Package ‘cdcatR’

April 2, 2020

Type Package

Title Congitive Diagnostic Computerized Adaptive Testing in R

Version 1.0.0

Date 2020-4-2

Description This package holds functions for conducting CD-CAT applications.

License GPL-3

LazyData TRUE

Depends R (>= 3.4.0), GDINA (>= 2.2.0), ggplot2, cowplot, doParallel (>= 1.0.15), CDM (>= 7.4.19), snow, NPCD

URL https://github.com/miguel-sorrel/cdcatR

BugReports miguel.sorrel@uam.es

RoxygenNote 7.1.0

Encoding UTF-8

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

Maintainer Miguel A. Sorrel <miguel.sorrel@uam.es>

R topics documented:

`att.plot` *Create plots for attribute mastery estimates*

Description

Create plots for attribute mastery estimates (X: item position, Y: mastery probability).

Usage

```
att.plot(cdcat.obj, i, k = NULL, ...)
```

Arguments

`cdcat.obj`	An object of class `cdcat`
`i`	Examinee to be plotted
`k`	Attribute/s to be plotted. Default is NULL, which plotts all attributes

Value

`att.plot` creates a plot.

`cdcat` *Cognitively based computerized adaptive test application*

Description

`cdcat` conducts a CD-CAT application for a given dataset. Different item selection rules can be used: the general discrimination index (GDI; de la Torre & Chiu, 2016; Kaplan et al., 2015), the Jensen-Shannon divergence index (JSD; Kang et al., 2017; Minchen & de la Torre, 2016; Yigit et al., 2018), the posterior-weighted Kullback-Leibler index (PWKL; Cheng, 2009), the modified PWKL index (MPWKL; Kaplan et al., 2015), the nonparametric item selection method (NPS; Chang et al., 2018), or random selection. Fixed length or fixed precision CD-CAT can be applied.

Usage

```
cdcat(
  fit = NULL,
  dat = NULL,
  Q = NULL,
  itemSelect = "GDI",
  MAXJ = 20,
  FIXED.LENGTH = TRUE,
  att.prior = NULL,
  post.initial = NULL,
  max.cut = 0.8,
  NPS.args = list(gate = "AND", pseudo.prob = T, w.type = 1, seed = NULL),
  n.cores = 2,
  i.print = 250,
  ...
)
```

Arguments

`fit`	Calibrated item bank with the GDINA or CDM package
`dat`	Dataset to be analyzed. If `is.null(dat)` it takes data from the fit object (i.e., post-hoc simulation)
`Q`	Q-matrix to be used in the analysis. If `is.null(Q)` it takes Q from the fit object
`itemSelect`	Item selection rule: GDI, JSD, MPWKL, PWKL, NPS, random
`MAXJ`	Maximum number of items to be applied. Default is 20
`FIXED.LENGTH`	Fixed CAT-length (TRUE) or fixed-precision (FALSE). Default is TRUE
`att.prior`	Prior distribution for MAP/EAP estimates
`post.initial`	Prior distribution for `itemSelect`
`max.cut`	Cutoff for fixed-precision (posterior pattern > max.cut). Default is .80
`NPS.args`	A list of options when `itemSelect = "NPS"`
`n.cores`	Number of cores to be used during parallelization. Default is 2
`i.print`	Print examinee information. Default is 250

Value

`cdcat` returns an object of class `cdcat`.

References

Chang, Y.-P., Chiu, C.-Y., & Tsai, R.-C. (2019). Nonparametric CAT for CD in educational settings with small samples. *Applied Psychological Measurement, 43*, 543-561.

Cheng, Y. (2009). When cognitive diagnosis meets computerized adaptive testing: CD-CAT. *Psychometrika, 74*, 619-632.

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

Kang, H.-A., Zhang, S., & Chang, H.-H. (2017). Dual-objective item selection criteria in cognitive diagnostic computerized adaptive testing. *Journal of Educational Measurement, 54*, 165-183.

Kaplan, M., de la Torre, J., & Barrada, J. R. (2015). New item selection methods for cognitive diagnosis computerized adaptive testing. *Applied Psychological Measurement, 39*, 167-188.

Minchen, N., & de la Torre, J. (2016, July). *The continuous G-DINA model and the Jensen-Shannon divergence*. Paper presented at the International Meeting of the Psychometric Society, Asheville, NC, United States.

Yigit, H. D., Sorrel, M. A., de la Torre, J. (2018). Computerized adaptive testing for cognitively based multiple-choice data. *Applied Psychological Measurement, 43*, 388-401.

