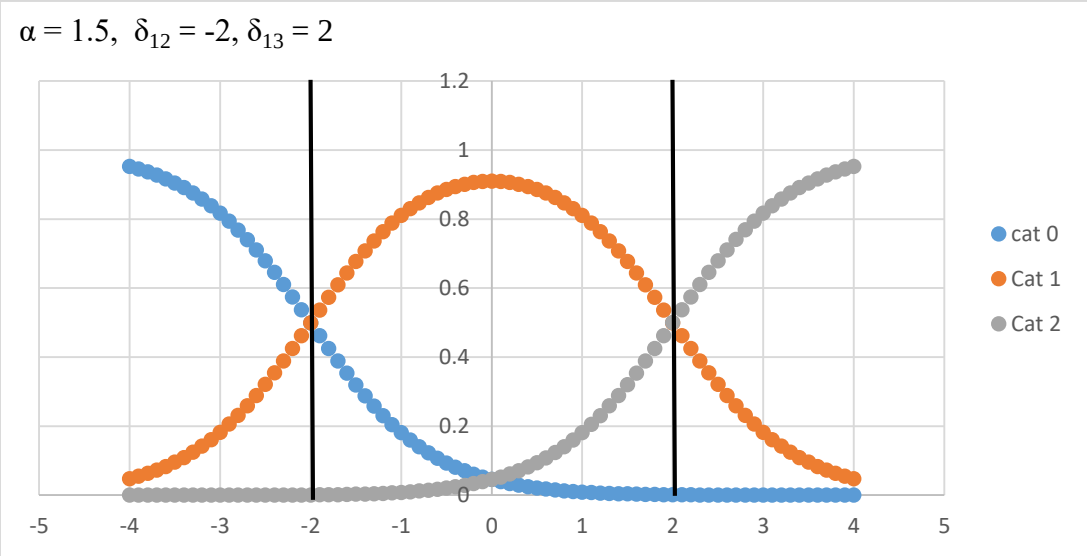
For polytomous unidimensional data, a typical model would include response categories per item, with such categories and their respective scores having option response functions (ORFs). These ORFs represent the probabilities of obtaining designated scores contingent on the level of the latent trait theta (θ). In ordered polytomous IRT models that divide responses into a set of ordered pairs of adjacent categories, transitioning between option response functions per item occurs at respective transition locations (δ_{ih}s) between the ordered category pairs. Given that such item transition locations separate the intervals associated with category scores with respect to (θ), the number of intervals per item will be (k + 1) relative to the item transition locations. The premise of having such transition locations be ordered in terms of their magnitude will be dependent upon the selected polytomous model, but such an assumption is not a necessary condition when calibrating polytomous models in general (De Ayala, 2009). Although there are many polytomous IRT models to introduce, only two will be briefly mentioned given their relevance to subsequent analyses. One of the two dominance models is selected due to its similarity to the GGUM, which is the main focus of the model fit analyses on the next chapter. The first model is the *generalized partial credit model* (GPCM) (Muraki, 1992), which is defined in the following equation:

$$\mathrm{P}(Y_{\mathrm{i}} = y \mid \theta_{\mathrm{j}}) = \frac{\exp\{\alpha_i \left[y\left(\theta_j - \delta_i\right) - \sum_{k=0}^{y} \tau_{ik}\right]\}}{\sum_{w=0}^{M}\{\exp\{\alpha_i \left[w\left(\theta_j - \delta_i\right) - \sum_{k=0}^{w} \tau_{ik}\right]\}\}} \qquad (3)$$

where $\sum_{k=0}^{y=M} \tau_{ik} = 0$, Y_i represents the probability of a response in item *i*'s *y* category, ((Y_i= y|θ) = 0, 1, 2, ..., *M*), *M* corresponds to the number of response categories minus 1, θ_j is the location

of person *j* on the latent continuum, δ_i is the location of item *i* on the latent continuum, α_i is the discrimination of item *i*, and τ_{ik} represents the location of *k*th response category threshold on the latent continuum with respect to the *i*th item location. In short, equation 3 divides the probability of selecting a specific response category given theta (θ) over the sum of all the probabilities corresponding to the locations of the response categories for a specific item conditional on theta (θ). Note that the GPCM response categories are separated by respective thresholds (i.e., τ_{ik}). These thresholds can be sequentially ordered in more constrained versions of the GPCM for respective response categories such as the rating scale model (RSM) (Masters, 1982), but are allowed to be unordered for the GPCM and ideal point models such as the GGUM (Roberts et al., 2000). Figure 2 displays the option response function (ORF) for a three-category hypothetical item calibrated using the GPCM model with $\alpha = 1.5$, $\delta_{12} = -2$, and $\delta_{13} = 2$:

Figure 2. *ORF for the GPCM Model*



The second polytomous IRT model for dominance data is the *graded response model* (GRM) (Samejima, 1969). This model differs from the GPCM in defining the probability of a response relative to the specified response categories per item. In GPCM, the premise was estimating the probability of a response in a specific response category, and how the probability

would change when transitioning to an adjacent response category accordingly as shown in Figure 2. The GRM compares response probabilities in a cumulative fashion, in which a specific point is selected on the latent continuum relative to the response categories that would define the comparison. For example, in a three-category item, a comparison of probability pertaining to a response might compare category 0 (i.e., obtaining a score of 0) to that of category one and two together (i.e., obtaining a score of 1 or higher). The estimation of the cumulative probabilities can be achieved via utilizing dichotomous models such as the 2PL (Samejima, 1969; De Ayala, 2009). This follows utilizing a series of 2PL models to a sequential series of responses, which eventually yield the expected probabilities for the GRM response categories. The probabilities for the response categories are complements to one other. For example, when examining an item with three-response categories, the probability of responding in any of the categories will be 1, while the probability of scoring in category 0 will be equal to 1 minus the probability of scoring in category 1 or higher. The probability of scoring in category 1 or 2 rather than category 0 will be equal to the difference between the probabilities of being in category 1 from that of being in category 2. Finally, the probability of being in category 2 or higher is just the probability of being in category 2 since the portability of being at a higher category is 0. For illustrative purposes, the following equation is taken from (De Ayala, 2009), which demonstrates how to obtain the probability of scoring in category 1 or 2 rather than category 0. Note that P* indicates cumulative probabilities, and δ_i, α_i, and θ are the category boundary location, item discrimination, and person location parameters respectively:

$$P_1 = P_1^* - P_2^* = p(x_i = \{1,2\}|\,\theta) - p((x_i = \{2\}|\,\theta) = \frac{e^{\alpha_i(\theta-\delta_1)}}{1 + e^{\alpha_i(\theta-\delta_1)}} - \frac{e^{\alpha_i(\theta-\delta_2)}}{1 + e^{\alpha_i(\theta-\delta_2)}} \qquad (4)$$

If the 2PL model is applied for each category boundary location δ_i separately with respect to θ, then we obtain cumulative probability curves corresponding to such boundary locations,

sometimes referred to as *category boundary curves*. Figure 3 illustrates the category boundary curves for a three-category item with $\alpha = 2$, $\delta_1 = -2$, $\delta_2 = 2$:

Figure 3. *Boundary Category Curves for the GRM Model*

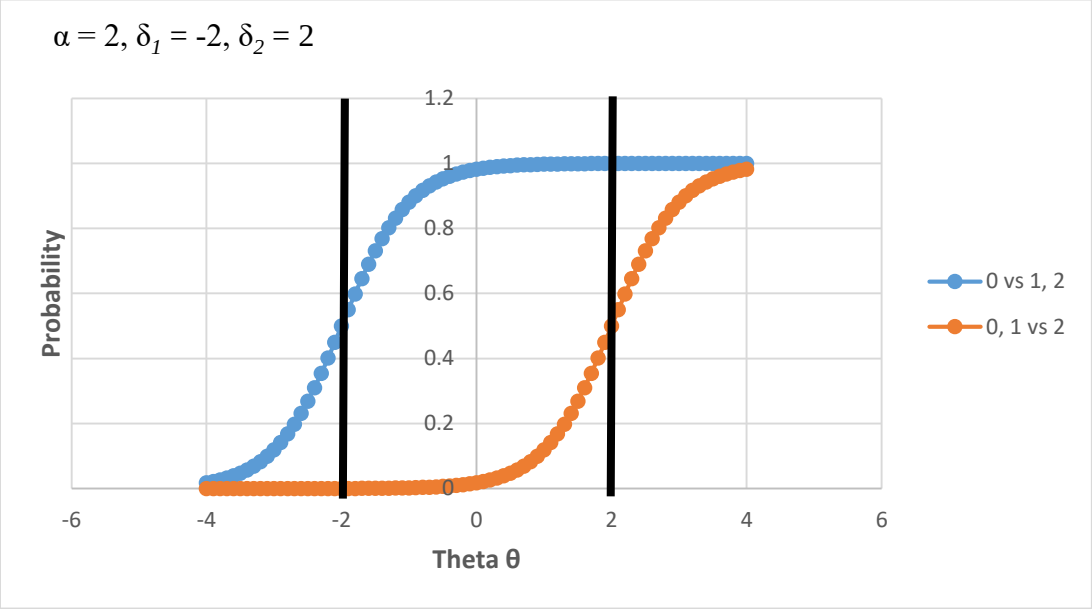
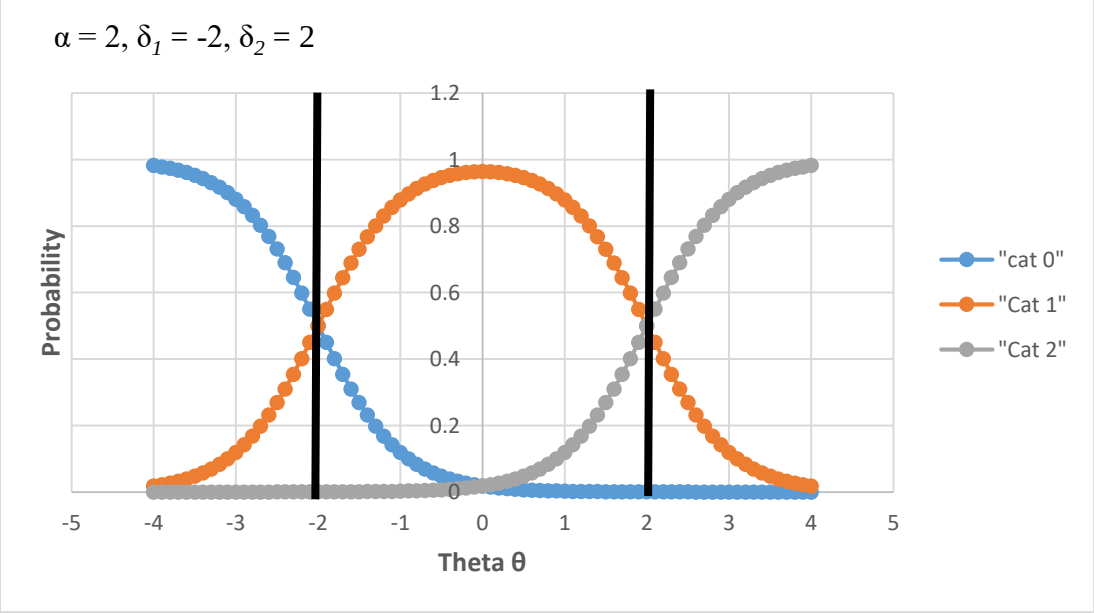


Figure 4 displays the ORFs for the same hypothetical item. Note that the orange curve in Figure 4 can be obtained by substituting the respective values of α and δ_i in equation 4 across θ:

Figure 4. *ORF for the GRM Model*



The process of fitting IRT models for polytomous data involves the estimation of both item and person parameters, which follows similar procedures to that of dichotomous data. MMLE and EAP estimation processes are also utilized for estimating item and person parameters respectively.

In the next chapter, The GRM will be selected as the IRT dominance model for subsequent model fit comparisons with an ideal point model (i.e., GGUM) for polytomous data. The GPCM on the other hand, represents one of the premises that defines the GGUM model, which is going to be introduced in the following section as the ideal point IRT model of choice. The GGUM will also be the calibrated model of choice for model fit analyses given the limited amount of research done to that effect.

IRT Ideal Point Models

The theory behind ideal point models was first suggested by Thurstone (1928) as a way of measuring attitudes, in which the endorsement of presented statements is related to how similar these statements are to the actual attitude of the individuals. As mentioned in the introduction, ideal point models in IRT are inspired from Coombs unfolding (Coombs, 1964), which works well with non-cognitive items (i.e., attitudes) in terms of assessing their psychological proximity to the actual attitudes of the responding individuals (Roberts & Laughlin, 1996). Ideal point models do not assume a cumulative monotonic response function as in dominance models, but rather an unfolding single-peaked response function (Roberts, et al., 2000). Many researchers argue that both dichotomous and polytomous attitude statements, in which some sort of self-reflection is required are better captured by ideal point models (Drasgow, Chernyshenko, & Stark, 2010; Nye et al., 2019; Roberts, et al., 2000). For example, Drasgow et al. (2010) suggests that in organizational research, inventories requiring employees

to introspect are better modeled by ideal point processes. Nye et al. (2019) also cites several studies demonstrating that ideal point models (i.e., unfolding models) are superior to dominance based models when assessing personality, vocational interests, person-organization fit, performance ratings, and job attitudes. Unfolding models are also useful for item-level analyses when investigating response sets (e.g., malingering) associated with non-cognitive assessment inventories (Liu & Zhang, 2020; Scherbaum, Sabet, Kern, & Agnello, 2013).

The generalized graded unfolding model (GGUM) is an ideal point IRT model introduced by Roberts and colleagues (Roberts et al., 2000), with a constrained version of the model known at the *graded unfolding model* (GUM) being introduced prior to the generalized version (Roberts & Laughlin, 1996). Both the GGUM and GUM were developed under four basic premises relative to the response process. Note that all of the explanations to follow assume a unidimensional latent trait, and are based on the explanations in Roberts et al. (2000).

The first premise is that expected agreements of respondents to items/statements will be contingent on the items' relative positions to respondents' actual positions on the latent continuum representing the construct. Put simply, as the values of the *i*th item δ_i and the *j*th person θ_j approach one another, the distance between them approaches 0 and it is expected that person *j*'s likelihood of agreement to item *i* will be high.

The second premise is that a person can select a specific response category (e.g., "disagree") for two reasons. The first has to do with the person having a more positive attitude than the item's content, hence disagreeing from above. The second reason has to do with a person holding a more negative attitude than the item's content, hence disagreeing from below. In other words, there are two subjective responses for every observable response on a rating scale.

The third premise is that subjective responses (e.g., disagreeing from "above or "below" an item) to statements follow a dominance (i.e., cumulative) item response model. Muraki's (1992) GPCM can model GGUM's subjective response functions, hence its introduction in the previous section as one of the assumptions defining GGUM. In short, the subjective response category probability functions follow a cumulative model. Also, the number of response categories will be doubled because of the two possibilities for each observable response category. For example, a hypothetical item with four observable response categories (ORCs): *strongly disagree, disagree, agree, and strongly agree* can be modeled using the GPCM with seven subjective response category (SRCs) thresholds (τ_{ik}s). There are two subjective responses for every observable response (i.e., eight intervals in total). Also, the dominance of the most likely subjective response within the intervals is determined by the discrimination parameter (α_i). As mentioned in Roberts et al. 2000, the model's SRCs must be transformed into an ORC format that is compatible with the graded agreement scale. Since the two subjective response categories are mutually exclusive, the probability of a response within an observed response category will be equal to the sum of the respective probabilities related to the two subjective response categories.

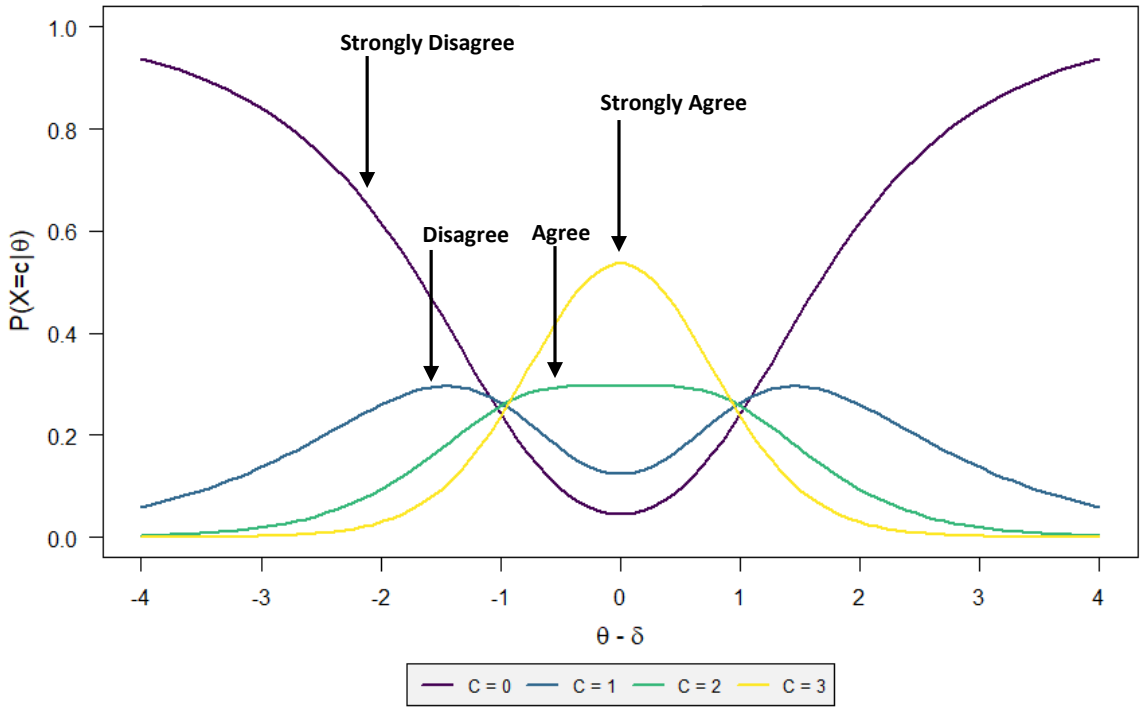
The fourth premise builds on the idea that subjective response categories must be defined in terms of the actual observable response category. The response category thresholds τ_{ik}s will be symmetric about the point $(\theta_j - \delta_i) = 0$. In short, premise four states that respondents have an equal probability of agreeing to an item situated along the latent continuum by either $-h$ or $+h$ units from their positions on the attitude continuum. By applying the forth premise, we get the following identity: $\sum_{k=0}^{M-c} \tau_{ik} = 0$. By integrating the following identity and taking into account that the sum of the mutually exclusive subjective categories would yield the observed response

category probability functions, the GPCM model in equation 3 can be modified to yield the formal definition the GGUM:

$$P(C_i = c \mid \theta_j) = \frac{\exp\{\alpha_i [c(\theta_j - \delta_i) - \sum_{k=0}^{c} \tau_{ik}]\} + \exp\{\alpha_i [(M-c)(\theta_j - \delta_i) - \sum_{k=0}^{c} \tau_{ik}]\}}{\sum_{w=0}^{B}\{\exp\{\alpha_i [w(\theta_j - \delta_i) - \sum_{k=0}^{w} \tau_{ik}]\}\} + \exp\{\alpha_i [(M-c)(\theta_j - \delta_i) - \sum_{k=0}^{c} \tau_{ik}]\}} \quad (5)$$

where C_i represents the observable response to item i, c = 0 (z = 0, 1, 2, ..., B) indicates the strongest level of disagreement, $c = B$ indicates the strongest level of agreement and is equal to the number of observable response categories minus 1, $M = 2B + 1$, θ_j is the location of person j on the latent continuum, δ_i is the location of item i on the latent continuum, α_i is the discrimination of item i, and τ_{ik} represents the location of kth response category threshold on the latent continuum with respect to the ith item location (Roberts et al., 2000). Figure 5 displays the observable response categories (ORC's) probability functions for a hypothetical four-category item as a function of $\theta_j - \delta_i$. C denotes the observed responses from 0 to 3:

Figure 5. *ORC Probability Functions for a GGUM Four-Category Item*



Note. This figure was produced in R using the package 'GGUM' by Tendeiro, J. N., and Castro-Alvarez, S. (2020). GGUM: Generalized Graded Unfolding Model. R package version 0.4-1. https://CRAN.R-project.org/package=GGUM

The graded unfolding model (GUM) is a constrained variant of the GGUM, in which the discrimination parameters are set to unity and the threshold parameters are equal across items (Roberts & Laughlin, 1996). Changing the discrimination parameter will alter the magnitude of the expected values function, in which larger values of α_i yield more peaked expected value functions reaching their upper bound. Also, changing the threshold values τ_{ik} by increasing the distance between them will also elevate the expected value function to its upper bound but will decrease its steepness in a simultaneous fashion (Roberts et al., 2000). The dichotomous unfolding data model is a simplified application of the polytomous variant using only two

response categories such as ‘agree’ or ‘disagree’, and can be modelled using both GUM or GGUM.

The process of fitting IRT ideal point models for data involves the estimation of both item and person parameters, which follows similar procedures to that of IRT dominance models. MMLE and EAP estimation processes are utilized for estimating item and person parameters respectively. Although JMLE was utilized to estimate the item parameters for the GUM in Roberts and Laughlin, (1996), subsequent publications utilized an MMLE estimation process in calibrating the item parameters for both the GUM and GGUM.

Since the GGUM assumes unidimensionality, a principal component analysis (PCA) can be performed to verify such an assumption. In Davison (1977), it is shown that responses adhering to a simple unfolding model will yield two principal components. In short, a unidimensional unfolding model can be inferred from a scree plot identifying two dominant eigenvalues from a polychoric correlation matrix (De Ayala & Hertzog, 1991). Another rule of thumb for assessing dimensionality in unfolding models entails estimating the final communalities of the first two principal components and examining whether the respective communality value is greater or equal to 0.3 (Roberts et al., 2000).

