Package ‘NPCDTools’

January 20, 2025

Title The Nonparametric Classification Methods for Cognitive Diagnosis

Version 1.0

Description Statistical tools for analyzing cognitive diagnosis (CD) data collected from small settings using the nonparametric classification (NPCD) framework. The core methods of the NPCD framework includes the nonparametric classification (NPC) method developed by Chiu and Douglas (2013) <DOI:10.1007/s00357-013-9132-9> and the general NPC (GNPC) method developed by Chiu, Sun, and Bian (2018) <DOI:10.1007/s11336-017-9595-4> and Chiu and Köhn (2019) <DOI:10.1007/s11336-019-09660-x>. An extension of the NPCD framework included in the package is the nonparametric method for multiple-choice items (MC-NPC) developed by Wang, Chiu, and Koehn (2023) <DOI:10.3102/10769986221133088>. Functions associated with various extensions concerning the evaluation, validation, and feasibility of the CD analysis are also provided. These topics include the completeness of Q-matrix, Q-matrix refinement method, as well as Q-matrix estimation.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Imports GDINA, NPCD, psych, SimDesign, stats, utils, gtools

NeedsCompilation no

Author Chia-Yi Chiu [aut, cph],
Weixuan Xiao [aut, cre],
Xiran Wen [aut],
Yu Wang [aut]

Maintainer Weixuan Xiao <wx2299@tc.columbia.edu>

Repository CRAN

Date/Publication 2024-09-23 13:01:53 UTC

Contents

AAR *Attribute-wise agreement rate*

Description

The function is used to compute the attribute-wise agreement rate between two sets of attribute profiles. They need to have the same dimensions.

Usage

```
AAR(x, y)
```

Arguments

x	One set of attribute profiles
y	The other set of attribute profiles

Value

The function returns the attribute-wise agreement rate between two sets of attribute profiles.

bestQperm *Column permutation of the estimated Q-matrix*

Description

Function `bestQperm` is used to rearrange the columns of the estimated Q so that the order of the columns best matches that of the true Q-matrix.

Usage

```
bestQperm(estQ, trueQ)
```

Arguments

`estQ`	The estimated Q-matrix.
`trueQ`	The true Q-matrix.

Value

The function returns a Q-matrix in which the order of the columns best matches that of the true Q-matrix.

`correction.rate` *Correction rate of a Q-matrix refinement method*

Description

This function computes the proportion of corrected q-entries that were originally misspecified in the provisional Q-matrix. This function is used only when the true Q-matrix is known.

Usage

```
correction.rate(ref.Q = ref.Q, mis.Q = mis.Q, true.Q = true.Q)
```

Arguments

`ref.Q`	the $J \times K$ binary Q-matrix obtained from applying the refinement procedure.
`mis.Q`	A $J \times K$ binary provisional Q-matrix.
`true.Q`	The $J \times K$ binary true Q-matrix.

Value

The function returns a value between 0 and 1 indicating the proportion of corrected q-entries in `ref.Q` that were originally missepcified in `mis.Q`.

`GNPC` *Estimation of examinees' attribute profiles using the GNPC method*

Description

Function `GNPC` is used to estimate examinees' attribute profiles using the general nonparametric classification (GNPC) method (Chiu, Sun, & Bian, 2018; Chiu & Koehn, 2019). It can be used with data conforming to any CDMs.

Usage

```
GNPC(
  Y,
  Q,
  initial.dis = c("hamming", "whamming"),
  initial.gate = c("AND", "OR", "Mix")
)
```

Arguments

`Y` A $N \times J$ binary data matrix consisting of the responses from N examinees to J items.

`Q` A $J \times K$ binary Q-matrix where the entry q_{jk} describing whether the kth attribute is required by the jth item.

`initial.dis` The type of distance used in the `AlphaNP` to carry out the initial attribute profiles for the GNPC method. Allowable options are `"hamming"` and `"whamming"` representing the Hamming and the weighted Hamming distances, respectively.

