Package 'cdmTools'

May 19, 2025

Type Package

Title Useful Tools for Cognitive Diagnosis Modeling

Version 1.0.6

Date 2025-04-24

Description Provides useful tools for cognitive diagnosis modeling (CDM). The package includes functions for empirical Q-matrix estimation and validation, such as the Hull method (Nájera, Sorrel, de la Torre, & Abad, 2021, <doi:10.1111/bmsp.12228>) and the discrete factor loading method (Wang, Song, & Ding, 2018, <doi:10.1007/978-3-319-77249-3_29>). It also contains dimensionality assessment procedures for CDM, including parallel analysis and automated fit comparison as explored in Nájera, Abad, and Sorrel (2021, <doi:10.3389/fpsyg.2021.614470>). Other relevant methods and features for CDM applications, such as the restricted DINA model (Nájera et al., 2023; <doi:10.3102/10769986231158829>), the general nonparametric classification method (Chiu et al., 2018; <doi:10.1007/s11336-017-9595-4>), and corrected estimation of the classification accuracy via multiple imputation (Kreitchmann et al., 2022; <doi:10.3758/s13428-022-01967-5>) are also available. Lastly, the package provides some useful functions for CDM simulation studies, such as random Q-matrix generation and detection of complete/identified Q-matrices.

License GPL-3

Depends R (>= 3.6.0)

Imports GDINA (>= 2.8.0), ggplot2 (>= 3.3.0), psych (>= 1.9.12), sirt (>= 3.9-4), parallel (>= 3.6.3), stats (>= 3.6.3), GPArotation (>= 2014.11-1), combinat (>= 0.0-8), fungible, foreach, doSNOW, plyr

URL https://github.com/pablo-najera/cdmTools

BugReports https://github.com/pablo-najera/cdmTools/issues

RoxygenNote 7.3.2

Encoding UTF-8

Author Pablo Nájera [aut, cre, cph],
Miguel A. Sorrel [aut, cph],
Francisco J. Abad [aut, cph],
Rodrigo S. Kreitchmann [ctb],
Kevin Santos [ctb]

Maintainer Pablo Nájera <p.najeraalvarez@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2025-05-19 08:10:02 UTC

Contents

CA.MI   *Calculate corrected classification accuracy with multiple imputation*

Description

This function calculates the test-, pattern-, and attribute-level classification accuracy indices based on integrated posterior probabilities from multiple imputed item parameters (Kreitchmann et al., 2022). The classification accuracy indices are the ones developed by Iaconangelo (2017) and Wang et al. (2015). It is only applicable to dichotomous attributes. The function is built upon the CA function from the GDINA package (Ma & de la Torre, 2020).

Usage

```
CA.MI(fit, what = "EAP", R = 500, n.cores = 1, verbose = TRUE, seed = NULL)
```

Arguments

`fit`	An object of class `RDINA` or `GDINA` (Ma & de la Torre, 2020).
`what`	What attribute estimates are used? The default is `"EAP"`.
`R`	Number of bootstrap samples and imputations. The default is 500.
`n.cores`	Number of processors to use to speed up multiple imputation. The default is 2.
`verbose`	Show progress. The default is `TRUE`.
`seed`	A seed for obtaining consistent results. If `NULL`, no seed is used. The default is `NULL`.

Value

`CA.MI` returns an object of class `CA`, with a list of elements:

`tau` Estimated test-level classification accuracy, see Iaconangelo (2017, Eq 2.2) (`vector`).

`tau_l` Estimated pattern-level classification accuracy, see Iaconangelo (2017, p. 13) (`vector`).

`tau_k` Estimated attribute-level classification accuracy, see Wang, et al (2015, p. 461 Eq 6) (`vector`).

`CCM` Conditional classification matrix, see Iaconangelo (2017, p. 13) (`matrix`).

Author(s)

Rodrigo S. Kreitchmann, Universidad Nacional de Educación a Distancia

References

Iaconangelo, C.(2017). *Uses of classification error probabilities in the three-step approach to estimating cognitive diagnosis models*. (Unpublished doctoral dissertation). New Brunswick, NJ: Rutgers University.

Kreitchmann, R. S., de la Torre, J., Sorrel, M. A., Nájera, P., & Abad, F. J. (2022). Improving reliability estimation in cognitive diagnosis modeling. *Behavior Research Methods*. https://doi.org/10.3758/s13428-022-01967-5

Ma, W., & de la Torre, J. (2020). GDINA: An R package for cognitive diagnosis modeling. *Journal of Statistical Software*, *93*(14). https://doi.org/10.18637/jss.v093.i14

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

Examples

```
library(GDINA)
dat <- sim10GDINA$simdat[1:100,]
Q <- sim10GDINA$simQ
fit <- GDINA(dat = dat, Q = Q, model = "GDINA")
ca.mi <- CA.MI(fit)
ca.mi
```

---

estQ   *Empirical Q-matrix estimation*

Description

Empirical Q-matrix estimation based on the *discrete factor loading* method (Wang, Song, & Ding, 2018) as used in Nájera, Abad, and Sorrel (2021). Apart from the conventional dichotomization criteria, the procedure based on loading differences described in Garcia-Garzon, Abad, and Garrido (2018) is also available. Furthermore, the bagging bootstrap implementation (Xu & Shang, 2018) can be applied; it is recommended when working with small sample sizes. The psych package (Revelle, 2020) is used for estimating the required exploratory factor analysis (EFA).

Usage

```
estQ(
  r,
  K,
  n.obs = NULL,
  criterion = "row",
  boot = FALSE,
  efa.args = list(cor = "tet", rotation = "oblimin", fm = "uls"),
  boot.args = list(N = 0.8, R = 100, verbose = TRUE, seed = NULL)
)
```

Arguments

r	A correlation matrix or raw data (matrix or data.frame). If a correlation matrix is used, it must have dimensions J items × J items. Please note that tetrachoric or polychoric correlations should be used when working with dichotomous or polytomous items, respectively. If raw data is used, it must have dimensions N individuals × J items. Missing values need to be coded as NA.
K	Number of attributes to use.
n.obs	Number of individuals if r is a correlation matrix. If n.obs is provided, r will be treated as a correlation matrix. Use NULL if r is raw data. The default is NULL.
criterion	Dichotomization criterion to transform the factor loading matrix into the Q-matrix. The possible options include "row" (for row means), "col" (for column means), "loaddiff" (for the procedure based on loading differences), or a value between 0 and 1 (for a specific threshold). The default is "row".
boot	Apply the bagging bootstrap implementation? Only available if r is raw data. If FALSE, the EFA will be applied once using the whole sample size. If TRUE, several EFAs will be applied with different subsamples; the estimated Q-matrix will be dichotomized from the bootstrapped Q-matrix, but the EFA fit indices, factor loadings, and communalities will be computed from the EFA with the whole sample size. The default is FALSE.
efa.args	A list of arguments for the EFA estimation:

cor Type of correlations to use. It includes "cor" (for Pearson correlations) and "tet" (for tetrachoric/polychoric correlations), among others. See fa function from the psych R package for additional details. The default is "tet".

rotation Rotation procedure to use. It includes "oblimin", "varimax", and "promax", among others. An oblique rotation procedure is usually recommended. See fa function from the psych R package for additional details. The default is "oblimin".

fm Factoring method to use. It includes "uls" (for unweighted least squares), "ml" (for maximum likelihood), and "wls" (for weighted least squares), among others. See fa function from the psych R package for additional details. The default is "uls".

boot.args   A list of arguments for the bagging bootstrap implementation (ignored if boot = FALSE):

N Sample size (or proportion of the total sample size, if lower than 1) to use in each bootstrap replication. The default is .8.