The GGUM for both dichotomous and polytomous data follows a parametric approach, which allows the computation of attitude estimates to be invariant of the respective items used in calibration. The invariance property is also applicable to the item locations, which are invariant to the responses of the examinees constituting the attitude of interest in a sample (Roberts et al., 2000). Item discrimination parameters are also invariant to the responses of the examinees, and are tested for invariance via different methods that mainly involve the examination of the interaction between item location and discrimination parameters between selected subsamples.

Interested readers are referred to De Ayala (2009) for an overview on assessing invariance. The applicability of the invariance property is only realized once the unfolding model fits the data (Hoijtink, 1990).

IRT Model Fit Statistics

In theory, the majority of estimation processes examining IRT model fit would involve comparing the individual-level residuals between the observed (x_{ni}) and predicted scores (P_{nix}). However, a case of perfect model fit will still be short from yielding individual-level residuals that are equal to zero. This occurs as a result of limiting the observed scores (x_{ni}) to a set of fixed values such as 0 or 1 for dichotomous data, while varying the respective item response function that is used for estimating the predicted scores (P_{nix}) to range from 0 to 1 (Ames & Penfield, 2015). This problem led statisticians to come up with different ways to estimate model fit for IRT models. One such solution is to sort individual scores into distinct groups *h* based on their ability estimate $\hat{\theta}$, sometimes referred to as '*binning*'. This process allows a comparison of observed and expected scores within each bin, hence allowing the residuals in theory to equal zero in cases of good model fit; the usefulness of such binning process is dependent on sample size.

As mentioned in the introduction, many of the model fit techniques utilize a chi-square approach, with the generic form of the estimation process for dichotomous data presented in the following equation:

$$\chi_i^2 = \sum_{h=1}^{H} N_{hi} \frac{(r_{hi})^2}{P_{hi}(1 - P_{hi})} \qquad (6)$$

where r_{hi} represents the bin-level residuals and N_{hi} refers to the number of respondents in bin *h* attempting item i. Equation 6 incorporates the residuals relative to the selected bins rather than

the whole item, which makes it possible to obtain zero residuals when estimating model fit. Three chi square model fit estimates will be introduced. The first two of these statistics will bin respondents into groups, with noticeable differences in the process of creating such bins, as well as in the approach of estimating the bin-level predicted responses P_{hi}.

Yen's Q1 statistic.

This fit statistic (Yen, 1981) is based on another chi-square fit statistic introduced in Bock (1960, 1972). The *Q1* statistic accounts for the ability level theta θ of respondents across an item *i* and sorts them into 10 relatively equal sized groups based on their ability estimates. The N_{hi} in Equation 6 corresponds to the number of respondents per group *h*, and r_{hi} is the difference between the observed and predicted (i.e., expected) score proportions to those answering item *i* correctly. The respective degrees of freedom needed to compare the obtained observed chi-square value to that indicated by the expected distribution (i.e., expected value) are 10 – (# of model parameters). For example, a 2PL model with its two parameters of δ_i, and α_i will incur 8 degrees of freedom. The main difference between *Q1* and Bock's chi-square has to do with selecting the number of bins. While *Q1* specifies the number of bins to be exactly 10 based on the respondents' ability levels, Bock's statistic can accommodate *H* number of bins. Also, *Q1* utilizes the average bin-level predicted responses P_{hi} for estimating P_{hi} per bin, while Bock's chi-square uses the median P_{hi} for respondents per bin (Ames & Penfield, 2015). The null hypothesis specifies perfect model data fit.

Although this fit method may allow researchers to obtain zero residuals with good fitting models, it has nevertheless been criticized in many respects. First, since the binning process is dependent on ability estimates, it is possible that the presence of a biased ability to begin with will produce an invalid fit statistic (Yen,1981). Second, the notion of binning into equal sized

groups will vary depending on the sample size, which can yield high Type I error rates (Orlando & Thissen, 2000). As mentioned earlier, larger sample sizes are usually more useful for applying *Q1* or Bock's χ^2. Third, some models such as those realized by ideal point processes may not fare well with such fit indices (Nye et al., 2019; Roberts, et al., 2000). It is possible to obtain relatively small expected frequencies for particular response categories when dealing with ideal point models. This could happen given the propensity of respondents to strongly agree with items that are close in proximity to their locations on the latent trait. In other words, creating bins of equal sample sizes will not work for such models. Nye et al. (2019) also adds that correcting this problem by combining response categories to increase respondents per bin will not be useful when the expected frequencies are small to begin with. Given such limitations, other fit indices that do not require binning on model-dependent θ estimates are recommended.

S – X^2 statistic.

Orlando and Thissen (2000) proposed binning examines into groups based on observed test scores rather than model-dependent θ values. This can be achieved via tabulating expected responses from the selected model's respective predictions for each item across all of theta θ intervals. The procedure would allow the expected responses to be compared to the observed ones. Such a process will not require the reliance on an estimate of θ for the binning process, and would avoid the potential issues associated with model-dependent binning that were mentioned earlier for *Q1*. The only issue that may occur with this binning approach is the inability to maintain a fixed degrees of freedom when testing for model fit using the chi-square statistic, since it is possible to incur dependencies among tables of observed counts for items on a single test (Orlando & Thissen, 2000).

The main difference between the $S - X^2$ and the aforementioned chi-square fit indices in terms of estimation involves the expected/predicted frequencies. In other words, the P_{hi} from equation 6 is estimated differently, and would not involve calculating the average or median bin-level predicted responses. Rather, a process involving the prediction of joint likelihood distributions for each observed test score is utilized (Thissen, Pommerich, Billeaud, & Williams, 1995). Thissen and colleagues developed a recursive algorithm, which utilizes the joint likelihood for selected groups based on their observed scores per item. Prediction using a joint likelihood approach was actually first introduced in (Lord & Wingersky, 1984) for test equating purposes, and was later modified for other IRT applications such as the $S - X^2$ statistic. This method involves omitting one item at a time when estimating the likelihood, and then adding the item back to calculate the proportion of test takers with a specific observed score answering item *i* correctly. $S - X^2$ follows a chi-square distribution with degrees of freedom equal to $I - 1 - m$. Where I is the number of items on a test and m is the number of parameters entailed by the IRT model for a given item. The estimation of P_{hi} takes the following form:

$$P_{hi} = \frac{\int T_i S_{h-1}^{*i} \Phi(\theta) d\theta}{\int S_h \Phi(\theta) d\theta} \qquad (7)$$

As described by Orlando and Thissen (2000): "the S_h is the observed score posterior distribution for score group h, T_i is the response function for item i, S_{h-1}^{*i} is the observed score posterior distribution for score group h-1 without the last item, and the integrals are estimated using rectangular quadrature over equally spaced increments of θ from -4.5 to 4.5" (pp. 53-54). The null hypothesis specifies a perfect model data fit.

It has been argued that the $S - X^2$ statistic works well for estimating model fit with logistic IRT models (Orlando & Thissen, 2000). Namely, the statistic exhibits low Type I error rates in general for such models. Also, power analyses involving the $S - X^2$ statistic demonstrated

good results in terms of detecting misfit across different conditions through varying the number of items exhibiting misfit (Orlando & Thissen, 2003). However, there are concerns about whether the $S - X^2$ works well with non-homogeneous groups based on the latent trait estimate. After all, the binning process in $S - X^2$ does not create homogeneous respondent groups with respect to the latent trait when varying models with non-equal discrimination parameters. This in turn might affect the power of the item fit estimate given the process of assigning respondents to different groups, which might be problematic as mixing respondents with misfitting responses with other respondents will inevitably occur (Roberts, 2008). Observed test scores (OTS) are used for the grouping process in $S - X^2$, and if the latent trait is heterogeneous, then it might be problematic to estimate item fit for ideal point models such as the GGUM. The reason is that according to Roberts (2008), such models are usually defined by item characteristic curves that are symmetric, do not follow a monotonic trajectory, and have their maximum values at $\theta_j - \delta_j = 0$. For such reasons, Roberts (2008) argues that it is possible to get identical expected OTSs from examinees with completely different θ_j when calibrating a GGUM model.

To resolve the issue of detecting misfit when performing power analyses, Roberts (2008) introduced a corrected version of $S - X^2$ that does not include the score of the examined item *i* (i.e., c $S - X^2$). Roberts also introduced variants of $S - X^2$ that utilized observed subset scores (OSS), which are calculated from extreme item scores rather than OTS. Surprisingly, the standard $S - X^2$ still yielded relatively comparable results to that of the corrected version in terms of exhibiting reasonable Type I error and power rates (Roberts, 2008). Such results were inferred from a simulation study that varied sample size and test length. The simulation study also compared the standard $S - X^2$ item fit statistic to the other fit indices using OSS in terms of

detecting Type I error and power, with $S - X^2$ showing better ability to detect misfit than any of the OSS fit indices. For the complete analysis, interested readers can refer to Roberts's article.

Adjusted chi-square χ^2 for item singlets, doublets, and triplets.

First introduced by Drasgow, Levine, Tsien, Williams, and Mead (1995), this fit method does not require binning examinees into groups for estimating observed and expected response frequencies. Rather, it requires summing up such frequencies over the number of response options (Drasgow, Levine, Williams, McLaughlin, & Candell, 1989; Nye et al., 2019). Conditional option response functions (CORFs) are usually utilized, which yields probability estimates of choosing an incorrect option in examinees answering an item incorrectly given θ. When calculating the χ^2 statistic, there can be an *I* number of such statistics for *I* items calculated separately, which can be referred to as *item singles*. The general form of expressing the chi square fit statistic for item singles for dichotomous data follows an ordinary χ^2 expression:

$$\chi_i^2 = \sum_{k=0}^{1} \frac{[O_i(k) - E_i(k)]^2}{E_i(k)} \qquad (8)$$

Where $O_i(k)$ is the observed frequency of option *k*, and is estimated by counting the number of times in which respondents selected option *k* in the sample. $E_i(k)$ represents the expected number of times in which respondents choose option *k*, which is estimated from the respective option response function by:

$$E_i(k) = N \int P(u_i = k|\theta) f(\theta) d(\theta) \qquad (9)$$

The *f* in the above expression refers to the θ density, which follows a standard normal given the scaling of the option response function with respect to the distribution, u_i refers to the response score. Research has shown that the chi-square statistic for single items is generally insensitive to detecting misfit under various conditions (Stark, Chernyshenko, Drasgow, & Williams, 2006;

Van den Wollenberg, 1982). For example, Nye et al. (2019) found that a chi-square statistic for single items is a poor indicator of misfit under most conditions pertaining to different sample sizes and number of items. Given its limited ability to detect misfit, Drasgow et al. (1995) introduced an χ^2 statistic from the expected frequency of item pairs via endorsing response options *k* and *k'* concurrently, referred to as *item doubles*. This follows estimating the observed frequencies from a two-way contingency table, with the expected frequencies obtained by expanding equation 9 to:

$$E_i(k, k') = N \int P(u_i = k|\theta)P(u_j = k'|\theta)f(\theta)d(\theta) \qquad (10)$$

Extending to χ2 items triples and beyond can be achieved by expanding equation 10. For example, a multiway contingency table can be used for estimating the χ2 using triples of items (Tay, Ali, Drasgow, & Williams, 2011). There are $\binom{I}{2}$ χ2 possible statistics for item doubles and $\binom{I}{3}$ χ2 for item triples. The possible combinations of item doubles and triples increases dramatically by increasing the number of items. For example, a test with 30 items yields 435 combinations for item doubles and 4,060 combinations for triples. To overcome this issue, Drasgow and colleagues (1995) divided the *I* test items into *I*/3 sets of three items. These sets were then used to compute the respective χ2 statistics for individual items, item pairs for doubles, and the whole set at once for triples. The degrees of freedom for the χ2 statistics equal to the number of cells minus one. For example, an item with three response categories will have two degrees of freedom. For item doubles, the χ2 statistic degrees of freedom with each item having three response categories will be eight (i.e., 9 – 1). As mentioned by LaHuis, Clark, and O'Brien (2009), a minimum expected frequency of five is maintained when collapsing over cells, and adjustments to the degrees of freedom are made to reflect the collapsing process.

Also, to account for the dependency of $\chi 2$ on sample size, as well as ensuring that the adjusted $\chi 2$ statistics are comparable across different sample sizes (Tay et al., 2011), the estimation of $\chi 2$ for item singles, doubles, and triples are adjusted to a sample size of 3,000. The $\chi 2$ for such items will be estimated using the ratio of the chi-square to the respective degrees of freedom $\chi 2/df$. The sample size adjustment is expressed in the following equation:

$$\frac{\chi_i^2}{df} = 3{,}000\frac{\chi_i^2 - df}{N} + df \qquad (11)$$

where *df* is the respective degrees of freedom. Although all of the presented IRT fit indices so far are for each item, model-fit estimation has been conducted for the adjusted $\chi 2$ fit statistics for item singles, doubles, and triples. The basic premise involves taking the mean of the $\chi 2/df$ ratios and comparing it with the value of 3 based on empirical findings using large cognitive ability data (Chernyshenko, Stark, Chan, Drasgow, & Williams, 2001; Drasgow et al., 1995). Mean ratios that are less than 3 for items singles, doubles, and triples indicate good model fit (Chernyshenko et al., 2001).

Studies assessing model fit using the adjusted $\chi 2$ fit statistics have been conducted (Drasgow et al., 1995; Tay et al., 2011). Results favor the use of $\chi 2$ fit for item doubles and triples in detecting misfit and not item singles given the inability to detect misfit in many conditions. For example, Tay and colleagues (2011) found that for both dichotomous and polytomous data generated from different IRT models adjusted $\chi 2$ fit tests for item pairs and triplets were able to identify the correct model well. These fit indices were successful in identifying the correct model for tests with relatively large numbers of items (i.e., 30 items). Nye and colleagues (2019) also found that the adjusted $\chi 2$ for item doubles and triples were among the most accurate indicators of misfit, even when generating different dichotomous and polytomous IRT models and calibrating them via the GGUM. However, the adjusted $\chi 2$ for

single items did not perform as well in detecting misfit. Nevertheless, power did improve for the adjusted $\chi 2$ for single items once the number of items were greater than 20.

The next fit indices are estimated using a likelihood-ratio approach. The first of these approaches utilizes a similar binning process to that of *Q1*. The second approach is a relative-fit method that compares different IRT models for best fit.

The G^2 statistic.

The simplest form of this likelihood ratio (LR) fit statistic is applicable to dichotomous items and is given by:

$$LR_i = 2\sum_{h=1}^{H}[N_{hi1}\ln\left(\frac{N_{hi1}}{N_{hi}P_{hi1}}\right) + N_{hi0}\ln\left(\frac{N_{hi0}}{N_{hi}(1-P_{hi1})}\right)] \qquad (12)$$

where N_{hi1} and N_{hi0} correspond to the number people per bin *h* responding to item *i* correctly and incorrectly, respectively (Ames & Penfield, 2015). For G^2 as proposed by McKinley and Mills (1985), P_{hi1} represents the probability of responding correctly at the average value of the ability level for respondents in bin *h*. Similar to the *Q1* test of fit, examinees are binned according to their ability estimate $\hat{\theta}$. However, the number of selected bins are not constrained to 10 as in *Q1*, hence examinees can be sorted in *h* number of bins according to their ability levels. G^2 is also distributed as chi-square such as the aforementioned fit indices, with respective degrees of freedom equal to the selected number of bins *H*. The null hypothesis assumes a perfect model data fit.

As mentioned by Ames and Penfield (2015), G^2 also has similar problems to that of *Q1* in terms of relying on model-dependent θ estimates for creating the bins, to which they cite DeMars (2005) criticisms on such a matter. Also, Roberts (2008) simulation study that compared different item fit statistics for the GGUM included the G^2 statistic. The results showed that

statistic behaved erratically in terms of yielding a higher Type I error rate with larger sample sizes. This did not occur with other fit indices such as the $S - X^2$ and those indices conditioned on subtest test scores.

Testing relative model fit can also be assessed through the G^2 statistic for nested IRT models, in which related models can be compared with one another. For example, the GUM is nested within the GGUM if the discrimination parameters are constrained to unity as well as making the threshold parameters identical across all items. The comparison of the likelihood ratios between the models using the difference of G^2s takes the following form:

$$\Delta G^2 = -2\ln(L_R) - (-2\ln(L_F)) \qquad (13)$$

As defined by (De Ayala, 2009), L_R is the likelihood for the constrained model (e.g., GUM) while L_F is the likelihood of the full model (e.g., GGUM). The main issue with this relative fit approach is that it is restricted to comparing models from the same family. Also, it doesn't penalize models with unnecessary parameters. Therefore, there is a problem with model over-parameterization. The next presented relative model fit statistics are supposed to handle the aforementioned issues.

AIC and BIC.

Both Akaike's information criterion (AIC) (Akaike, 1973) and Bayesian information criterion (BIC) (Schwarz, 1978) are relative model fit indices, which are determined by the number of parameters in the tested model. AIC is calculated as:

$$AIC = -2logL + 2v \qquad (14)$$

where log L is the log-likelihood and v refers to the number of parameters in the selected model. The BIC is calculated as:

$$BIC = -2logL + v\ log\ n \qquad (15)$$

where *n* corresponds to the sample size. The *log* n in BIC incurs greater likelihood values when compared to AIC, hence being the more stringent fit index when compared to AIC (i.e., yields larger values). As mentioned by Nye et al. (2019), these relative fit statistics have shown promising results for correctly identifying fit for dichotomous IRT models (Kang, Cohen, & Sung, 2009). However, there has been less research done on the effectiveness of such relative fit statistics in identifying the correct IRT model when calibrating ideal point models such as the GGUM. Nye et al. (2019) is among the few studies that utilized both AIC and BIC in detecting fit and misfit on generated data from dominance models that were calibrated by the GGUM. Results from the study showed that such fit statistics are able to detect both Type I error and power 100 percent of the time across replications, while not being influenced by either the number of items or sample size.

The final fit statistic is often used in the SEM literature. However, it has been utilized to a lesser degree in the IRT literature albeit its promising capabilities in identifying the correct model under various conditions pertaining to different sample sizes and number of items on a test.

Standardized root mean square residual (SRMSR).

This fit statistic is appropriate for both large nominal and ordinal data, and is usually utilized in factor analysis. Also, it addresses some of the problems inherent with chi-square fit statistics such as sensitivity to sample size. Maydeu-Olivares and Joe (2014) demonstrated the SRMSR's applicability for estimating approximate fit, which can be used for evaluating model fit for IRT models. The SRMSR is simply the square root of the average squared residual correlations between a set of item pairs *i* and *j*. The residual correlation is the sample or

population correlation minus the expected correlation. The population SRMSR for item pair is defined as follows:

$$SRMSR = \sqrt{\sum_{i<j} \frac{\left(\hat{\rho}_{ij}^{T} - \hat{\rho}_{ij}^{0}\right)^{2}}{n(n-1)/2}} \qquad (16)$$

where $\hat{\rho}_{ij}^{T}$ refers to the population correlation, and $\hat{\rho}_{ij}^{0}$ is the expected correlation. This statistic and its extension to ordinal data are more useful over other limited information goodness of fit statistics such as the M_2 and M_{ORD}, since the former two can be computed without any degrees of freedom. M_2 is a limited information fit statistic that can be used for sparse dichotomous data with large number of items (i.e., many empty cells in a frequency table). According to Xu, Paek, and Xia (2017),

> M_2 follows asymptotically a central chi-square distribution under the null hypothesis with asymptotically normal consistent estimators. Its degrees of freedom is equal to the number of used multivariate moments (or the number of the margins up to 2) minus the number of model parameters (p. 633).

It utilizes the means and cross-products (i.e., bivariate information) to estimate fit. M_{ORD} is an extension of the M_2 statistic that uses a different asymptotic covariance matrix and matrix of derivatives when estimating parameters, and accommodates large number of items with multiple response categories per item (i.e., ordinal data). Interested readers are referred to Maydeu-Olivares and Joe (2014) for computing M_2 and M_{ORD}.

The root mean square error of approximation (RMSEA) is a fit statistic that compares the difference between a hypothesized model and a perfect model (Browne & Cudeck, 1992). Though often utilized as a goodness of fit approximation in multivariate contexts and can be applied to IRT models, its sampling distribution is only approximated with asymptotic methods

in small models (Maydeu-Olivares & Joe, 2014). An alternative fit statistic $RMSEA_2$ resolves this issue by using only bivariate information as the M_2 statistic instead of the full information needed to calculate RMSEA.

SRMSR is shown to be linearly related to $RMSEA_2$, with an average R^2 of 97% (Maydeu-Olivares & Joe, 2014). Such a relationship is useful since $RMSEA_2$ can be estimated using the M_2 statistic. An SRMSR ≤ 0.05 points toward a tested IRT model that approximately represents the data of interest. Based on both simulated and empirical data (Maydeu-Olivares & Joe, 2014; Nye et al., 2019), the SRMSR was selected to test for IRT model fit with favorable results in terms identifying the correct model and detecting misfit (Nye et al., 2019).

Although less utilized when compared to the other model fit indices when examining IRT calibrated data, the SRMSR may be among the most useful model fit statistics around. It can accommodate different IRT models tested under different conditions pertaining to different sample sizes and number of items (Nye et al., 2019). Also, few IRT software packages are equipped with either approximate or limited fit information indices such as the SRMSR or M_2 for IRT models such as mirt (Chalmers, 2012) and flexMIRT (Cai, 2017), which make them less likely to be utilized for performing model fit analyses.

