MEASUREMENT: INTERDISCIPLINARY RESEARCH AND PERSPECTIVES
2018, VOL. 16, NO. 1, 71–77
https://doi.org/10.1080/15366367.2018.1437243

Routledge
Taylor & Francis Group

SOFTWARE REVIEW

GDINA and CDM Packages in R

André A. Rupp[a] and Peter W. van Rijn[b]

[a]Educational Testing Service (ETS); [b]Educational Testing Service (ETS) Global

ABSTRACT

We review the GIDNA and CDM packages in R for fitting cognitive diagnosis / diagnostic classification models. We first provide a summary of their core capabilities and then use both simulated and real data to compare their functionalities in practice. We found that the most relevant routines in the two packages appear to be more similar than different with the Shiny R app of GDINA making the program usage very user-friendly for key tasks. However, working with complex parametric latent-variable models in both packages will always be a task best suited for well-trained data scientists

KEYWORDS

Cognitive diagnosis models; diagnostic classification models; software review; GDINA package; CDM package

Introduction

In the last 20 years, the field of educational measurement has seen the continual expansion of scientific research into the so-called cognitive diagnosis or diagnostic classification models (DCMs), which are the topic of this special issue (for applied overviews see, e.g., Bradshaw, 2016; Rupp, Templin, & Henson, 2010). Statistically speaking, the current methodological state of the art revolves around the parametric specification, estimation, criticism, and refinement of DCMs within unified latent-variable frameworks, rather than individual component models that used to be the de facto approach in the early days of this work.

There are currently three prominent modeling frameworks available to interested users, which are the log-linear cognitive diagnosis model (LCDM) framework articulated by Henson, Templin, and Willse (2009), the generalized diagnostic input noisy "and-gate" (GDINA) framework articulated by De La Torre (2011), and the general diagnostic model (GDM) framework articulated by von Davier (2005). Even though the scientists who have developed these routines sometimes disagree with one another about certain properties of these frameworks and the ways they are operationalized in different software packages, it is fair to say that the three frameworks share many more similarities than features that set them apart.

Formally, following the notation of the GDINA model, the probability that a person with attribute vector $\boldsymbol{\alpha}$ of length K will answer item j correctly under the identity link function is given by

$$P\left(X_j = 1|\alpha_{lj}^*\right) = \delta_{j0} + \sum_{k=1}^{K_j^*} \delta_{jk}\alpha_{lk} + \sum_{k=1}^{K_j^*-1}\sum_{k'=k+1}^{K_j^*} \delta_{jkk'}\alpha_{lk}\alpha_{lk'} + \ldots + \delta_{j12\ldots K_j'}\prod_{k=1}^{K_j^*}\alpha_{lk} \quad (1)$$

where j indexes items, k indexes attributes, α_{lj}^* is the reduced attribute pattern consisting of the columns of attributes required by item j as specified in the $J \times K$ Q-matrix for the assessment with $l = 1, \ldots, 2^{K_j^*}$, αs are the attribute parameters, and δs are the item parameters (intercept, main effects, and interaction effects); note that log and logit links can also be employed.

From a data scientist's standpoint, it is most important to identify a suitable software package for working with these models. Note that we are deliberately using the term "data scientist" rather than "practitioner" in this review to underscore that a minimum proficiency with a programming

CONTACT Peter W. van Rijn pvanrijn@etsglobal.org ETS Global, Strawinskylaan 929, 1077XX Amsterdam, The Netherlands.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/hmes.

environment like R is needed in order to reap the full benefits of almost any of the software packages/routines on the market. For the GDM, an associated MDLTM software package can currently be accessed for free with a special licensing agreement from the Educational Testing Service in Princeton, New Jersey. For the LCDM, a common solution has been to use the commercial program Mplus (Muthen & Muthen, 2017; www.statmodel.com). For the GDINA model, early software consisted of freely available code in an object-oriented matrix language called Ox (http://www.doornik.com/ox/).

Fortunately, there are nowadays several routines available within the popular programming environment R (www.r-project.org), which make access to computational routines for DCMs much easier. In this brief software review, we focus on two of these packages CDM (Robitzsch, Kiefer, George, & Uenlue, 2017; version 5.9–27) and GDINA (Ma & de la Torre, 2017; version 1.4.2) because they have relatively broad coverage of various parametric DCMs and continue to be updated frequently by the developer teams. Specifically, we first provide a brief comparison of the key features of the two software packages, including some technical estimation details that affect their performance for more complex data structures. We then compare their statistical output and the associated versatility of their embedded routines on simulated datasets that represent a few key contrasting cases that data scientists might encounter in large-scale assessment contexts. We then apply these routines to a real dataset that has been previously analyzed for peer-reviewed publications to evaluate how helpful these routines are with data that do not have an obvious true generating model. We close the review with a few recommendations for data scientists and developers of these packages.