R Number of bootstrap replications. The default is 100.

verbose Show progress? The default is TRUE.

seed A seed for obtaining consistent results. If NULL, no seed is used. The default is NULL.

Value

estQ returns an object of class estQ.

est.Q Estimated Q-matrix (matrix).

efa.loads Factor loading matrix (matrix).

efa.comm EFA communalities (vector).

efa.fit EFA model fit indices (vector).

boot.Q Bagging bootstrap Q-matrix before dichotomization. Only if boot = TRUE (matrix).

is.Qid Q-matrix identifiability information (list).

specifications Function call specifications (list).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas

References

Garcia-Garzon, E., Abad, F. J., & Garrido, L. E. (2018). Improving bi-factor exploratory modelling: Empirical target rotation based on loading differences. *Methodology*, *15*, 45–55. https://doi.org/10.1027/1614-2241/a000163

Nájera, P., Abad, F. J., & Sorrel, M. A. (2021). Determining the number of attributes in cognitive diagnosis modeling. *Frontiers in Psychology*, *12*:614470. https://doi.org/10.3389/fpsyg.2021.614470

Revelle, W. (2019). *psych: Procedures for Psychological, Psychometric, and Personality Research*. R package version 1.9.12. https://CRAN.R-project.org/package=psych.

Wang, W., Song, L., & Ding, S. (2018). An exploratory discrete factor loading method for Q-matrix specification in cognitive diagnosis models. In: M. Wilberg, S. Culpepper, R. Janssen, J. Gonzalez, & D. Molenaar (Eds.), *Quantitative Psychology. IMPS 2017. Springer Proceedings in Mathematics & Statistics* (Vol. 233, pp. 351–362). Springer.

Xu, G., & Shang, Z. (2018). Identifying latent structures in restricted latent class models. *Journal of the American Statistical Association*, *113*, 1284–1295. https://doi.org/10.1080/01621459.2017.1340889

Examples

```
library(GDINA)
dat <- sim30GDINA$simdat
Q <- sim30GDINA$simQ

#-----------------------------
# Using default specifications
#-----------------------------
sugQ1 <- estQ(r = dat, K = 5) # Estimate Q-matrix
sugQ1$est.Q <- orderQ(sugQ1$est.Q, Q)$order.Q # Reorder Q-matrix attributes
mean(sugQ1$est.Q == Q) # Check similarity with the generating Q-matrix

#-----------------------------------
# Using the bagging bootstrap method
#-----------------------------------
# In boot.args argument, R >= 100 is recommended (R = 20 is here used for illustration purposes)
sugQ2 <- estQ(r = dat, K = 5, boot = TRUE, boot.args = list(R = 20, seed = 123)) # Estimate Q-matrix
sugQ2$est.Q <- orderQ(sugQ2$est.Q, Q)$order.Q # Reorder Q-matrix attributes
sugQ2$boot.Q # Proportion of replicas a q-entry was specified in the estimated Q-matrix
mean(sugQ2$est.Q == Q) # Check similarity with the generating Q-matrix
```

FCGDINA	*G-DINA model for forced-choice blocks*

Description

Estimation of the G-DINA model for forced-choice responses according to Nájera et al. (2024). Block polarity (i.e., statement direction) and initial values for parameters can be specified to determine the design of the forced-choice blocks. The GDINA package (Ma & de la Torre, 2020) is used to estimate the model via expectation maximumation (EM) algorithm if no priors are used. To estimate the *forced-choice diagnostic classification model* (FC-DCM; Huang, 2023) using Bayes modal estimation, please check the codes provided in https://osf.io/h6x9e/. Only unidimensional statements (i.e., bidimensional blocks) are currently supported.

Usage

```
FCGDINA(
  dat,
  Q,
  polarity = NULL,
```

```
  polarity.initial = 1e-04,
  att.dist = "saturated",
  att.prior = NULL,
  verbose = 1,
  higher.order = list(),
  catprob.parm = NULL,
  control = list()
)
```

Arguments

dat   A *N* individuals x *J* items (matrix or data.frame). Missing values need to be coded as NA. Caution is advised if missing data are present.

Q   A *F* blocks x *K* attributes Q-matrix (matrix or data.frame). Each q-vector must measure two attributes, reflecting the attributes measured by its statements.

polarity   A *F* blocks x 2 (matrix or data.frame). Each row reflects the direction of the first and second statement, where 1 and -1 corresponds to direct and inverse statements, respectively. Default is NULL, denoting that all statements are direct.

polarity.initial
                A numeric value that indicates the initial value for the estimation of the probability of endorsement for the latent group whose ideal response is equal to 0. The initial value for the latent group whose ideal response is equal to 1 will be 1 - polarity.initial. The initial value for latent groups without a clear ideal response is always equal to 0.5. This argument is ignored if catprob.parm != NULL. Default is 1e-4.

att.dist   How is the joint attribute distribution estimated? It can be "saturated", "higher.order", "fixed", "independent", and "loglinear". Only considered if EM estimation is used. Default is "saturated". See the GDINA package documentation for more information.

att.prior   A vector of length 2^K to specicy attribute prior distribution for the latent classes. Only considered if EM estimation is used. Default is NULL. See the GDINA package documentation for more information.

verbose   How to print calibration information after each EM iteration? Can be 0, 1 or 2, indicating to print no information, information for current iteration, or information for all iterations.

higher.order   A list specifying the higher-order joint attribute distribution with the following components. Only considered if EM estimation is used. See the GDINA package documentation for more information.

catprob.parm   A list of initial values for probabilities of endorsement for each nonzero category. Default is NULL. See the GDINA package documentation for more information.

control   A list of control parameters. Only considered if EM estimation is used. See the GDINA package documentation for more information.

Value

FCGDINA returns an object of class FCGDINA.

GDINA.obj Estimation output from the GDINA function of the GDINA.MJ (Ma & Jiang, 2021) function, depending on whether EM or BM estimation has been used (`list`).

technical Information about initial values (`list`).

specifications Function call specifications (`list`).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas

References

Huang, H.-Y. (2023). Diagnostic Classification Model for Forced-Choice Items and Noncognitive Tests. *Educational and Psychological Measurement*, *83*(1), 146-180. https://doi.org/10.1177/00131644211069906

Ma, W., & de la Torre, J. (2020). GDINA: An R package for cognitive diagnosis modeling. *Journal of Statistical Software*, *93*(14). https://doi.org/10.18637/jss.v093.i14

Nájera, P., Kreitchmann, R. S., Escudero, S., Abad, F. J., de la Torre, J., & Sorrel, M. A. (2025). A General Diagnostic Modeling Framework for Forced-Choice Assessments. *British Journal of Mathematical and Statistical Psychology*.

