Eurasian Journal of Educational Research 80 (2019) 171-192

Eurasian Journal of Educational Research
www.ejer.com.tr

Implementation of Cognitive Diagnosis Modeling using the GDINA R Package

Jimmy de la TORRE[1], Lokman AKBAY[2]

ARTICLE INFO

Article History:
Received: 28 Dec. 2018
Received in revised form: 27 Feb. 2019
Accepted: 08 Mar. 2019
DOI: 10.14689/ejer.2019.80.9

Keywords
CDM, cognitive diagnosis modeling, GDINA package, R implementation

ABSTRACT

Purpose: Well-designed assessment methodologies and various cognitive diagnosis models (CDMs) to extract diagnostic information about examinees' individual strengths and weaknesses have been developed. Due to this novelty, as well as educational specialists' lack of familiarity with CDMs, their applications are not widespread. This article aims at presenting the fundamentals of CDM and demonstrating the various implementations using a freeware R package, namely, the GDINA. Present article explains the basics of CDM and provide sufficient details on the implementations so that it may guide novice researchers in CDM applications.

Research Methods: The manuscript starts with presenting the CDM terminology, including input and output of a CDM analysis. The introduction section is followed by generalized deterministic noisy and gate model framework. A brief description of the package GDINA is also provided. Then, numerical examples on various CDM analyses are provided using the R package with a graphical user interface. The paper is concluded by some additional functions and concluding remarks. **Results and Implications for Research and Practice:** Although other software programs are also available, using the GDINA package offers users some flexibilities such as allowing estimation of a wide range of CDMs and allowing nonprogrammers to benefit from this package through the GUI. In addition to ordinary CDM analyses, GDINA package further allows users to apply model selection at the test- and item-level to make sure that the most appropriate CDM (i.e., CDM that best explains the attribute interactions in the item) is fitted to the response data. Furthermore, to identify possible item-attribute specification mistakes in the Q-matrix, implementation of an empirical Q-matrix validation method is available in the GDINA package. Lastly, this package offers various handy graphs, which can be very useful in emphasizing important information and comparing various parameters and/or statistics.

[1] The University of Hong Kong, HONG KONG, e-mail: j.delatorre@hku.hk, ORCID: https://orcid.org/0000-0002-0893-3863
[2] Corresponding author, Burdur Mehmet Akif Ersoy University, TURKEY, e-mail: lokmanakbay@gmail.com, ORCID: https://orcid.org/0000-0003-4026-5241

Introduction

Assessments, when purposefully designed, can provide opportunities for collecting relevant diagnostic information. Interpretations based on such information then enable educational specialists to take precise remedial actions. Recently developed formative assessments are expected to provide students and teachers with detailed feedback on what students are able to do yielding information that can optimize instruction and learning. In other words, a formative assessment should identify individual strengths and weaknesses in a particular content, which results in enhanced teaching and learning environment (DiBello & Stout, 2007). For formative assessment to fulfil this task, in addition to well-designed assessment methodologies, various cognitive diagnosis models (CDMs) to extract diagnostic information from diagnostic assessments are needed. These models are regarded as latent class models, which can be used to detect mastery and nonmastery of multiple fine-grained skills or attributes in a particular content domain (de la Torre, 2009).

Although there are several software programs available to estimate CDMs, there are considerable amounts of benefits of using R for estimation purposes: (1) Conducting estimation of various CDMs are available in R; (2) Unlike the R packages, many software programs handle only one type of CDM (e.g., MDLTM for general diagnostic model [GDM; von Davier, 2006], Arpeggio Suite for noncompensatory-RUM [NC-RUM; Hartz, 2002], Mplus for log-linear CDM [LCDM; Henson, Templin & Willse, 2009]); (3) Many software programs are either commercial or only obtained by contacting to authors; and (4) Syntax preparation for some of these programs may require substantial effort. Although several R packages available for the CDM analyses; one of the most comprehensive packages is the GDINA package.

This article aims at explaining the fundamentals of CDMs as well as demonstrating the various implementations using GDINA package (Ma & de la Torre, 2018). The current article explains the basics of CDM and provides sufficient details on the implementations and may be used to guide novice researchers in CDMs related studies. We start by introducing CDM terminologies, and input and output of a CDM analysis. This will be followed by presentation of the G-DINA model framework including several specific CDMs, which can be derived from the G-DINA. A brief description of the package GDINA will then be provided. The fourth section will demonstrate how various analyses are conducted using the R package with a graphical user interface. In the fifth section, the paper summarizes additional features of the GDINA R package. The final section will provide some concluding remarks.

Input and Output in a CDM Analysis

Two input matrices are needed for a basic specification of a CDM. The first matrix consists of examinees' item responses, and may be called response matrix. This matrix is composed of examinees' binary (in the simplest form) responses to items on a test. This is typically an IxJ matrix $\boldsymbol{X}$, where the element x_{ij} indicates whether examinee i correctly responded item j ($x_{ij} = 1$) or not ($x_{ij} = 0$). The second matrix, which specifies relationship between each item on a test and content related attributes, is called Q-matrix (Tatsuoka, 1983). For instance, for a test consisting of $j = 1,2, \ldots, J$ items and

measuring $k = 1,2,\ldots,K$ attributes, the elements of a IxJ Q-matrix are composed of binary variables, where $q_{jk} = 1$ indicates that examinees must have the *kth* attribute to be able to achieve a correct response to *jth* item. Similarly, element of $q_{jk} = 0$ indicates that the *kth* attribute is not required for a successful response to *jth* item.

The aim of conducting a CDM analysis is to be able to make inferences about examinees' mastery status of each of the K attributes. Typically, but not always, attributes are dichotomous, and the K attributes form 2^K attribute patterns, which are called latent classes and denoted as $\boldsymbol{\alpha}_l = [\alpha_{l1},\ldots,\alpha_{lK}]$, where $l = 1,\ldots,2^K$. Each element $\boldsymbol{\alpha}_{lk}$ indicates whether members of latent class $\boldsymbol{\alpha}_l$ possess *kth* attribute. At the end of the analysis, each examinee is assigned with an attribute profile indicating which specific attributes the examinee has and has not mastered. In addition, the analysis provides information on (1) proportion of examinees mastered a specific attribute, and (2) proportion of examinees within each latent class.

Generalized DINA Model Framework

A wide range of saturated and reduced CDMs have been introduced in the literature. One way of distinguishing these various models pertains to attribute-effects considered in the response process. In a general (i.e., saturated) model, all main and interaction effects of measured attributes contribute to the item response function. The *generalized deterministic, noisy and gate* model (G-DINA: de la Torre, 2011) is a general CDM from which, more specific models can be derived. For example, *deterministic input, noisy "and" gate* model (DINA: Junker & Sijtsma, 2001), *deterministic input, noisy "or" gate* model (DINO: Templin & Henson, 2006), and Additive-CDM (ACDM: de la Torre, 2011) are derived from the G-DINA model.

Within the G-DINA model framework, a baseline probability (the probability of success when an examinee has not mastered any required attributes for item *j*), main effect terms (change in the success probability when a required attribute is mastered) and possible interaction terms (change in the success probability when more than one attribute is mastered) are specified. To derive the A-CDM, DINA model and DINO model from the G-DINA model, one needs to place specific constraints on the G-DINA item response function. Specifically, to obtain the A-CDM, all interaction terms in the G-DINA model item response function are set to zero. Likewise, to derive the DINA model, all item parameters but the baseline and highest order interaction are set to zero. Finally, to obtain the DINO model, the main and interaction effects are constrained to be equal with alternating signs. Above constraints result in $K_j^* + 1$ item parameters for the A-CDM and just two item parameters for the DINA and DINO models.