`initial.gate` The type of relation between examinees' attribute profiles and the items. Allowable relations are `"AND"`, `"OR"`, and `"Mix"`, representing the conjunctive, disjunctive, and mixed relations, respectively

Value

The function returns a series of outputs, including

att.est The estimates of examinees' attribute profiles

class The estimates of examinees' class memberships

weighted.ideal The weighted ideal responses

weight The weights used to compute the weighted ideal responses

GNPC algorithm

A weighted ideal response $\eta^{(w)}$, defined as the convex combination of $\eta^(c)$ and $\eta^(d)$, is proposed. Suppose item j requires $K_j^* \leq K$ attributes that, without loss of generality, have been permuted to the first K_j^* positions of the item attribute vector $\boldsymbol{q_j}$. For each item j and $\mathcal{C}_l$, the weighted ideal response $\eta_{ij}^{(w)}$ is defined as the convex combination $\eta_{ij}^{(w)} = w_{lj}\eta_{lj}^{(c)} + (1 - w_{lj})\eta_{lj}^{(d)}$ where $0 \leq w_{lj} \leq 1$. The distance between the observed responses to item j and the weighted ideal responses $w_{lj}^{(w)}$ of examinees in $\mathcal{C}_l$ is defined as the sum of squared deviations: $d_{lj} = \sum_{i \in \mathcal{C}_l}(y_{ij} - \eta_{lj}^{(w)})^2 = \sum_{i \in \mathcal{C}_l}(y_{ij} - w_{lj}\eta_{lj}^{(c)} - (1 - w_{lj})\eta_{lj}^{(d)})$ Thus, $\widehat{w_{lj}}$ can be minimizing d_{lj}: $\widehat{w_{lj}} = \frac{\sum_{i \in \mathcal{C}_l}(y_{ij} - \eta_{lj}^{(d)})}{\|\mathcal{C}_l\|(\eta_{lj}^{(c)} - \eta_{lj}^{(d)})}$

As a viable alternative to $\boldsymbol{\eta^{(c)}}$ for obtaining initial estimates of the proficiency classes, Chiu et al. (2018) suggested to use an ideal response with fixed weights defined as $\eta_{lj}^{(fw)} = \frac{\sum_{k=1}^{K}\alpha_k q_{jk}}{K}\eta_{lj}^{(c)} + (1 - \frac{\sum_{k=1}^{K}\alpha_k q_{jk}}{K})\eta_{lj}^{(d)}$

NPC *Estimation of examinees' attribute profiles using the NPC method*

Description

The function is used to estimate examinees' attribute profiles using the nonparametric classification (NPC) method (Chiu, & Douglas, 2013). It uses a distance-based algorithm on the observed item responses for classifying examiness. This function estimates attribute profiles using nonparametric approaches for both the "AND gate" (conjunctive) and the "OR gate" (disjunctive) cognitive diagnostic models. These algorithms select the attribute profile with the smallest loss function value (plain, weighted, or penalized Hamming distance, see below for details) as the estimate. If more than one attribute profiles have the smallest loss function value, one of them is randomly chosen.

Usage

```
NPC(
  Y,
  Q,
  gate = c("AND", "OR"),
  method = c("Hamming", "Weighted", "Penalized"),
  wg = 1,
  ws = 1
)
```

Arguments

`Y` A matrix of binary responses. Rows represent persons and columns represent items. 1=correct, 0=incorrect.

`Q` The Q-matrix of the test. Rows represent items and columns represent attributes. 1=attribute required by the item, 0=attribute not required by the item.

`gate` A character string specifying the type of gate. It can be one of the following:

"AND" The examinee needs to possess all required attributes of an item in order to answer it correctly.

"OR" The examinee needs to possess only one of the required attributes of an item in order to answer it correctly.

`method` The method of nonparametric estimation.

"Hamming" The plain Hamming distance method

"Weighted" The Hamming distance weighted by inversed item variance

"Penalized" The Hamming distance weighted by inversed item variance and specified penalizing weights for guess and slip.