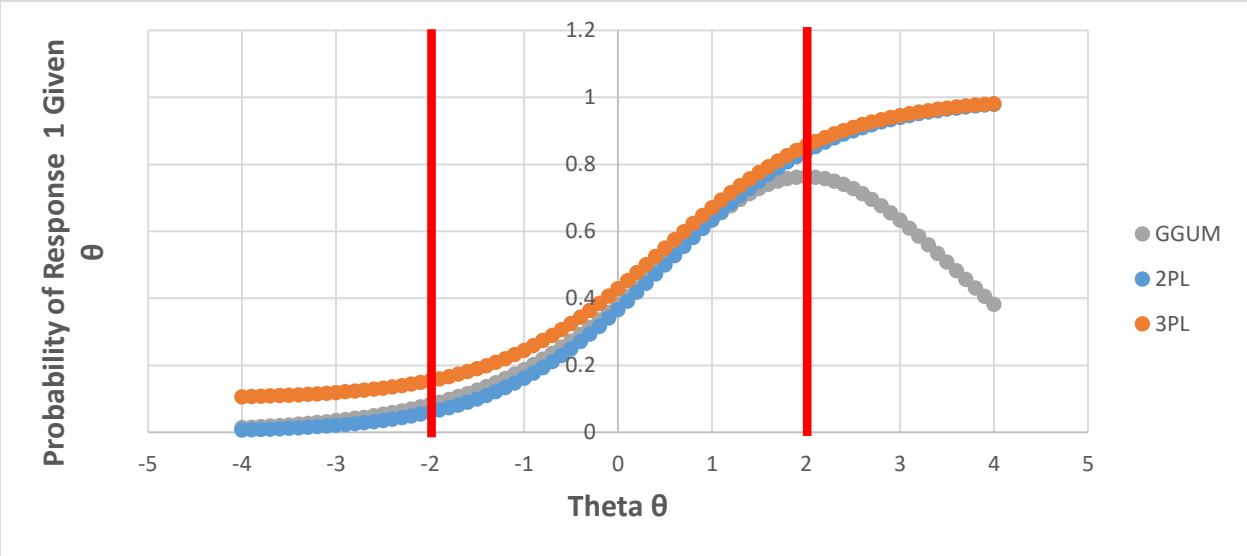
In this respect, it would be conducive to IRT research if the aforementioned item and model fit indices are compared in terms of correct model identification and the detection of misfit for unfolded models such as the GGUM. After all, the advantageous properties inherent within IRT models will only be applicable and valid if the pre-selected model fits the data.

Overlap of Item Response Functions between IRT Models

Although research has shown in theory and practice that ideal point IRT models such as the GGUM are better suited for attitude and survey data given the possibility of obtaining higher

observed δs (Coombs, 1964; Roberts & Laughlin, 1996; Thurstone, 1928), IRT data obtained from dominance models such as the 2PL and 3PL models may fit the GGUM model well (Tay et al., 2011). Actually, earlier research advocated fitting the 2PL model to self-report data such as those assessing attitudes and personality (i.e., noncognitive items) since model fit estimation did not indicate misfit (Fraley, Waller, & Brennan, 2000; Reise & Waller, 1990; Tay et al., 2011). In short, an ideal point model such as the GGUM can fit generated data from the 2PL and 3PL models well for high δ_i values without indicating misfit. Figure 6 illustrates how the IRFs of the GGUM, 2PL, and 3PL are pretty much overlapping across the θ continuum, albeit the divergence of the IRFs paths between the GGUM and the dominance models as θ becomes greater than 2. Such a divergence represents a minority of respondents and would not affect model-fit (Tay et al., 2011).

Figure 6. *GGUM/2PL/3PL Item Response Function*



Note. The GGUM IRF in gray is computed with $\alpha_i = 0.9$, $\delta_i = 2$, and $\tau_i = -1.3$. The 2PL IRF in blue is computed with $\alpha_i = 1.1$ and $\delta_i = 0.5$. The 3PL IRF in orange is computed with $\alpha_i = 1.1$, $\delta_i = 0.5$, and $\chi_i = 0.1$.

Given such overlap between IRT models, subsequent analyses of model fit would entail specifying different ranges of generated δs between the GGUM and the dominance models across replications, which might to a certain extent allow the GGUM model to differentiate between the generated data from the different IRT models in terms of fit.

The Current Study

The purpose of this study is to compare the aforementioned IRT model fit indices by fitting the Generalized Graded Unfolding Model (GGUM) to different generated IRT data models, and test their Type I error and power rates respectively. The generated IRT data from the models will include both dichotomous and polytomous variants, and they will be unidimensional. Based on the results, suggestions will be made as to which fit statistics are the most useful for the IRT unfolding model.

Research questions to be addressed:

1) How comparable are the different model fit indices in terms of identifying the correct model (i.e., Type I error) when the generated data are actually from the GGUM?
2) How comparable are the different model fit indices in terms of detecting misfit (i.e., power) when the generated data actually is from another IRT model calibrated by the GGUM?
3) How would the different model fit indices fare when varying numbers pertaining to sample size, items, and response categories (i.e., dichotomous vs. polytomous) on detection rates of fit and misfit?
4) How comparable are relative fit indices (i.e., AIC and BIC) when the GGUM model is fit to generated GGUM data utilizing different δ ranges and compared to dichotomous dominance models.

CHAPTER 3

METHODS

Variables

A simulation study was performed utilizing four variables: sample size (500, 1000, 2000, 3000), number of items (10, 20, 40), type of IRT model used to generate data, and type of data (dichotomous or polytomous). For dichotomous IRT models, data were generated from the 2PL model, 3PL model, and GGUM comprised of two response categories (i.e., coded 0 or 1). For polytomous IRT models, data were generated from the graded response model (GRM) and the GGUM comprised of four response categories. Within each condition, data were generated and an IRT model was fitted to the data for each of 100 replications. All simulations pertaining to data generation, fitting the model to the generated data, and estimation of model fit indices were conducted in R (Chalmers, 2012; Nydick, 2014; Tendeiro & Castro-Alvarez, 2020).

Data Generation

Item and person parameters.

The distributions from which the generated item parameters were obtained from previous IRT simulation studies (Nye et al., 2019; Roberts, Donoghue & Laughlin, 2002; Tay et al., 2011). For dichotomous dominance IRT models (i.e., 2PL and 3PL), the discrimination parameters were generated from a log-normal [0, 0.5] distribution and dividing by 1.702. The item locations were generated from a random uniform distribution [-2, 2]. For the 3PL model, the guessing parameter χ was obtained from a random uniform distribution [0, 0.3]. For the GGUM model, the discrimination parameters were generated from a uniform distribution [0.5, 2] distribution.. The threshold parameter (τ) was sampled from a uniform random distribution [-1.4. -0.4]. For polytomous IRT models, the discrimination parameters were generated from a log-normal [0,

0.5] distribution and dividing by 1.702, while the discrimination parameters for the GGUM were generated from a random uniform distribution [0.5, 2]. The threshold parameters for the GRM (δ_1, δ_2, δ_3) were generated from random uniform distributions [-2, -0.5], [-0.5, 0.5], [0.5, 2], respectively (Kieftenbeld & Natesan, 2012; Nye et al., 2019). For the GGUM as indicated by Roberts et al. (2002), the threshold parameters (τ_{ik}) were generated independently for each item. For a selected item, the highest threshold parameter (τ_{iB} or τ_3) was drawn from a uniform distribution [-1.4. -0.4]. Successive τ parameters for each item (i.e., τ_2 or τ_1) were sampled using the following recursive formula:

$$\tau_{ik-1} = \tau_{ik} - 0.25 + e_{ik-1}, \text{ for } k = 2, 3, \qquad (17)$$

where e_{ik-1} represents a random error term sampled from a normal distribution $N(0, 0.04)$. The item parameters will vary across replications to test as many different ranges of items and observe whether the detection of fit/misfit will hold across different generated items. Although varying the item parameters might produce less consistent patterns when it comes to detecting misfit, such as when fitting the GGUM to the incorrect generated data (i.e., large sample size and number of items might not always yield the highest rates of misfit), it would nevertheless be useful for generalizability purposes in terms of testing the accuracy of the model fit indices in detecting fit/misfit under different parameter ranges. For all IRT models, person parameters (θ) are generated from a random normal distribution $N(0, 1)$, which also varies per replication.

Item location parameters for the GGUM.

The item location δs parameters for the GGUM were generated using 3 different ranges from random uniform distributions. The first parameters was generated from a random uniform distribution [-2, 2] as specified by Roberts et al. (2002). The second set of item location parameters were also generated from a random uniform distribution [-2, 2] as specified by

Roberts et al. (2002). Nevertheless, these item locations do not include values ranging from -1 to 1 in order to prevent possible overlap between small values of δs between competing models, which supposedly reduces the detection of misfit. The third set of item locations were generated from a random uniform distribution [-3, 3], which has a greater range of generated δs to that of the other competing dominance models. This is done to reduce the possibility of incurring large δs for the dominance models, which might lead to higher proportions of false negatives. The different sets were compared in terms of relative fit (i.e., comparative fit) to one another via AIC and BIC fit indices, and the set with the smallest fit values were selected for subsequent fit analyses. Based on the expected overlap between the IRF's of the GGUM and the dichotomous dominance models, it is expected that Roberts et al. (2002) recommended range of a uniform distribution [-2, 2] will fare worse than the other two sets in terms of relative fit when utilizing the marginal maximum likelihood algorithm for parameter estimation.

Response data generation.

Responses from simulated data for dichotomous items were generated through comparing the item response probabilities from each model to a random uniform distribution [0, 1]. As indicated by Nye et al. (2019), a score of 1 was assigned to a response data for a dichotomous item if the response probability is greater than the generated number from the uniform distribution, while a score of 0 was assigned if the response probability is less than the generated number from the uniform distribution. For dominance IRT models with dichotomous items such as the 2PL and 3PL, response data were generated by the "catIRT" package in R (Nydick, 2014). . For the GGUM model with dichotomous items, response data were generated by the "GGUM" package in R by setting the category threshold indicator *C* to 1 (Tendeiro & Castro-Alvarez, 2020).

For polytomous items with four response categories, a score of 3 was assigned to a response data if the sum of the response probabilities for categories 0, 1, and 2 was less than the randomly sampled uniform number. If the random generated number was less than the sum of the probabilities for categories 0, 1, and 2 but greater than the sum of the probabilities for categories 0, and 1, then a score of 2 was assigned, and so forth. For dominance IRT models such as the GRM, response data were also generated by the "catIRT" package in R (Nydick, 2014). However, response data generated for the GRM by the "catIRT' package assigned a value of 1 instead of 0 to the lowest response category. Hence, an adjustment was made in which a value of 1 is deducted from each of the response category values ranging from 1 to 4, which in turn yielded response values ranging from 0 to 3. For the GGUM model with four response category items, response data were also generated by the "GGUM" package in R, but via setting the category threshold indicator *C* to 3 (Tendeiro & Castro-Alvarez, 2020).

Model Parameters Calibration

Once the generated response data for the selected IRT models were created, the GGUM model was fit to the data accordingly by the marginal maximum likelihood algorithm as specified by Roberts et al. (2000), which is based on an expectation-maximization (EM) approach. R packages "GGUM" and "mirt" were utilized to calibrate the item parameters (Chalmers, 2012; Tendeiro & Castro-Alvarez, 2020). The reason for using two different packages to perform the calibration process has to do with subsequent model fit estimation approaches that are available in one of the packages but not the other. For instance, the GGUM package is only able to estimate the *Adjusted Chi-square* χ^2 and relative fit statistics (i.e., *AIC* and *BIC*), while the mirt package is only able to estimate fit statistics such as *Q1*, $S-X^2$, G^2, *SRMSR*, and the relative fit statistics.

In the GGUM package, the selected number of nodes used for numerical integration is 60. The selected maximum number of EM outer and inner iterations are 200 and 30, respectively. The convergence tolerance is set to 0.001. The selected number of nodes, iterations, and convergence tolerance values follows those utilized in Tay et al. (2011).

In the mirt package, the GGUM model is fit to generated data by setting the type of density form for the latent parameters to 'empircalhist', which utilizes an empirical histogram as described by Bock and Aitkin (1981). This form is only applicable for unidimensional models estimated using the EM algorithm (Chalmers, 2012). The numerical optimizer is set to 'nlminb'. The Newton-Raphson optimizer is desired since it also follows Bock and Aitkin (1981), but is not utilized since it is less stable in yielding converged solutions in mirt. Sixty quadrature points are used for item estimation and the convergence tolerance is also set to 0.001 as in the GGUM package. The number of N cycles is set to 10000.

Model Fit Indices

The calibrated GGUM was fit to generated IRT data and is evaluated using the following model fit indices: *Q1*, $S - X^2$, G^2, *Adjusted Chi-square* χ^2 (i.e., adjusted to a sample size of 3000), *SRMSR*, *AIC*, and *BIC*. As suggested by Nye et al. (2019), fit indices were calculated using estimated item and person parameters in each replication. To compare item-level fit statistics (e.g., *Q1*, $S - X^2$, G^2, and *Adjusted Chi-square* χ^2 for item singles) to scale-level fit statistics (e.g., *Adjusted Chi-square* $\chi 2$ for item doubles and triples, SRMSR, *AIC*, and *BIC*), model-data fit is calculated via examining the proportion of items exhibiting misfit per replication, and subsequently averaging the proportions across the replications.

Critical values for evaluating model fit within each index are as follows. The *Q1* statistic for each item on a single replication is compared to a chi-square distribution with 10 – *m* degrees

of freedom; *m* is the number of estimated parameters for an item. The $S - X^2$ statistic for each item on a single replication is compared to chi-square distribution with $I - 1 - m$ degrees of freedom, where I is the number of items on a test. The G^2 statistic for each item on a single replication is compared to chi-square distribution with *h* degrees of freedom, where *h* equals to the selected number of bins. The selected bins correspond to grouped individuals on a test or measure based on specified ranges corresponding to ability θ (Ames & Penfield, 2015).

The average *Adjusted Chi-square* χ^2 for item singles, doubles, and triples are divided by their degrees of freedom and ratios greater than 3 indicate misfit (Chernyshenko et al., 2001). For the *SRMSR*, values greater than 0.05 indicate misfit as noted by Maydeu-Olivares and Joe (2014). For the relative model-fit statistics *AIC* and *BIC*, indices' values are compared against one another, with the model yielding the lowest information criterion considered the best fitting model. Across replications (i.e., 100 replications), the number of times that the fitted model with the lowest values of AIC and BIC is reported.

For all of the aforementioned model fit analyses except for AIC and BIC fit statistics, respective proportions of Type I error rates and power across replications for each of the fit indices are calculated and reported as an indicator of model fit/misfit. For Type I error, the proportion of times in which the GGUM model falsely rejects the null hypothesis of model fit when calibrated to GGUM generated data across the 100 replications is reported (i.e., the number of false rejections divided by 100). For estimating power, the proportion of times in which the GGUM model correctly rejects the null hypothesis of model fit when calibrated to a dominance IRT model generated data across the 100 replications is reported (i.e., the number of correct rejections divided by 100). For example, for model fit indices such as *Adjusted Chi-square* χ^2 and *SRMSR*, if the GGUM model is fit to different GGUM generated data 100 times, and it was

found that the null hypothesis of model fit is only rejected four times across the 100 replications, then the type I error is 0.04 or 4 percent. Similarly, if the GGUM model is fit to the 2PL generated data 100 times, and it was found that the null hypothesis of model fit is rejected 89 times across the 100 replications, then power is 0.89 or 89 percent. For item fit statistics such as *Q1*, $S - X^2$, and G^2, each item is examined and counted as 1 if the null hypothesis of item fit is rejected. The number of rejected null hypotheses on a single replication is counted and averaged across the number of items. For example, if 4 out of 20 items had their null hypotheses of item fit rejected, then 0.2 or 20 percent of items from the set of 20 items are presumed to exhibit misfit. This process was followed across the 100 replications with the final type I error rate or power obtained by averaging the proportions of items' misfit across replications. For AIC and BIC, models are compared to each other. Across the 100 replications, the number of times in which each of the competing models has the lowest AIC and BIC is reported. For example across the 100 replications, if the GGUM model had the lowest BIC value when fitting to a GGUM data 89 percent of the time, while the 2PL data incurred the lowest BIC 6 percent of the time, and the 3PL data incurred the lowest BIC 5 percent of the time, then it can be said that the BIC is able to correctly detect model fit 89 for the percent of the time for GGUM generated data.

Technical Considerations and Seed Selection

For dichotomous generated data, the default selected seed is set to 2875 for both packages that are used to generate data (i.e., GGUM and catIRT). However, some of the calibrations did not converge in conditions with small number of items, and a different seed had to be assigned to achieve convergence (Table 1). For polytomous generated data, the default selected seed is also 2875 for both packages that are used to generate data (i.e., GGUM and catIRT). However, many of the simulated data led to nonconvergence and had to be assigned a

different seed to achieve convergence (Table 2). Also, fitting polytomous data to the GGUM model is time consuming and therefore utilized UNL's Holland Computing Center supercomputer for the computation process. The RStudio interactive sessions on Holland's supercomputer only allows a maximum of 8 hours per sessions. For such reasons, the total number of replications are segmented into mini sessions for conditions with large number of items, hence prompting the assignment of different seeds for the segmented runs. For example, generated data with 40 items via the catIRT package required segmenting the 100 replications into 4 sessions, with each session consisting of 25 calibrations. Table 2 displays the number of segments per condition with its respective seed.

Table 1. Seed Values for catIRT and GGUM Packages for Dichotomous Generated Data

I	*N*	**Package**	Seed (catIrt)	Seed (GGUM)
10	500	GGUM	2875	2875
		2PL	2875	7777
		3PL	2875	2875
	1000	GGUM	7777	2875
		2PL	2875	2875
		3PL	2875	2875
	2000	GGUM	2875	2875
		2PL	2875	2875
		3PL	2875	2875
	3000	GGUM	2875	2875
		2PL	2875	2875
		3PL	2875	2875
I		**Package**	Seed (catIrt)	Seed (GGUM)
20	500	GGUM	2875	2875
		2PL	2875	2875
		3PL	2875	2875
	1000	GGUM	2875	2875
		2PL	2875	2875
		3PL	2875	2875
	2000	GGUM	2875	2875
		2PL	2875	2875
		3PL	2875	2875
	3000	GGUM	2875	2875
		2PL	2875	2875
		3PL	2875	2875

I		**Package**	Seed (catIrt)	Seed (GGUM)
40	500	GGUM	2875	2875
		2PL	2875	2875
		3PL	2875	2875
	1000	GGUM	2875	2875
		2PL	2875	2875
		3PL	2875	2875
	2000	GGUM	2875	2875
		2PL	2875	2875
		3PL	2875	2875
	3000	GGUM	2875	2875
		2PL	2875	2875
		3PL	2875	2875

I = number of items; *N* = sample size.

Table 2**.** Seed Values for catIRT and GGUM Packages for Polytomous Generated Data

I	*N*	**Package**	Seed 1 (catIrt)	Seed 2 (catIrt)	Seed 3 (catIrt)	Seed 4 (catIrt)	Seed 1 (GGUM)	Seed 2 (GGUM)	Seed 3 (GGUM)	Seed 4 (GGUM)	Seed 5 (GGUM)	Seed 6 (GGUM)
10	500	GGUM	7777	-	-	-	7777	-	-	-	-	-
		GRM	7777	-	-	-	7777	-	-	-	-	-
	1000	GGUM	2875	-	-	-	2875	-	-	-	-	-
		GRM	2875	-	-	-	2875	-	-	-	-	-
	2000	GGUM	2875	-	-	-	7777	-	-	-	-	-
		GRM	7777	-	-	-	7777	-	-	-	-	-
	3000	GGUM	2875	-	-	-	2875	-	-	-	-	-
		GRM	7777	-	-	-	2875	7777	-	-	-	-
I	*N*	**Package**	Seed 1 (catIrt)	Seed 2 (catIrt)	Seed 3 (catIrt)	Seed 4 (catIrt)	Seed 1 (GGUM)	Seed 2 (GGUM)	Seed 3 (GGUM)	Seed 4 (GGUM)	Seed 5 (GGUM)	Seed 6 (GGUM)
20	500	GGUM	2875	7777	-	-	2875	-	-	-	-	-
		GRM	2875	1111	-	-	2875	7777	1111	-	-	
	1000	GGUM	2875	7777	-	-	2875	-	-	-	-	-
		GRM	1234	7777	-	-	2875	7777	1111	-	-	
	2000	GGUM	9997	7777	-	-	2875	-	-	-	-	-
		GRM	2875	7777	-	-	2875	7777	1111	-	-	
	3000	GGUM	9997	7777	-	-	2875	-	-	-	-	-
		GRM	2875	7777	-	-	2875	7777	1111	-	-	-

I	*N*	Package	Seed 1 (catIrt)	Seed 2 (catIrt)	Seed 3 (catIrt)	Seed 4 (catIrt)	Seed 1 (GGUM)	Seed 2 (GGUM)	Seed 3 (GGUM)	Seed 4 (GGUM)	Seed 5 (GGUM)	Seed 6 (GGUM)
40	500	GGUM	2875	7777	7887	1111	2875	-	-	-	-	-
		GRM	2875	7777	7887	1111	2875	7777	1997	1111	1234	-
	1000	GGUM	2875	7777	7887	1111	2875	-	-	-	-	-
		GRM	2875	7777	7887	1111	2875	7777	1997	1111	1234	-
	2000	GGUM	2875	7777	3232	1111	2875	7777	-	-	-	-
		GRM	2875	7777	3232	1111	2875	7777	1997	1111	1234	-
	3000	GGUM	2875	7777	2001	1111	2875	7777	-	-	-	-
		GRM	2875	7777	2001	1111	2875	7777	1997	1111	1234	1212

I = number of items; *N* = sample size.

CHAPTER 4

RESULTS

Relative Fit between Dichotomous GGUM Generated Data

Table 3 presents the results of relative fit between two GGUM simulated data sets, with the generated set of δs ranging from a uniform distribution [-2, 2] while excluding the interval [-1, 1] being a better fit than the generated set of δs ranging from a uniform distribution [-2, 2] as specified by Roberts et al. (2002). This table shows that the generated set of δs that do not include the interval [-1, 1] are better calibrated by the GGUM model about 95 percent of the time, and almost 100 percent of the time when the number of items are 20 and above. Note that in the condition specifying a sample size of 3000 and 20 items as well as a sample size of 500 and 40 items, the generated set of δs that do not include the interval [-1, 1] have the lowest AIC and BIC values in 99 replications out of a 100. Conditions with smaller sample sizes with 20 items do show a slightly better fit of 100 percent. This may be due to varying the model parameters per replication as mentioned in the previous section.