Examples

```
library(GDINA)
set.seed(123)
# Q-matrix for the unidimensional statements
Q.items <- do.call("rbind", replicate(5, diag(5), simplify = FALSE))
# Guessing and slip
GS <- cbind(runif(n = nrow(Q.items), min = 0.1, max = 0.3),
            runif(n = nrow(Q.items), min = 0.1, max = 0.3))
n.blocks <- 30 # Number of forced-choice blocks


#----------------------------------------------------------------------------------------
# Illustration with simulated data using only direct statements (i.e., homopolar blocks)
#----------------------------------------------------------------------------------------

# Block polarity (1 = direct statement; -1 = indirect statement)
polarity <- matrix(1, nrow = n.blocks, ncol = 2)
sim <- simFCGDINA(N = 1000, Q.items, n.blocks = n.blocks, polarity = polarity,
                  model = "GDINA", GDINA.args = list(GS = GS), seed = 123)
Q <- sim$Q # Generated Q-matrix of forced-choice blocks
dat <- sim$dat # Generated responses
att <- sim$att # Generated attribute profiles

fit <- FCGDINA(dat = dat, Q = Q, polarity = polarity) # Fit the G-DINA model with EM estimation
ClassRate(personparm(fit$GDINA.obj), att) # Classification accuracy

#-----------------------------------------------------------------------------------------
# Illustration with simulated data using some inverse stataments (i.e., heteropolar blocks)
#-----------------------------------------------------------------------------------------

polarity <- matrix(1, nrow = n.blocks, ncol = 2)
```

```
# Including 15 inverse statements
polarity[sample(x = 1:(2*n.blocks), size = 15, replace = FALSE)] <- -1
sim <- simFCGDINA(N = 1000, Q.items, n.blocks = n.blocks, polarity = polarity,
                  model = "GDINA", GDINA.args = list(GS = GS), seed = 123)
Q <- sim$Q
dat <- sim$dat
att <- sim$att

fit <- FCGDINA(dat = dat, Q = Q, polarity = polarity)
ClassRate(personparm(fit$GDINA.obj), att)
```

---

genQ   *Generate Q-matrix*

Description

Generates a Q-matrix. The criteria from Chen, Liu, Xu, & Ying (2015) and Xu & Shang (2018) can be used to generate identifiable Q-matrices. Only binary Q-matrix are supported so far. Useful for simulation studies.

Usage

```
genQ(J, K, Kj, I = 2, min.JK = 3, max.Kcor = 1, Qid = "none", seed = NULL)
```

Arguments

J	Number of items.
K	Number of attributes.
Kj	A vector specifying the number (or proportion, if summing up to 1) of items measuring 1, 2, 3, ..., attributes. The first element of the vector determines the number (or proportion) of items measuring 1 attribute, and so on. See Examples.
I	Number of identity matrices to include in the Q-matrix (up to column permutation). The default is 2.
min.JK	Minimum number of items measuring each attribute. It can be overwritten by I, if I is higher than min.JK. The default is 3.
max.Kcor	Maximum allowed tetrachoric correlation among the columns to avoid overlapping (Nájera, Sorrel, de la Torre, & Abad, 2020). The default is 1.
Qid	Assure that the generated Q-matrix is generically identifiable. It includes "none" (for no identifiability assurance), "DINA", "DINO", or "others" (for other CDMs identifiability). The default is "none".
seed	A seed for obtaining consistent results. If NULL, no seed is used. The default is NULL.

Value

genQ returns an object of class genQ.

gen.Q The generated Q-matrix (matrix).

JK Number of items measuring each attribute (vector).

Kcor Tetrachoric correlations among the columns (matrix).

is.Qid Q-matrix identifiability information (list).

specifications Function call specifications (list).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas

References

Chen, Y., Liu, J., Xu, G., & Ying, Z. (2015). Statistical analysis of Q-matrix based diagnostic classification models. *Journal of the American Statistical Association*, *110*, 850-866. https://doi.org/10.1080/01621459.2014.9348

Nájera, P., Sorrel, M. A., de la Torre, J., & Abad, F. J. (2020). Balancing fit and parsimony to improve Q-matrix validation. *British Journal of Mathematical and Statistical Psychology*. https://doi.org/10.1111/bmsp.12228

Xu, G., & Shang, Z. (2018). Identifying latent structures in restricted latent class models. *Journal of the American Statistical Association*, *113*, 1284-1295. https://doi.org/10.1080/01621459.2017.1340889

Examples

```
Kj <- c(15, 10, 0, 5) # 15 one-att, 10 2-atts, 0 3-atts, and 5 four-atts items
Q <- genQ(J = 30, K = 4, Kj = Kj, Qid = "others", seed = 123)
```

GNPC *General nonparametric classification method*

Description

Attribute profile estimation using the *general nonparametric classification method* (GNPC; Chiu, Sun, & Bian, 2018). The GNPC can be considered as a robust alternative to the parametric G-DINA model with low sample sizes. The AlphaNP function from the NPCD package (Zheng & Chiu, 2019; Chiu, Sun, & Bian, 2018) using weighted Hamming distances is used to initiate the procedure.

Usage

```
GNPC(
  dat,
  Q,
  initiate = "AND",
  min.change = 0.001,
  maxitr = 1000,
  verbose = TRUE
)
```

Arguments

dat	A *N* individuals x *J* items (matrix or data.frame). Missing values need to be coded as NA. Caution is advised if missing data are present.
Q	A *J* items x *K* attributes Q-matrix (matrix or data.frame).
initiate	Should the conjunctive ("AND") or disjunctive ("OR") NPC be used to initiate the procedure? Default is "AND".
min.change	Minimum proportion of modified attribute profiles to use as a stopping criterion. Default is .001.
maxitr	Maximum number of iterations. Default is 1000.
verbose	Print information after each iteration. Default is TRUE.

Value

GNPC returns an object of class GNPC.

alpha.est Estimated attribute profiles (matrix).

loss.matrix The distances between the weighted ideal responses from each latent class (rows) and examinees' observed responses (columns) (matrix).

eta.w The weighted ideal responses for each latent class (rows) on each item (columns) (matrix).

w The estimated weights, used to compute the weighted ideal responses (matrix).

n.ite Number of iterations required to achieve convergence (double).

hist.change Proportion of modified attribute profiles in each iteration (vector).

specifications Function call specifications (list).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas

References

Chiu, C.-Y., & Douglas, J. (2013). A nonparametric approach to cognitive diagnosis by proximity to ideal response patterns. *Journal of Classification*, *30*, 225-250. DOI: 10.1007/s00357-013-9132-9

Chiu, C.-Y., Sun, Y., & Bian, Y. (2018). Cognitive diagnosis for small education programs: The general nonparametric classification method. *Psychometrika*, *83*, 355-375. DOI: 10.1007/s11336-017-9595-4

Zheng, Y., & Chiu, C.-Y. (2019). *NPCD: Nonparametric methods for cognitive diagnosis*. R package version 1.0-11. https://cran.r-project.org/web/packages/NPCD/.