`wg` Additional argument for the "penalized" method. It is the weight assigned to guessing in the DINA or DINO models. A large value of weight results in a stronger impact on Hamming distance (larger loss function values) caused by guessing.

`ws` Additional input for the "penalized" method. It is the weight assigned to slipping in the DINA or DINO models. A large value of weight results in la stronger impact on Hamming distance (larger loss function values) caused by slipping.

Value

The function returns a series of outputs, including:

alpha.est Estimated attribute profiles. Rows represent persons and columns represent attributes. 1=examinee masters the attribute, 0=examinee does not master the attribute.

est.ideal Estimated ideal response to all items by all examinees. Rows represent persons and columns represent items. 1=correct, 0=incorrect.

est.class The class number (row index in pattern) for each person's attribute profile. It can also be used for locating the loss function value in loss.matrix for the estimated attribute profile for each person.

n.tie Number of ties in the Hamming distance among the candidate attribute profiles for each person. When we encounter ties, one of the tied attribute profiles is randomly chosen.

pattern All possible attribute profiles in the search space.

loss.matrix The matrix of the values for the loss function (the plain, weighted, or penalized Hamming distance). Rows represent candidate attribute profiles in the same order with the pattern matrix; columns represent different examinees.

NPC algorithm with three distacne methods

Proficiency class membership is determined by comparing an examinee's observed item response vector $\boldsymbol{Y}$ with each of the ideal item response vectors of the realizable $2^K = M$ proficiency classes. The ideal item responses are a function of the Q-matrix and the attribute vectors characteristic of the different proficiency classes. Hence, an examinee's proficiency class is identified by the attribute vector $\boldsymbol{\alpha}_m$ underlying that ideal item response vector which is closest—or most similar—to an examinee's observed item response vector. The ideal response to item j is the score that would be obtained by an examinee if no perturbation occurred.

Let $\boldsymbol{\eta}_i$ denote the J-dimensional ideal item response vector of examinee i, and the $\hat{\boldsymbol{\alpha}}$ of an examinee's attribute vector is defined as the attribute vector underlying the ideal item response vector that among all ideal item response vectors minimizes the distance to an examinee's observed item response vector: $\hat{\boldsymbol{\alpha}} = \arg\min_{m \in \{1,2,\ldots,M\}} d(\boldsymbol{y_i}, \boldsymbol{\eta}_m)$

A distance measure often used for clustering binary data is the Hamming distance that simply counts the number of disagreements between two vectors: $d_H(\boldsymbol{y}, \boldsymbol{\eta}) = \sum_{j=1}^{J} |y_j - \eta_j|$

If the different levels of variability in the item responses are to be incorporated, then the Hamming distances can be weighted, for example, by the inverse of the item sample variance, which allows for larger impact on the distance functions of items with smaller variance: $d_{wH}(\boldsymbol{y}, \boldsymbol{\eta}) = \sum_{j=1}^{J} \frac{1}{\overline{p_j}(1-\overline{p_j})} |y_j - \eta_j|$

Weighting weighting differently for departures from the ideal response model that would result from slips versus guesses is also considered: $d_{gs}(\boldsymbol{y}, \boldsymbol{\eta}) = \sum_{j=1}^{J} w_g I[y_j = 1]|y_j - \eta_j| + \sum_{j=1}^{J} w_s I[y_j = 0]|y_j - \eta_j|$

References

Chiu, C. (2011). Flexible approaches to cognitive diagnosis: nonparametric methods and small sample techniques. Invited session of cognitive diagnosis and item response theory at 2011 Joint Statistical Meeting.

Chiu, C. Y., & Douglas, J. A. (2013). A nonparametric approach to cognitive diagnosis by proximity to ideal response patterns. Journal of Classification 30(2), 225-250.

