Package ‘cdmTools’

March 21, 2022

Type Package

Title Useful Tools for Cognitive Diagnosis Modeling

Version 1.1.0

Date 2022-03-21

Description Provides useful tools for cognitive diagnosis modeling (CDM). The packages includes the discrete factor loading method for Q-matrix estimation (Wang, Song, & Ding, 2018, <doi:10.1007/978-3-319-77249-3_29>) and the Hull method for Q-matrix validation (Nájera, Sorrel, de la Torre, & Abad, 2021, <doi:10.1111/bmsp.12228>). It also provides dimensionality assessment procedures for determining the number of attributes underlying CDM data, including parallel analysis and automated CDM fit comparison as explored in Nájera, Abad, and Sorrel (2021, <doi:10.3389/fpsyg.2021.614470>). 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

LazyData FALSE

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), foreach, doSNOW

Suggests CDM

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

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

RoxygenNote 7.1.1

Encoding UTF-8

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

Maintainer Pablo Nájera <`pablo.najera@uam.es`>

R topics documented:

`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., manuscript submitted for publication). The classification accuracy indices are the ones delevoped 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`	A G-DINA model fit object from the `GDINA` package (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 Autónoma de Madrid

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. (manuscript submitted for publication). Improving reliability estimation in cognitive diagnosis modeling.

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*, emph52 , 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` Is the generated Q-matrix identifiable under the DINA/DINO models or others CDMs? (`vector`).

`specifications` Function call specifications (`list`).

Author(s)

Pablo Nájera, Universidad Autónoma de Madrid

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 = 30 is here used for illustration purposes)
sugQ2 <- estQ(r = dat, K = 5, boot = TRUE, boot.args = list(R = 30, 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
```

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 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` Is the generated Q-matrix identifiable under the DINA/DINO models or others CDMs? (`vector`).

`specifications` Function call specifications (`list`).

Author(s)

Pablo Nájera, Universidad Autónoma de Madrid

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/10.1080/01621459.2014.9

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.122

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)
```

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

Description

Checks whether a Q-matrix is complete (Köhn & Chiu, 2017, 2018) and identifiable according to the criteria from Chen, Liu, Xu, & Ying (2015) and Xu & Shang (2018).

Usage

```
is.Qid(Q, model = "others", verbose = TRUE)
```

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). The default is `"others"`.
`verbose`	Should a message about the identifiability of the Q-matrix be printed? The default is `TRUE`.

Value

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

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

`comp.Q` Is the Q-matrix complete? (`logical`).

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

`message` A message about the identifiability of the Q-matrix and references (`string`).

`specifications` Function call specifications (`list`).

Author(s)

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

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/10.1080/01621459.2014.9

Köhn, H.-F., & Chiu, C.-Y. (2017). A procedure for assessing the completeness of the Q-matrices of cognitively diagnostic tests. *Psychometrika*, *82*, 112-132. https://doi.org/10.1007/s11336-016-9536-7

Köhn, H.-F., & Chiu, C.-Y. (2018). How to build a complete Q-matrix for a cognitively diagnostic test. *Journal of Classification*, *35*, 273-299. https://doi.org/10.1007/s00357-018-9255-0

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)
Q <- genQ(J = 30, K = 4, Kj = Kj, Qid = "others", seed = 123)$gen.Q
idQ <- is.Qid(Q)
```

`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 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` Is the generated Q-matrix identifiable under the DINA/DINO models or others CDMs? (`vector`).

`specifications` Function call specifications (`list`).

Author(s)

Pablo Nájera, Universidad Autónoma de Madrid

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 Autónoma de Madrid
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.122

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 congruent 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 Autónoma de Madrid

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 simulated 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 Autónoma de Madrid
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
```

---

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 dimensional 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 implementation (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 Autónoma de Madrid
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.122

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