As explained in de la Torre (2011), in the specification of general CDMs, several link functions may be used. In saturated forms, the G-DINA model can be expressed using the *identity* link, *logit* link, and *log* link, all of which provide identical model-data fit. *Additive* models of log-linear CDM and log CDM models are the *linear logistic model* (LLM; Maris, 1999) and the *reduced reparametrized unified model* (R-RUM; Hartz, 2002),

respectively. Although ACDM, LLM, and R-RUM have the same number of item parameters, unlike their general models, they do not provide identical model-data fit as they assume different underlying processes. For details on saturated and reduced models and their parameter estimations, readers may refer to de la Torre (2011).

GDINA Package

The package 'GDINA' composed of a set of psychometric tools for cognitive diagnosis modeling. It has the capability of handling both dichotomous and polytomous response data. The G-DINA model, the sequential G-DINA (Ma & de la Torre, 2016) and many CDMs subsumed by these two can be estimated using this package. More specifically, in addition to the models subsumed by the G-DINA model, (i.e., the DINA, DINO, A-CDM, LLM, R-RUM), *multiple-strategy DINA model* [de la Torre & Douglas, 2008], many extensions of the G-DINA model such as the sequential G-DINA model for ordinal and nominal data [Ma & de la Torre, 2016], polytomous G-DINA model for polytomous attributes [Chen & de la Torre, 2013], multiple group G-DINA model for individuals from multiple groups [Ma, Terzi, Lee & de la Torre, 2017], and diagnostic tree model for polytomous response data with multiple strategies [Ma, 2018] can also be handled by the GDINA package. Marginal maximum likelihood estimation with expectation-maximization algorithm (MMLE/EM) is used for item parameter estimation. This package allows user to assign different CDMs to different items in a single test. It is also flexible in terms of handling independent, saturated, higher-order, and structured joint attribute distributions. Offering a *graphical user interface* is a notable feature of the package. Along with providing person and item parameter estimates and model-fit statistics, this package allows users to conduct various analyses including Q-matrix validation, item-fit evaluation, item and test-level model comparisons, and differential item functioning.

Demonstrations

To be able to follow the demonstrations below, R software (R 3.5.0 or higher version) along with R studio (a graphic user interface for R) need to be downloaded from https://cran.r-project.org/ and https://rstudio/com, respectively. After installing R and Rstudio, the GDINA package (version 2.2.0) with its all dependencies must be installed. Then, command > library(GDINA) is run to start working with the GDINA package. To help nonprogrammer users, we intend to apply the main functions of the package using *Shiny*, which is a web application framework that is used to build interactive web applications directly from R. To use Shiny with the GDINA function, one need to run the command > startGDINA(). Then the GDINA graphical user interface (GUI) opens in a new window.

Reading Input Data

Table 1
Generating Q-matrix

Item	A1	A2	A3	A4	A5	A6	Item	A1	A2	A3	A4	A5	A6
1	1	0	0	0	0	0	*10*	0	0	0	1	1	0
2	0	1	0	0	0	0	*11*	0	0	0	0	1	1
3	0	0	1	0	0	0	*12*	1	0	0	0	0	1
4	0	0	0	1	0	0	*13*	1	1	1	0	0	0
5	0	0	0	0	1	0	*14*	0	1	1	1	0	0
6	0	0	0	0	0	1	*15*	0	0	1	1	1	0
7	1	1	0	0	0	0	*16*	0	0	0	1	1	1
8	0	1	1	0	0	0	*17*	1	0	0	0	1	1
9	0	0	1	1	0	0	*18*	1	1	0	0	0	1

For simulation purposes, given a uniformly distributed attribute patterns, we generated a response data set based on the GDINA, ACDM, and DINA models. To generate the data, we used a hypothetical Q-matrix given in Table 1. Eighteen items requiring one, two, or three attributes constitute the Q-matrix measuring a total of six attributes. Thus, in the data generation, 1000 attribute patterns from 2^6 possible attribute patterns were drawn from a uniform distribution. Then, based on this sample, response data were generated following the DINA model, additive model, and G-DINA model. Responses for items with multiple attributes distributed among the three generating models. More specifically, items seven to 14 followed the DINA model, items 15 and 16 followed the ACDM, and last two items followed the G-DINA model. For the data generation purposes, lower and upper bound of probability corrects were drawn from a random uniform distribution of U(.05, .20). Then, final response data involved 1000 examinees' binary responses to 18 items. The R code used for data generation is given in Figure 1.

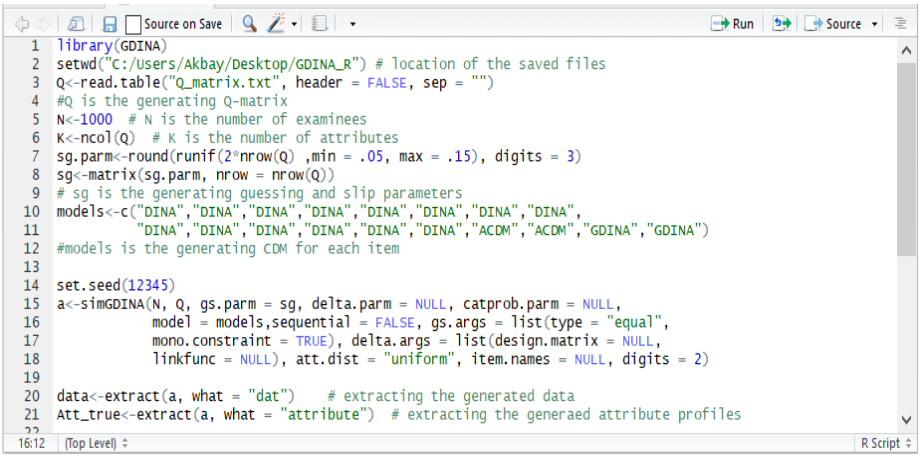


Figure 1. R Code for Data Generation

To read the input data (i.e., response data and Q-matrix) by the GUI, we need to go to the input tab and select our files from the folder where they are kept. There are three panels in this window. In the upper panel, shown in Figure 2, input files are read. Data in the input files need to be separated with one of the following: Tab, comma, semicolon, or space. If our input file has header, *Header* box must be checked. The remaining two panels (not given in the figure) are used to check whether the data are read properly. Specifically, middle and lower panels show the first six rows of the response data and the Q-matrix, respectively.

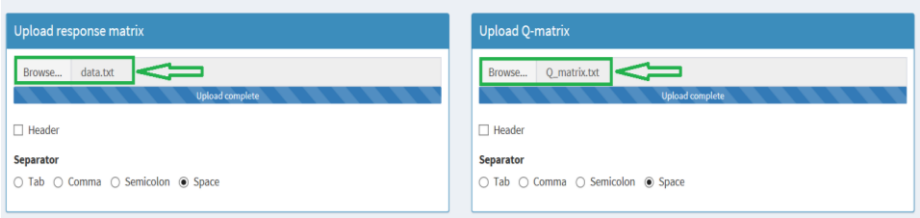


Figure 2. Upper Panel of the Input Tab.

Model Estimation

In many cases, item parameters along with examinees' attribute patterns are not known. In this situation, item parameters and examinee attribute profiles are estimated together. To fulfill this, GDINA package employs marginal maximum likelihood estimation (MMLE) through expectation-maximization (EM) algorithm. The *Estimation Settings* tab of the GUI provide us with several lists and options to select from. This tab is given in Figure 3. First, we need to select a CDM to fit among the eight available models: GDINA, logit GDINA, log GDINA, DINA, DINO, ACDM, LLM, and RRUM. There is a ninth option here, which allows user to specify a vector of models (comma delimited without quotation mark).