Examples

```
library(GDINA)
Q <- sim30GDINA$simQ # Q-matrix
K <- ncol(Q)
J <- nrow(Q)
set.seed(123)
GS <- data.frame(guessing = rep(0.1, J), slip = rep(0.1, J))
```

```
sim <- simGDINA(200, Q, GS)
simdat <- sim$dat # Simulated data
simatt <- sim$attribute # Generating attributes
fit.GNPC <- GNPC(simdat, Q) # Apply the GNPC method
ClassRate(fit.GNPC$alpha.est, simatt) # Check classification accuracy
```

---

is.CDMid   *Check whether a CDM is identified*

Description

Uses a post-hoc simulation approach to check whether a cognitive diagnosis model is identified (i.e., all latent classes are distinguishable; de la Torre et al., 2023).

Usage

```
is.CDMid(fit, N = 10000, timesJ = 20, Wald = FALSE, verbose = TRUE)
```

Arguments

fit	An object of class RDINA or GDINA (Ma & de la Torre, 2020).
N	A *numeric* value that indicates the number of respondents to simulate. Default is 10000.
timesJ	A *numeric* value that indicates the number of times the test length is multiplied. Default is 20.
Wald	A logical value that indicates whether the Wald method should be used to find the best model for each item (only applicable if fit is of class GDINA). Default is FALSE.
verbose	A logical value that indicates whether information about the process should be printed or not. Default is TRUE.

Value

is.CDMid returns an object of class is.CDMid.

total Overall classification accuracy (CCA) and number of posterior multiple modes (PMM). A CCA = 1 indicates that all latent classes are identified (vector).

class Classification accuracy (CCA) and number of posterior multiple modes (PMM) for each latent class. A CCA = 1 indicates that the latent class is identified (data.frame).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas

References

de la Torre, J., Sorrel, M. A., & Nájera, P. (2023, July). *Cognitive diagnosis modeling*. Workshop at the VII International Psychometric Summer School "Applied Psychometrics in Psychology and Education". Yerevan, Armenia.

Examples

```
library(GDINA)
dat <- sim30GDINA$simdat
Q <- sim30GDINA$simQ
fit <- GDINA(dat, Q)
id <- is.CDMid(fit)
```

---

is.Qid	*Check whether a Q-matrix is identifiable*

Description

Checks whether a Q-matrix fulfills the conditions for strict and generic identifiability according to Gu & Xu (2021).

Usage

```
is.Qid(Q, model)
```

Arguments

Q	A *J* items x *K* attributes Q-matrix (`matrix` or `data.frame`).
model	CDM to be considered. It includes `"DINA"`, `"DINO"`, or `"others"` (for other CDMs; e.g., G-DINA, A-CDM).

Value

is.Qid returns an object of class `is.Qid`.

strict Is the Q-matrix strictly identifiable? (`logical`).

generic Is the Q-matrix generically identifiable? (`logical`).

conditions Identifiability criteria and whether they are fulfilled or not (`vector`).

specifications Function call specifications (`list`).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas
Miguel A. Sorrel, Universidad Autónoma de Madrid

References

Gu, Y., & Xu, G. (2021). Sufficient and necessary conditions for the identifiability of the Q-matrix. *Statistica Sinica*, *31*, 449-472. https://www.jstor.org/stable/26969691

Examples

```
Kj <- c(15, 10, 0, 5)
Q <- genQ(J = 30, K = 4, Kj = Kj, Qid = "others", seed = 123)$gen.Q
idQ <- is.Qid(Q, model = "DINA")
```

---

missQ   *Introduce random misspecifications in Q-matrix*

Description

Introduces random misspecifications in a Q-matrix. Only binary Q-matrix are supported so far. Useful for simulation studies.

Usage

```
missQ(Q, qjk, retainJ = 0, Qid = "none", seed = NULL)
```

Arguments

Q	A *J* items x *K* attributes Q-matrix (`matrix` or `data.frame`).
qjk	Number (or proportion, if lower than 1) of q-entries to modify in the Q-matrix.
retainJ	Number of items to retain (i.e., not modify) in the Q-matrix. It will retain the first `retainJ` items. It is useful for assuring the completeness of the misspecified Q-matrix if the first items conform one or more identity matrices. The default is 0.
Qid	Assure that the generated Q-matrix is generically identifiable. It includes `"none"` (for no identifiability assurance), `"DINA"`, `"DINO"`, or `"others"` (for other CDMs identifiability). The default is `"none"`.
seed	A seed for obtaining consistent results. If NULL, no seed is used. The default is NULL.

Value

`missQ` returns an object of class `missQ`.

miss.Q The misspecified Q-matrix (`matrix`).

Q The input (true) Q-matrix (`matrix`).

JK Number of items measuring each attribute (`vector`).

Kcor Tetrachoric correlations among the columns (`matrix`).

is.Qid Q-matrix identifiability information (`list`).

specifications Function call specifications (`list`).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas

References

Xu, G., & Shang, Z. (2018). Identifying latent structures in restricted latent class models. *Journal of the American Statistical Association*, *113*, 1284-1295. https://doi.org/10.1080/01621459.2017.1340889

Examples

```
Kj <- c(15, 10, 0, 5) # 15 one-att, 10 2-atts, 0 3-atts, and 5 four-atts items
Q <- genQ(J = 30, K = 4, Kj = Kj, Qid = "others", seed = 123)
miss.Q <- missQ(Q = Q$gen.Q, qjk = .20, retainJ = 4, seed = 123)
```

---

modelcompK   *CDM fit comparison - dimensionality assessment method*

Description

A procedure for determining the number of attributes underlying CDM using model fit comparison. For each number of attributes under exploration, a Q-matrix is estimated from the data using the *discrete factor loading* method (Wang, Song, & Ding, 2018), which can be further validated using the *Hull* method (Nájera, Sorrel, de la Torre, & Abad, 2020). Then, a CDM is fitted to the data using the resulting Q-matrix, and several fit indices are computed. After the desired range of number of attributes has been explored, the fit indices are compared. A suggested number of attributes is given for each fit index. The AIC index should be preferred among the other fit indices. For further details, see Nájera, Abad, & Sorrel (2021). This function can be also used by directly providing different Q-matrices (instead of estimating them from the data) in order to compare their fit and select the most appropriate Q-matrix. Note that, if Q-matrices are provided, this function will no longer serve as a dimensionality assessment method, but just as an automated model comparison procedure.