Table 3. Relative Fit Indices Rates of GGUM Data Models AIC and BIC with Generated δs from a Uniform Distribution [-2, 2] That Do Not Include [-1, 1] Against Generated δs from a Uniform Distribution [-2, 2]

I	*N*	**Gen. Model**	*AIC*	*BIC*
10	500	GGUM (No -1 to 1)	0.95	0.95
		GGUM (-2 to 2)	0.05	0.05
	1000	GGUM (No -1 to 1)	0.94	0.94
		GGUM (-2 to 2)	0.06	0.06
	2000	GGUM (No -1 to 1)	0.96	0.96
		GGUM (-2 to 2)	0.04	0.04
	3000	GGUM (No -1 to 1)	0.96	0.96
		GGUM (-2 to 2)	0.04	0.04

		Gen. Model	*AIC*	*BIC*
20	500	GGUM (No -1 to 1)	1.00	1.00
		GGUM (-2 to 2)	0	0
	1000	GGUM (No -1 to 1)	1.00	1.00
		GGUM (-2 to 2)	0	0
	2000	GGUM (No -1 to 1)	1.00	1.00
		GGUM (-2 to 2)	0	0
	3000	GGUM (No -1 to 1)	0.99	0.99
		GGUM (-2 to 2)	0.01	0.01
		Gen. Model	*AIC*	*BIC*
40	500	GGUM (No -1 to 1)	0.99	0.99
		GGUM (-2 to 2)	0.01	0.01
	1000	GGUM (No -1 to 1)	1.00	1.00
		GGUM (-2 to 2)	0	0
	2000	GGUM (No -1 to 1)	1.00	1.00
		GGUM (-2 to 2)	0	0
	3000	GGUM (No -1 to 1)	1.00	1.00
		GGUM (-2 to 2)	0	0

I = number of items; *N* = sample size; Gen. Model = data generation models; GGUM (No -1 to 1) = GGUM generated data with δs ranging from a uniform distribution [-2, 2] that do not Include [-1, 1]; GGUM (-2 to 2) = GGUM generated data with δs ranging from a uniform distribution [-2, 2].

Table 4 also presents the results of relative fit between two GGUM simulated data sets, with the generated set of δs ranging from a uniform distribution [-3, 3] having even a better fit than the generated set of δs ranging from a uniform distribution [-2, 2] when compared to Table 3. In this comparison, the generated set of δs from a uniform distribution [-3, 3] are better calibrated by the GGUM model in about 100 percent of the replications across all conditions. This may be due to the fact that the GGUM data are better calibrated when they include more items with extreme item responses such as those common in noncognitive measures of attitudes and personality surveys (Coombs, 1964; Thurstone, 1928). Hence, extending the δ range facilitate capturing the more extreme items.

Table 4. Relative Fit Indices Rates of GGUM Data Models AIC and BIC with Generated δs from a Uniform Distribution [-3, 3] Against Generated δs from a Uniform Distribution [-2, 2]

I	*N*	**Gen. Model**	*AIC*	*BIC*
10	500	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0
	1000	GGUM (-3 to 3)	0.99	0.99
		GGUM (-2 to 2)	0.01	0.01
	2000	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0
	3000	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0
		Gen. Model	*AIC*	*BIC*
20	500	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0
	1000	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0
	2000	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0
	3000	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0
		Gen. Model	*AIC*	*BIC*
40	500	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0
	1000	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0
	2000	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0
	3000	GGUM (-3 to 3)	1.00	1.00
		GGUM (-2 to 2)	0	0

I = number of items; *N* = sample size; Gen. Model = data generation models; GGUM (-3 to 3) = GGUM generated data with δs ranging from a uniform distribution [-3, 3]; GGUM (-2 to 2) = GGUM generated data with δs ranging from a uniform distribution [-2, 2].

Table 5 compares the relative fit between two GGUM simulated data sets, with the generated set of δs ranging from a uniform distribution [-3, 3] being a better fit than the generated set of δs ranging from a uniform distribution [-2. 2] that do not include the interval [-1, 1]. This table shows that the generated set of δs ranging from a uniform distribution [-3, 3] are better calibrated by the GGUM model about 80 percent of the time, and above 95 percent of the time when the number of items is 40. Based on these comparisons, dichotomous GGUM generated data with δs ranging from a uniform distribution [-3, 3] is selected for subsequent comparisons of absolute fit against dichotomous dominance models.

Table 4. Relative Fit Indices Rates of GGUM Data Models AIC and BIC with Generated δs from a Unif. orm Distribution [-3, 3] Against Generated δs from a Uniform Distribution [-2, 2] that do not Include [-1, 1*]*

I	*N*	**Gen. Model**	*AIC*	*BIC*
10	500	GGUM (-3 to 3)	0.80	0.80
		GGUM (No -1 to 1)	0.20	0.20
	1000	GGUM (-3 to 3)	0.82	0.82
		GGUM (No -1 to 1)	0.18	0.18
	2000	GGUM (-3 to 3)	0.80	0.80
		GGUM (No -1 to 1)	0.20	0.20
	3000	GGUM (-3 to 3)	0.83	0.83
		GGUM (No -1 to 1)	0.17	0.17
		Gen. Model	*AIC*	*BIC*
20	500	GGUM (-3 to 3)	0.80	0.80
		GGUM (No -1 to 1)	0.20	0.20
	1000	GGUM (-3 to 3)	0.88	0.88
		GGUM (No -1 to 1)	0.12	0.12
	2000	GGUM (-3 to 3)	0.90	0.90
		GGUM (No -1 to 1)	0.10	0.10

	3000	GGUM (-3 to 3)	0.91	0.91
		GGUM (No -1 to 1)	0.09	0.09
		Gen. Model	*AIC*	*BIC*
40	500	GGUM (-3 to 3)	0.96	0.04
		GGUM (No -1 to 1)	0.96	0.04
	1000	GGUM (-3 to 3)	0.96	0.04
		GGUM (No -1 to 1)	0.96	0.04
	2000	GGUM (-3 to 3)	0.93	0.07
		GGUM (No -1 to 1)	0.93	0.07
	3000	GGUM (-3 to 3)	0.96	0.04
		GGUM (No -1 to 1)	0.96	0.04

I = number of items; *N* = sample size; Gen. Model = data generation models; GGUM (-3 to 3) = GGUM generated data with δs ranging from a uniform distribution [-3, 3]; GGUM (No -1 to 1) = GGUM generated data with δs ranging from a uniform distribution [-2, 2] that do not Include [-1, 1].

Absolute Fit Indices for Dichotomous Data

GGUM package fit indices.

Table 6 presents the results of the average *Adjusted Chi-square* χ^2 for item singles, doubles, and triples across the 100 replications from the GGUM package. When the GGUM model is correctly fit to GGUM generated data, *Adjusted Chi-square* χ^2 fit statistics exhibit low type I error rates for all item variants, with an almost zero rejection rate of model fit for all cases when utilizing 20 items or above, irrespective of sample size. Albeit useful, these *Adjusted Chi-square* χ^2 fit statistics exhibit low rejection rates when the GGUM model is fit to incorrect data models such as the 2PL and 3PL models. In other words, power to detect misfit is poorly realized by the *Adjusted Chi-square* χ^2 for item singles, doubles, and triples. This poses a problem when using real data since low rejection rates of model fit might represent a case of low power to detect misfit when the data actually comes from other IRT models.

Table 5. Type I Error Rates and Power of Absolute Model Fit Indices When the GGUM Model is Fit to Dichotomous IRT Data Models

I	*N*	**Gen. Model**	*Q1*	*S-X²*	*G²*	X2 Singles	X2 Doubles	X2 Triples	SRMSR
10	500	GGUM	0.95 (27)	0.75 (2)	0.94 (43)	0.01	0.01	0.01	0.90
		2PL	0.97 (35)	0.57	0.98 (55)	0.02	0.02	0.02	0.56
		3PL	0.93 (33)	0.55	0.93 (56)	0	0	0	0.50
	1000	GGUM	1.00 (13)	0.85	1.00 (23)	0.01	0.03	0.02	0.89
		2PL	1.00 (24)	0.56	1.00 (50)	0	0	0	0.34
		3PL	0.98 (18)	0.47	0.99 (39)	0	0	0	0.23
	2000	GGUM	1.00 (2)	0.91	1.00 (12)	0.01	0.02	0.02	0.87
		2PL	1.00 (8)	0.62	1.00 (39)	0	0	0	0.09
		3PL	1.00 (9)	0.54	1.00 (37)	0	0	0	0.07
	3000	GGUM	1.00 (4)	0.93	1.00 (8)	0.01	0.01	0.01	0.82
		2PL	1.00 (7)	0.65	1.00 (28)	0	0	0	0.04
		3PL	1.00 (5)	0.56	1.00 (26)	0	0	0	0
		Gen. Model	*Q1*	*S-X²*	*G²*	X2 Singles	X2 Doubles	X2 Triples	SRMSR
20	500	GGUM	0.76 (27)	0.76 (5)	0.77 (29)	0	0	0.01	0.98
		2PL	0.89 (39)	0.44	0.90 (44)	0	0	0	0.69
		3PL	0.84 (39)	0.43	0.82 (43)	0	0	0	0.57
	1000	GGUM	0.94 (11)	0.84 (4)	0.94 (19)	0	0	0	0.97
		2PL	0.99 (23)	0.44	0.98 (33)	0	0	0	0.24
		3PL	0.97 (30)	0.48	0.97 (32)	0	0	0	0.24
	2000	GGUM	0.99 (1)	0.89 (1)	0.99 (1)	0	0	0	0.98
		2PL	1.00 (8)	0.53	1.00 (15)	0	0	0	0.06
		3PL	1.00 (24)	0.50	1.00 (27)	0	0	0	0.08
	3000	GGUM	1.00	0.92	1.00	0	0	0	0.99
		2PL	1.00 (9)	0.57	1.00 (15)	0	0	0	0.03
		3PL	1.00 (17)	0.54	1.00 (19)	0	0	0	0.07
		Gen. Model	*Q1*	*S-X²*	*G²*	X2 Singles	X2 Doubles	X2 Triples	SRMSR
40	500	GGUM	0.44 (40)	0.69 (14)	0.46 (43)	0	0	0	1.00
		2PL	0.83 (24)	0.32	0.82(29)	0.04	0.04	0.04	0.53
		3PL	0.76 (37)	0.35	0.74 (38)	0	0	0	0.60
	1000	GGUM	0.67 (10)	0.77 (4)	0.70 (13)	0	0	0	1.00
		2PL	0.95(25)	0.34	0.95 (29)	0.02	0.02	0.02	0.14
		3PL	0.92 (25)	0.36	0.92 (25)	0	0	0	0.37
	2000	GGUM	0.89 (1)	0.85	0.90 (1)	0	0	0	1.00
		2PL	0.98 (18)	0.42	0.98(18)	0.03	0.03	0.03	0.04
		3PL	0.98 (24)	0.42	0.98 (24)	0	0	0	0.27
	3000	GGUM	0.95	0.89	0.96	0	0	0	1.00
		2PL	0.99 (17)	0.47	0.99 (17)	0.03	0.03	0.03	0.05
		3PL	0.99 (16)	0.48	0.99 (17)	0	0	0	0.27

I = number of items; *N* = sample size; Gen. Model = data generation models. Shaded cells in light blue indicate the correct data model (i.e., GGUM generated data); ( ) = number of uncounted replications; *Q1* = Yen's Q1 (1981) statistic; *S-X²* = Orlando and Thissen (2000) fit statistic; *G²* = McKinley and Mills (1985) fit statistic; Singles, doubles, and triples are Drasgow et al.'s (1995) adjusted chi-square model fit statistics; SRMSR = Maydeu-Olivares and Joe (2014) standardized root mean square residual fit statistic.

Mirt package fit indices.

Table 6 also presents the results of the average *Q1*, $S - X^2$, G^2, and *SRMSR* across the 100 replications from the mirt package. Upon examining the results and analyzing the calibrations, it seems that utilizing the empirical histogram density form as described by Bock and Aitkin (1981) is not recommended for dichotomous data, albeit being originally utilized in the marginal maximum likelihood algorithm, which is based on an expectation-maximization (EM) approach. Note that the EM approach is also the default method specified by Roberts et al. (2000) for calibrating the GGUM parameters. In addition, it should be mentioned that although all of the calibrations achieved convergence, many of them under the mirt package produced warnings indicating possible issues with parameters' stability, which prompted changing the seeds constantly. This can be observed by the erratic patterns within Table 6, in which type I error for the *SRMSR* is really high for the GGUM generated data while power gets lower as the number of items increase for the 2PL and 3PL data models. Also, item fit statistics *Q1* and G^2 had difficulty estimating respective chi-square values for many items within each replication. The brackets in corresponding cells show the number of replications omitted when calculating the average fit values. For example, the *Q1* fit type I error for the GGUM generated data with 10 items and 500 simulees exclude 27 replications from the 100 replications to estimate the average number of model fit rejections. These omissions occur if the bin-level predicted responses P_{hi} for a particular item cannot be estimated accurately given a few number of subjects per bin corresponding a specific estimated ability range. The $S - X^2$ item fit statistic had less of its replications omitted when compared to *Q1* and G^2 due to its reliance on observed scores rather than the estimated ability level. Still, the $S - X^2$ item fit statistic tends to overestimates type I error for the GGUM generated data and somewhat underestimates power for 2PL and 3PL, though to a lesser extent than *SRMSR*. Based on the fit results from both

packages, the only accurate absolute fit statistics obtained from comparing parameters to those estimated through calibrating the GGUM model via the (EM) approach are the *Adjusted Chi-square* χ^2 for item singles, doubles, and triples for the GGUM generated (i.e., correct) data model.

Relative Fit Indices for Dichotomous Models

Table 7 presents the results for the relative fit indices AIC and BIC, with the GGUM generated set of δs ranging from a uniform distribution [-3, 3]. As shown, The GGUM model is best fit to generated GGUM data when compared to dominance models. For 10 items, the GGUM generated data have the lowest AIC and BIC values in 81, 87, 83, and 85 of the time out of a 100 for sample sizes 500, 1000, 2000, and 3000, respectively when compared to the 2PL and 3PL data models. The percentages went up in the 90's range when the number of item is increased to 20, and all the way up to 100 percent when the number of items is 40. As mentioned earlier, conditions with smaller sample sizes may sometime yield higher percentages due to varying the model parameters per replication. Having said that, these relative fit indices do produce favorable results in terms of specifying the correct data model to the GGUM model.

Table 6. Relative Fit Indices Rates of GGUM Data Model AIC and BIC with Generated δs from a Uniform Distribution [-3, 3] Against Generated data from 2PL and 3PL Data Models

I	*N*	**Gen. Model**	*AIC*	*BIC*
10	500	GGUM	0.81	0.81
		2PL	0.12	0.12
		3PL	0.07	0.07
	1000	GGUM	0.87	0.87
		2PL	0.10	0.10
		3PL	0.03	0.03
	2000	GGUM	0.83	0.83
		2PL	0.13	0.13
		3PL	0.04	0.04
	3000	GGUM	0.85	0.85
		2PL	0.12	0.12
		3PL	0.03	0.03

		Gen. Model	*AIC*	*BIC*
20	500	GGUM	0.97	0.97
		2PL	0.02	0.02
		3PL	0.01	0.01
	1000	GGUM	0.95	0.95
		2PL	0.04	0.04
		3PL	0.01	0.01
	2000	GGUM	0.95	0.95
		2PL	0.03	0.03
		3PL	0.02	0.02
	3000	GGUM	0.94	0.94
		2PL	0.04	0.04
		3PL	0.02	0.02
		Gen. Model	*AIC*	*BIC*
40	500	GGUM	1.00	1.00
		2PL	0	0
		3PL	0	0
	1000	GGUM	1.00	1.00
		2PL	0	0
		3PL	0	0
	2000	GGUM	1.00	1.00
		2PL	0	0
		3PL	0	0
	3000	GGUM	1.00	1.00
		2PL	0	0
		3PL	0	0

I = number of items; *N* = sample size; Gen. Model = data generation models; GGUM = GGUM generated data with δs ranging from a uniform distribution [-3, 3].

Table 8 presents the results for the relative fit indices AIC and BIC, with the GGUM generated set of δs ranging from a uniform distribution [-2, 2] while excluding the interval [-1, 1]. As shown, The GGUM model is best fit to generated GGUM data when compared to dominance models, but is less able to predict the correct model for smaller number of items when compared to the previous generated GGUM data model in Table 7. For 10 items, the GGUM generated data in Table 8 have the lowest AIC and BIC values in 58, 74, 71, and 74 of the time out of a 100 for sample sizes 500, 1000, 2000, and 3000, respectively when compared to the 2PL and 3PL data models. The percentages go above 85 percent when the number of item is

increased to 20, and almost all the way to 100 percent when the number of items is 40. Again, conditions with smaller sample sizes can sometime yield higher percentages due to varying the model parameters per replication. Having said that, these relative fit indices do somewhat produce favorable results in terms of specifying the correct data model to the GGUM model.

Table 7. Relative Fit Indices Rates of GGUM Data Model AIC and BIC with Generated δs from a Uniform Distribution [-2, 2] that do not Include [-1, 1] Against Generated data from 2PL and 3PL Data Models

I	*N*	**Gen. Model**	*AIC*	*BIC*
10	500	GGUM (No 1 to -1)	0.58	0.58
		2PL	0.25	0.25
		3PL	0.17	0.17
	1000	GGUM (No 1 to -1)	0.74	0.74
		2PL	0.20	0.20
		3PL	0.06	0.06
	2000	GGUM (No 1 to -1)	0.71	0.71
		2PL	0.22	0.22
		3PL	0.07	0.07
	3000	GGUM (No 1 to -1)	0.74	0.74
		2PL	0.17	0.17
		3PL	0.09	0.09
		Gen. Model	*AIC*	*BIC*
20	500	GGUM (No 1 to -1)	0.90	0.90
		2PL	0.07	0.07
		3PL	0.03	0.03
	1000	GGUM (No 1 to -1)	0.92	0.92
		2PL	0.06	0.06
		3PL	0.03	0.03
	2000	GGUM (No 1 to -1)	0.88	0.88
		2PL	0.07	0.07
		3PL	0.05	0.05
	3000	GGUM (No 1 to -1)	0.87	0.87
		2PL	0.08	0.08
		3PL	0.05	0.05
		Gen. Model	*AIC*	*BIC*
40	500	GGUM (No 1 to -1)	1.00	1.00
		2PL	0	0
		3PL	0	0
	1000	GGUM (No 1 to -1)	1.00	1.00
		2PL	0	0
		3PL	0	0
	2000	GGUM (No 1 to -1)	0.99	0.99

		2PL	0	0
		3PL	0.01	0.01
		GGUM (No 1 to -1)	1.00	1.00
	3000	2PL	0	0
		3PL	0	0

I = number of items; *N* = sample size; Gen. Model = data generation models; GGUM (No -1 to 1) = GGUM generated data with δs ranging from a uniform distribution [-2, 2] that do not Include [-1, 1].

Table 9 presents the results for the relative fit indices AIC and BIC, with the original GGUM generated set of δs ranging from a uniform distribution [-2, 2] as specified by Roberts et al. (2002). As observed, the GGUM model does not fit generated GGUM data better than the dominance models when the number of items is 10. It is the 2PL generated data that the GGUM model best fits to, followed by the 3PL model when the sample size is 500. The GGUM generated data gains traction with 10 items in terms of being identified as a better fitting data by having a lower AIC and BIC relative to the 3PL with increasing sample size. However, the 2PL generated data still have the lowest relative fit values in the 10-item condition, irrespective of increases in sample size. The GGUM generated data are better identified as the best fitting data model as the number of items increases to 20, followed by the 2PL and 3PL generated data models, respectively. The highest percentage in which the GGUM generated data is identified as the best fitting data by the model for 20 items is 57 percent at a sample size of 3000. This is considered a low value when compared to percentages on the previous GGUM generated data from tables 7 and 8, in which the percentages were in the high 80's and even 90's. When the number of item is increased to 40, the GGUM generated data are better identified as having the lowest AIC and BIC, with identification percentages going all the way up to 86 percent at a sample size of 1000. Again, conditions with larger sample sizes such as 3000 can sometime yield lower percentages due to varying the model parameters per replication.

Table 8. Relative Fit Indices Rates of GGUM Data Model AIC and BIC with Generated δs from a Uniform Distribution [-2, 2] Against Generated data from 2PL and 3PL Data Models

I	*N*	**Gen. Model**	*AIC*	*BIC*
10	500	GGUM (-2 to 2)	0.20	0.20
		2PL	0.48	0.48
		3PL	0.32	0.32
	1000	GGUM (-2 to 2)	0.39	0.39
		2PL	0.43	0.43
		3PL	0.18	0.18
	2000	GGUM (-2 to 2)	0.32	0.32
		2PL	0.44	0.44
		3PL	0.24	0.24
	3000	GGUM (-2 to 2)	0.35	0.35
		2PL	0.45	0.45
		3PL	0.20	0.20
		Gen. Model	*AIC*	*BIC*
20	500	GGUM (-2 to 2)	0.46	0.46
		2PL	0.33	0.33
		3PL	0.21	0.21
	1000	GGUM (-2 to 2)	0.44	0.44
		2PL	0.33	0.33
		3PL	0.23	0.23
	2000	GGUM (-2 to 2)	0.49	0.49
		2PL	0.26	0.26
		3PL	0.25	0.25
	3000	GGUM (-2 to 2)	0.57	0.57
		2PL	0.20	0.20
		3PL	0.23	0.23
		Gen. Model	*AIC*	*BIC*
40	500	GGUM (-2 to 2)	0.78	0.78
		2PL	0.15	0.15
		3PL	0.07	0.07
	1000	GGUM (-2 to 2)	0.86	0.86
		2PL	0.10	0.10
		3PL	0.04	0.04
	2000	GGUM (-2 to 2)	0.76	0.76
		2PL	0.18	0.18
		3PL	0.06	0.06
	3000	GGUM (-2 to 2)	0.77	0.77
		2PL	0.15	0.15
		3PL	0.08	0.08

I = number of items; *N* = sample size; Gen. Model = data generation models; GGUM (No -2 to 1) = GGUM generated data with δs ranging from a uniform distribution [-2, 2].