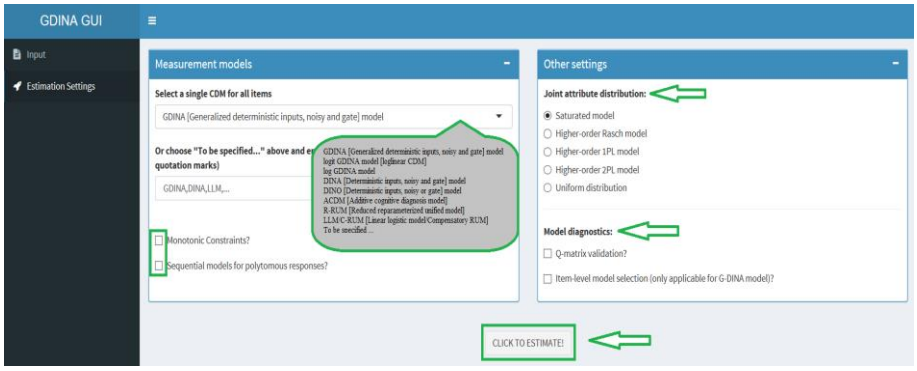


Figure 3. Estimation Specifications

Then, we need to make a decision on the attribute distribution. There are five options available in the GUI: Saturated, higher-order Rasch, higher-order 1PL, higher-order 2PL, and uniform distribution. The three higher-order attribute distribution types are defined based on the item response models for continuous ability. One- and

two-parameter logistic models, as well as the Rasch model are available options here. In addition, there are four more *optional* settings. We need to check the boxes corresponding to options such as applying *monotonic constraints* (i.e., mastering an additional attribute does not lead to a lower success probability) to the fitted model, *Q-matrix validation*, and *item-level model selection*. We then run model estimation by clicking the *Click to Estimate!* button.

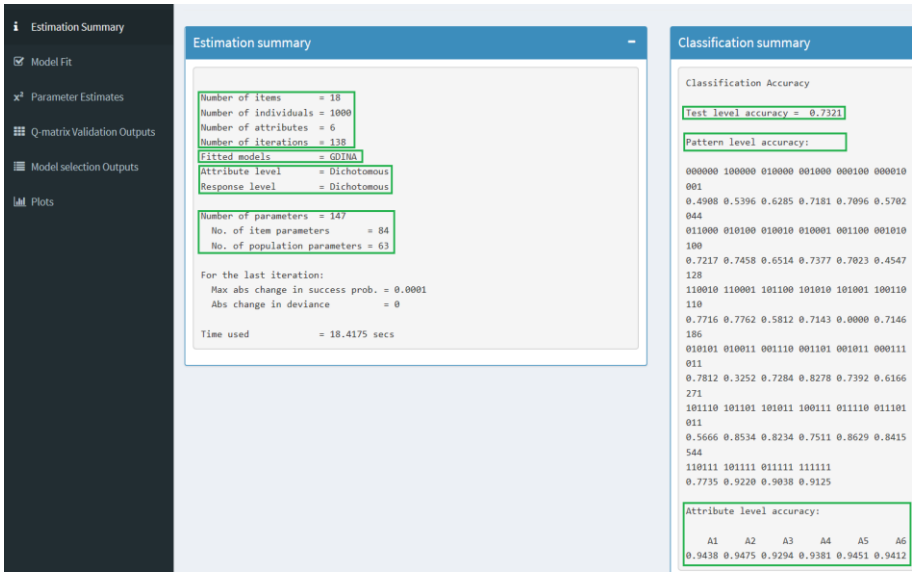


Figure 4. Estimation Summary

In this specific example, we fitted the GDINA model with saturated attribute distribution and applied monotonic constraints. We further requested Q-matrix validation and item-level model selection. Estimation results are found in the *Estimation Summary* tab when the estimation is completed. As shown in Figure 4, there are two panels on this tab. The left panel shows the estimation summary and right reports the classification summary. Estimation summary starts with some descriptive statistics such as the number of items, number of examinees, number of attributes, and number of iterations needed for the model to converge. For this specific example, 18 items, 1000 examinees, six measured attributes, and 138 iterations are reported. Then, information on the fitted model by item, attribute type, and response type are reported. For our example, we see that G-DINA model fitted to all items, and both attribute and response data are dichotomous.

In the estimation summary further information about the number of total, item, and person parameters are also displayed. Sixty-three person parameters and 84 item parameters were estimated. Notice that because we have a total number of six attributes to measure, in saturated attribute distribution, we must estimate 63 (2^6-1) latent classes, as well as all item parameters defined by the fitted model (84 delta

parameters defined by the GDINA model). Test and pattern level classification accuracy rates are reported on the right side under the classification summary panel.

Model-fit evaluation

Estimated CDM parameters are interpretable to the extent that model explains the data (Chen, de la Torre & Zhang, 2013). Thus, inferences based on any CDM are valid as long as the model fits to data. Data-model fit can be checked by examining *relative-* and *absolute- fit* statistics. The former statistics indicate the most appropriate model when competing models are available, whereas the latter shows whether model itself fits the data. Relative and absolute fit statistics are found under the *Model Fit* tab of the GUI. As shown in Figure 5, the relative test fit statistics, log-likelihood, Akaike information criterion (AIC; Akaike, 1974) and Bayesian information criterion (BIC; Schwarz, 1976) with the corresponding penalties are given on the relative test fit window.

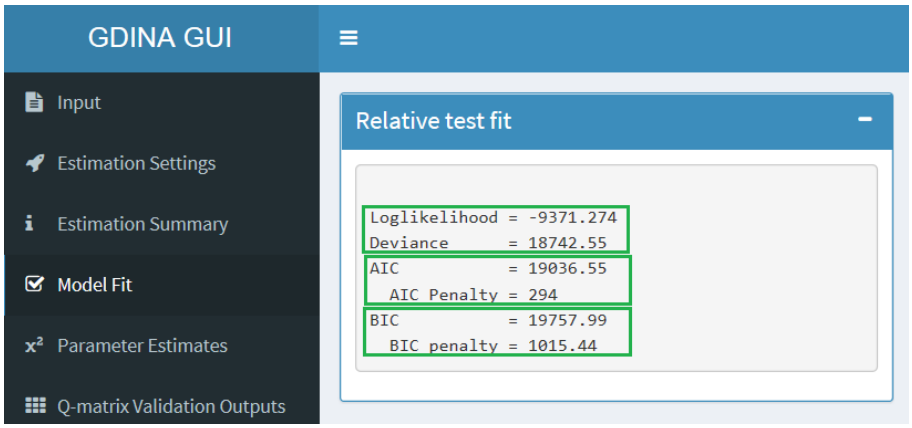


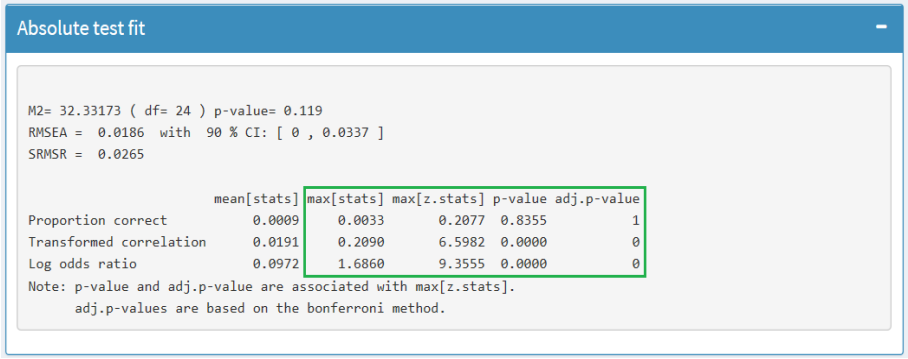
Figure 5. Relative Fit Statistics

When we have rival models that may fit the data, we need to check relative fit statistics to compare these models. To do so, we can look at the AIC and BIC values of nonnested models in which, the model with the least information criteria is preferred. When rival models are nested (e.g., GDINA vs. DINA), one can make a decision based on a likelihood ratio test: $LR = -2[LL_{reduced} - LL_{saturated}]$, where LR is the likelihood ratio of saturated and reduced models, which is compared against χ^2 distribution with a given significance level (e.g., .05) and degrees of freedom determined by the parameters difference in the compared models. Our null hypothesis for this test is the reduced model fits the data as good as the full model, and we reject the null when LR is larger than the critical χ^2 value.