Examples

```
# Generate item and examinee profiles

natt <- 3
nitem <- 4
nperson <- 5
Q <- rbind(c(1, 0, 0), c(0, 1, 0), c(0, 0, 1), c(1, 1, 1))
alpha <- rbind(c(0, 0, 0), c(1, 0, 0), c(0, 1, 0), c(0, 0, 1), c(1, 1, 1))

# Generate DINA model-based response data
slip <- c(0.1, 0.15, 0.2, 0.25)
guess <- c(0.1, 0.15, 0.2, 0.25)
my.par <- list(slip=slip, guess=guess)
data <- matrix(NA, nperson, nitem)
eta <- matrix(NA, nperson, nitem)
for (i in 1:nperson) {
  for (j in 1:nitem) {
    eta[i, j] <- prod(alpha[i,] ^ Q[j, ])
    P <- (1 - slip[j]) ^ eta[i, j] * guess[j] ^ (1 - eta[i, j])
    u <- runif(1)
    data[i, j] <- as.numeric(u < P)
    }
}

# Using the function to estimate examinee attribute profile
alpha.est.NP.H <- NPC(data, Q, gate="AND", method="Hamming")
alpha.est.NP.W <- NPC(data, Q, gate="AND", method="Weighted")
alpha.est.NP.P <- NPC(data, Q, gate="AND", method="Penalized", wg=2, ws=1)

nperson <- 1   # Choose an examinee to investigate
print(alpha.est.NP.H) # Print the estimated examinee attribute profiles
```

PAR *Pattern-wise agreement rate*

Description

The function is used to compute the pattern-wise agreement rate between two sets of attribute profiles. They need to have the same dimensions.

Usage

```
PAR(x, y)
```

Arguments

x	One set of attribute profiles
y	The other set of attribute profiles

Value

The function returns the pattern-wise agreement rate between two sets of attribute profiles.

Q.generate *Generation of Dichotomous Q-Matrix*

Description

The function generates a complete Q-matrix based on a pre-specified probability of getting a one.

Usage

```
Q.generate(K, J, p, single.att = TRUE)
```

Arguments

K	The number of attributes
J	The number of items
p	The probability of getting a one in the Q-matrix
single.att	Whether all the single attribute patterns are included. If T, the completeness of the Q-matrix is guaranteed.

Value

The function returns a complete dichotomous Q-matrix

Examples

```
q = Q.generate(3,20,0.5,single.att = TRUE)
q1 = Q.generate(5,30,0.6,single.att = FALSE)
```

QR *Refine the Q-matrix by Minimizing the RSS*

Description

We estimate memberships using the non-parametric classification method (weighted hamming), and comparisons of the residual sum of squares computed from the observed and the ideal item responses.

Usage

```
QR(Y, Q, gate = c("AND", "OR"), max.ite = 50)
```

Arguments

`Y`	A matrix of binary responses (1=correct, 0=incorrect). Rows represent persons and columns represent items.
`Q`	The Q-matrix of the test. Rows represent items and columns represent attributes.
`gate`	A string, "AND" or "OR". "AND": the examinee needs to possess all related attributes to answer an item correctly. "OR": the examinee needs to possess only one of the related attributes to answer an item correctly.
`max.ite`	The number of iterations to run until all RSS of all items are stationary.

Value

A list containing:

`initial.class`	Initial classification
`terminal.class`	Terminal classification
`modified.Q`	The modified Q-matrix
`modified.entries`	The modified q-entries

The Q-Matrix Refinment (QR) Method

This function implements the Q-matrix refinement method developed by Chiu (2013), which is also based on the aforementioned nonparametric classification methods (Chiu & Douglas, 2013). This Q-matrix refinement method corrects potential misspecified entries of the Q-matrix through comparisons of the residual sum of squares computed from the observed and the ideal item responses.

The algorithm operates by minimizing the RSS. Recall that Y_{ij} is the observed response and η_{ij} is the ideal response. Then the RSS of item j for examinee i is defined as

$$RSS_{ij} = (Y_{ij} - \eta_{ij})^2$$

. The RSS of item j across all examinees is therefor

$$RSS_j = \sum_{i=1}^{N}(Y_{ij} - \eta_{ij})^2 = \sum_{m=1}^{2^k} \sum_{i \in C_m} (Y_{ij} - \eta_{jm})^2$$

where C_m is the latent proficiency-class m, and N is the number of examinees. Chiu(2013) proved that the expectation of RSS_j is minimized for the correct q-vector among the $2^K - 1$ candidates. Please see the paper for the justification.