Although the relative fit indices in Table 9 do somewhat produce favorable results in terms of specifying the correct data model to the GGUM model when the number of items are relatively high, that is not the case in conditions with smaller set of items. In other words, data from dominance IRT models such as the 2PL may be fit by the GGUM as well. This goes back to the possible overlap between the response functions of both dominance and ideal point IRT models (Tay et al., 2011). As shown, one of the solutions to the overlap issue is to expand the range of the δs values for the GGUM generated data in order to capture the differences in the response functions between the models as they become more apparent on the extremes. Another solution would be to exclude the δs that are possibly overlapping, mainly the ones located near 0 (i.e., omitting δs between -1 and 1). Based on comparing the different GGUM generated data, the first solution yielded the lowest AIC and BIC values. To summarize, generated GGUM data with δs ranging from a uniform distribution [-3, 3] in Table 7 had the lowest AIC and BIC values with relatively high percentages across all replications and conditions.

Absolute Fit Indices for Polytomous Data

GGUM package fit indices.

Table 10 presents the results of the average *Adjusted Chi-square* χ^2 for item singles, doubles, and triples across the 100 replications from the GGUM package for polytomous data. When the GGUM model is correctly fit to GGUM generated data, *Adjusted Chi-square* χ^2 fit statistics exhibit low type I error rates for all item variants, with a zero rejection rate of model fit across all conditions, irrespective of the number of items and sample size. When fitting the GGUM model to GRM data, the rate of detecting misfit (i.e., power) vary as a function of the specific model fit index, the number of items, and sample size. *Adjusted Chi-square* χ^2 for item singles and doubles are better able to detect misfit than their item triples counterpart, with the

highest rates of detecting misfit being in the 20 items conditions instead of those with 40 items, except for the condition of item doubles with 40 items and 3000 simulees, in which the detection rate of misfit is 78 percent. This might be due to varying the model parameters per replication as mentioned in previous sections. Also, the rate of detecting misfit increased with increasing sample size for item singles and doubles fit indices within each item category, while surprisingly decreasing with the item triples fit index.

These results do not agree with what was found in previous studies about *Adjusted Chi-square* χ^2 for item doubles and triples as being the more useful fit indices in identifying the correct model when compared to item singles (Drasgow et al., 1995; Tay et al., 2011). Although the detection of misfit improved greatly by the *Adjusted Chi-square* χ^2 fit statistics for polytomous data when compared to those generated by dichotomous models, the performance of such fit statistics are yet to be considered high, with the highest detection rate barely reaching 80 percent. Also, the decreasing detection rate of item triples fit index with increasing sample size is problematic and should be noted accordingly.

Mirt package fit indices.

Table 10 also presents the results of the average *Q1*, $S-X^2$, G^2, and *SRMSR* across the 100 replications from the mirt package for polytomous data. As in the case for dichotomous data, the type I error rate is really high by the *SRMSR* fit index for the GGUM generated data, as well the other fit indices. Also, mirt produced the same warnings with dichotomous data indicating possible issues with parameters' stability, which prompted changing the seeds constantly. It can also be noted that for item fit statistics *Q1*, $S-X^2$, and G^2, the number of excluded replications still exist but in smaller quantities. However, when fitting the GGUM to GRM generated data, the ability of the *SRMSR*, *Q1*, and G^2 to detect misfit increased as the number of items and sample size increased.

Table 9. Type I Error Rates and Power of Absolute Model Fit Indices When the GGUM Model is Fit to Polytomous IRT Data Models

I	*N*	**Gen. Model**	*Q1*	*S-X²*	*G²*	X2 Singles	X2 Doubles	X2 Triples	SRMSR
10	500	GGUM	0.88 (8)	0.82 (1)	0.91 (12)	0	0	0	1.00
		GRM	0.94 (2)	0.51	0.93 (9)	0.23	0.38	0.44	0.71
	1000	GGUM	0.97 (3)	0.90 (2)	0.98 (3)	0.04	0.04	0.02	1.00
		GRM	0.98	0.52	0.99 (2)	0.45	0.53	0.41	0.70
	2000	GGUM	0.99 (1)	0.96 (5)	0.99 (2)	0	0	0	1.00
		GRM	1.00	0.67	1.00 (2)	0.49	0.58	0.31	0.78
	3000	GGUM	1.00 (2)	0.97 (1)	1.00 (2)	0.03	0.03	0.01	1.00
		GRM	1.00	0.70	1.00 (2)	0.65	0.69	0.33	0.79
		Gen. Model	*Q1*	*S-X²*	*G²*	X2 Singles	X2 Doubles	X2 Triples	SRMSR
20	500	GGUM	0.72 (7)	0.85 (18)	0.74 (9)	0	0	0	1.00
		GRM	0.94 (1)	0.35	0.91 (4)	0.61	0.64	0.58	0.94
	1000	GGUM	0.85 (1)	0.89 (14)	0.87 (1)	0	0	0	1.00
		GRM	0.98 (2)	0.47	0.98 (3)	0.67	0.63	0.46	0.97
	2000	GGUM	0.93	0.96 (10)	0.95	0	0	0	1.00
		GRM	1.00	0.51	1.00 (1)	0.72	0.76	0.39	1.00
	3000	GGUM	0.98 (1)	0.98 (9)	0.98 (1)	0	0	0	1.00
		GRM	1.00 (1)	0.575	1.00 (1)	0.69	0.74	0.48	0.99
		Gen. Model	*Q1*	*S-X²*	*G²*	X2 Singles	X2 Doubles	X2 Triples	SRMSR
40	500	GGUM	0.55 (7)	0.83 (12)	0.56 (8)	0	0	0	1.00
		GRM	0.93	0.29	0.88 (1)	0.21	0.49	0.50	1.00
	1000	GGUM	0.66 (1)	0.92 (19)	0.66 (1)	0	0	0	1.00
		GRM	0.99 (2)	0.36	0.97 (3)	0.40	0.59	0.43	1.00
	2000	GGUM	0.80	0.98 (10)	0.80	0	0	0	1.00
		GRM	1.00 (4)	0.43	0.99 (5)	0.50	0.68	0.36	1.00
	3000	GGUM	0.86	0.99 (14)	0.87	0	0	0	1.00
		GRM	1.00 (5)	0.47	1.00 (6)	0.64	0.78	0.38	1.00

I = number of items; *N* = sample size; Gen. Model = data generation models. Shaded cells in light blue indicate the correct data model (i.e., GGUM generated data); ( ) = number of uncounted replications; *Q1* = Yen's Q1 (1981) statistic; *S-X²* = Orlando and Thissen (2000) fit statistic; *G²* = McKinley and Mills (1985) fit statistic; Singles, doubles, and triples are Drasgow et al.'s (1995) adjusted chi-square model fit statistics; SRMSR = Maydeu-Olivares and Joe (2014) standardized root mean square residual fit statistic.

As observed in Table 10, the ability of the *SRMSR* to detect misfit went up from 71 to 79 percent on the 10-item condition with increasing sample size. For 20 items, the detection of misfit for the *SRMSR* went up from 94 to 100 percent. For 40 items, the detection of misfit is observed across all replications 100 of the time, irrespective of sample size. For item fit statistics *Q1* and G^2, the detection of misfit is also high and mostly in the 90's range, even displaying a perfect rate of detecting misfit for large sample sizes, irrespective of the number of items. However, some of the calibrations for these item fit statistics are omitted when calculating the percentage of misfit out of the total calibrations as was the case for dichotomous data, which makes *SRMSR* a better model fit index for detecting misfit. Having said that, it should be noted that omitted calibrations for the *Q1* and G^2 item fit statistics are few and might still be considered as good estimators of model misfit for polytomous data models. This of course, is not the case for dichotomous data. For the $S– X^2$ item fit statistic, the ability to detect misfit is not consistent across conditions, and ranged from 29 percent all the way to 70 percent in an unsystematic progression across the conditions with some omitted calibrations. Hence, the $S – X^2$ item fit statistic is the least performing fit statistic in detecting misfit when calibrating polytomous data. Although some of these results are promising for detecting misfit with absolute fit indices when compared to the calibrations with dichotomous data, utilizing the marginal maximum likelihood algorithm in mirt still fails to identify the correct polytomous generated data when the GGUM model is calibrated to GGUM generated data,

Relative Fit Indices for Polytomous Models

Table 11 presents the results for the relative fit indices AIC and BIC when fitting polytomous data. As shown, The GGUM model is best fit to generated GGUM data when compared to those generated by the GRM. Across all conditions, irrespective of the number of

items and sample size, the GGUM generated data are almost identified perfectly by the GGUM model across the 100 replications as having lower relative AIC and BIC values when compared to data generated from the GRM model. Again, these results show that relative fit indices seem to be more reliable in identifying the correct data model when compared to their absolute fit counterparts, even more so when testing polytomous data. In Table 11, it can be seen that the ability to identify the correct model is at almost 100 percent across replications for just 10 items, while being in the 80's range in terms of identification percentages for dichotomous data. A possible explanation for correct higher identification rates by generated polytomous data might have to do with the lower probability of overlap between observable response categories (ORCs) from different polytomous IRT models. Also, dichotomizing graded data prior to model calibration might risk in decreasing the precision of person estimates (Roberts & Laughlin, 1996). The aforementioned statement holds if there is a theoretical rationale behind utilizing graded (i.e., polytomous) data for data collection.

Table 10. Relative Fit Indices Rates of GGUM Polytomous Data Model AIC and BIC Against Generated data from the GRM Data Model

I	*N*	**Gen. Model**	*AIC*	*BIC*
10	500	GGUM	1.00	1.00
		GRM	0	0
	1000	GGUM	0.99	0.99
		GRM	0.01	0.01
	2000	GGUM	1.00	1.00
		GRM	0	0
	3000	GGUM	1.00	1.00
		GRM	0	0
		Gen. Model	*AIC*	*BIC*
20	500	GGUM	1.00	1.00
		GRM	0	0
	1000	GGUM	1.00	1.00
		GRM	0	0
	2000	GGUM	1.00	1.00
		GRM	0	0
	3000	GGUM	1.00	1.00
		GRM	0	0
		Gen. Model	*AIC*	*BIC*
40	500	GGUM	1.00	1.00
		GRM	0	0
	1000	GGUM	1.00	1.00
		GRM	0	0
	2000	GGUM	1.00	1.00
		GRM	0	0
	3000	GGUM	1.00	1.00
		GRM	0	0

I = number of items; *N* = sample size; Gen. Model = data generation models; GGUM = GGUM polytomous generated data with 4 response categories.

Based on calibrating the model parameters via MMLE and subsequently testing their model fit, it seems that polytomous datasets are better identified by the GGUM model as either being generated by the correct data model or otherwise. Despite issues with some of the absolute fit indices in detecting fit/misfit, it seems that relative fit indices do perform relatively well in identifying the correct data model for both dichotomous and polytomous datasets.

CHAPTER 5

DISCUSSION

General Discussion of Results

This study tested the performance of several IRT model fit indices when the GGUM is fit to both dominance and ideal point IRT generated data for both dichotomous and polytomous item responses. Currently, the only attempt that compared different IRT model fit indices directly for dominance and ideal point data models in terms detecting misfit was conducted by Nye et al. (2019). They also examined multidimensional generated IRT data. However, their analyses did not include the S– X2 and G2 fit statistics. In addition, they calibrated the 2PL and GRM models to different generated IRT data models and the calibration involved Markov chain Monte Carlo (MCMC) estimation with Metropolis-Hastings within Gibbs sampling (Patz & Junker, 1999) instead of the EM approach used in this study.

The results showed that the ability of the absolute model fit indices to detect misfit as well as to identify the correct data model was best realized by the SRMSR and Adjusted Chi-square $\chi 2$ model fit statistics for polytomous data, respectively. As for dichotomous generated data, the *Adjusted Chi-square* χ^2 fit statistics was the only accurate absolute fit indices yielding low type I error rates when comparing GGUM estimates to their. These results are based on selecting a desired nominal rate of 0.05 for type I error and 0.80 for power as cutoff points for determining the 'low' and "high" values for such indices. As mentioned and demonstrated earlier, utilizing the empirical histogram density form for dichotomous data when using MMLE yielded high type I error rates and low power, irrespective of the number of items and sample size. Also, it is possible that the overlap between the IRF's of dichotomous dominance IRT models such as the

2PL and 3PL with ideal point models such as the GGUM makes it more difficult for the fit indices to detect misfit when the model calibration is performed via the EM algorithm.

Prior to performing the analyses for the dichotomous generated items, model calibration was tested using the 'Gaussian' density form in mirt. This produced lower type I error rates than the empirical histogram density form but still resulted in low power for dominance models. Also, many of the calibrations did either not converge or produced warnings indicating possible issues with parameters' stability. Table 12 presents the type I error and power rates for trial calibrations of sample sizes 500 and 2000 when the GGUM model is fit to both GGUM and 2PL generated data while utilizing the Gaussian density form within the EM algorithm. As observed, model convergence could not be achieved for the 10 and 20 item conditions when the sample size is 500. Given that, it seems that the empirical histogram density form leads to more cases of model convergence than does the Gaussian density form, even if issues of model stability do sometimes emerge when fitting the GGUM model to its generated data in mirt. Also, since the GGUM package uses the EM algorithm, the empirical histogram density form was kept in the mirt package to closely align the estimation settings in terms of parameter calibrations between both packages.

Table 11. Trial Calibrations Type I Error Rates and Power of Absolute Model Fit Indices When the GGUM Model is Fit to Dichotomous IRT Data Models using the 'Gaussian' Density Form in mirt.

I	*N*	**Gen. Model**	*Q1*	$S\text{-}X^2$	G^2	SRMSR
10	500	GGUM	-	-	-	-
		2PL	-	-	-	-
	2000	GGUM	1.00 (8)	0.28	1.00 (15)	0.39
		2PL	1.00 (13)	0.14	1.00 (32)	0.01
		Gen. Model	*Q1*	$S\text{-}X^2$	G^2	SRMSR
20	500	GGUM	-	-	-	-
		2PL	-	-	-	-
	2000	GGUM	0.84 (1)	0.25	0.85 (2)	0.30
		2PL	0.98 (3)	0.11	0.99 (20)	0
		Gen. Model	*Q1*	$S\text{-}X^2$	G^2	SRMSR
40	500	GGUM	0.12 (39)	0.11	0.13 (43)	0.38
		2PL	0.76 (23)	0.09	0.68 (28)	0.04
	2000	GGUM	0.30	0.21	0.32	0.19
		2PL	0.95 (4)	0.11	0.96(7)	0

I = number of items; *N* = sample size; Gen. Model = data generation models. Shaded cells in light blue indicate the correct data model (i.e., GGUM generated data); ( ) = number of uncounted replications; blank cells = unconverged calibrations; *Q1* = Yen's Q1 (1981) statistic; $S\text{-}X^2$ = Orlando and Thissen (2000) fit statistic; G^2 = McKinley and Mills (1985) fit statistic; SRMSR = Maydeu-Olivares and Joe (2014) standardized root mean square residual fit statistic.

For polytomous generated items, type I error rates were also high when fitting the GGUM model to its generated data in mirt. To exclude the possibility that such high rates may be caused by the negative generated taus (τ_{ik}) in the GGUM package, which are used to generate item responses along with the other model parameters in equation 5, GGUM data were also simulated using the mirt package's 'simdata' feature on trial calibrations. Table 13 presents both

negative and positive taus generated for 20 hypothetical items, in which the first three columns are the ones used to generate item responses in the GGUM package. Although 'simdata' utilizes positive taus (τ_{ik}) for generating polytomous response data (i.e., columns 5 to 7 in Table 13), the results are still similar in terms of type I error rates between the packages when the GGUM model is fit to its data.

Table 12. Positive and Negative Taus (τik) Generated for 20 Hypothetical Items Using GGUM Package

Item #	Negative taus				Positive taus		
	tau 1	tau 2	tau 3	tau 0	tau 3	tau 2	tau 1
1	-0.9405	-0.6859	-0.4389	0	0.4389	0.6859	0.9405
2	-1.3475	-1.1412	-0.9136	0	0.9136	1.1412	1.3475
3	-1.3656	-1.0732	-0.714	0	0.714	1.0732	1.3656
4	-1.3928	-1.1802	-0.9625	0	0.9625	1.1802	1.3928
5	-0.9462	-0.6939	-0.478	0	0.478	0.6939	0.9462
6	-1.3669	-1.1155	-0.8359	0	0.8359	1.1155	1.3669
7	-1.0729	-0.802	-0.4565	0	0.4565	0.802	1.0729
8	-1.0097	-0.7635	-0.5384	0	0.5384	0.7635	1.0097
9	-1.0162	-0.7263	-0.4857	0	0.4857	0.7263	1.0162
10	-1.1297	-0.8754	-0.6223	0	0.6223	0.8754	1.1297
11	-1.3484	-1.0977	-0.8576	0	0.8576	1.0977	1.3484
12	-1.4863	-1.2365	-0.9858	0	0.9858	1.2365	1.4863
13	-1.2769	-1.0755	-0.8628	0	0.8628	1.0755	1.2769
14	-1.207	-0.9614	-0.7787	0	0.7787	0.9614	1.207
15	-1.291	-1.0061	-0.7277	0	0.7277	1.0061	1.291
16	-1.1376	-0.9244	-0.6208	0	0.6208	0.9244	1.1376
17	-1.107	-0.9176	-0.621	0	0.621	0.9176	1.107
18	-1.3235	-1.0603	-0.8205	0	0.8205	1.0603	1.3235

19	-1.0073	-0.7705	-0.5673	0	0.5673	0.7705	1.0073
20	-1.397	-1.1895	-0.9375	0	0.9375	1.1895	1.397

Table 14 shows the model fit results of 5 selected calibrations (i.e., replications) using both packages with 20 items and a sample size of 2000, with *SRMSR* statistic from the mirt package being compared to the *Adjusted Chi-square* χ^2 statistics from the GGUM package.

Table 13. Model Fit Results of 5 Selected Calibrations via Simulating GGUM Data from the 'simdata' Syntax in mirt with 20 Polytomous Items and a Sample size of 2000

(I = 20) (N = 2000) (C = 3) (Seed = 2875)	**Gen. Model**	X2 Singles (mean)	X2 Doubles (mean)	X2 Triples (mean)	SRMSR
Replication #1	GGUM	0	0.3425	0.7054	0.3162446
Replication #2		0	0.3237	0.6789	0.2467794
Replication #3		0	0.4355	0.6547	0.3191819
Replication #4		0	0.3131	0.5934	0.291728
Replication #5		0	0.3159	0.4957	0.2890404

I = number of items; *N* = sample size; *C* = number of response categories -1; Gen. Model = data generation models; GGUM = GGUM polytomous generated data with 4 response categories; Singles, doubles, and triples are Drasgow et al.'s (1995) adjusted chi-square model fit statistics; SRMSR = Maydeu-Olivares and Joe (2014) standardized root mean square residual fit statistic.

As observed, all of the 5 calibrations produced *SRMSR* values that are greater than 0.05, which incorrectly indicate misfit (i.e., high type I error). In contrast, all of the mean *Adjusted Chi-square* χ^2 statistics values for item singles, doubles, and triples are less than 3, which indicate that the model fits the data. These results are similar to those obtained on the actual analysis, which generated data using the GGUM package.

The GGUM package ability to detect misfit for *Adjusted Chi-square* χ^2 fit statistics barely approached 80 percent in the condition of 40 items and sample size of 3000. However, this was only the case for the item doubles fit statistics. Also, conditions with 20 items for item singles and doubles fit statistics had generally higher detection rates of misfit than those with 40 items. This might be due to varying the model parameters per replications, which might cause a combination of item and person generated parameters that yield better fit results, even in cases with smaller number of items.

Item level fit statistics *Q1*, $S - X^2$, and G^2 did perform poorly in detecting misfit for dichotomous generated data, with many of the items being excluded from the analysis of *Q1* and G^2 due to the possible lack of a minimum number of subjects to be assigned to an ability group during the binning process when estimating fit. However, both *Q1* and G^2 were able to correctly detect misfit a high percentage of the time with polytomous generated items. Omitted items are also excluded in calibrations with polytomous data but to a lesser degree than their dichotomous counterparts. The $S - X^2$ item fit statistic incurred less instances of excluded items since its grouping process is based on total scores rather than simulees' abilities. However, its performance in detecting misfit is weak when compared to *Q1* and G^2. This result contradicts a body of research demonstrating its efficiency and accuracy over the aforementioned item fit statistics in detecting misfit (Ames & Penfield, 2015; Orlando & Thissen, 2000, 2003; Roberts, 2008).

Although previous research has shown that the $S - X^2$ fit statistic performed well in detecting misfit using MMLE and EM, none of the papers except for Roberts (2008) that actually investigated its performance in calibrating the GGUM parameters. For instance, the series of papers published by Orlando and Thissen investigated the performance of the $S - X^2$ fit statistic

for nested dominance models only. As for Roberts (2008), his study differed from the current one by comparing different $S - X^2$ fit statistics variants to nested GGUM models via fixing the discrimination and threshold parameters for some conditions across 1000 replications per cell. Also, six response categories were used instead of four in his analysis, which by default excludes testing the fit of dichotomous items. Roberts (2008) results also excluded items that did not have cases in particular bins to perform the fit estimation. New set of parameters were generated accordingly, which might have led to higher proportions of misfit detection between nested GGUM models. This practice is not utilized in the current study, which might explain the poor performance of the $S - X^2$ fit statistic.

Adjusted Chi-square χ^2 statistics also performed better for the GGUM in terms of detecting misfit when compared to the current study for dichotomous items in Tay et al. (2011) using MMLE and EM. The disparity in the results between the studies is probably due to preselecting and omitting the middle ranged δ values within each generated data per calibration in Tay et al. (2011); a practice that is also not followed in the current study. Also, cross validation data is utilized in Tay et al. (2011) using the same generated item parameters (i.e., fixing the parameters across replications). These generated item parameters do not include simulees with zero endorsements when estimating doubles and triples fit statistics. The aforementioned data setup is also not followed in the current study.