Relative fit statistics of our simulation is given in Figure 5. For demonstration purposes, we also fitted the DINA model to the data that we have been analyzing. The log-likelihood, AIC, and BIC results when the DINA is fitted are: -9674.58, 19547.15, and 20033.02, respectively. AIC and BIC reported in Figure 5 are lower than these values obtained from the DINA model fit. Furthermore, computed $LR =$

$-2[-9674.58 + 9371.27] = 606.62$, given the model parameters 99 and 147 for the DINA and GDINA models, need to be compared against $\chi^2_{(.05;48)} = 65.17$. Because the computed LR is larger than the critical value, we reject the null hypothesis and conclude that the GDINA model fits the data better than the DINA model.

For absolute fit evaluation, GDINA package offers three statistics: proportion correct (*p*), log-odds ratio (*l*), and transformed correlation (*r*). Specifically, *p* stands for residuals between the observed and predicted proportion correct of individual items; *l* is computed based on log-odds ratio of item pairs; and *r* is computed based on the residuals between the observed and predicted correlations of item pairs (Chen, de la Torre & Zhang, 2013; de la Torre & Douglas, 2008; Sinharay & Almond, 2007). When these statistics are close to zero, it indicates that the model fits the data.


Absolute test fit

M2= 32.33173 (df= 24) p-value= 0.119
RMSEA = 0.0186 with 90 % CI: [0 , 0.0337]
SRMSR = 0.0265

	mean[stats]	max[stats]	max[z.stats]	p-value	adj.p-value
Proportion correct	0.0009	0.0033	0.2077	0.8355	1
Transformed correlation	0.0191	0.2090	6.5982	0.0000	0
Log odds ratio	0.0972	1.6860	9.3555	0.0000	0

Note: p-value and adj.p-value are associated with max[z.stats].
adj.p-values are based on the bonferroni method.

Figure 6. Absolute Fit Statistics

To use *p*, *l*, and *r*, their standard errors are also needed. Please refer to Chen, de la Torre and Zhang (2013) for derivation of the statistics and their respective standard errors. The *z*-scores of the statistics are used to test whether the residuals are significantly different from zero. Because *J*-1 proportion correct and *J*(*J*-1)/2 pairwise log-odds ratio and transformed correlations are available for a test with *J* items, only the maximum z-scores of each statistics are tested and the test results are provided by the GUI under the model fit tab. Here in our example, summary of absolute fit analysis is provided in Figure 6.

Means, maximums, maximum of z-scores, and corresponding p-values and adjusted p-values are reported in the figure. When we check maximum z-statistics for all three absolute fit statistics and corresponding p- and adj.p-values, we reject the null hypotheses that the model fit the data based on *l* and *r*. Only, *p* statistic retains the null hypothesis, which is claimed to be either unreliable or results in low rejection rates (Chen, de la Torre & Zhang, 2013). We should note there that adj.p-value is the bonferroni adjusted p-value as we implicitly test *J*-1 proportion correct and *J*(*J*-1)/2 pairwise log-odds ratio and transformed correlations by testing the maximum of the z-scores.

The above results indicate that there is at least one item that cannot be fitted to the model, which is expected because our data were generated based on the items

following the DINA, ACDM, and G-DINA models. However, if we fit DINA model to the data corresponding to the first 14 items using the first 14 rows of the Q-matrix, all three absolute fit statistics indicate that the model fit the data (p-values for the maximum z-statistics are .878, .133, and .124 for proportion correct, log-odds and transformed correlations, respectively. These results, therefore, indicate that if only the first 14 items confirming the DINA model constituted the test, all aforementioned absolute fit indices would yield model-data fit for all items. We should note that the GUI also provides a heatmap plot depicting the p-values log-odds and transformed correlation obtained from item pairs.

Parameter Estimation

Both the item and person parameters estimates are readily available under the *Parameter Estimates* tab of the GUI. This tab, given in Figure 7, offers a list of item parameter estimates with or without standard errors: *success probabilities of reduced latent classes*; *guessing and slip parameters*; *delta parameters;* and *success probabilities of all latent classes*. To obtain the standard errors along with the estimates, *Estimate S.E?* box must be checked. Figure 7 demonstrates that we requested guessing (*g*) and slip (*s*) parameters with corresponding standard errors. Under G-DINA model, *g* and 1-*s* parameters stand for lower and upper bound of success probabilities.

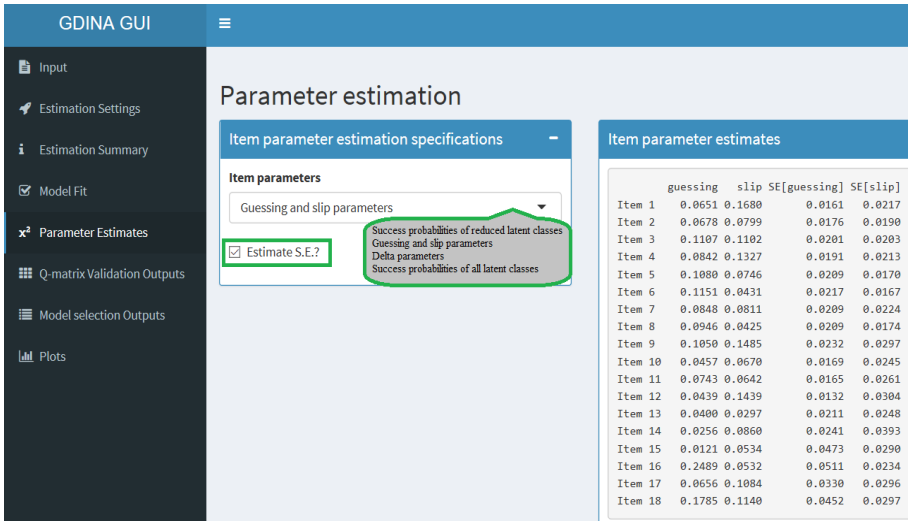


Figure 7. Item Parameter Estimation Specifications

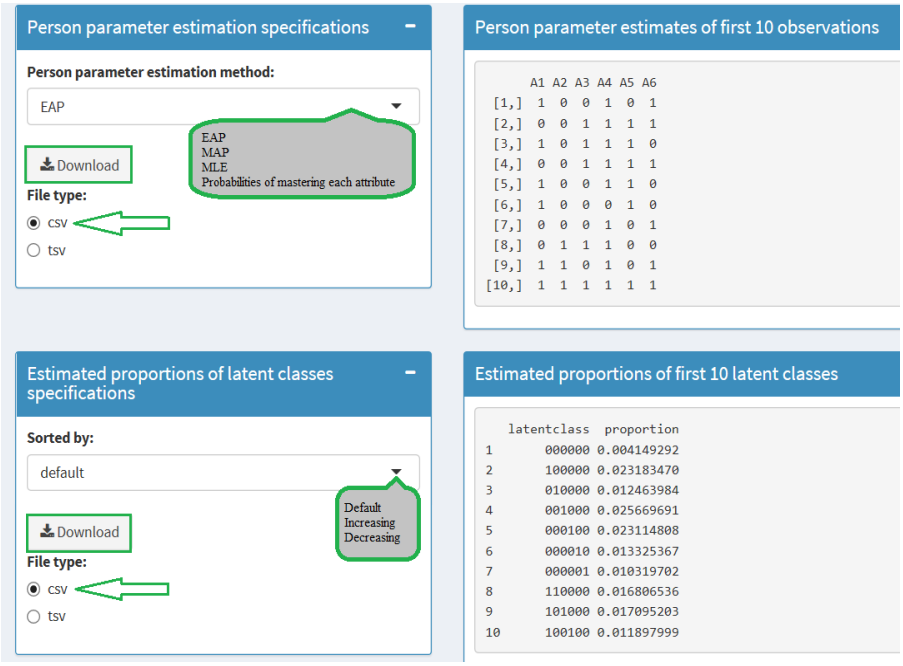


Figure 8. Person Parameter Estimation Specifications

In addition to item parameters, the *Parameter Estimates* tab of the GUI provides us with person parameters as well. Examinees' estimated attribute profiles for the first 10 examinees are shown in the middle panel of this tab. Attribute profile estimates based on maximum likelihood estimation (MLE), expected a priori (EAP), and maximum a priori (MAP) are computed and may be selected from the list given in the *Person parameter estimation method*. This list also includes the *Probabilities of mastering each attribute* option. In our example given in Figure 8, we see the first ten examinees' estimated attribute vectors. The GUI allows for all these types of person parameters to be downloaded as a .csv or .tsv file. To do so, we need to select one of the two file types, and hit the *Download* button.