Feature comparison

Technical capabilities

In this section, we compare the two software packages in terms of inputs, outputs, technical specifications, and documentation; see Table 1 for a high-level overview of these characteristics.

To briefly evaluate the implementation of estimation routines and their associated statistical properties empirically, we compared the performance of the two packages with three datasets: (1) a small simulated dataset with 500 persons, 12 items, and four attributes, (2) a large simulated dataset with 5,000 persons, 55 items, and 10 attributes, and (3) a real dataset from the administration of a diagnostic math test with 70 items and four attributes to 2,032 persons as described by Kunina-Habenicht, Rupp, and Wilhelm (2017). For the simulations, the Q-matrix for the fraction-subtraction data was used as a starting point (see, e.g., de la Torre, 2011; Table 2) with the guessing and slipping parameters set at 0.1 for all items and the number of replications set to 100. In order to keep things manageable, we only used the DINA model with identity link. All generating parameters are available from the second author upon request.

In terms of being able to reproduce simulation results exactly, a software package should allow the user to set the seed of the random number generator; in R, the base function set.seed is generally recommended for this. However, only the simGDINA routine in the GDINA package returns the same data if the seed for the random number generator is set this way, whereas the sim.din routine in the CDM package does not. Similarly, it is generally preferable that a simulation function can generate data for all models that the package can otherwise estimate. This is the case for the GDINA package, which can handle a variety of models in a unified manner, but not for the CDM package.

For these two reasons, we employed the simulation function in the GDINA package to perform the simulations. In calling the estimation functions from both packages, we used package-specific default settings as much as possible. However, we did set the maximum absolute change in item parameters to 1E-6 for determining convergence in both packages because the default settings of

Table 1. Key features of CDM and GDINA packages.

Package Feature	CDM	GDINA
Input properties		
# Items	Any	Any
Scales of item variables	Dichotomous, polytomous	Dichotomous, polytomous
# Attributes	Any	Any
Coding of attributes	Any	Any
# groups	> 1 (for MC-DINA)	1
Package feature	CDM	GDINA
Model properties		
Models estimated	DINA, DINO, GDINA, pGDINA, mg-GDINA, GDM, HO-GDINA, MC-DINA, MLC-IRT, SLCA	A-CDM, DINA, DINO, GDINA, sequential GDINA, HO-GDINA, LLM/CRUM, RRUM
Attribute hierarchies/structural model	Yes (no covariates)	Yes (no covariates)
Estimation approaches	MML with E-M algorithm, WLS, ULS, JML	MML with E-M algorithm
Simulation functionalities	DINA, DINO, GDINA; dichotomous item variables and attributes only	A-CDM, DINA, DINO, GDINA, sequential GDINA, HO-GDINA, LLM/CRUM, RRUM
Fit assessment properties		
Item fit statistics	RMSEA, RMSD, MAD, MD, entropy, Wald test, DIF test, S-X2, distance measures	Wald test, DIF test, LR test
Person fit statistics	Appropriateness statistics	None
Absolute fit statistics	Mean RMSEA, SRMSR, MAD_Cor, MAD_Residcov, MAD_Q3, MAD_aQ3	Wald test
Residual dependence statistics	X^2 statistics	Correlations, proportions, log-odds
Relative fit statistics	AIC, BIC, LR tests	AIC, BIC, deviance
Automated criticism and refinement	Q-matrix validation method (basic), skill space reduction	Q-matrix validation method (extended)
Package feature	CDM	GDINA
Output Properties		
Latent class information	Membership probabilities, classification consistency	Membership probabilities
Attribute information	Mastery probabilities, attribute classifications, attribute reliability index, tetrachoric and polychoric attribute correlations	Mastery probabilities, attribute classifications
Item parameters	Slipping, guessing, item discrimination index, item discrimination matrices	Slipping, guessing, item discrimination index
Graphical output	Expected vs. observed skill probabilities, latent class distribution, skill mastery probabilities for response patterns and attributes	Bivariate heatmap for item fit, probabilities for latent classes, mesa plot for Q-matrix validation
Additional technical properties		
Other	Based on S3 system; includes routines for identification of equivalency classes, identification of equivalent DINA models, GDD classification, creation of pseudo dichotomous items from polytomous items, creation of reduced skill space, ideal response pattern, jackknife procedures	Based on S3 system; includes routines for Q-matrix validation, model estimation and refinement, generation of all possible attribute patterns, comparison of saturated with reduced models, monotonicity checks, unrestricted Q-matrix

Table 2. Key performance information for simulated data.