When identifying the correct data model based on relative fit indices AIC and BIC, the GGUM model did identify the correct model consistently for dichotomous items as the number of items and sample size increased. However, given the overlap between the IRF's of the dichotomous IRT models, the choice of item location ranges δs for the GGUM generated data determined the percentage of correct data model identifications. As was shown, a range of δs [-3,

3] from a uniform distribution produced the highest percentages of correct identification in terms of the GGUM data model having the lowest AIC and BIC when compared to those generated by dominance models, even in conditions where the number of items was 10. A range of δs [-2, 2] from a uniform distribution produced the least percentages of correct identifications in terms of exhibiting higher AIC and BIC values when compared to data generated form dominance models. This was most likely to be observed when the number of items is 10, in which 2PL generated data had the lowest AIC and BIC values. For polytomous items, relative fit indices almost always identified the correct model 100 percent of the time, with GGUM generated data resulting in lower AIC and BIC values than GRM generated data across all conditions, irrespective of the number of items and sample size. In short, the relative (i.e., comparative) model fit indices AIC and BIC are the most consistent and efficient indices in identifying the correct data model. In this study, this applies to both dichotomous data with generated δs [-3, 3] from a uniform distribution, and to a larger extent to polytomous data when the EM algorithm is used for model calibration.

Given that AIC and BIC indices are relative fit indices, their utility might only be realized when interpreted along with absolute fit indices. One useful strategy to test data models is to compare their relative fit indices first, and then testing the data model yielding the lowest AIC and BIC values using some measure of absolute fit index such as the *SRMSR* or the *Adjusted Chi-square* χ^2 fit statistics. As mentioned in Nye et al. (2019), the practice of testing multiple fit indices to assess model fit is common in SEM literature (Hu & Bentler, 1999). However, such a practice is less realized in IRT literature and might assist researchers in identifying the appropriate IRT model for implemented data (Nye et al., 2019).

Recommendations for Future Studies

Assessing model fit for ideal point models such as the GGUM is of paramount importance, particularly when the assumptions of the selected IRT model are assumed to be true. Unfortunately, there is a shortage of analyses pertaining to model fit for ideal point models. Based on comparing different model fit indices in this study, the majority of such fit indices were not able to detect misfit for dichotomous data when the GGUM was fit to dominance model. However, more promising results were obtained for polytomous data in terms of detecting misfit, particularly when utilizing the *SRMSR* fit statistic. Also, low type I error rates were only observed by the *Adjusted Chi-square* χ^2 fit statistics. In general, the best performing fit statistics were the relative (i.e., comparative) ones such as AIC and BIC, with the selection of appropriate δ ranges affecting the rate of correct identification for dichotomous data, while correctly identifying the data model for polytomous data in across all conditions. However, previous attempts to compare different fit indices for both dominance and ideal point IRT models did produce more promising results in terms of detecting fit/misfit for dichotomous data (Nye et al., 2019).

One possible issue in the current study that might have led to the poor performance of the absolute model fit indices in detecting misfit is the implementation of the marginal maximum likelihood EM algorithm to calibrate the parameters using the empirical histogram density form (Bock & Aitkin, 1981). In Nye et al. (2019), Bayesian procedures were implemented to calibrate the model parameters. Possible future studies can compare the accuracy of the calibrated parameters using both the EM algorithm and one of the Bayesian methods such as the Markov chain Monte Carlo (MCMC) method. The R package ‘bggum’ utilizes a Bayesian approach to calibrate the GGUM parameters, and can be used to perform a comparative analysis of model fit

indices using both model calibration algorithms (Brandon, Duck-Mayr, & Montgomery, 2020). For example, it would be interesting to observe the performance of the item fit statistic $S - X^2$ when the calibrated parameters are calculated using a Bayesian estimator, since this fit statistic is usually more accurate in detecting misfit than more traditional ones such as *Q1* and G^2 (Orlando & Thissen, 2000). Such an outcome is not realized in the current study when the EM algorithm is used given its different research objectives and data generation process from the aforementioned studies. In short, previous research testing model fit indices for the GGUM utilized conditions to data generation that increased the proportion of detecting misfit across calibrations. Though promising, the results of such studies are less generalizable than the current one. As mentioned earlier, the only study that compared model fit indices between different IRT models as the current one is Nye et al. (2019), which utilized a Bayesian approach to calibrate the GGUM parameters.

Future studies can also fix model parameters such as item or person parameters to narrow down the possible sources of disturbance associated with the data, which was apparent at several conditions where higher detection rates of misfit were observed under smaller number of items and sample sizes. Though useful, it should be noted that fixing any of the item parameters may lower the generalizability of results. In other words, having a particular model fit index detect misfit in a consistent manner while varying item and person parameters along their respective distribution spectrums makes it more viable as a measure of fit. Still, future studies could generate multiple sets of response data and control the degree to which parameters are fixed within datasets. These datasets would then be compared to one another in terms of detecting fit.

The overlap between the IRFs of the dichotomous models made it difficult for the relative fit indices to identify the correct data model, particularly between GGUM and the 2PL-generated data.

Future researchers might be interested in testing a series of different item parameters ranges other than the items' locations δs as was investigated in this study. Such an overlap might explain why some attitude and survey data can still be well calibrated by dominance IRT models such as the 2PL model (Tay et al., 2011).

It would also be interesting to investigate how other polytomous IRT data models such as Muraki's (1992) GPCM would fare in terms of fit when calibrated by the GGUM model. As mentioned earlier, the GPCM can model the GGUM's model subjective response categories as described by the third premise when defining the GGUM model (Roberts et al., 2000). Given that, researchers can investigate possible ranges in which specific ORFs between the GGUM and the GPCM overlap.

Although simple IRT dominance models such as the Rasch or the 1PL models cannot be compared directly to the GGUM model in terms of fit given that the latter does not assume a fixed slope (i.e., discrimination) across items, they can nevertheless be possibly compared to the GUM variant of the model. The GUM model assumes a fixed slope across items (Roberts & Laughlin, 1996). Hence, generated GUM data can probably be compared to the Rasch or 1PL models by the *INFIT* and *OUTFIT* fit indices (Ames & Penfield, 2015; Masters, 1982).

Conclusion

Although the results provide some promising methods for assessing fit, particularly those assessing relative model fit both dichotomous and polytomous item responses, they also point out the limitations with several absolute fit indices when the marginal maximum likelihood EM algorithm is used to calibrate the model parameters. Also, it is difficult for absolute model fit indices to detect misfit when the GGUM model is fit to dichotomous generated data such as the 2PL and 3PL model given their IRFs' overlap with that of the GGUM data. However, expanding

the ranges of item locations δs for GGUM generated data might lead to identifying the correct data model by relative fit statistics such as AIC and BIC more frequently. For polytomous response data, *SRMSR* fit statistic is useful in detecting misfit when the GGUM model is fit to the GRM data, while the *Adjusted Chi-square* χ^2 fit statistics are useful in detecting fit when the GGUM model is fit to its data, even when the EM algorithm is used to calibrate the model parameters.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Csaki (Eds.), *Proceedings of the 2nd International Symposium on Information Theory* (pp. 267-281). Budapest: Akademiai Kiado.

Ames, A., & Penfield, R. (2015). An NCME instructional module on item-fit statistics for item response theory models. *Educational Measurement: Issues and Practice, 34*(3), 39-48.

Bandalos, D. L. (2018). *Measurement theory and applications for the social sciences*. New York, NY: The Guilford Press.

Bennett, J., & Hays, W. (1960). Multidimensional unfolding: Determining the dimensionality of ranked preference data. *Psychometrika, 25*(1), 27-43.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord & M. R. Novick, *Statistical theories of mental test scores*. Reading, MA: Addison-Wesley.

Bock, R. D. (1960), *Methods and applications of optimal scaling*. Chapel Hill, NC: L. L. Thurstone Psychometric Laboratory Memorandum, No.25.

Bock, R. (1972). Estimating item parameters and latent ability when responses are scored in two or more nominal categories. *Psychometrika, 37*(1), 29-51.

Bock, R., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika, 46*(4), 443-459.

Bock, R., & Lieberman, M. (1970). Fitting a response model for *n* dichotomously scored items. *Psychometrika, 35*(2), 179-197.

Bock, R., & Mislevy, R. (1982). Adaptive EAP estimation of ability in a microcomputer environment. *Applied Psychological Measurement, 6*(4), 431-444.

Bolt, D. (2002). A Monte Carlo comparison of parametric and nonparametric polytomous DIF detection methods. *Applied Measurement in Education, 15*(2), 113-141.

Borsboom, D. (2009). *Measuring the mind* (1st ed.). Cambridge: Cambridge University Press.

Brandon, J., Duck-Mayr, & Montgomery, J. (2020). bggum: Bayesian estimation of generalized graded unfolding model parameters. R package version 1.0.2. https://CRAN.R-project.org/package=bggum

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science, 16*(3), 199-231.

Browne, M., & Cudeck, R. (1992). Alternative ways of assessing model fit. *Sociological Methods & Research, 21*(2), 230-258.

Cai, L. (2017). flexMIRT® version 3.51: Flexible multilevel multidimensional item analysis and test scoring [Computer software]. Chapel Hill, NC: Vector Psychometric Group.

Camilli, G. (1994). Origin of the scaling constant d = 1.7 in item response theory. *Journal of Educational and Behavioral Statistics, 19*(3), 293.

Chalmers, R. P. (2012). mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software, 48*(6), 1-29.

Chernyshenko, O., Stark, S., Chan, K., Drasgow, F., & Williams, B. (2001). Fitting item response theory models to two personality inventories: Issues and insights. *Multivariate Behavioral Research, 36*(4), 523-562.

Cook, L., & Eignor, D. (1989). Using item response theory in test score equating. *International Journal Of Educational Research, 13*(2), 161-173.

Coombs, C. H. (1964). *A theory of data.* New York, NY: John Wiley & Sons, Inc.

Coombs, C., Dawes, R., & Tversky, A. (1970). *Mathematical psychology*. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Cronbach, L., & Meehl, P. (1955). Construct validity in psychological tests. *Psychological Bulletin, 52*(4), 281-302.

Cronbach, L. J., Rajaratnam, N., & Gleser, G. C. (1963). Theory of generalizability: A liberalization of reliability theory. *British Journal of Statistical Psychology, 16*(2), 137 - 163.

Davison, M. (1977). On a metric, unidimensional unfolding model for attitudinal and developmental data. *Psychometrika, 42*(4), 523-548.

de Ayala, R. (2009). *The theory and practice of item response theory*. New York, NY: The Guilford Press.

De Ayala, R., & Hertzog, M. (1991). The assessment of dimensionality for use in item response theory. *Multivariate Behavioral Research, 26*(4), 765-792.

De Ayala, R., Dodd, B., & Koch, W. (1992). A comparison of the partial credit and graded response models in computerized adaptive testing. *Applied Measurement in Education*, *5*(1), 17-34.

DeMars, C. (2005). Type I error rates for Parscale's fit index. *Educational and Psychological Measurement*, *65*(1), 42-50.

Drasgow, F., Chernyshenko, O., & Stark, S. (2010). 75 years after Likert: Thurstone was right!. *Industrial and Organizational Psychology*, *3*(4), 465-476.

Drasgow, F., Levine, M., Tsien, S., Williams, B., & Mead, A. (1995). Fitting polytomous item response theory models to multiple-choice tests. *Applied Psychological Measurement*, *19*(2), 143-166.

Drasgow, F., Levine, M., Williams, B., McLaughlin, M., & Candell, G. (1989). Modeling incorrect responses to multiple-choice items with multilinear formula score theory. *Applied Psychological Measurement*, *13*(3), 285-299.

Foster, G., Min, H., & Zickar, M. (2017). Review of item response theory practices in organizational research. *Organizational Research Methods*, *20*(3), 465-486.

Fraley, R., Waller, N., & Brennan, K. (2000). An item response theory analysis of self-report measures of adult attachment. *Journal of Personality And Social Psychology*, *78*(2), 350-365.

Hoijtink, H. (1990). A latent trait model for dichotomous choice data. *Psychometrika*, *55*(4), 641-656.

Hu, L., & Bentler, P. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journal*, *6*(1), 1-55.

Janssen, T. (2001). Frege, contextuality and compositionality. *Journal of Logic, Language, and Information, 10*(1), 115-136.

Kang, T., Cohen, A., & Sung, H. (2009). Model selection indices for polytomous items. *Applied Psychological Measurement*, *33*(7), 499-518.

Kaskowitz, G., & De Ayala, R. (2001). The effect of error in item parameter estimates on the test response function method of linking. *Applied Psychological Measurement*, *25*(1), 39-52.

Keppel, G., & Wickens, T. (2004). *Design and analysis* (4th ed.). Upper Saddle River, N.J.: Pearson Prentice Hall.

Kieftenbeld, V., & Natesan, P. (2012). Recovery of graded response model parameters. *Applied Psychological Measurement*, *36*(5), 399-419.

Kripke, S. (2008). Frege's theory of sense and reference: Some exegetical notes. *Theoria*, *74*(3), 181-218.

LaHuis, D., Clark, P., & O'Brien, E. (2009). An examination of item response theory item fit indices for the graded response model. *Organizational Research Methods*, *14*(1), 10-23.

Linden, W., & Glas, C. (2000). *Computerized adaptive testing*. Dordrecht: Kluwer Academic.

Liu, J., & Zhang, J. (2020). An item-level analysis for detecting faking on personality tests: appropriateness of ideal point item response theory models. *Frontiers In Psychology*, *10*.

Loevinger, J. (1957). Objective tests as instruments of psychological theory. *Psychological Reports, 3*(3), 635-694.

Lord, F. (1952). A theory of test scores (psychometric monograph no. 7). Richmond, VA: Psychometric Corporation. Retrieved from http://www.psychometrika.org/journal/online/MN07.pdf

Lord, F., & Wingersky, M. (1984). Comparison of IRT true-score and equipercentile observed-score "equating". *Applied Psychological Measurement*, *8*(4), 453-461.

Masters, G. (1982). A Rasch model for partial credit scoring. *Psychometrika*, *47*(2), 149-174.

Maydeu-Olivares, A., & Joe, H. (2014). Assessing approximate fit in categorical data analysis. *Multivariate Behavioral Research*, *49*(4), 305-328.

McIver, J., & Carmines, E. (1981). *Unidimensional scaling*. Beverly Hills, CA: Sage Publications.

McKinley, R., & Mills, C. (1985). A comparison of several goodness-of-fit statistics. *Applied Psychological Measurement*, *9*(1), 49-57.

Muraki, E. (1992). A generalized partial credit model: application of an EM Algorithm. *Applied Psychological Measurement*, *16*(2), 159-176.

Muthén, L. K., & Muthén, B. O. (1998-2017). M*plus* user's guide. (6th ed.). Los Angeles, CA: Muthén & Muthén.

Nydick, S. W. (2014). catIrt: An R package for simulating RIT-based computerized adaptive Tests. R package version 0.5-0. https://CRAN.R-project.org/package=catIrt

Nye, C., Joo, S., Zhang, B., & Stark, S. (2019). Advancing and evaluating irt model data fit indices in organizational research. *Organizational Research Methods, 23*(3), 457-486.

Orlando, M., & Thissen, D. (2000). Likelihood-based item-fit indices for dichotomous item response theory models. *Applied Psychological Measurement, 24*(1), 50-64.

Orlando, M., & Thissen, D. (2003). Further investigation of the performance of s - x2: an item fit index for use with dichotomous item response theory models. *Applied Psychological Measurement, 27*(4), 289-298.

Patz, R., & Junker, B. (1999). A straightforward approach to Markov chain Monte Carlo methods for item response models. *Journal of Educational and Behavioral Statistics, 24*(2), 146.

Raschka, S., & Mirjalili, V. (2019). *Python machine learning : machine learning and deep learning with python, scikit-learn, and tensorflow 2 (3rd ed.)*. Birmingham, UK: Packt Publishing, Limited.

Reise, S., & Waller, N. (1990). Fitting the two-parameter model to personality data. *Applied Psychological Measurement, 14*(1), 45-58.

Roberts, J. (2008). Modified likelihood-based item fit statistics for the generalized graded Unfolding Model. *Applied Psychological Measurement, 32*(5), 407-423.

Roberts, J., & Laughlin, J. (1996). A unidimensional item response model for unfolding responses from a graded disagree-agree response scale. *Applied Psychological Measurement, 20*(3), 231-255.

Roberts, J., Donoghue, J., & Laughlin, J. (2000). A general item response theory model for unfolding unidimensional polytomous responses. *Applied Psychological Measurement*, *24*(1), 3-32.

Roberts, J., Donoghue, J., & Laughlin, J. (2002). Characteristics of MML/EAP parameter estimates in the generalized graded unfolding model. *Applied Psychological Measurement*, *26*(2), 192-207.

Roberts, J., Laughlin, J., & Wedell, D. (1999). Validity issues in the Likert and Thurstone approaches to attitude measurement. *Educational and Psychological Measurement*, *59*(2), 211-233.

Rubin, D. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, *12*(4), 1151-1172.

Samejima, F. (1969). Estimation of latent ability using a response pattern of graded scores. *Psychometrika Monograph Supplement*, No. 17.

Scherbaum, C., Sabet, J., Kern, M., & Agnello, P. (2013). Examining faking on personality inventories using unfolding item response theory models. *Journal of Personality Assessment*, *95*(2), 207-216.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461-464.

Sinharay, S., Johnson, M., & Stern, H. (2006). Posterior predictive assessment of item response theory models. *Applied Psychological Measurement*, *30*(4), 298-321.

Stark, S., Chernyshenko, O., Drasgow, F., & Williams, B. (2006). Examining assumptions about item responding in personality assessment: Should ideal point methods be considered for scale development and scoring?. *Journal of Applied Psychology*, *91*(1), 25-39.

Tay, L., Ali, U., Drasgow, F., & Williams, B. (2011). Fitting IRT models to dichotomous and polytomous data: Assessing the relative model–data fit of ideal point and dominance models. *Applied Psychological Measurement*, *35*(4), 280-295.

Tay, L., Meade, A., & Cao, M. (2014). An overview and practical guide to IRT measurement equivalence analysis. *Organizational Research Methods*, *18*(1), 3-46.

Tendeiro, J. N., & Castro-Alvarez, S. (2020). GGUM: Generalized graded unfolding model. R package version 0.4-1. https://CRAN.R-project.org/package=GGUM

Thissen, D., Pommerich, M., Billeaud, K., & Williams, V. (1995). Item response theory for scores on tests including polytomous items with ordered responses. *Applied Psychological Measurement*, *19*(1), 39-49.

Thurstone, L. (1928). Attitudes can be measured. *American Journal of Sociology*, *33*(4), 529-554.

Traub, R. (2005). Classical test theory in historical perspective. *Educational Measurement: Issues And Practice*, *16*(4), 8-14.

Van den Wollenberg, A. (1982). Two new test statistics for the Rasch model. *Psychometrika*, *47*(2), 123-140.

Wang, W., & Wu, S. (2015). Confirmatory multidimensional IRT unfolding models for graded-response items. *Applied Psychological Measurement*, *40*(1), 56-72.

Wright, B. (1979). *Best test design*. Chicago, IL: Mesa Press.

Wright, B., & Masters, G. (1982). *Rating scale analysis*. Chicago, IL: Mesa Press.

Xu, J., Paek, I., & Xia, Y. (2017). Investigating the behaviors of M2 and RMSEA2 in fitting a unidimensional model to multidimensional data. *Applied Psychological Measurement*, *41*(8), 632-644.

Yen, W. (1981). Using simulation results to choose a latent trait model. *Applied Psychological Measurement*, *5*(2), 245-262.