Q-matrix Validation

In general, implementation of the CDMs requires a Q-matrix mapping attributes to items. This matrix embodies the cognitive specifications in test construction (Leighton, Gierl & Hunka, 2004). Because this input plays a crucial role in incorporating cognitive theories into psychometric practice, it needs to be correctly specified. Only then, CDMs can provide maximum diagnostic information (de la Torre, 2008). However, due to subjective judgments of content experts, Q-matrix construction process might not be fulfilled easily and successfully. This process may result in some misspecifications in the Q-matrix such that a required attribute may be missed, or an unnecessary attribute may be added in the matrix. These two types of

misspecifications are known as under- and over-specifications, respectively (de la Torre & Chiu, 2016).

Another possible error may emerge when both under- and over-specifications coexist for an item, which is referred to as over-and-under-specification (de la Torre & Chiu, 2016). Negative effects of misspecified Q-matrices in item calibration, examinee classification, and model misfit have been shown in the literature (e.g., Chiu, 2013; de la Torre, 2008; de la Torre & Chiu, 2016). To address misspecifications due to subjectivity in the process, a general Q-matrix validation procedure (i.e., de la Torre & Chiu, 2016) along with several others specifically developed for reduced models (e.g., Chiu, 2013; de la Torre, 2008) have been proposed.

An item specific discrimination index, ς_j^2, is proposed and used in the general Q-matrix validation method. ς_j^2 is a measure of weighted variance of success probability for a particular attribute distribution. De la Torre and Chiu (2016) showed that a correct q-vector yields homogenous latent groups in terms of probability of success. The idea behind use of this index for Q-matrix validation lies in the fact that correct q-vector should produce the highest variability in success probability. Their method suggests selecting a q-vector that approximates the maximum ς_j^2 with the fewest attribute specifications.

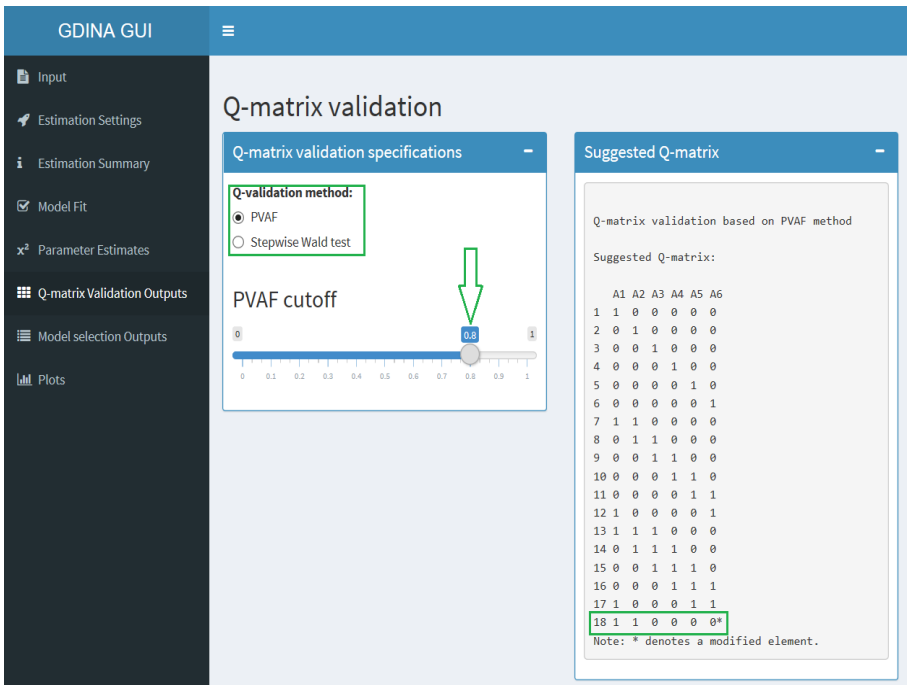


Figure 9. Setting PVAF Cutoff and Suggested Q-matrix

For the search algorithm to select a parsimonious q-vector producing approximately the highest ς_j^2 rather than the q-vector yielding the highest ς_j^2, a

stopping rule is implemented as a part of the algorithm. This stopping rule is defined by the proportion of variance accounted for (PVAF) by a particular q-vector relative to the highest ς_j^2. Because maximum ς_j^2 is produced by the q-vector requiring all measured attributes, a q-vector is selected by this method when PVAF is higher than a predetermined cutoff (i.e., ϵ).

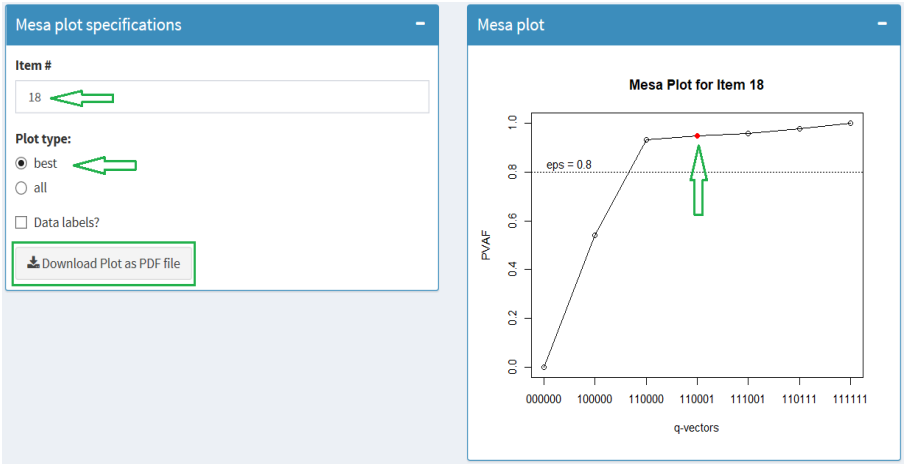


Figure 10. Mesa Plot with Specifications

The GDINA package implements the general Q-matrix validation method. Validation results (when requested in the estimation specifications tab) are found in the *Q-matrix Validation Outputs* tab in the GUI. On this output page, first PVAF cutoff needs to be specified so that suggested Q-matrix will be produced based on this cutoff. Here in our example given in Figure 9, we set the PVAF to .80 and the corresponding suggested Q-matrix yielded correction in one item (i.e., item 18). Stepwise implementation of the test is also available through the option *Stepwise Wald test*. Remember that although the validation method has suggested this q-vector, the final decision on correcting the Q-matrix should be based on the expert decision after careful consideration of related item, q-vector, and the attributes measured by the test.

Down on the same page, we are able to see Mesa Plot (de la Torre & Ma, 2018) for each item. Item 18 is shown in Figure 10 as an example, for which we plotted the *best q-vectors* option. We could also see *all q-vectors* option by selecting the plot type from *Mesa Plot Specifications*. PVAF values for this item indicates that a q-vector of 110000 is as good as the specified q-vector (110001). Notice that the original q-vector is marked with filled circle. Here we also have an option to print *data labels* by clicking the corresponding box in the specifications window. Furthermore, the GUI allows us to download Mesa plots as PDF file.

Item-level Model Comparison

GDINA package allows item-level model selection so that CDMs need not be specified a priori. To perform model selection between saturated and reduced models at the item-level, Wald test is used to compare G-DINA model against the fits of the

specific CDMs (i.e., DINA, DINO, ACDM, LLM, and RRUM). The Wald test proposed by de la Torre (2011) is an item level procedure, which can be performed for items requiring at least two attributes. Wald test requires a restriction matrix of **R** constraining the GDINA parameterization to derive the reduced models. This restriction matrix is used in computation of Wald statistic. For further information on the Wald statistics, readers may refer to de la Torre and Lee (2013) and Ma, Iaconangelo and de la Torre (2016).