Model	Package	Mean (SD) deviance	Mean bias $\hat{\delta}$	Mean RMSE $\hat{\delta}$	Mean variance ratio	Mean (SD) classification accuracy	Mean (SD) computational time (in seconds)	Mean (SD) number of iterations
DINA (small)	GDINA	5,102.9 (98.5)	0.001	0.037	NR[1]	75.1 (1.1)	0.38 (0.16)	57 (26)
	CDM	5,102.9 (98.5)	0.001	0.037	1.34	74.2 (1.2)	0.64 (0.35)	82 (58)
DINA (large)	GDINA	121,689.4 (388.7)	0.000	0.008	NR[1]	98.4 (0.1)	790.86 (223.03)	54 (15)
	CDM	121,689.4 (388.7)	0.000	0.008	1.05	98.4 (0.1)	15.79 (0.78)	9 (1)

[1]Due to irregularities in the standard errors, this number is not reported (NR).

GDINA and CDM were different. Moreover, we noted that the gdina function of the CDM package considers a maximum absolute change in deviance, but a relative criterion would be preferred because the value of the log likelihood is dependent on the number of items and persons. Finally, neither package provides a warning if the convergence criterion has not been reached after the maximum number of iterations.

We compared the deviance, estimates of the δ parameters, and overall computational performance. We also compared the standard errors of the δ parameters, but we found that these were sometimes very large for the GDINA package. These problems were not encountered with the CDM package and, although the se.delta entry of the output object is empty, SEs are indeed provided in the item entry of the output object.

Computational performance

Table 2 shows a summary of the results for the simulations. It displays the mean deviance as well as mean bias, mean RMSE, and mean variance ratio for the δ estimates. The mean variance ratio is the mean of the ratio of the variance of estimated δ parameters over replications to the mean of the estimated variance of estimated δ parameters (i.e., SE2). In short, this number gives an indication of the quality of the standard errors such that deviations from 1 can be interpreted as a percentage. If its value is larger than 1, the SEs are underestimated; if it is smaller than one, the SEs are overestimated. Finally, Table 2 shows the mean computation time and mean numbers of iterations across replications.

Table 2 shows that both packages perform very similarly on these data with the exception of two characteristics: (1) For the large dataset, the computational time for the GDINA package was much longer than that for the CDM package and (2) the GDINA package had a few irregularities in standard error estimates. As we note below, we want to caution against overgeneralizing these results however because our simulation study design was limited.

For the real dataset, following the empirical results described in Kunina et al. (2017), we estimated a model with additive main effects only:

$$\text{logit}[P(X_j = 1|\boldsymbol{\alpha})] = \delta_{j0} + \sum_{k=1}^{K_j^*} \delta_{jk}\boldsymbol{\alpha}_{lk} \tag{2}$$

This model is specified differently in both packages, namely by using the LLM ("linear logistic model") specification in the GDINA package versus the ACDM ("additive cognitive diagnosis model") specification within the gdina routine in the CDM package. The resulting log-likelihoods were quite close albeit not exactly equal (i.e., −35,850.9 for GDINA, −35,859.9 for CDM), while the estimates of the δ parameters were rather similar across the two packages as the scatterplot in Figure 1 illustrates. The agreement between attribute profiles was 98%. In terms of model fit, model

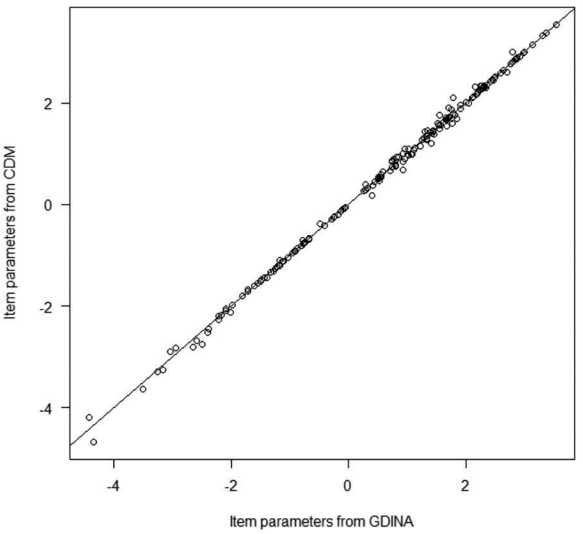

Figure 1. Item parameter estimates for real dataset from GDINA and CDM packages.