Usage

```
modelcompK(
  dat,
  exploreK = 1:7,
  Qs = NULL,
  stop = "none",
  val.Q = TRUE,
 estQ.args = list(criterion = "row", cor = "tet", rotation = "oblimin", fm = "uls"),
 valQ.args = list(index = "PVAF", iterative = "test.att", maxitr = 5, CDMconv = 0.01),
  verbose = TRUE
)
```

Arguments

dat	A *N* individuals x *J* items (matrix or data.frame). Missing values need to be coded as NA.
exploreK	Number of attributes to explore. The default is from 1 to 7 attributes.
Qs	A list of Q-matrices to compare in terms of fit. If Qs is used, exploreK is ignored.
stop	A fit index to use for stopping the procedure if a model leads to worse fit than a simpler one. This can be useful for saving time without exploring the whole exploreK when it is probable that the correct dimensionality has been already visited. It includes "AIC", "BIC", "CAIC", "SABIC", "M2", "SRMSR", "RMSEA2", or "sig.item.pairs". The latter represents the number of items that show bad fit with at least another item based on the transformed correlations (see itemfit function in the GDINA package; Ma & de la Torre, 2020). It can be also "none", which means that the whole exploreK will be examined. The default is "none".
val.Q	Validate the estimated Q-matrices using the *Hull* method? Note that validating the Q-matrix is expected to increase its quality, but the computation time will increase. The default is TRUE.
estQ.args	A list of arguments for the *discrete factor loading* empirical Q-matrix estimation method (see the estQ function):
	criterion Dichotomization criterion to transform the factor loading matrix into the Q-matrix. The possible options include "row" (for row means), "col" (for column means), "loaddiff" (for the procedure based on loading differences), or a value between 0 and 1 (for a specific threshold). The default is "row".
	cor Type of correlations to use. It includes "cor" (for Pearson correlations) and "tet" (for tetrachoric/polychoric correlations), among others. See fa function from the psych R package for additional details. The default is "tet".
	rotation Rotation procedure to use. It includes "oblimin", "varimax", and "promax", among others. An oblique rotation procedure is usually recommended. See fa function from the psych R package for additional details. The default is "oblimin".
	fm Factoring method to use. It includes "uls" (for unweighted least squares), "ml" (for maximum likelihood), and "wls" (for weighted least squares), among others. See fa function from the psych R package for additional details. The default is "uls".
valQ.args	A list of arguments for the *Hull* empirical Q-matrix validation method. Only applicable if valQ = TRUE (see the valQ function):
	index What index to use. It includes "PVAF" or "R2". The default is "PVAF".
	iterative (Iterative) implementation procedure. It includes "none" (for non-iterative), "test" (for test-level iterations), "test.att" (for test-level iterations modifying the least possible amount of q-entries in each iteration), and "item" (for item-level iterations). The default is "test.att".
	maxitr Maximum number of iterations if an iterative procedure has been selected. The default is 5.

CDMconv Convergence criteria for the CDM estimations between iterations (only if an iterative procedure has been selected). The default is 0.01.

verbose Show progress? The default is TRUE.

Value

modelcompK returns an object of class modelcompK.

sug.K The suggested number of attributes for each fit index (vector). Only if Qs = NULL.

sel.Q The suggested Q-matrix for each fit index (vector).

fit The fit indices for each fitted model (matrix).

exp.exploreK Explored dimensionality (vector). It can be different from exploreK if stop has been used.

usedQ Q-matrices used to fit each model (list). They will be the estimated (and validated) Q-matrices if Qs = NULL. Otherwise, they will be Qs.

specifications Function call specifications (list).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas
Miguel A. Sorrel, Universidad Autónoma de Madrid
Francisco J. Abad, Universidad Autónoma de Madrid

References

Ma, W., & de la Torre, J. (2020). GDINA: An R package for cognitive diagnosis modeling. *Journal of Statistical Software*, *93*(14). https://doi.org/10.18637/jss.v093.i14

Nájera, P., Abad, F. J., & Sorrel, M. A. (2021). Determining the number of attributes in cognitive diagnosis modeling. *Frontiers in Psychology*, *12*:614470. https://doi.org/10.3389/fpsyg.2021.614470

Nájera, P., Sorrel, M. A., de la Torre, J., & Abad, F. J. (2020). Balancing fit and parsimony to improve Q-matrix validation. *British Journal of Mathematical and Statistical Psychology*. https://doi.org/10.1111/bmsp.12228

Wang, W., Song, L., & Ding, S. (2018). An exploratory discrete factor loading method for Q-matrix specification in cognitive diagnosis models. In: M. Wilberg, S. Culpepper, R. Janssen, J. González, & D. Molenaar (Eds.), *Quantitative Psychology. IMPS 2017. Springer Proceedings in Mathematics & Statistics* (Vol. 233, pp. 351-362). Springer.

Examples

```
library(GDINA)
dat <- sim30GDINA$simdat
Q <- sim30GDINA$simQ

#------------------------------------
# Assess dimensionality from CDM data
#------------------------------------
mcK <- modelcompK(dat = dat, exploreK = 4:7, stop = "AIC", val.Q = TRUE, verbose = TRUE)
mcK$sug.K # Check suggested number of attributes by each fit index
mcK$fit # Check fit indices for each K explored
```

```
sug.Q <- mcK$usedQ[[paste0("K", mcK$sug.K["AIC"])]] # Suggested Q-matrix by AIC
sug.Q <- orderQ(sug.Q, Q)$order.Q # Reorder Q-matrix attributes
mean(sug.Q == Q) # Check similarity with the generating Q-matrix

#-------------------------------------------------
# Automatic fit comparison of competing Q-matrices
#-------------------------------------------------
trueQ <- Q
missQ1 <- missQ(Q, .10, seed = 123)$miss.Q
missQ2 <- missQ(Q, .20, seed = 456)$miss.Q
missQ3 <- missQ(Q, .30, seed = 789)$miss.Q
Qs <- list(trueQ, missQ1, missQ2, missQ3)
mc <- modelcompK(dat = dat, Qs = Qs, verbose = TRUE)
mc$sel.Q # Best-fitting Q-matrix for each fit index
mc$fit # Check fit indices for each Q explored
```

---

orderQ   *Reorder Q-matrix columns*

Description

Reorders Q-matrix columns according to a target matrix (e.g., another Q-matrix). Specifically, it
provides a reordered Q-matrix which columns show the lowest possible average Tucker index con-
gruent coefficient with the target columns. Reordering a Q-matrix is alike relabeling the attributes
and it does not change the model. Useful for simulation studies (e.g., comparing a validated Q-
matrix with the generating Q-matrix).

Usage

```
orderQ(Q, target)
```

Arguments

Q	A *J* items x *K* attributes Q-matrix (`matrix` or `data.frame`). This is the Q-matrix that will be reordered.
target	A *J* items x *K* attributes Q-matrix (`matrix` or `data.frame`). This could be the "true", generating Q-matrix.

Value

orderQ returns an object of class orderQ.

order.Q The reordered Q-matrix (`matrix`).

configs Comparison information between the different column configurations of the Q-matrix and
the target Q-matrix, including the average absolute difference and the average Tucker index
of factor congruence (`matrix`). The function will not look for all possible specifications if a
perfect match is found.

specifications Function call specifications (`list`).

Author(s)

Francisco J. Abad, Universidad Autónoma de Madrid
Pablo Nájera, Universidad Pontificia Comillas

Examples

```
library(GDINA)
dat <- sim30GDINA$simdat
Q <- sim30GDINA$simQ
sugQ1 <- estQ(r = dat, K = 5) # Estimate Q-matrix
sugQ1$est.Q <- orderQ(sugQ1$est.Q, Q)$order.Q # Reorder Q-matrix attributes
mean(sugQ1$est.Q == Q) # Check similarity with the generating Q-matrix
```

---

paK   *Parallel analysis - dimensionality assessment method*

Description

Parallel analysis with column permutation (i.e., resampling) as used in Nájera, Abad, & Sorrel (2021). It is recommended to use principal components, Pearson correlations, and mean criterion (Garrido, Abad, & Ponsoda, 2013; Nájera, Abad, & Sorrel, 2021). The parallel analysis based on principal axis factor analysis is conducted using the `fa.parallel` function of the `psych` R package (Revelle, 2020). The tetrachoric correlations are efficiently estimated using the `sirt` R package (Robitzsch, 2020). The graph is made with the `ggplot2` package (Wickham et al., 2020).