APPENDIX A

R Code

Starting Codes for the GGUM Package

GGUM package data generation code *GEN.D1* for δs ranging from a [-3, 3] uniform distribution. For a uniform distribution with δs ranging from [-2, 2], the *italicized* line of code below is replaced by *`delta <- sort(round(runif(I, -2, 2), 4))`*. For a uniform distribution with δs ranging from [-2, 2] but not including [-1, 1], the *italicized* line if code below is replaced by *`delta <- sort(round(c(runif(I2, -2, -1),runif(I2, 1, 2)), 4))`*, with I2 equal to the number of items divided by 2. The rest of the code is identical to that from the GGUM package by Tendeiro and Castro-Alvarez (2020).

```
GEN.D1<-function (N, I, C, model = "GGUM", seed = 2875)
{
  set.seed(seed)
  if (model == "GGUM")
    alpha <- round(runif(I, 0.5, 2), 4)
  if (model == "GUM")
    alpha <- rep(1, I)
  delta <- sort(round(runif(I, -3, 3), 4))
  if (length© == 1)
    C <- rep(C, I)
  C.max <- max©
  if (model == "GGUM") {
    tau.half <- matrix(NA, nrow = I, ncol = C.max)
    tau.half[, 1] <- round(runif(I, 0.4, 1.4), 4)
    if (C.max >= 2) {
```

```
      for (i in 2:C.max) {
        tau.half[, i] <- (i <= C) * (tau.half[, i -
                                                1] + 0.25 +
round(rnorm(I, 0, 0.04), 4))
      }
    }
    taus <- cbind(-tau.half[, C.max:1], 0, tau.half)
  }
  if (model == "GUM") {
    tau.half <- rep(NA, C.max)
    tau.half[1] <- round(runif(1, 0.4, 1), 4)
    if (C.max >= 2) {
      for (i in 2:C.max) {
        tau.half[i] <- tau.half[i - 1] + 0.25 + round(rnorm(1,
                                                            0,
0.04), 4)
      }
    }
    taus <- c(0, tau.half)
    taus <- matrix(rep(taus, I), nrow = I, byrow = TRUE)
    for (i in 1:I) {
      if (C[i] < C.max)
        taus[i, (C[i] + 2):(C.max + 1)] <- 0
    }
    taus <- cbind(-taus[, (C.max + 1):2], taus)
  }
  theta <- round(rnorm(N, 0, 1), 4)
  M <- 2 * C + 1
  probs.array <- array(NA, dim = c(N, I, C.max + 1))
  for (z in 0:C.max) {
    probs.array[, , z + 1] <- P.GGUM(z, alpha, delta, taus,
```

```
                                        theta, C)
  }
  res <- apply(probs.array, 1:2, function(vec)
which(rmultinom(1,

1, vec) == 1) - 1)
  return(list(alpha.gen = alpha, delta.gen = delta, taus.gen =
taus,
              theta.gen = theta, data = res))
}
```

To obtain the P.GGUM function above, the original source code from the GGUM package has to run first. The following syntax corresponds to Tendeiro and Castro-Alvarez (2020) source code:

```
# GPCM (base for GUM and GGUM) ----
#    y     : Either scalar (whose value is then replicated I
times), or vector
#            of length I
#    alpha : Vector of length I
#    delta : Vector of length I
#    taus  : Either vector of length M (which is then replicated
I times), or
#           (generalized) matrix I x max(M)
#    theta : Vector of length N
#    M     : Either scalar (whose value is then replicated I
times), or vector
#            of length I
#
# GPCM applies to GGUM in its most general form.

# P.GPCM ----
P.GPCM <- function(y, alpha, delta, taus, theta, M)
```

```
{
  N         <- length(theta)
  I         <- length(delta)
  if (length(y) == 1) y <- rep(y, I)
  if (is.vector(taus)) taus <- matrix(rep(taus, I), nrow = I,
byrow = TRUE)
  if (length(M) == 1) M <- rep(M, I)
  taus.zero <- cbind(0, taus)
  taus.cum  <- t(apply(taus.zero, 1, cumsum))
  part      <- function(i, w)
  {
    if ((0 <= w) && (w <= M[i])) {
      exp(alpha[i] * (w * (theta - delta[i]) - taus.cum[i,
((max(M) - M[i])/2) + w + 1]))
    } else rep(0, N)
  }
  num       <- sapply(1:I,      function(i) part(i, y[i]),
simplify = "array")
  tmp       <- sapply(0:max(M), function(w) sapply(1:I,
function(i) part(i, w)))
  den       <- matrix(rowSums(tmp, na.rm = TRUE), ncol = I,
byrow = FALSE)
  return(num / den)
}


# P.GGUM ----
P.GGUM <- function(z, alpha, delta, taus, theta, C)
{
  N         <- length(theta)
  I         <- length(delta)
  if (length(z) == 1) z <- rep(z, I)
```

```
  if (is.vector(taus)) taus <- matrix(rep(taus, I), nrow = I,
byrow = TRUE)

  if (length(C) == 1) C <- rep(C, I)

  M         <- 2 * C + 1

  mat.ind   <- matrix(rep(z <= C, N), nrow = N, byrow = TRUE)

  return( mat.ind * (P.GPCM(z, alpha, delta, taus, theta, M) +
P.GPCM(M - z, alpha, delta, taus, theta, M)) )

}


# probs.GGUM ----

#' @title Compute model probabilities for the GGUM

#'

#' @description \code{probs.GGUM} computes model probabilities
for the GGUM (and

#'   the GUM) for given item and person parameters.

#'

#' @param alpha A vector of length \eqn{I} with the
discrimination parameters.

#' @param delta A vector of length \eqn{I} with the difficulty
parameters.

#' @param taus An \eqn{I\times M}{IxM} matrix with the threshold
parameters

#'   (\eqn{M = 2\times\max{C}+1}{M = 2*max(C)+1}).

#' @param theta A vector of length \eqn{N} with the person
parameters.

#' @param C \eqn{C} is the number of observable response
categories minus 1

#'   (i.e., the item scores will be in the set \eqn{\{0, 1, ...,
C\}}). It

#'   should either be a vector of \eqn{I} elements or a scalar.
In the latter

#'   case, it is assumed that \eqn{C} applies to all items.

#'
```

```
#' @return The function returns an \eqn{N\times I\times
K}{NxIxK} array with the

#'   GGUM probabilities, with \eqn{K=\max{C}+1}{K=max(C)+1}. To
retrieve the

#'   GUM-based probabilities just constrain alpha to a unit
vector of length {I}

#'   (i.e., \code{alpha = rep(1, I)}). In this case, make sure
\code{C} is

#'   constant across items.

#'

#' @section Details: This function computes the GGUM-based
probabilities for all

#'   (person, item, response category) combinations. For the
GGUM formula see

#'   the help for function \code{GGUM}
(\code{\link[GGUM]{GGUM}}).

#'

#' @author Jorge N. Tendeiro, \email{j.n.tendeiro@rug.nl}

#'

#' @examples

#' C <- c(3, 3, 3, 5, 5)

#' gen <- GenData.GGUM(10, 5, C, seed = 456)

#' gen.alpha <- gen$alpha.gen

#' gen.delta <- gen$delta.gen

#' gen.taus  <- gen$taus.gen

#' gen.theta <- gen$theta.gen

#'

#' # Compute model probabilities for the parameters above:

#' Ps <- probs.GGUM(gen.alpha, gen.delta, gen.taus, gen.theta,
C)

#' Ps

#' # In particular, the sum of the probabilities across all
response options
```

```
#' # (i.e., the third dimension) should be 1 for all (person,
item) combinations:
#' apply(Ps, 1:2, sum)
#' @export
probs.GGUM <- function(alpha, delta, taus, theta, C)
{
  # Sanity check - parameters:
  Sanity.params(alpha, delta, taus, theta, C)


  N     <- length(theta)
  I     <- length(alpha)
  C.max <- max(C)
  res <- array(0, dim = c(N, I, C.max + 1))
  for (c in 0:C.max) res[, , c+1] <- P.GGUM(c, alpha, delta,
taus, theta, C)
  dimnames(res)[[1]] <- paste0("N", 1:N)
  dimnames(res)[[2]] <- paste0("I", 1:I)
  dimnames(res)[[3]] <- paste0("C=", 0:C.max)
  return(res)
}


# P.GRM ----
P.GRM <- function(C, IP, theta)
{
  N         <- length(theta)
  I         <- nrow(IP)
  alpha     <- IP[, ncol(IP)]
  betas     <- IP[, -ncol(IP)]
  res.cum   <- array(NA, c(N, I, C))
  for (i in 1:I)
  {
```

```
    for (c in 1:C)
    {
      arg              <- alpha[i] * (theta - betas[i, c])
      res.cum[ , i, c] <- exp(arg) / (1 + exp(arg))
    }
  }
  res.cum <- array(c(matrix(1, N, I), res.cum, matrix(0, N, I)),
dim = c(N, I, C + 2))
  res     <- array(NA, c(N, I, C + 1))
  for (c in 1:(C + 1)) res[, , c] <- res.cum[, , c] - res.cum[,
, c + 1]
  return(res)
}
```

Simulation Codes for Absolute Fit Indices

Fitting the GGUM model to GGUM dichotomous data and testing for *Adjusted Chi-square* χ^2 fit statistics via the GGUM Package.

```
# First, make sure to run the source code for GEN.D1, which is
an edited code from the GGUM package utilizing a different delta
distribution (Tendeiro & Castro-Alvarez, 2020) #
GGUM_fit_GGUM_Dich_Data <- function(F, N, I)
{

# F = # of replications, N = sample size, I = # of items #
library(GGUM)
# C = # of response categories - 1 #
C <- 1
# Generate GGUM data after adjusting item location & Tau #
list1 <- vector("list", length=F)
set.seed(2875)
```

```
for (i in 1:F) {

    list1[[i]] <- GEN.D1(N, I, C, "GGUM", seed = sample(1:50000,
1, replace = FALSE))

}

# Subset the response matrices from list 1 #

list2 <- vector("list", length=F)

for (i in 1:F) {

  list2[[i]] <- subset(list1[[i]][["data"]])

}

# Fit the GGUM model to the generated data #

list3 <- vector("list", length=F)

for (i in 1:F) {

  list3[[i]] <- GGUM(list2[[i]], 1, N.nodes = 60, max.outer =
200, max.inner = 30)

}

# Calculating model fit using Adj Chi-square statistics #

list4 <- vector("list", length=F)

for (i in 1:F) {

  list4[[i]] <- MODFIT(list3[[i]])

}

# Tabulating results and calculating proportion of Type I error
and power #

x <-
data.frame("sin"=double(),"Dob"=double(),"Tri"=double(),"SinT"=d
ouble(),"DobT"=double(),"TriT"=double())

for (i in 1:F) {

  x[i,1]<- list4[[i]][["Summary.table"]][1,8]

}

for (i in 1:F) {

  x[i,2]<- list4[[i]][["Summary.table"]][2,8]

}

for (i in 1:F) {
```

```
  x[i,3]<- list4[[i]][["Summary.table"]][3,8]
}
x[,4:6] <- ifelse(x[,1:3]>= 3,1,0)
y<-
data.frame("Singlets"=mean(x[,4]),"Doublets"=mean(x[,5]),"Triple
ts"=mean(x[,6]))
list_final <- list(list1, list2, list3, list4, x, y)
return(list_final)
}
```

Fitting the GGUM model to dominance IRT data models (i.e., 2PL & 3PL) and testing for *Adjusted Chi-square* χ^2 fit statistics via the GGUM package.

```
GGUM_fit_2PL_Data <- function(F, N, I)
{
# F = # of replications, N = sample size, I = # of items #
  library(catIrt)
  library(GGUM)
  # Generating item parameters, for 3PL, replace c = 0 with c =
runif(I, 0, 0.3) #
  list1 <- vector("list", length=F)
  set.seed(2875)
  for (i in 1:F) {
    list1[[i]] <- cbind(a = (rlnorm(I, meanlog = 0, sdlog =
0.5))/1.702, b = runif(I, -2, 2), c = 0)
  }
# Simulating 2PL IRT response data using the catIRT package and
subsetting the data into a list #
  list2 <- vector("list", length=F)
  for (i in 1:F) {
    list2[[i]] <- simIrt(theta = rnorm(N),  params = list1[[i]],
mod = "brm")
```

```
  }
  list3 <- vector("list", length=F)
  for (i in 1:F) {
    list3[[i]] <- subset(list2[[i]][["resp"]])
  }
  # Fitting the GGUM model to the generated 2PL data #
  list4 <- vector("list", length=F)
  for (i in 1:F) {
    list4[[i]] <- GGUM(list3[[i]], 1, N.nodes = 60, max.outer =
200, max.inner = 30)
  }
  # Estimating model fit using adjusted chi-square indices for
item singles, double, triples #
  list5 <- vector("list", length=F)
  for (i in 1:F) {
    list5[[i]] <- MODFIT(list4[[i]])
  }
  # Indexing values of results and tabulating the proportion of
Type I error & power #
  x <-
data.frame("sin"=double(),"Dob"=double(),"Tri"=double(),"SinT"=d
ouble(),"DobT"=double(),"TriT"=double())
  for (i in 1:F) {
    x[i,1]<- list5[[i]][["Summary.table"]][1,8]
  }
  for (i in 1:F) {
    x[i,2]<- list5[[i]][["Summary.table"]][2,8]
  }
  for (i in 1:F) {
    x[i,3]<- list5[[i]][["Summary.table"]][3,8]
  }
  x[,4:6] <- ifelse(x[,1:3]>= 3,1,0)
```

```
  y<-
data.frame("Singlets"=mean(x[,4]),"Doublets"=mean(x[,5]),"Triple
ts"=mean(x[,6]))


  list_final <- list(list1, list2, list3, list4, list5, x, y)


  return(list_final)
}
```

Fitting the GGUM model to GGUM dichotomous data and testing for *Q1*, $S - X^2$, G^2, and *SRMSR* fit statistics via the mirt Package.

```
# First, make sure to run the source code for GEN.D1, which is

an edited code from the GGUM package utilizing a different delta

distribution (Tendeiro & Castro-Alvarez, 2020) #


mirt_fit_GGUM_Dich_Data <- function(F, N, I)

{

  # F = # of replications, N = sample size, I = # of items.
Also, manually adjusting the code under list3 for the number of
items is required #

  # Seed number can be edited in the set.seed command under
list1

library(mirt)

library(GGUM)

# C = # of response categories - 1 #

C <- 1

# Generating item parameters and GGUM response data #

list1 <- vector("list", length=F)

set.seed(2875)

for (i in 1:F) {
```

```
  list1[[i]] <- GEN.D1(N, I, C, "GGUM", seed = sample(1:50000,
1, replace = FALSE))

}

# subsetting the GGUM data into a list #

list2 <- vector("list", length=F)

for (i in 1:F) {

  list2[[i]] <- subset(list1[[i]][["data"]])

}

# Fitting the GGUM model to GGUM data with 20 items. for x
number of items, replace  Model.Dich<-'F1=1-x' & paste0("Item",
1:x) #

list3 <- vector("list", length=F)

for (i in 1:F) {

  Model.Dich<-'F1=1-20'

  colnames(list2[[i]]) <- paste0("Item", 1:20)

  list3[[i]]<-mirt(list2[[i]], model = Model.Dich,itemtype =
"ggum", method = "EM", dentype = 'empiricalhist', TOL = 0.001,
quadpts = 60, technical = list(NCYCLES = 10000))

}

# SRMSR Statistic ----------- #

list4 <- vector("list", length=F)

for (i in 1:F) {

  list4[[i]] <- M2(list3[[i]])

}

# Q1 Statistic -------------- #

list5 <- vector("list", length=F)

for (i in 1:F) {

  list5[[i]] <- itemfit(list3[[i]], 'X2', group.bins = 10)

}

list6 <- vector("list", length=F)

for (i in 1:F) {

  list6[[i]] <- ifelse(list5[[i]][,5]<= 0.05,1,0)
```

```
}
# S-X2 Statistic -------------- #
list7 <- vector("list", length=F)
for (i in 1:F) {
 list7[[i]] <- itemfit(list3[[i]], 'S_X2')
}
list8 <- vector("list", length=F)
for (i in 1:F) {
  list8[[i]] <- ifelse(list7[[i]][,5]<= 0.05,1,0)
}
# G2 Statistic -------------- #
list9 <- vector("list", length=F)
for (i in 1:F) {
  list9[[i]] <- itemfit(list3[[i]], 'G2')
}
list10 <- vector("list", length=F)
for (i in 1:F) {
  list10[[i]] <- ifelse(list9[[i]][,5]<= 0.05,1,0)
}
# Tabulation and proportions ---------- #
x <-data.frame("SRMSR"=double(),"Q1"=double(),"S-
X2"=double(),"G2"=double(),
               "SRMSR_P"=double())
for (i in 1:F) {
  x[i,1]<- list4[[i]][1,7]
}
for (i in 1:F) {
  x[i,2]<- mean(list6[[i]])
}
for (i in 1:F) {
```

```
  x[i,3]<- mean(list8[[i]])
}
for (i in 1:F) {
  x[i,4]<- mean(list10[[i]])
x[,5] <- ifelse(x[,1]>= 0.05,1,0)
y<-data.frame("SRMSR"=mean(x[,5]),"Q1"=mean(x[,2], na.rm =
TRUE),"S-X2"=mean(x[,3], na.rm = TRUE), "G2"=mean(x[,4], na.rm =
TRUE))
list_final <- list(list1, list2, list3, list4, list5, list6,
list7, list8, list9, list10, x, y)
return(list_final)
}
```

Fitting the GGUM model to dominance IRT data models (i.e., 2PL & 3PL) and testing for *Q1*, *S* – X^2, G^2, and *SRMSR* fit statistics via the mirt Package.

```
mirt_fit_2PL_Data <- function(F, N, I)
{
# F = # of replications, N = sample size, I = # of items. Also,
manually adjusting the code under list4 for the number of items
is required #
# Seed number can be edited in the set.seed command under list1
library(mirt)
library(catIrt)
# Generating item parameters, for 3PL, replace c = 0 with c =
runif(I, 0, 0.3)  #
list1 <- vector("list", length=F)
set.seed(2875)
for (i in 1:F) {
  list1[[i]] <- cbind(a = (rlnorm(I, meanlog = 0, sdlog =
0.5))/1.702, b = runif(I, -2, 2), c = 0)
}
```

```
# Simulating 2PL IRT response data using the catIRT package and
subsetting the data into a list #

list2 <- vector("list", length=F)

for (i in 1:F) {

  list2[[i]] <- simIrt(theta = rnorm(N),  params = list1[[i]],
mod = "brm")

}

list3 <- vector("list", length=F)

for (i in 1:F) {

  list3[[i]] <- subset(list2[[i]][["resp"]])

}

# Fitting the GGUM model to 2PL data with 20 items. for x number
of items, replace  Model.Dich<-'F1=1-x' & paste0("Item", 1:x) #

# for a Gaussian density type, replace "dentype =
'empiricalhist'" with 'dentype = 'Gaussian'" #

list4 <- vector("list", length=F)

for (i in 1:F) {

  Model.Dich<-'F1=1-20'

  colnames(list3[[i]]) <- paste0("Item", 1:20)

  list4[[i]]<-mirt(list3[[i]], model = Model.Dich,itemtype =
"ggum", method = "EM", dentype = 'empiricalhist', TOL = 0.001,
quadpts = 60, technical = list(NCYCLES = 10000))

}

# SRMSR Statistic ----------- #

list5 <- vector("list", length=F)

for (i in 1:F) {

  list5[[i]] <- M2(list4[[i]])

}

# Q1 Statistic -------------- #

list6 <- vector("list", length=F)

for (i in 1:F) {

  list6[[i]] <- itemfit(list4[[i]], 'X2', group.bins = 10)
```

```
}
list7 <- vector("list", length=F)
for (i in 1:F) {
  list7[[i]] <- ifelse(list6[[i]][,5]<= 0.05,1,0)
}
# S-X2 Statistic -------------- #
list8 <- vector("list", length=F)
for (i in 1:F) {
  list8[[i]] <- itemfit(list4[[i]], 'S_X2')
}
list9 <- vector("list", length=F)
for (i in 1:F) {
  list9[[i]] <- ifelse(list8[[i]][,5]<= 0.05,1,0)
}
# G2 Statistic -------------- #
list10 <- vector("list", length=F)
for (i in 1:F) {
  list10[[i]] <- itemfit(list4[[i]], 'G2')
}
list11 <- vector("list", length=F)
for (i in 1:F) {
  list11[[i]] <- ifelse(list10[[i]][,5]<= 0.05,1,0)
}
# Tabulation and proportions ---------- #
x <-data.frame("SRMSR"=double(),"Q1"=double(),"S-
X2"=double(),"G2"=double(),
               "SRMSR_P"=double())
for (i in 1:F) {
  x[i,1]<- list5[[i]][1,7]
}
```

```
for (i in 1:F) {
  x[i,2]<- mean(list7[[i]])
}
for (i in 1:F) {
  x[i,3]<- mean(list9[[i]])
}
for (i in 1:F) {
  x[i,4]<- mean(list11[[i]])
}
x[,5] <- ifelse(x[,1]>= 0.05,1,0)
y<-data.frame("SRMSR"=mean(x[,5]),"Q1"=mean(x[,2], na.rm =
TRUE),"S-X2"=mean(x[,3], na.rm = TRUE), "G2"=mean(x[,4], na.rm =
TRUE))
list_final <- list(list1, list2, list3, list4, list5, list6,
list7, list8, list9, list10, list11, x, y)
return(list_final)
}
```

Fitting the GGUM model to GGUM polytomous data and testing for *Adjusted Chi-square* χ^2 fit statistics via the GGUM Package.

```
# First, make sure to run the source code for GEN.D1, which is
an edited code from the GGUM package utilizing a different delta
distribution (Tendeiro & Castro-Alvarez, 2020) #
GGUM_fit_GGUM_Poly_Data <- function(F, N, I)
{
  # F = # of replications, N = sample size, I = # of items #
  library(GGUM)
  # C = # of response categories - 1 #
  C <- 3
  # Generate GGUM data after adjusting item location & Tau #
  list1 <- vector("list", length=F)
```

```
  set.seed(2875)
  for (i in 1:F) {
    list1[[i]] <- GEN.D1(N, I, C, "GGUM", seed = sample(1:50000,
1, replace = FALSE))
  }
  # Subset the response matrices from list 1 #
  list2 <- vector("list", length=F)
  for (i in 1:F) {
    list2[[i]] <- subset(list1[[i]][["data"]])
  }
  # Fit the GGUM model to the generated data #
  list3 <- vector("list", length=F)
  for (i in 1:F) {
    list3[[i]] <- GGUM(list2[[i]], C, N.nodes = 60, max.outer =
200, max.inner = 30)
  }
  # Calculating model fit using Adj Chi-square statistics #
  list4 <- vector("list", length=F)
  for (i in 1:F) {
    list4[[i]] <- MODFIT(list3[[i]])
  }
  # Tabulating results and calculating proportion of Type I
error and power #
  x <-
data.frame("sin"=double(),"Dob"=double(),"Tri"=double(),"SinT"=d
ouble(),"DobT"=double(),"TriT"=double())
  for (i in 1:F) {
    x[i,1]<- list4[[i]][["Summary.table"]][1,8]
  }
  for (i in 1:F) {
    x[i,2]<- list4[[i]][["Summary.table"]][2,8]
  }
```

```
  for (i in 1:F) {
    x[i,3]<- list4[[i]][["Summary.table"]][3,8]
  x[,4:6] <- ifelse(x[,1:3]>= 3,1,0)
  y<-
data.frame("Singlets"=mean(x[,4]),"Doublets"=mean(x[,5]),"Triple
ts"=mean(x[,6]))
  list_final <- list(list1, list2, list3, list4, x, y)
  return(list_final)
}
```