fit approaches in both packages are not quite unified across different models and require detailed knowledge about the different statistics from a data scientist. For example, the overall Wald test (modelcomp in GDINA and gdina.wald in CDM) only work if the GDINA model is used and produced different results when this model was run with the two packages.

Overall, then, the empirical performance of the routines in the two packages appears to be more similar than not, although a more comprehensive simulation study design would be needed to investigate this more thoroughly.

User-friendliness

From an applied user perspective, perhaps the most striking difference between the two packages is that GDINA includes a Shiny R graphical user interface that allows the data scientist to select key input, estimation, and output options directly without having to create a series of R commands; Figure 2 shows a screenshot of the different components of this interface. Several particularly nice features about these settings are that output can be saved into different files for exporting, that both numerical and graphical output is available for certain aspects such as response probability estimates, and that certain parameters can be expressed on different scales to aid in interpretation.

For example, item parameter estimates are displayed on four different scales, with associated standard errors if desired: (1) as equivalence-class-specific response probabilities, (2) as guessing and slipping probabilities, (3) as differences in mastery probabilities across equivalence classes, and (4) as a full matrix of response probabilities across all latent classes. As the user interface gets further expanded, it will be nice to see additional graphical representations of some of the key numerical output and see further integration of the key statistics directly along with the graphical output (e.g., display the item parameter estimates in their different parameterizations directly underneath a bar graph that shows response probabilities for latent classes; see, e.g., Bradshaw, 2016, for a few examples).

Summary

In short, working with complex parametric latent-variable models such as DCMs will always be a task for well-trained data scientists, rather than nonquantitative applied experts who are interested in