Usage

```
paK(
  dat,
  R = 100,
  fa = "pc",
  cor = "both",
  cutoff = "mean",
  fm = "uls",
  plot = TRUE,
  verbose = TRUE,
  seed = NULL
)
```

Arguments

dat   A *N* individuals x *J* items (`matrix` or `data.frame`). Missing values need to be coded as `NA`.

R   Number of resampled datasets (i.e., replications) to generate. The default is 100.

fa	Extraction method to use. It includes "pc" (for principal components analysis), "fa" (for principal axis factor analysis), and "both". The default is "pc".
cor	What type of correlations to use. It includes "cor" (for Pearson correlations), "tet" (for tetrachoric/polychoric correlations), and "both". The default is "both".
cutoff	What criterion to use as the cutoff. It can be "mean" (for the average generated eigenvalues) or a value between 0 and 100 (for a percentile). A vector with several criteria can be used. The default is "mean".
fm	Factoring method to use. It includes "uls" (for unweighted least squares), "ml" (for maximum likelihood), and "wls" (for weighted least squares), among others. The default is "uls".
plot	Print the parallel analysis plot? Note that the plot might be messy if many variants are requested. The default is TRUE.
verbose	progress. The default is TRUE.
seed	A seed for obtaining consistent results. If NULL, no seed is used. The default is NULL.

Value

paK returns an object of class paK.

sug.K The suggested number of attributes for each variant (vector).

e.values The sample and reference eigenvalues (matrix).

plot The parallel analysis plot. Only if plot = TRUE (plot).

specifications Function call specifications (list).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas
Miguel A. Sorrel, Universidad Autónoma de Madrid
Francisco J. Abad, Universidad Autónoma de Madrid

References

Garrido, L. E., Abad, F. J., & Ponsoda, V. (2013). A new look at Horn's parallel analysis with ordinal variables. *Psychological Methods*, *18*, 454-474. https://doi.org/10.1037/a0030005

Nájera, P., Abad, F. J., & Sorrel, M. A. (2021). Determining the number of attributes in cognitive diagnosis modeling. *Frontiers in Psychology*, *12*:614470. https://doi.org/10.3389/fpsyg.2021.614470

Revelle, W. (2019). *psych: Procedures for Psychological, Psychometric, and Personality Research*. R package version 1.9.12. https://CRAN.R-project.org/package=psych.

Robitzsch, A. (2020). *sirt: Supplementary Item Response Theory Models*. R package version 3.9-4. https://CRAN.R-project.org/package=sirt.

Wickham, H., et al. (2020). *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. R package version 3.3.2. https://CRAN.R-project.org/package=ggplot2.

Examples

```
library(GDINA)
dat <- sim30GDINA$simdat
Q <- sim30GDINA$simQ
# In paK, R = 100 is recommended (R = 30 is here used for illustration purposes)
pa.K <- paK(dat = dat, R = 30, fa = "pc", cutoff = c("mean", 95), plot = TRUE, seed = 123)
pa.K$sug.K # Check suggested number of attributes by each parallel analysis variant
pa.K$e.values # Check eigenvalues
pa.K$plot # Show parallel analysis plot
```

---

personFit	*Calculate standardized log-likelihood statistic (lZ) for person fit evaluation*

Description

This function calculates the standardized log-likelihood statistic (lZ; Cui & Li, 2015; Drasgow et al. 1985) and the proposals for correcting its distribution discussed in Santos et al. (2019).

Usage

```
personFit(fit, att.est = "MLE", sig.level = 0.05, p.adjust.method = "BH")
```

Arguments

fit	An object of class RDINA or GDINA (Ma & de la Torre, 2020).
att.est	What attribute estimates are used? The default is "MLE".
sig.level	Scalar numeric. Alpha level for decision. Default is 0.05.
p.adjust.method	
	Scalar character. Correction method for p-values. Possible values include "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", and "none". See p.adjust function from the stats R package for additional details. Default is BH.

Value

personFit returns an object of class personFit, with a list of elements:

stat Person fit statistics (data.frame).

p p-values (two-sided test) for the person fit statistics (data.frame).

sigp Scalar vectors denoting the examinees for which the person fit statitic is significant (p-value) (list).

sigadjp Scalar vectors denoting the examinees for which the person fit statitic is significant (adjusted p-value) (list).

Author(s)

Miguel A. Sorrel, Universidad Autónoma de Madrid,
Kevin Santos, University of the Philippines,
Pablo Nájera, Universidad Pontificia Comillas

References

Cui, Y., & Li, J. (2015). Evaluating person fit for cognitive diagnostic assessment. *Applied Psychological Measurement*, *39*, 223–238. https://doi.org/10.1177/0146621614557272

Drasgow, F., Levine, M. V., & Williams, E. A. (1985). Appropriateness measurement with polychotomous item response models and standardized indices. *British Journal of Mathematical and Statistical Psychology*, *38*, 67–86. https://psycnet.apa.org/doi/10.1111/j.2044-8317.1985.tb00817.x

Santos, K. C. P., de la Torre, J., & von Davier, M. (2020). Adjusting person fit index for skewness in cognitive diagnosis modeling. *Journal of Classification*, *37*, 399-420. https://doi.org/10.1007/s00357-019-09325-5

Examples

```
library(GDINA)
dat <- sim10GDINA$simdat[1:20, ]
Q <- sim10GDINA$simQ
fit <- GDINA(dat = dat, Q = Q, model = "GDINA")
res.personFit <- personFit(fit)
res.personFit
```

RDINA   *Restricted DINA model*

Description

Estimation of the *restricted deterministic input, noisy "and" gate* model (R-DINA; Nájera et al., 2023). In addition to the non-compensatory (i.e., conjunctive) condensation rule of the DINA model, the compensatory (i.e., disjunctive) rule of the DINO model can be also applied (i.e., R-DINO model). The R-DINA/R-DINO model should be only considered for applications involving very small sample sizes (N < 100; Nájera et al., 2023), and model fit evaluation and comparison with competing models (e.g., DINA/DINO, G-DINA) is highly recommended.