Fitting the GGUM model to the GRM data model and testing for *Adjusted Chi-square* χ^2 fit statistics via the GGUM package.

```
GGUM_fit_GRM_Data <- function(F, N, I)
{
# F = # of replications, N = sample size, I = # of items #
 library(catIrt)
  library(GGUM)
  # Generating item parameters #
  list1 <- vector("list", length=F)
  set.seed(2875)
  for (i in 1:F) {
    list1[[i]] <- cbind(a = (rlnorm(I, meanlog = 0, sdlog =
0.5))/1.702, b1 = runif(I, -2, -0.5), b2 = runif(I, -0.5, 0.5),
                        b3 = runif(I, 0.5, 2))
  }
  # Simulating GRM IRT response data using the catIRT package
and subsetting the data into a list by adjusting the responses
to range from category 0 to 3 #
  list2 <- vector("list", length=F)
  for (i in 1:F) {
```

```
    list2[[i]] <- simIrt(theta = rnorm(N),  params = list1[[i]],
mod = "brm")
  }
  list3 <- vector("list", length=F)
  for (i in 1:F) {
    list3[[i]] <- subset(list2[[i]][["resp"]]-1)
  }
  # Fitting the GGUM model to the generated GRM data #
  list4 <- vector("list", length=F)
  for (i in 1:F) {
    list4[[i]] <- GGUM(list3[[i]], 3, N.nodes = 60, max.outer =
200, max.inner = 30)
  }
  # Estimating model fit using adjusted chi-square indices for
item singles, double, triples #
  list5 <- vector("list", length=F)
  for (i in 1:F) {
    list5[[i]] <- MODFIT(list4[[i]])
  }
  # Indexing values of results and tabulating the proportion of
Type I error & power #
  x <-
data.frame("sin"=double(),"Dob"=double(),"Tri"=double(),"SinT"=d
ouble(),"DobT"=double(),"TriT"=double())
  for (i in 1:F) {
    x[i,1]<- list5[[i]][["Summary.table"]][1,8]
  }
  for (i in 1:F) {
    x[i,2]<- list5[[i]][["Summary.table"]][2,8]
  }
  for (i in 1:F) {
    x[i,3]<- list5[[i]][["Summary.table"]][3,8]
```

```
  }
  x[,4:6] <- ifelse(x[,1:3]>= 3,1,0)
  y<-
data.frame("Singlets"=mean(x[,4]),"Doublets"=mean(x[,5]),"Triple
ts"=mean(x[,6]))
  list_final <- list(list1, list2, list3, list4, x, y)
  return(list_final)
}
```

Fitting the GGUM model to GGUM polytomous data and testing for *Q1*, $S - X^2$, G^2, and *SRMSR* fit statistics via the mirt Package.

```
# First, make sure to run the source code for GEN.D1, which is
an edited code from the GGUM package utilizing a different delta
distribution (Tendeiro & Castro-Alvarez, 2020) #
mirt_fit_GGUM_Poly_Data <- function(F, N, I)
{
  # F = # of replications, N = sample size, I = number of items.
Also, manually adjusting the code under list3 for the number of
items is required #
  # Seed number can be edited in the set.seed command under
list1
  library(mirt)
  library(GGUM)
  # C = # of response categories - 1 #
  C <- 3
  # Generating item parameters and GGUM response data #
  list1 <- vector("list", length=F)
  set.seed(2875)
  for (i in 1:F) {
    list1[[i]] <- GEN.D1(N, I, C, "GGUM", seed = sample(1:50000,
1, replace = FALSE))
```

```
  }
  # subsetting the GGUM data into a list #
  list2 <- vector("list", length=F)
  for (i in 1:F) {
    list2[[i]] <- subset(list1[[i]][["data"]])
  }
  # Fitting the GGUM model to GGUM data with 20 items. for x
number of items, replace  Model.Dich<-'F1=1-x' & paste0("Item",
1:x) #
  list3 <- vector("list", length=F)
  for (i in 1:F) {
    Model.Poly<-'F1=1-20'
    colnames(list2[[i]]) <- paste0("Item", 1:20)
    list3[[i]]<-mirt(list2[[i]], model = Model.Poly,itemtype =
"ggum", method = "EM", dentype = 'empiricalhist', TOL = 0.001,
quadpts = 60, technical = list(NCYCLES = 10000))
  }
  # SRMSR Statistic ----------- #
  list4 <- vector("list", length=F)
  for (i in 1:F) {
    list4[[i]] <- M2(list3[[i]])
  }
  # Q1 Statistic -------------- #
  list5 <- vector("list", length=F)
  for (i in 1:F) {
    list5[[i]] <- itemfit(list3[[i]], 'X2', group.bins = 10)
  }
  list6 <- vector("list", length=F)
  for (i in 1:F) {
    list6[[i]] <- ifelse(list5[[i]][,5]<= 0.05,1,0)
  }
```

```
  # S-X2 Statistic -------------- #
  list7 <- vector("list", length=F)
  for (i in 1:F) {
    list7[[i]] <- itemfit(list3[[i]], 'S_X2')
  }
  list8 <- vector("list", length=F)
  for (i in 1:F) {
    list8[[i]] <- ifelse(list7[[i]][,5]<= 0.05,1,0)
  }
  # G2 Statistic -------------- #
  list9 <- vector("list", length=F)
  for (i in 1:F) {
    list9[[i]] <- itemfit(list3[[i]], 'G2')
  }
  list10 <- vector("list", length=F)
  for (i in 1:F) {
    list10[[i]] <- ifelse(list9[[i]][,5]<= 0.05,1,0)
  }
  # Tabulation and proportions ---------- #
  x <-data.frame("SRMSR"=double(),"Q1"=double(),"S-
X2"=double(),"G2"=double(),
                 "SRMSR_P"=double())
  for (i in 1:F) {
    x[i,1]<- list4[[i]][1,7]
  }
  for (i in 1:F) {
    x[i,2]<- mean(list6[[i]])
  }
  for (i in 1:F) {
    x[i,3]<- mean(list8[[i]])
```

```
  }
  for (i in 1:F) {
    x[i,4]<- mean(list10[[i]])
  }
  x[,5] <- ifelse(x[,1]>= 0.05,1,0)
  y<-data.frame("SRMSR"=mean(x[,5]),"Q1"=mean(x[,2], na.rm =
TRUE),"S-X2"=mean(x[,3], na.rm = TRUE), "G2"=mean(x[,4], na.rm =
TRUE))
  list_final <- list(list1, list2, list3, list4, list5, list6,
list7, list8, list9, list10, x, y)
  return(list_final)
}
```

Fitting the GGUM model to GRM data and testing for *Q1*, $S - X^2$, G^2, and *SRMSR* fit statistics via the mirt Package.

```
mirt_fit_GRM_Data <- function(K, N, I)
{
  # K = # of replications, N = sample size, I = number of items.
Also, manually adjusting the code under list4 for the number of
items is required #
  # Seed number can be edited in the set.seed command under
list1
  library(mirt)
  library(catIrt)
  # Generating item parameters #
list1_GRM <- vector("list", length=K)
set.seed(2875)
for (i in 1:K) {
  list1_GRM[[i]] <- cbind(a = (rlnorm(I, meanlog = 0, sdlog =
0.5))/1.702, b1 = runif(I, -2, -0.5), b2 = runif(I, -0.5, 0.5),
                          b3 = runif(I, 0.5, 2))
```

```
}

# Simulating GRM IRT response data using the catIRT package and
subsetting the data into a list by adjusting the responses to
range form category 0 to 3 #

list2_GRM <- vector("list", length=K)

for (i in 1:K) {

  list2_GRM[[i]] <- simIrt(theta = rnorm(N),  params =
list1_GRM[[i]], mod = "grm")

}

list3_GRM <- vector("list", length=K)

for (i in 1:K) {

  list3_GRM[[i]] <- subset((list2_GRM[[i]][["resp"]])-1)

}

# Fitting the GGUM model to 3PL data with 20 items. for x number
of items, replace  Model.Dich<-'F1=1-x' & paste0("Item", 1:x) #

list4 <- vector("list", length=K)

for (i in 1:K) {

  Model.Poly<-'F1=1-20'

  colnames(list3_GRM[[i]]) <- paste0("Item", 1:20)

  list4[[i]]<-mirt(list3_GRM[[i]], model = Model.Poly,itemtype =
"ggum", method = "EM", dentype = 'empiricalhist', TOL = 0.001,
quadpts = 60, technical = list(NCYCLES = 10000))

}

# SRMSR Statistic ----------- #

list5 <- vector("list", length=K)

for (i in 1:K) {

  list5[[i]] <- M2(list4[[i]])

}

# Q1 Statistic -------------- #

list6 <- vector("list", length=K)

for (i in 1:K) {

  list6[[i]] <- itemfit(list4[[i]], 'X2', group.bins = 10)
```

```
}
list7 <- vector("list", length=K)
for (i in 1:K) {
  list7[[i]] <- ifelse(list6[[i]][,5]<= 0.05,1,0)
}
# S-X2 Statistic -------------- #
list8 <- vector("list", length=K)
for (i in 1:K) {
  list8[[i]] <- itemfit(list4[[i]], 'S_X2')
}
list9 <- vector("list", length=K)
for (i in 1:K) {
  list9[[i]] <- ifelse(list8[[i]][,5]<= 0.05,1,0)
}
# G2 Statistic -------------- #
list10 <- vector("list", length=K)
for (i in 1:K) {
  list10[[i]] <- itemfit(list4[[i]], 'G2')
}
list11 <- vector("list", length=K)
for (i in 1:K) {
  list11[[i]] <- ifelse(list10[[i]][,5]<= 0.05,1,0)
}
# Tabulation and proportions ---------- #
x <-data.frame("SRMSR"=double(),"Q1"=double(),"S-
X2"=double(),"G2"=double(),
               "SRMSR_P"=double())
for (i in 1:K) {
  x[i,1]<- list5[[i]][1,7]
}
```

```
for (i in 1:K) {
  x[i,2]<- mean(list7[[i]])
}
for (i in 1:K) {
  x[i,3]<- mean(list9[[i]])
}
for (i in 1:K) {
  x[i,4]<- mean(list11[[i]])
}
x[,5] <- ifelse(x[,1]>= 0.05,1,0)
y<-
data.frame("SRMSR"=mean(x[,5]),"Q1"=mean(x[,2],na.rm=TRUE),"S-
X2"=mean(x[,3],na.rm=TRUE), "G2"=mean(x[,4],na.rm=TRUE))
list_final <- list(list1, list2, list3, list4, list5, list6,
list7, list8, list9, list10, list11, x, y)
return(list_final)
}
```

Simulation Codes for Relative Fit Indices

AIC & BIC fit indices for dichotomous data.

```
# Open required files in the global environment and save the
following lists containing AIC and BIC indices in the working
directory #
saveRDS(list3, "GGUM.rds")
saveRDS(list4, "2PL.rds")
saveRDS(list4, "3PL.rds")
# read the following files in the global environment only #
GGUM_D <- readRDS("GGUM.rds")
twoPL <- readRDS("2PL.rds")
threePL <- readRDS("3PL.rds")
```

```
# specify the number of rows in the dataframe for subsequent
indexing #

K <- 100

# Tabulation and indexing using ifelese statements #

w <-
data.frame("AIC_GGUM"=double(),"AIC_2PL"=double(),"AIC_3PL"=doub
le(),


"BIC_GGUM"=double(),"BIC_2PL"=double(),"BIC_3PL"=double(),

               "AIC_GGUM_T"=double(), "AIC_2PL_T"=double(),
"AIC_3PL_T"=double(),

               "BIC_GGUM_T"=double(), "BIC_2PL_T"=double(),
"BIC_3PL_T"=double())

for (i in 1:K) {

  w[i,1]<- GGUM_D[[i]][["InformationCrit"]][1,3]

}

for (i in 1:K) {

  w[i,2]<- twoPL[[i]][["InformationCrit"]][1,3]

}

for (i in 1:K) {

  w[i,3]<- threePL[[i]][["InformationCrit"]][1,3]

}

for (i in 1:K) {

  w[i,4]<- GGUM_D[[i]][["InformationCrit"]][1,4]

}

for (i in 1:K) {

  w[i,5]<- twoPL[[i]][["InformationCrit"]][1,4]

}

for (i in 1:K) {

  w[i,6]<- threePL[[i]][["InformationCrit"]][1,4]

}
```

```
w$AIC_GGUM_T <- ifelse(w$AIC_GGUM < w$AIC_2PL & w$AIC_GGUM <
w$AIC_3PL, 1,0)

w$AIC_2PL_T <- ifelse(w$AIC_2PL < w$AIC_GGUM & w$AIC_2PL <
w$AIC_3PL, 1,0)

w$AIC_3PL_T <- ifelse(w$AIC_3PL < w$AIC_GGUM & w$AIC_3PL <
w$AIC_2PL, 1,0)

w$BIC_GGUM_T <- ifelse(w$BIC_GGUM < w$BIC_2PL & w$BIC_GGUM <
w$BIC_3PL, 1,0)

w$BIC_2PL_T <- ifelse(w$BIC_2PL < w$BIC_GGUM & w$BIC_2PL <
w$BIC_3PL, 1,0)

w$BIC_3PL_T <- ifelse(w$BIC_3PL < w$BIC_GGUM & w$BIC_3PL <
w$BIC_2PL, 1,0)

y<-data.frame("AIC_GGUM"=mean(w$AIC_GGUM_T),
"AIC_2PL"=mean(w$AIC_2PL_T), "AIC_3PL"=mean(w$AIC_3PL_T),

              "BIC_GGUM"=mean(w$BIC_GGUM_T),
"BIC_2PL"=mean(w$BIC_2PL_T),"BIC_3PL"=mean(w$BIC_3PL_T))
```

AIC & BIC fit indices for GGUM generated data using different delta ranges.

```
# Open required files in the global environment and save the
following lists containing AIC and BIC indices in the working
directory #

# These lists contain AIC and BIC results for GGUM datasets
utilizing different delta ranges for response generation #

saveRDS(list3, "GGUM.rds")

saveRDS(list3, "G.rds")

# read the following files in the global environment only #

GGUM_D <- readRDS("GGUM.rds")

G <- readRDS("G.rds")

# specify the number of rows in the dataframe for subsequent
indexing #

K <- 100

# Tabulation and indexing using ifelese statements #

x <-data.frame("AIC_GGUM"=double(),"AIC_G"=double(),
```

```
                "BIC_GGUM"=double(),"BIC_G"=double(),
                "AIC_GGUM_T"=double(), "AIC_G_T"=double(),
                "BIC_GGUM_T"=double(), "BIC_G_T"=double())
for (i in 1:K) {
  x[i,1]<- GGUM_D[[i]][["InformationCrit"]][1,3]
}
for (i in 1:K) {
  x[i,2]<- G[[i]][["InformationCrit"]][1,3]
}
for (i in 1:K) {
  x[i,3]<- GGUM_D[[i]][["InformationCrit"]][1,4]
}
for (i in 1:K) {
  x[i,4]<- G[[i]][["InformationCrit"]][1,4]
}
x$AIC_GGUM_T <- ifelse(x$AIC_GGUM < x$AIC_G, 1,0)
x$AIC_G_T <- ifelse(x$AIC_G < x$AIC_GGUM, 1,0)
x$BIC_GGUM_T <- ifelse(x$BIC_GGUM < x$BIC_G, 1,0)
x$BIC_G_T <- ifelse(x$BIC_G < x$BIC_GGUM, 1,0)
y<-data.frame("AIC_GGUM"=mean(x$AIC_GGUM_T),
"AIC_G"=mean(x$AIC_G_T),
              "BIC_GGUM"=mean(x$BIC_GGUM_T),
"BIC_G"=mean(x$BIC_G_T))
```

AIC & BIC fit indices for polytomous data.

```
# Open required files in the global environment and save the
following lists containing AIC and BIC indices in the working
directory #
saveRDS(list3, "GGUM.rds")
saveRDS(list4, "GRM.rds")
# read the following files in the global environment only #
```

```
GGUM_D <- readRDS("GGUM.rds")

GRM <- readRDS("GRM.rds")

# specify the number of rows in the dataframe for subsequent
indexing #

K <- 100

# Tabulation and indexing using ifelese statements #

x <-data.frame("AIC_GGUM"=double(),"AIC_GRM"=double(),

               "BIC_GGUM"=double(),"BIC_GRM"=double(),

               "AIC_GGUM_T"=double(), "AIC_GRM_T"=double(),

               "BIC_GGUM_T"=double(), "BIC_GRM_T"=double())

for (i in 1:K) {

  x[i,1]<- GGUM_D[[i]][["InformationCrit"]][1,3]

}

for (i in 1:K) {

  x[i,2]<- GRM[[i]][["InformationCrit"]][1,3]

}

for (i in 1:K) {

  x[i,3]<- GGUM_D[[i]][["InformationCrit"]][1,4]

}

for (i in 1:K) {

  x[i,4]<- GRM[[i]][["InformationCrit"]][1,4]

}

x$AIC_GGUM_T <- ifelse(x$AIC_GGUM < x$AIC_GRM, 1,0)

x$AIC_GRM_T <- ifelse(x$AIC_GRM < x$AIC_GGUM, 1,0)

x$BIC_GGUM_T <- ifelse(x$BIC_GGUM < x$BIC_GRM, 1,0)

x$BIC_GRM_T <- ifelse(x$BIC_GRM < x$BIC_GGUM, 1,0)

y<-data.frame("AIC_GGUM"=mean(x$AIC_GGUM_T),
"AIC_GRM"=mean(x$AIC_GRM_T),

              "BIC_GGUM"=mean(x$BIC_GGUM_T),
"BIC_GRM"=mean(x$BIC_GRM_T))
```

Additional Codes

This code provides an example of how to test for model fit using segmented runs (i.e., replications) for polytomous items when performed under a supercomputer. In this particular example, the *Adjusted Chi-square* χ^2 fit statistic is presented. However, the same process can be performed to test for the other fit indices in mirt.

```
# Open required files in the global environment and save the
following lists for subsequent model fit computation #
saveRDS(list4A, "GGUM_A.rds")
saveRDS(list4B, "GGUM_B.rds")
saveRDS(list4C, "GGUM_C.rds")
saveRDS(list4D, "GGUM_D.rds")
saveRDS(list4E, "GGUM_E.rds")
saveRDS(list4F, "GGUM_F.rds")
# read the following files in the global environment only #
L_A <- readRDS("GGUM_A.rds")
L_B <- readRDS("GGUM_B.rds")
L_C <- readRDS("GGUM_C.rds")
L_D <- readRDS("GGUM_D.rds")
L_E <- readRDS("GGUM_E.rds")
L_F <- readRDS("GGUM_F.rds")
F <- 100
# Index the read files into a single list #
list4 <- vector("list", length = 100)
list4<- c(L_A, L_B, L_C, L_D, L_E)
library(GGUM)
# Estimating model fit using adjusted chi-square indices for
item singles, double, triples #
list5 <- vector("list", length=F)
```

```
for (i in 1:F) {
  list5[[i]] <- MODFIT(list4[[i]])
}
# Indexing values of results and tabulating the proportion of
Type I error & power #
x <-
data.frame("sin"=double(),"Dob"=double(),"Tri"=double(),"SinT"=d
ouble(),"DobT"=double(),"TriT"=double())
for (i in 1:F) {
  x[i,1]<- list5[[i]][["Summary.table"]][1,8]
}
for (i in 1:F) {
  x[i,2]<- list5[[i]][["Summary.table"]][2,8]
}
for (i in 1:F) {
  x[i,3]<- list5[[i]][["Summary.table"]][3,8]
}
x[,4:6] <- ifelse(x[,1:3]>= 3,1,0)
y<-
data.frame("Singlets"=mean(x[,4]),"Doublets"=mean(x[,5]),"Triple
ts"=mean(x[,6]))
```

This code tests for the fit indices from both the GGUM and mirt packages when simulating GGUM data from the 'simdata' syntax in mirt. The comparison goes through 5 replications.

Table 14 values are directly obtained from this code.

```
library(mirt)
library(GGUM)
# F = # of replications, N = sample size, I = number of items, C
= # of response categories - 1 #
F <- 5
```

```
N <- 2000

I <- 20

C <- 3

# Generate discrimination parameters from a uniform distribution
#

lista <- vector("list", length=F)

for (i in 1:F) {

  lista[[i]] <- round(runif(I, 0.5, 2), 4)

}

# Generate item location parameters from a uniform distribution
#

listb <- vector("list", length=F)

for (i in 1:F) {

  listb[[i]] <- sort(round(runif(I, -3, 3), 4))

}

# Generate GGUM data seperately to using the GGUM package to
obtain generated taus #

list_GGUM_Gen <- vector("list", length=F)

for (i in 1:F) {

  list_GGUM_Gen[[i]] <- GEN.D1(N, I, C, "GGUM", seed =
sample(1:50000, 1, replace = FALSE))

}

# subset the positive taus from the previous list #

list_tau <- vector("list", length=F)

for (i in 1:F) {

  list_tau[[i]] <- list_GGUM_Gen[[i]][["taus.gen"]][,7:5]

}

# generate GGUM data in mirt using generated item discrimination
from lista, item location from listb, and generated taus from
the GGUM package #

list_dat <- vector("list", length=F)

for (i in 1:F) {
```

```
  list_dat[[i]] <- simdata(lista[[i]], listb[[i]], N, 'ggum',
t=list_tau[[i]])

}

# Calibrate the GGUM model using mirt #

list_mirt <- vector("list", length=F)

for (i in 1:F) {

  Model.Poly<-'F1=1-20'

  colnames(list_dat[[i]]) <- paste0("Item", 1:20)

  list_mirt[[i]] <- mirt(list_dat[[i]], model = Model.Poly,
'ggum', dentype = 'empiricalhist', TOL = 0.001, quadpts = 60,
technical = list(NCYCLES = 10000))

}

# calculate SRMSR fit statistic from the mirt package #

list_M2 <- vector("list", length=F)

for (i in 1:F) {

  list_M2[[i]] <- M2(list_mirt[[i]])

}

# calibrate the GGUM model using the the genrared data above #

list_GGUM_Fit <- vector("list", length=F)

for (i in 1:F) {

  list_GGUM_Fit[[i]] <- GGUM(list_dat[[i]], C)

}

# calculate the adjusted chi-sqaure statistics from the GGUM
package #

list_MODFIT <- vector("list", length=F)

for (i in 1:F) {

  list_MODFIT[[i]] <- MODFIT(list_GGUM_Fit[[i]])

}
```