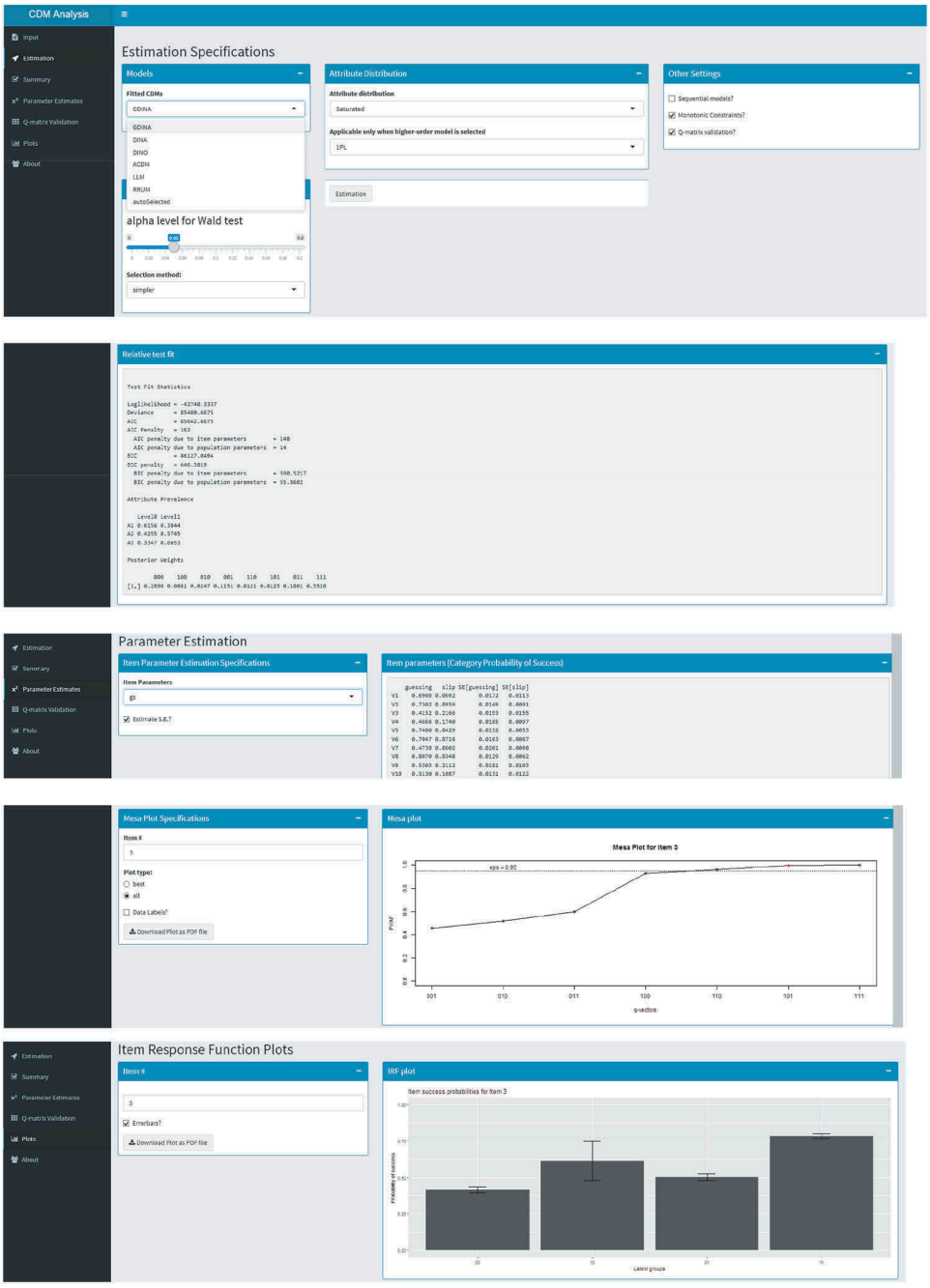

Figure 2. GDINA user interface components from the Shiny R dashboard.

merely using the results from DCMs for assessment-based decision-making. Nowadays, data scientists are proficient in programming and the computational facilities that are available via open-source and commercial packages are incredibly powerful. With the CDM and GDINA packages, data scientists have two often-aligned resources available with somewhat different technical capabilities and user features.

If we had to choose one of these two over the other at all costs, we would currently prefer the GDINA package for a few related reasons. One, the development team behind the package, most notably Dr. de la Torre, has spent many years building up an extensive repertoire of scientific insight, computational routines, and practical experience with these models. Two, related to this, it is likely that this package will continue to be revised in upcoming years, thus making the user interface and its underlying capabilities even more powerful as the foundational scientific research around the GDINA modeling framework is ongoing with dissemination at conferences and peer-reviewed outlets. Three, the user-friendly interface allows data scientists to interact with the package easily and to discuss results in interdisciplinary teams using numerical and graphical representations.

That being said, the developers of the CDM package have shown an astute awareness of the scientific state of the art as well and have worked hard to implement current computational routines into their package as well. Furthermore, based on our small example above, there were few critical differences in the performance of the two packages and, if anything, a slightly longer running time and a few unusual standard error estimates for the GDINA package for our examples. What these results show to us, more than anything, is that, for either package, care is required when certain routines are called and output is inspected since not all functionalities are necessarily available for all models. This is where the interplay of well-trained data scientists, psychometricians, and substantive assessment development experts is critical.

References

Bradshaw, L. (2016). Diagnostic classification models. In A. A. Rupp, & J. P. Leighton (Eds.), The handbook of cognition and assessment: Frameworks, methodologies, and applications (pp. 297–327). West Sussex, UK: Wiley.

de la Torre, J. (2011). The generalized DINA model framework. Psychometrika, 76, 179–199. doi:10.1007/s11336-011-9207-7

Henson, R., Templin, J., & Willse, J. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. Psychometrika, 74, 191–210. doi:10.1007/s11336-008-9089-5

Kuninia-Habenicht, O., Rupp, A. A., & Wilhelm, O. (2017). Incremental validity of multidimensional proficiency scores from diagnostic classification models: An illustration for elementary school mathematics. International Journal of Testing. doi:10.1080/15305058.2017.1291517

Ma, W., & De La Torre, J. (2017, April). GDINA [software package in R]. Available online at https://cran.r-project.org/web/packages/

Muthén, L. K., & Muthén, B. O. (2017). Mplus user's guide (8th ed.). Los Angeles, CA: Muthén & Muthén.

Robitzsch, A., Kiefer, T., George, A. C., & Uenlue, A. (2017, October). CDM [software package in R]. Available online at https://cran.r-project.org/web/packages/

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

von Davier, M. (2005). A general diagnostic model applied to language test data (Research Report RR-05-16). Princeton, NJ: Educational Testing Service.