Usage

```
RDINA(
  dat,
  Q,
  gate = "AND",
  att.prior = NULL,
  est = "Brent",
  tau.alpha = "MAP",
```

```
    emp.bayes = FALSE,
    boot = FALSE,
    n.boots = 500,
    n.cores = 1,
    maxitr = 1000,
    conv.crit = 1e-04,
    init.phi = 0.2,
    bound.p = 1e-06,
    verbose = TRUE,
    seed = NULL
)
```

Arguments

dat	A *N* individuals x *J* items (`matrix` or `data.frame`). Missing values need to be coded as `NA`. Caution is advised if missing data are present.
Q	A *J* items x *K* attributes Q-matrix (`matrix` or `data.frame`).
gate	Either a conjunctive (`"AND"`) or disjunctive (`"OR"`) condensation rule to estimate the RDINA or RDINO model, respectively. Default is `"AND"`.
att.prior	A 2^K attributes vector containing the prior distribution for each latent class. The sum of all elements does not have to be equal to 1, since the vector will be normalized. Default is `NULL`, which is a uniform prior distribution.
est	Use the Brent's method (`"Brent"`) or the expectation-maximization algorithm (`"EM"`) to estimate the model? Default is `"Brent"`, since it is faster and both algorithms are virtually equivalent for the RDINA/RDINO model.
tau.alpha	Attribute profile estimator (either `"MAP"`, `"EAP"`, or `"MLE"`) used to calculate the estimated classification accuracy as done with the `CA` function of the `GDINA` package (Ma & de la Torre, 2020).
emp.bayes	Use empirical Bayes estimation for structural parameters. Default is `FALSE`.
boot	Use bootstrapping to increase robustness in posterior probabilities estimation. Default is `FALSE`.
n.boots	Number of bootstrapping samples. Default is 500.
n.cores	Number of CPU processors to speed up computation when bootstrapping is used. Default is 1.
maxitr	Maximum number of iterations. Default is 1000.
conv.crit	Convergence criterion regarding the maximum absolute change in either the phi parameter estimate or the marginal posterior probabilities of attribute mastery. Default is 0.0001.
init.phi	Initial value for the phi parameter. Default is 0.2.
bound.p	Lowest value for probability estimates (highest would be 1 - bound.p). Default is 1e-06.
verbose	Print information after each iteration. Default is `TRUE`.
seed	Random number generation seed (e.g., to solve ties in case they occur with MLE or MAP estimation). Default is `NULL`, which means that no specific seed is used.

Value

RDINA returns an object of class RDINA.

MLE Estimated attribute profiles with the MLE estimator (matrix).

MAP Estimated attribute profiles with the MAP estimator (matrix).

EAP Estimated attribute profiles with the EAP estimator (matrix).

phi Phi parameter estimate (numeric).

post.probs A (list) containing the estimates of the posterior probability of each examinee in each latent class (pp), marginal posterior probabilities of attribute mastery (mp), and posterior probability of each latent class (lp).

likelihood A (list) containing the likelihood of each examinee in each latent class (lik_il) and the model log-likelihood (logLik).

test.fit Relative model fit indices (list).

class.accu A (list) containing the classification accuracy estimates at the test-level (tau), latent class-level (tau_l), and attribute-level (tau_k).

specifications Function call specifications (list).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas

References

Ma, W., & de la Torre, J. (2020). GDINA: An R package for cognitive diagnosis modeling. *Journal of Statistical Software*, *93*(14). https://doi.org/10.18637/jss.v093.i14

Nájera, P., Abad, F. J., Chiu, C.-Y., & Sorrel, M. A. (2023). The Restricted DINA model: A Comprehensive Cognitive Diagnostic Model for Classroom-Level Assessments. *Journal of Educational and Behavioral Statistics*.

Examples

```
library(GDINA)
Q <- sim30GDINA$simQ # Q-matrix
K <- ncol(Q)
J <- nrow(Q)
set.seed(123)
GS <- data.frame(guessing = rep(0.2, J), slip = rep(0.2, J))
sim <- simGDINA(20, Q, GS, model = "DINA")
simdat <- sim$dat # Simulated data
simatt <- sim$attribute # Generating attributes
fit.RDINA <- RDINA(simdat, Q) # Apply the GNPC method
ClassRate(fit.RDINA$EAP, simatt) # Check classification accuracy
```

RDINA2GDINA   *Translate RDINA object into GDINA object*

Description

This function translates an object of class RDINA to an object of class GDINA, so that the estimated R-DINA object is compatible with most of the functions in the GDINA package (Ma & de la Torre, 2020), including model fit, item fit, and Q-matrix validation.

Usage

```
RDINA2GDINA(fit)
```

Arguments

fit   An object of class RDINA.

Value

RDINA2GDINA returns an object of class GDINA. See the GDINA package for more information.

Author(s)

Pablo Nájera, Universidad Pontificia Comillas

References

Ma, W., & de la Torre, J. (2020). GDINA: An R package for cognitive diagnosis modeling. *Journal of Statistical Software*, *93*(14). https://doi.org/10.18637/jss.v093.i14

Examples

```
library(GDINA)
dat <- sim30DINA$simdat
Q <- sim30DINA$simQ
fit1 <- RDINA(dat, Q)
fit2 <- RDINA2GDINA(fit1)
modelfit(fit2) # Model fit evaluation
itemfit(fit2) # Item fit evaluation
```

---

simFCGDINA   *Forced-choice data simulation based on the G-DINA model*

Description

Simulate forced-choice (FC) responses based on the G-DINA model (de la Torre, 2011) and the FC-DCM (Huang, 2023). This function accommodates FC responses to the `simGDINA` function from the `GDINA` package (Ma & de la Torre, 2020).

Usage

```
simFCGDINA(
  N,
  Q.items,
  n.blocks = NULL,
  polarity = NULL,
  att = NULL,
  model = "GDINA",
  GDINA.args = list(GS = NULL, GS.items = c(1/3, 1/3), AC = 0, AT = 0),
  FCDCM.args = list(d0 = c(0.2, 0.2), sd = c(0.15, 0.15), a = c(0, 0), b = 0),
  seed = NULL
)
```

Arguments

N	A `numeric` value indicating the sample size.
Q.items	A binary `matrix` of dimensions J statements x K attributes indicating what statements measure what attributes.
n.blocks	A `numeric` value indicating the number of forced-choice blocks.
polarity	A `matrix` of dimensions `n.blocks` x 2 indicating whether each statament in each block is direct (1) or inverse (-1). Default is a matrix full of 1 (i.e., all statements are direct).
att	A `matrix` of dimensions N individuals x K attributes indicating the attribute profile of each individual. Default is `NULL`, meaning that attribute profiles will be simulated based on the specifications listed on `GDINA.args` or `FCDCM.args`.
model	Use the G-DINA model (`"GDINA"`) or the FC-DCM (`"FCDCM"`) as the generating model. Default is `"GDINA"`.
GDINA.args	A `list` of arguments used if `model = "GDINA"`.
	GS A J statements x 2 matrix indicating the guessing and slip parameter of each statement. Default is `NULL`.
	GS.items Only used if `GDINA.args$GS = NULL`. A `vector` of length 2 indicating the minimum and maximum value for the random generation of guessing and slip parameters for each statement. Default is `c(1/3, 1/3)`.
	AC A `numeric` value indicating the attribute correlations in line with the multivariate normal threshold model (Chiu et al., 2009). Default is 0.

AT A numeric value indicating the attribute thresholds in line with the multi-variate normal threshold model (Chiu et al., 2009). Default is 0.

FCDCM.args A `list` of arguments used if `model = "FCDCM"`.

d0 A `vector` of length 2 indicating the minimum and maximum value for the baseline probability for each FC block (see Huang, 2023). Default is `c(0.2, 0.2)`.

sd A `vector` of length 2 indicating the minimum and maximum value for the statement utility parameters (see Huang, 2023). Default is `c(0.15, 0.15)`.

a A `numeric` value indicating the minimum and maximum discrimination parameter for the higher-order model. Default is `c(0, 0)`.

b A `numeric` value indicating the location parameter for the higher-order model. Default is 0.

seed Random number generation seed. Default is `NULL`.