References

Chiu, C. Y. (2013). Statistical Refinement of the Q-matrix in Cognitive Diagnosis. *Applied Psychological Measurement, 37(8)*, 598-618.

`retention.rate` *Retention rate of a Q-matrix refinement method*

Description

This function computes the proportion of correctly specified q-entries in a provisional Q-matrix that remain correctly specified after a Q-matrix refinement procedure is applied. This function is used only when the true Q-matrix is known.

Usage

```
retention.rate(ref.Q = ref.Q, mis.Q = mis.Q, true.Q = true.Q)
```

Arguments

`ref.Q`	the $J \times K$ binary Q-matrix obtained from applying a refinement procedure.
`mis.Q`	A $J \times K$ binary provisional Q-matrix.
`true.Q`	The $J \times K$ binary true Q-matrix.

Value

The function returns a value between 0 and 1 indicating the proportion of correctly specified q-entries in `mis.Q` that remain correctly specified in `ref.Q` after a Q-matrix refinement procedure is applied to `mis.Q`.

RR *Entry-wise and vector-wise recovery rates*

Description

Function RR is used to compute the recovery rates for an estimate Q-matrix. In general, it can be used to compute the agreement rate between two matrices with identical dimensionalities.

Usage

```
RR(Q1, Q2)
```

Arguments

Q1	The first Q-matrix.
Q2	The second Q-matrix that has the same dimensionality as Q1.

Value

The function returns

entry.wise The entry-wise recovery rate

item.wise The item-wise recovery rate

TSQE *Two-step Q-matrix Estimation Method*

Description

The function is used to estimate the Q-matrix based on the data (responses) using the two-step Q-matrix estimation method.

Usage

```
TSQE(
  Y,
  K,
  input.cor = c("tetrachoric", "Pearson"),
  ref.method = c("QR", "GDI"),
  GDI.model = c("DINA", "ACDM", "RRUM", "GDINA"),
  cutoff = 0.8
)
```

Arguments

`Y`	A $N \times J$ binary data matrix consisting of responses from N examinees to J items
`K`	The number of attributes in the Q-matrix
`input.cor`	The type of correlation used to compute the input for the exploratory factor analysis. It could be the tetrachoric or Pearson correlation.
`ref.method`	The refinement method use to polish the provisional Q-matrix obtained from the EFA. Currently available methods include the Q-matrix refinement (QR) method and the G-DINA discrimination index (GDI).
`GDI.model`	The CDM used in the GDI algorithm to fit the data. Currently available models include the DINA model, the ACDM, the RRUM, and the G-DINA model
`cutoff`	The cutoff used to dichotomize the entries in the provisional Q-matrix

Value

The function returns the estimated Q-matrix

Estimation Method

The TSQE method merges the Provisional Attribute Extraction (PAE) algorithm with a Q-matrix refinement-and-validation method including the Q-Matrix Refinement (QR) Method and the G-DINA Model Discrimination Index (GDI). Specifically, the PAE algorithm relies on classic exploratory factor analysis (EFA) combined with a unique stopping rule for identifying a provisional Q-matrix, and the resulting provisional Q-Matrix will be "polished" with a refinement method to derive the final estimation of Q-matrix.

The Provisional Attribute Extraction (PAE) Algorithm

The initial step of the algorithm is to aggregating the collected Q-Matrix into an inter-item tetrachoric correlation matrix. The reason for using tetrachoric correlation is that the examinee responses are binary, so it is more appropriate than the Pearson product moment correlation coefficient. See Chiu et al. (2022) for details. The next step is to use factor analysis on the item-correlation matrix, and treat the extracted factors as proxies for the latent attributes. The third step concerns identifying which specific attributes are required for which item:

(1) Initialize the item index as $j = 1$.