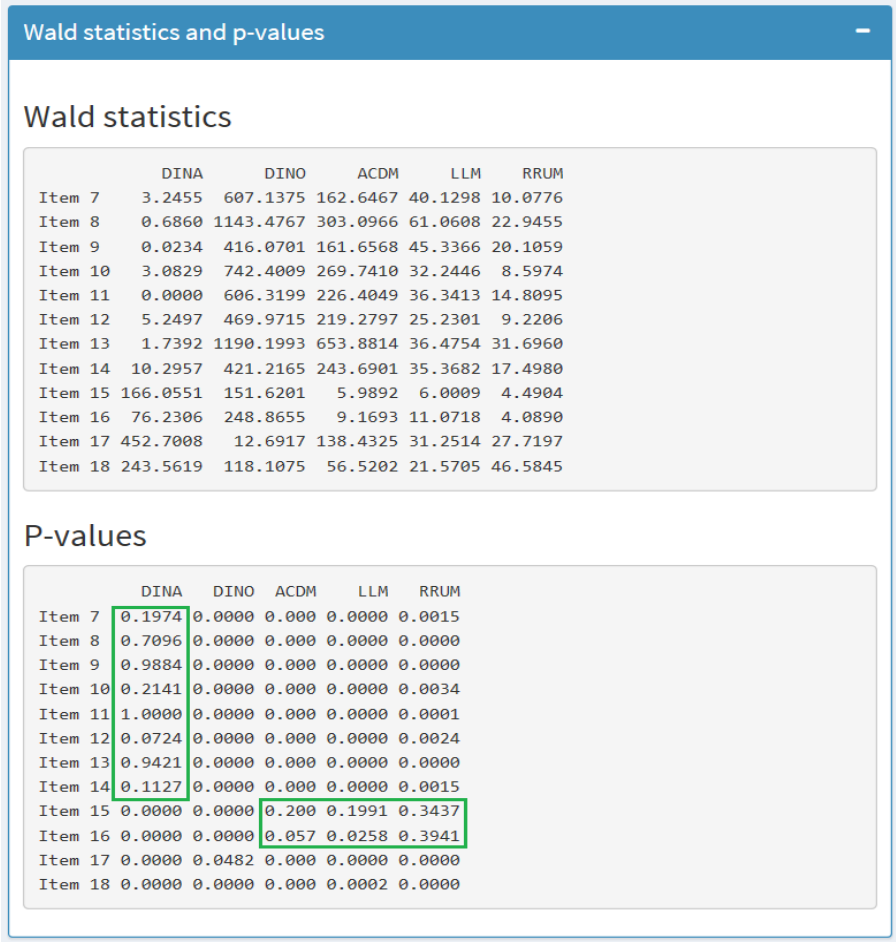


Figure 11. Item-level Model Comparison

For our case, the Wald statistics and corresponding p-values for the different reduced models, given in Figure 11, are reported under the *Model Selection Output* tab of the GUI. Notice that these statistics only printed for the items requiring at least two attributes for which more than one CDM can be fitted. Wald statistics and p-values indicates that items 7-14 are DINA and items 17-18 are the GDINA items. Remaining two items (i.e., 15-16) conform to at least two of the following CDMs: RRUM, ACDM

and LLM. Although the data for these two items are simulated using the ACDM, because these three models have the same number of item parameters under different *link functions* (see de la Torre, 2011 for different link functions), it may be hard to make a perfect selection.

Plots

The GUI allows users to generate various types of graphs via the *Plots* tab. These readily available graphs are useful tools to highlight important information and enable users to compare various parameters or statistics. For example, *Plot of probability of mastery for individuals* is given in Figure 12; which allows users to compare individuals' mastery probabilities. To do so, one need to enter a vector of individuals whose mastery probabilities are to be compared. This vector must be comma delimited, and no quatation marks is needed. Another type of graph is generated to display *individual posterior probabilities*. This bar graph allows user to display the mastery probabilities in increasing or decreasing order.

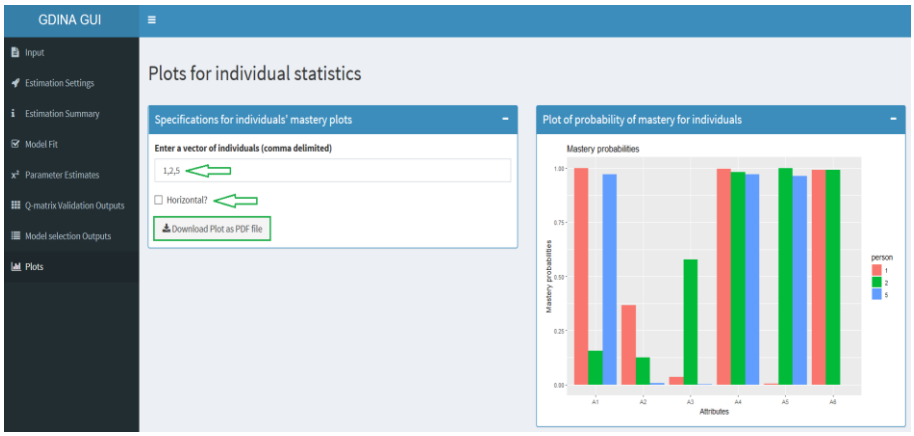


Figure 12. Individuals' Mastery Probability Plots

The next graph is the *Plot of proportions of latent classes*, which indicates the latent classes' mastery rates. The mastery proportions of the latent classes may also be displayed in decreasing or increasing order. All these graphs are displayed vertically as default; however, user can switch to the horizontal positioning by clicking the *Horizontal* box. The last graph is the *Item response function plot*, which displays the item statistics of the selected item. Error bars can also be added to this bar graph by clicking the *Error bars* box.

Additional Functions Offered by the Package

This section provides additional flexibilities offered by the package not covered by the GUI. R scripts are needed to use additional functions. By writing a script additional features of the package become available, and additional analyses that are not part of the GUI can be conducted. Note that, because our goal is to serve unexperienced

audience, we only focus on the fundamental functions used in dichotomous attribute cases. Information in this section is useful for readers who have some level of R programming skills.

We can start with the function *simGDINA( )* that is used to simulate examinee response data. Sample size (*N*), Q-matrix (*Q*), and lower and upper bound of probability correct (*gs.parm*) must be provided, where *gs.parm* is a matrix or data frame for lower and upper bound probability correct parameters (i.e., guessing and slip parameters). The contribution of the delta parameters of the additive models is specified by the *gs.args* argument for which two options, "equal" and "random", are available. The *model* argument allows users to assign a CDM to each item. *model* is a character vector for each item (or a scalar when all items are the same type), for which the available options are: "GDINA", "DINA", "DINO", "ACDM", "LLM", "RRUM", and "MSDINA". The *att.dist* argument is used to indicate the attribute distribution for simulation, where "uniform", "higher.order", "mvnorm", and "multinomial" are available options for the uniform, higher order, multivariate normal, and multinomial distributions, respectively. When attribute distribution happens to be multinomial, the probability of each attribute pattern needs to be specified through *att.prior*. If we would like attribute distribution to be higher-order, then, higher order distribution of attributes need to be specified through *higher.order.parm*. Lastly, a list of parameters for multivariate normal distribution is provided by *mvnorm.parm* when the attribute distribution to be generated is a multivariate normal.

The GDINA package allows users to generate hierarchical attribute structures and, to provide prior joint attribute distribution with specified hierarchical relations among the attributes. For this purpose, the *att.structure( )* function can be used. The *hierarchy.list* is a list used for specifying hierarchical structure among the attributes. Elements of this list specify direct prerequisite relations between the two attributes. For example, when three attributes (e.g., A1, A2, and A3) have a linear relationship where A1 is prerequisite for A2 (and A3), and A2 is also prerequisite for A3; this hierarchical structure is defined by the *list(c(1,2), c(2,3))*. The number of attributes *K* must also be specified to use this function. The *att.prob* argument is used in this function to set latent class probabilities as either "random" or "uniform."