Value

simFCGDINA returns an object of class `simFCGDINA`.

dat Generated FC responses (`matrix`).

att Generated attribute profiles (`matrix`).

Q Generated Q-matrix of FC blocks (`matrix`).

LCprob Generated block response probabilities for each latent class (`matrix`).

item.pairs Statements used in each FC block (`matrix`).

q_att Attribute measured by each statement as used by Huang (2023) (`matrix`).

q_sta Relative position of each statement as used by Huang (2023) (`matrix`).

simGDINA Object of class `simGDINA` (`list`).

polarity Polarity matrix indicating the direction of each statement in each block (`matrix`).

GS Generated guessing and slip parameter for each statement (`matrix`).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas

References

Huang, H.-Y. (2023). Diagnostic Classification Model for Forced-Choice Items and Noncognitive Tests. *Educational and Psychological Measurement*, *83*(1), 146-180. https://doi.org/10.1177/00131644211069906

Ma, W., & de la Torre, J. (2020). GDINA: An R package for cognitive diagnosis modeling. *Journal of Statistical Software*, *93*(14). https://doi.org/10.18637/jss.v093.i14

Examples

```
library(GDINA)
set.seed(123)
# Q-matrix for the unidimensional statements
Q.items <- do.call("rbind", replicate(5, diag(5), simplify = FALSE))
# Guessing and slip
GS <- cbind(runif(n = nrow(Q.items), min = 0.1, max = 0.3),
            runif(n = nrow(Q.items), min = 0.1, max = 0.3))
n.blocks <- 30 # Number of forced-choice blocks
# Block polarity (1 = direct statement; -1 = indirect statement)
polarity <- matrix(1, nrow = n.blocks, ncol = 2)
sim <- simFCGDINA(N = 1000, Q.items, n.blocks = n.blocks, polarity = polarity,
                  model = "GDINA", GDINA.args = list(GS = GS), seed = 123)
```

---

valQ   *Empirical Q-matrix validation*

Description

Empirical Q-matrix validation using the *Hull* method (Nájera, Sorrel, de la Torre, & Abad, 2020a).
The procedure can be used either with the PVAF (de la Torre & Chiu, 2016) or McFadden's pseudo
R-squared (McFadden, 1974). The PVAF is recommended (Nájera, Sorrel, de la Torre, & Abad,
2020a). Note that the pseudo R-squared might not be computationally feasible for highly dimen-
sional Q-matrices, say more than 10 attributes. Different iterative implementations are available,
such as the test-level implementation (see Terzi & de la Torre, 2018), attribute-test-level implemen-
tation (Nájera, Sorrel, de la Torre, & Abad, 2020a), and item-level implementation (Nájera, Sorrel,
de la Torre, & Abad, 2020b). If an iterative implementation is used, the GDINA R package (Ma & de
la Torre, 2020) is used for the calibration of the CDMs.

Usage

```
valQ(
  fit,
  index = "PVAF",
  iterative = "test.att",
  emptyatt = TRUE,
  maxitr = 100,
  CDMconv = 1e-04,
  verbose = TRUE
)
```

Arguments

fit   A G-DINA model fit object from the GDINA package (Ma & de la Torre, 2020).

index   What index to use. It includes "PVAF" or "R2". The default is "PVAF".

iterative	(Iterative) implementation procedure. It includes "none" (for non-iterative), "test" (for test-level iterations), "test.att" (for attribute-test-level), and "item" (for item-level iterations). The default is "test.att".
emptyatt	Is it possible for the suggested Q-matrix to have an empty attribute (i.e., an attribute not measured by any item)? Although rarely, it is possible for iterative procedures to provide a suggested Q-matrix in which one or more attributes are empty. This might indicate that the original Q-matrix had more attributes than necessary. If FALSE, then at least one item (i.e., the one that is most likely) will measure each attribute in the suggested Q-matrix. The default is TRUE.
maxitr	Maximum number of iterations if an iterative procedure has been selected. The default is 100.
CDMconv	Convergence criteria for the CDM estimations between iterations (only if an iterative procedure has been selected). The default is 0.0001.
verbose	Print information after each iteration if an iterative procedure is used. The default is TRUE.

Value

valQ returns an object of class valQ.

sug.Q Suggested Q-matrix (matrix).

Q Original Q-matrix (matrix).

sugQ.fit Several fit indices from the model obtained with the suggested Q-matrix (vector).

index PVAF or pseudo R-squared (depending on which one was used) for each item (matrix).

iter.Q Q-matrices used in each iteration (list). Provided only if an iterative procedure has been used.

iter.index PVAF or pseudo R-squared (depending on which one was used) for each item in each iteration (list). Provided only if an iterative procedure has been used.

n.iter Number of iterations used (double). Provided only if an iterative procedure has been used.

convergence Convergence information (double). It can be 1 (convergence), 2 (lack of convergence: maximum number of iterations achieved), 3 (lack of convergence: empty attribute obtained), and 4 (lack of convergence: loop Q-matrices). Provided only if an iterative procedure has been used.

time Initial and finish time (vector).

time.used Total computation time (difftime).

specifications Function call specifications (list).

Author(s)

Pablo Nájera, Universidad Pontificia Comillas
Miguel A. Sorrel, Universidad Autónoma de Madrid
Francisco J. Abad, Universidad Autónoma de Madrid

References

de la Torre, J., & Chiu, C.-Y. (2016). A general method of empirical Q-matrix validation. *Psychometrika*, *81*, 253-273. https://doi.org/10.1007/s11336-015-9467-8

Ma, W., & de la Torre, J. (2020). GDINA: An R package for cognitive diagnosis modeling. *Journal of Statistical Software*, *93*(14). https://doi.org/10.18637/jss.v093.i14

McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. In P. Zarembka (Ed.), *Frontiers in Economics* (pp. 105-142). Academic Press.

Nájera, P., Sorrel, M. A., de la Torre, J., & Abad, F. J. (2020a). Balancing fit and parsimony to improve Q-matrix validation. *British Journal of Mathematical and Statistical Psychology*. https://doi.org/10.1111/bmsp.12228

Nájera, P., Sorrel, M. A., de la Torre, J., & Abad, F. J. (2020b). Improving robustness in Q-matrix validation using an iterative and dynamic procedure. *Applied Psychological Measurement*, *46*, 431-446. https://doi.org/10.1177/0146621620909904

Terzi, R., & de la Torre, J. (2018). An iterative method for empirically-based Q-matrix validation. *International Journal of Assessment Tools in Education*, *5*, 248-262. https://doi.org/10.21449/ijate.407193

Examples

```
library(GDINA)
dat <- sim30GDINA$simdat
Q <- sim30GDINA$simQ # Generating Q-matrix
miss.Q <- missQ(Q = Q, qjk = .30, retainJ = 5, seed = 123)$miss.Q # Misspecified Q-matrix
fit <- GDINA(dat, miss.Q) # GDINA object
sug.Q <- valQ(fit = fit, verbose = TRUE) # Hull method for Q-matrix validation
mean(sug.Q$sug.Q == Q) # Check similarity with the generating Q-matrix
```

Index