(2) Let l_{jk} denote the loading of item j on factor k, where $k = 1, 2, ..., K$.

(3) Arrange the loadings in descending order. Define a mapping function $f(k) = t$, where t is the order index. Hence, $l_{j(1)}$ will indicate the maximum loading, while $l_{j(K)}$ will indicate the minimum loading.

(4) Define

$$p_j(t) = \frac{\sum_{h=1}^{t} l_{j(h)}^2}{\sum_{k=1}^{K} l_{jk}^2}$$

as the proportion of the communality of item j accounted for by the first t factors.

(5) Define

$$K_j = \min\{t \mid p_j(t) \geq \lambda\}$$

, where λ is the cut-off value for the desired proportion of item variance-accounted-for. Then, the ordered entries of the provisional q-vector of item j are obtained as

$$q^*_{j(t)} = \begin{cases} 1 & \text{if } t \leq K_j \\ 0 & \text{if } t > K_j \end{cases}$$

.

(6) Identify $q^*_j = (q^*_{j1}, q^*_{j2}, ..., q^*_{jK})$ by rearranging the ordered entries of the q-vector using the inverse function $k = f^{-1}(t)$.

(7) Set $j = j + 1$ and repeat (2) to (6) until $j = J$. Then denote the provisional Q-matrix as $\mathbf{Q}^*$.

The Q-Matrix Refinment (QR) Method

This function implements the Q-matrix refinement method developed by Chiu (2013), which is also based on the aforementioned nonparametric classification methods (Chiu & Douglas, 2013). This Q-matrix refinement method corrects potential misspecified entries of the Q-matrix through comparisons of the residual sum of squares computed from the observed and the ideal item responses.

The algorithm operates by minimizing the RSS. Recall that Y_{ij} is the observed response and η_{ij} is the ideal response. Then the RSS of item j for examinee i is defined as

$$RSS_{ij} = (Y_{ij} - \eta_{ij})^2$$

. The RSS of item j across all examinees is therefor

$$RSS_j = \sum_{i=1}^{N} (Y_{ij} - \eta_{ij})^2 = \sum_{m=1}^{2^k} \sum_{i \in C_m} (Y_{ij} - \eta_{jm})^2$$

where C_m is the latent proficiency-class m, and N is the number of examinees. Chiu(2013) proved that the expectation of RSS_j is minimized for the correct q-vector among the $2^K - 1$ candidates. Please see the paper for the justification.

The G-DINA Model Discrimination Index (GDI)

The GDI is an extension of de la Torre's (2008) δ-method, which has a limitation that it cannot be used with CDMs that devide examinees into more than two groups. In response to the limitation, de la Torre and Chiu (2016) porposed to select that item attribute vector which maximizes the weighted variance of the probabilities of a correct response for the different groups defined as

$$\zeta^2_j = \sum_{l=1}^{2^{K_j}} P(\alpha_{lj}) \left[P(Y_{ij} = 1 \mid \alpha_{lj}) - \bar{P}_j \right]^2$$

where $P(\alpha_{lj})$ is the posterior probability for the proficiency class α_{lj}, and $\bar{P}_j = \sum_{l=1}^{2^{K_j}} P(\alpha_{lj}) P(Y_{ij} = 1 \mid \alpha_{lj})$, where $l = 1, 2, ..., 2^{K_j}$. De la Torre and Chiu (2016) called ζ^2 the GDI, which can be applied to any CDM that can be reparameterized in terms of the G-DINA model.

References

Chiu, C. Y. (2013). Statistical Refinement of the Q-matrix in Cognitive Diagnosis. *Applied Psychological Measurement, 37(8)*, 598-618.

Chiu, C. Y., & Douglas, J. A. (2013). A nonparametric approach to cognitive diagnosis by proximity to ideal response patterns. *Journal of Classification 30(2)*, 225-250.

de la Torre, J., & Chiu, C.-Y. (2016) A general method of empirical Q-matrix validation. *Psychometrika, 81*, 253-73.

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

Index