The *GDINA( )* function calibrates the GDINA and reduced CDMs. Here we would like to focus on the features available using specific functions that are not available in the GUI. Along with the CDMs available in the GUI, multiple-strategy DINA model (MSDINA; de la Torre & Douglas, 2008) is also estimated by this function. This function allows hierarchically structured attributes in the estimation, for which *att.prior* must be specified such that prior weights 0 are assigned to impossible latent classes. This option is only available for calibration of the DINA, DINO, or G-DINA models.

To evaluate whether a specific CDM can replace the G-DINA model without significant loss in data-model fit, the *modelcomp( )* function can be used. Although model comparison can also be conducted in the item and test level using the GUI, use of this function allows one to specify the items for which the model comparison is

requested. Furthermore, we can also specify the reduced models to be considered in the model selection. Unlike in the GUI, Wald statistics is not the only option under this function. Two more statistics, namely, Lagrange multiplier and likelihood ratio, are also available for model comparison.

The *extract( )* function is used to extract information or data such as att.prior, discrim, prevalence, prosterior.prob, and delta.parm (i.e., attribute priors, discriminations, attribute prevalence, posterior probabilities, and delta parameters, respectively). To obtain such information, objects from class GDINA, itemfit, modelcomp, or simGDINA must be specified by the *object* argument. The *what* argument is then used to call the information we would like to extract.

The GDINA package permits one to evaluate differential item functioning (DIF), which may occur when the success probabilities on an item are different for the examinees from different groups with the same attribute mastery profile (Hou, de la Torre & Nandakumar, 2014). The dif() function is used to detect DIF based on the models in the GDINA function using the Wald test (Hou, de la Torre, & Nandakumar, 2014) and the likelihood ratio test (Ma, et. al, 2017). It should also be noted that the current version of the package only allows detection of DIF for two groups.

Concluding Remarks

In this manuscript, we first discussed relevant literature in cognitive diagnosis, which includes a saturated model and several reduced models. It was followed by presentation of the GDINA R package, which can be conveniently used for CDM analyses. We demonstrated step-by-step basic CDM analyses using the GUI with an aim of guiding novice researchers and practitioners in the field. Within the demonstration, we provided fundamental information on the statistics used and interpretations of the output. Although the analyses and options are limited when one uses the GUI, more advanced methods for CDMs are available in the package for users who are more familiar with R.

Several steps must be taken to implement cognitive diagnosis modeling successfully. The first step requires specification of the domain-specific attributes for which researchers may need to look into theories within the content domain, review the relevant literature, and conduct a protocol analysis with the think-aloud procedure (Akbay, Terzi, Kaplan, & Karaaslan, 2017; Leighton & Gierl, 2007; Tjoe & de la Torre, 2014). In general, finer grain sized attributes provide richer information (Alderson, 2005); however, an increase in the number of attributes measured by a single test requires a larger number of items which, in turn, requires much larger sample sizes for the model parameters to be estimated accurately.

Once the attributes are determined, items measuring single attributes and combinations of attributes are developed and the relationships between these items and the attributes must be carefully specified through the construction of a Q-matrix. It should be noted that the usefulness of the diagnostic information provided by the CDM analysis is highly dependent on the theoretical and empirical soundness of the Q-matrix (de la Torre, 2008; de la Torre, & Chiu 2016; Lee & Sawaki, 2009). Therefore,

item-attribute relationships must be specified carefully and correctly. In certain situations, researchers have expectations regarding how attributes interact in the item response process. In these situations, another fundamental step involves deciding a priori which CDMs should be fitted to item response data to test one's hypotheses.

Although other software programs are also available, using the GDINA package offers users some flexibilities such as allowing estimation of a wide range of CDMs and allowing nonprogrammers to benefit from this package through the GUI. Using the GDINA, one can obtain diagnostic information about individual examinees' mastery or nonmastery status of attributes; proportion of examinees who have mastered a specific attribute; and proportion of examinees that are in a specific latent class. In addition to ordinary CDM analyses, GDINA package further allows users to apply model selection at the test- and item-level to make sure that the most appropriate CDM (i.e., CDM that best explains the attribute interactions in the item) is fitted to the response data. Furthermore, to identify possible item-attribute specification mistakes in the Q-matrix, implementation of an empirical Q-matrix validation method is available in the GDINA package. Lastly, this package offers various handy graphs, which can be very useful in emphasizing important information and comparing various parameters and/or statistics.

References

Alderson, J. C. (2005). *Diagnosing foreign language proficiency: The interface between learning and assessment*. A&C Black.

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

Akbay, L., Terzi, R., Kaplan, M., & Karaaslan, K. G. (2017). Expert-based attribute identification and validation on fraction subtraction: A cognitively diagnostic assessment application. *Journal on Mathematics Education*, *8*, 103-120.

Chen, J. & de la Torre, J. (2013). A general cognitive diagnosis model for expert-defined polytomous attributes. *Applied Psychological Measurement*, *37*, 419-437.

Chen, J., de la Torre, J., & Zhang, Z. (2013). Relative and absolute fit evaluation in cognitive diagnosis modeling. *Journal of Educational Measurement*, *50*, 123-140.

Chiu, C. Y. (2013). Statistical refinement of the q-matrix in cognitive diagnosis. *Applied Psychological Measurement*, *37*, 598-618.

de la Torre, J. (2008). An empirically based method of Q-matrix validation for the DINA model: Development and applications. *Journal of Educational Measurement*, *45*, 343-362.

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

de la Torre, J. (2011). The generalized DINA model framework. *Psychometrika*, *76*, 179-199.

de la Torre, J., & Chiu, C. Y. (2016). General method of empirical Q-matrix validation. *Psychometrika*, *81*, 253-273.

de la Torre, J., & Douglas, J. A. (2008). Model evaluation and multiple strategies in cognitive diagnosis: An analysis of fraction subtraction data. *Psychometrika*, *73*, 595.

de la Torre, J., & Lee, Y. S. (2013). Evaluating the Wald test for item-level comparison of saturated and reduced models in cognitive diagnosis. *Journal of Educational Measurement*, *50*, 355-373.

de la Torre, J., & Ma, W. (2016, August). *Cognitive diagnosis modeling: A general framework approach and its implementation in R.* A Short Course at the Fourth Conference on Statistical Methods in Psychometrics, Columbia University, New York, NY.

DiBello, L. V., & Stout, W. (2007). Guest editors' introduction and overview: IRT-based cognitive diagnostic models and related methods. *Journal of Educational Measurement*, *44*, 285-291.

Hartz, S. M. (2002). *A Bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practicality* (Unpublished doctoral dissertation).

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

Hou, L., de la Torre, J. D., & Nandakumar, R. (2014). Differential item functioning assessment in cognitive diagnostic modeling: Application of the Wald test to investigate DIF in the DINA model. *Journal of Educational Measurement*, *51*, 98-125.

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

Leighton, J. P., & Gierl, M. J. (2007). Defining and evaluating models of cognition used in educational measurement to make inferences about examinees' thinking processes. *Educational Measurement: Issues and Practice*, *26*(2), 3-16.

Leighton, J. P., Gierl, M. J., & Hunka, S. M. (2004). The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka's rule-space approach. *Journal of Educational Measurement*, *41*, 205-237.

Ma, W. (2018). A diagnostic tree model for polytomous responses with multiple strategies. *British Journal of Mathematical and Statistical Psychology*. https://doi.org/10.111/bmsp.12137

Ma, W., & de la Torre, J. (2016). A sequential cognitive diagnosis model for polytomous responses. *British Journal of Mathematical and Statistical Psychology, 69,* 253-275.

Ma, W., & de la Torre, J. (2018). GDINA: The generalized DINA model framework, R package version 2.1.15. Retrived from https://CRAN.R-project.org/package=GDINA

Ma, W., Iaconangelo, C., & de la Torre, J. (2016). Model similarity, model selection, and attribute classification. *Applied Psychological Measurement*, *40*, 200-217.

Ma, W., Terzi, R., Lee, S. Y., & de la Torre, J. (2017, April). Multiple group cognitive diagnosis models and their applications in detecting differential item functioning. The annual meeting of American Educational Research Association, San Antonio, TX.

Maris, E. (1999). Estimating multiple classification latent class models. *Psychometrika*, *64*, 187-212.

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

Sinharay, S., & Almond, R. G. (2007). Assessing fit of cognitive diagnostic models a case study. *Educational and Psychological Measurement*, *67*, 239-257.

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

Team, R. C. (2015). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.

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

Tjoe, H., & de la Torre, J. (2014). The identification and validation process of proportional reasoning attributes: an application of a cognitive diagnosis modeling framework. *Mathematics Education Research Journal*, *26*, 237-255.

GDINA R Paketi Kullanarak Bilişsel Tanı Modelleri Uygulaması

Atıf:

de la Torre, J., & Akbay, L. (2019). Implementation of cognitive diagnosis modeling using the GDINA R package. *Eurasian Journal of Educational Research, 80,* 171-192, DOI: 10.14689/ejer.2019.80.9

Özet

Formatif değerlendirme için hazırlanmış olan ölçme araçları, amaçlı bir şekilde tasarlandığında, tanılayıcı bilgi toplamamıza imkân sağlarlar. Bu tür bilgilere dayanarak yapılan çıkarımlar eğitimcilerin telafi edici eylemler gerçekleştirmelerini sağlar. Son zamanlarda geliştirilen formatif değerlendirme yöntemleri öğrencilere ve öğretmenlere öğrencilerin hangi bilgi ve beceri parçacıklarına (bilişsel niteliklere) sahip oldukları ya da olmadıklarıyla ilgili ayrıntılı geri bildirim sağlayabilmektedir. Bu geribildirimlerin öğretimi ve öğrenmeyi optimize etmek amacıyla kullanılması beklenmektedir. Formatif ölçme ve değerlendirmelerin sonuçlarından tanısal bilgi edinebilmek için iyi tasarlanmış sınavların uygulanmasının yanı sıra çeşitli bilişsel tanı modellerinin kullanımına da ihtiyaç vardır. Ancak, bu modellerin alan yazına yeni kazandırılmış olması ve eğitimcilerin henüz yeterince aşina olmamaları nedeniyle, bilişsel tanı modellemelerinin (BTM) parametre kestirimlerinin ve diğer ilgili analizlerin uygulamaları yeterince yaygınlaşmamıştır.

Bu makalenin amacı BTM'ye aşina olmayan eğitimci ve araştırmacılara BTM'nin temel prensiplerini tanıtmak ve ücretsiz bir yazılım olan GDINA R paketi kullanılarak yapılabilecek çeşitli BTM uygulamalarını yeterince detaylı olarak göstermektir. BTM analizleri yapan bazı yazılım programlarının mevcut olmasına rağmen, alanla ilgili en kapsamlı paketlerden biri olan GDINA R paketinin kullanımının sağlayacağı avantajlar arasında şunlar sayılabilir: (1) Birçok bilişsel tanı modelinin kestiriminin R ile mümkün olması; (2) Diğer birçok yazılım programlarının tek tip modelin kestirimine imkan veriyor olması; (3) Diğer programların çoğunun ticari olması veya ancak yazarın kendisiyle irtibat kurularak temin edilebilir olması; ve (4) Sözdizimi (sintaks) hazırlamanın bazı programlarda oldukça zahmet verici olması.

Bu makalede öncelikle BTM analizinin girdi ve çıktıları da dahil olmak üzere BTM terminolojisi tartışılmakta, sonrasında GDINA model yapısı tanıtılmakta ve GDINA R paketi sunulmaktadır. Sonrasında ise nümerik bir örnek veri setinden yola çıkılarak, GDINA R paketi ve bu paketin sunduğu grafiksel kullanıcı ara yüzü (GUI: graphical user interface) kullanılarak yapılabilecek analiz türleri adım adım takip edilebilecek şekilde sunulmuştur. Ayrıca gerekli görülen noktalarda yapılan iş ve işlemlerin teorik bilgisine ve elde edilen sonuçların yorumlanmasına dair bilgiler verilmiştir. İlerleyen bölümlerde, GUI ile yapılamayan ancak GDINA paketinin sunduğu ve R kullanımına aşina olan kullanıcıların sözdizimi yazarak yapabilecekleri ilave analizlere ve paketin ek özelliklerine yer verilmiştir. Makalenin son bölümünde ise bazı hatırlatma ve tespitler yapılmıştır.

G-DINA modeli ve GDINA paketi

Genel ya da kısıtlanmış olarak birçok bilişsel tanı modelinin ayrımları öğrencinin maddeleri cevaplama sürecinde bilişsel niteliklerin (ölçülen bilgi ya da becerilerin) etkileşimlerine bağlı olarak yapılır. G-DINA gibi genel modellerde bütün ana etkiler ve etkileşim etkileri madde tepki fonksiyonuna katkıda bulunur. Bu tür genel modellerin parametrelerinde yapılacak kısıtlamalarla daha kısıtlı ya da daha sade modeller elde edilebilir. Bunun yanı sıra, genel modellerde birden farklı *link* fonksiyonlarının kullanımı da söz konusudur. Örneğin, G-DINA model identity, logit ve log linkleriyle farklı isimlerle sunulabilir, ancak farklı linklerle ortaya konulan genel modeller özdeş model-veri uyumuna sahiptirler.

GDINA paketi GUI aracılığıyla farklı linkler altında oluşturulabilecek olan GDINA, logit GDINA, log GDINA'nın yanı sıra, bu genel modellerde ortaya koyulacak kısıtlamalarla oluşturulabilen DINA, DINO, ACDM, LLM, ve RRUM modellerinin kestirimine imkan verir. Ayrıca, GUI aracılığıyla olmasa da, R kullanımına aşina olan ve sözdizimi yazabilen kullanıcılar GDINA R paketini kullanarak farklı stratejilerin kullanımı durumunu göz önünde bulunduran MS-DINA ve G-DINA'nın farklı uzantılarından oluşan modellerin (ör. sınıflama ve sıralama ölçekleriyle elde edilen veriler için *sequential* G-DINA ve kısmi puanlamalı nitelikleri için *polytomous* G-DINA) kestirimini yapabilirler.

GDINA'yı kullanarak, testi alanların nitelikleri kazanmış ya da kazanamamış olma durumlarıyla ilgili tanılayıcı bilgi; belirli bir niteliğin kazandırılma oranı; ve belirli bir profile sahip olan kişilerin oranı ile ilgili bilgiler elde edilebilir. Bunların dışında, temel CDM analizlerinin yanı sıra, madde ve nitelikler arasındaki etkileşimi en iyi açıklayan BTM'nin kullanılabilmesi adına GDINA paketi araştırmacıların test- ve madde-düzeyinde model seçimi yapabilmelerine imkan tanır. GDINA paketi aynı zamanda madde-nitelik eşleştirme matrisinde yapılabilecek muhtemel yanlışları ortaya çıkarmaya yardımcı olmak adına ampirik olarak Q-matrisin uygunluğunun doğrulamasını sağlayan metotların kullanılmasına imkan sağlar. Son olarak, bu paket, çeşitli kolay ve kullanışlı grafikler ortaya koyar. Bu grafikler elde edilen analiz sonuçlarında önemli noktaların vurgulanmasına ve elde edilen istatistik ve parametrelerin kolaylıkla karşılaştırılmasına olanak sağlar.

Anahtar sözcükler: Bilişsel tanı modelleri, GDINA, R paketi, BTM uygulamaları.