Examples

```
######################################
# Example 1.                         #
# CD-CAT simulation for a GDINA obj  #
######################################

#-----------Data generation----------#
Q <- sim180GDINA$simQ
K <- ncol(Q)
dat <- sim180GDINA$simdat
```

```
att <- sim180GDINA$simalpha

#----------Model estimation----------#
fit <- GDINA(dat = dat, Q = Q, verbose = 0) # GDINA package
# fit <- gdina(data = dat, q.matrix = Q, progress = 0) # CDM package

#---------------CD-CAT---------------#
res.FIXJ <- cdcat(fit = fit, dat = dat, FIXED.LENGTH = TRUE, n.cores = 4)
res.VARJ <- cdcat(fit = fit, dat = dat, FIXED.LENGTH = FALSE, n.cores = 4)

#---------------Results--------------#
res.FIXJ$est[[1]] # estimates for the first examinee (fixed-length)
res.VARJ$est[[1]] # estimates for the first examinee (fixed-precision)
att.plot(res.FIXJ, i = 1) # plot for estimates for the first examinee (fixed-length)
att.plot(res.VARJ, i = 1) # plot for estimates for the first examinee (fixed-length)
# FIXJ summary
res.FIXJ.sum.real <- cdcat.summary(cdcat.obj = res.FIXJ, alpha = att) # vs. real accuracy
res.FIXJ.sum.real$recovery$plotPCV
res.FIXJ.sum.real$recovery$plotPCA
# VARJ summary
res.VARJ.sum.post <- cdcat.summary(cdcat.obj = res.VARJ, alpha = att)
res.VARJ.sum.post$CATlength$stats
res.VARJ.sum.post$CATlength$plot
res.VARJ.sum.post$recovery
# Post-hoc CAT simulation (only if dat is fit$options$dat)
att.J <- personparm(fit, "MAP")[, -(K+1)] # GDINA package
# att.J <- t(sapply(strsplit(as.character(fit$pattern$map.est), ""), as.numeric)) # CDM package
class.J <- ClassRate(att, att.J) # upper-limit for accuracy
res.FIXJ.sum.post <- cdcat.summary(cdcat.obj = res.FIXJ, alpha = att.J)
res.FIXJ.sum.post$recovery$plotPCV + geom_hline(yintercept = class.J$PCV[K], color = "red")
res.FIXJ.sum.post$recovery$plotPCA + geom_hline(yintercept = class.J$PCA, color = "red")

######################################
# Example 2.                         #
# CD-CAT simulation for multiple     #
# GDINA objs and comparison of       #
# performance on a validation sample #
######################################

#----------------Data----------------#
Q <- sim155complex$simQ
K <- ncol(Q)
parm <- sim155complex$simcatprob.parm
dat.c <- sim155complex$simdat.c
att.c <- sim155complex$simalpha.c
dat.v <- sim155complex$simdat.v
att.v <- sim155complex$simalpha.v

#-----(multiple) Model estimation----#
fitTRUE <- GDINA(dat = dat.c, Q = Q, catprob.parm = parm, control = list(maxitr = 0), verbose = 0)
fitGDINA <- GDINA(dat = dat.c, Q = Q, verbose = 0)
fitDINA <- GDINA(dat = dat.c, Q = Q, model = "DINA", verbose = 0)
fitDINO <- GDINA(dat = dat.c, Q = Q, model = "DINO", verbose = 0)
fitACDM <- GDINA(dat = dat.c, Q = Q, model = "ACDM", verbose = 0)
LR2step <- LR_2step(fitGDINA)
models <- LR2step$models.adj.pvalues
fitLR2 <- GDINA(dat = dat.c, Q = Q, model = models, verbose = 0)
```

```
#---------------CD-CAT---------------#
fit.l <- list(fitTRUE, fitGDINA, fitDINA, fitDINO, fitACDM, fitLR2)
res.FIXJ.l <- res.VARJ.l <- list()
for(mm in 1:length(fit.l)) {
 fit <- fit.l[[mm]]
 res.FIXJ.l[[mm]] <- cdcat(fit = fit, dat = dat.v, FIXED.LENGTH = TRUE, n.cores = 4)
 res.VARJ.l[[mm]] <- cdcat(fit = fit, dat = dat.v, FIXED.LENGTH = FALSE, n.cores = 4)
}

#---------------Results--------------#
fitbest <- GDINA(dat = dat.v, Q = Q, catprob.parm = parm, control = list(maxitr = 1), verbose = 0)
fitbest.acc <- personparm(fitbest, "MAP")[, -(K+1)]
class.J <- ClassRate(att.v, fitbest.acc) # upper-limit for accuracy
# FIXJ comparison
res.FIXJ.sum.post.comp <- cdcat.comp(cdcat.obj.l = res.FIXJ.l, alpha = att.v)
res.FIXJ.sum.post.comp$PCVcomp + geom_hline(yintercept = class.J$PCV[K], color = "red")
res.FIXJ.sum.post.comp$PCAmcomp + geom_hline(yintercept = class.J$PCA, color = "red")
# VARJ comparison
res.VARJ.sum.post.comp <- cdcat.comp(cdcat.obj.l = res.VARJ.l, alpha = att.v)
res.VARJ.sum.post.comp$stats
res.VARJ.sum.post.comp$plots
res.VARJ.sum.post.comp$recovery

######################################
# Example 3.                         #
# Nonparametric CD-CAT for           #
# small-scale assessment             #
######################################

#-----------Data generation----------#
Q <- sim180DINA$simQ
K <- ncol(Q)
N <- 50
dat <- sim180DINA$simdat[1:N,]
att <- sim180DINA$simalpha[1:N,]

#--------Nonparametric CD-CAT--------#
res.NPS <- cdcat(dat = dat, Q = Q, itemSelect = "NPS", MAXJ = 30,
              NPS.args = list(gate = "AND", pseudo.prob = TRUE, w.type = 1, seed = 12345),
                n.cores = 4)

#---------------Results--------------#
res.NPS$est[[1]] # estimates for the first examinee
att.plot(res.NPS, i = 1) # plot for estimates for the first examinee
# FIXJ summary
res.NPS.sum.real <- cdcat.summary(cdcat.obj = res.NPS, alpha = att) # vs. real accuracy
res.NPS.sum.real$recovery$plotPCV
res.NPS.sum.real$recovery$plotPCA
# Post-hoc CAT simulation
fit <- AlphaNP(Y = dat, Q = Q, gate = "AND")
att.J <- fit$alpha.est
class.J <- ClassRate(att, att.J) # upper-limit for accuracy
res.NPS.sum.post <- cdcat.summary(cdcat.obj = res.NPS, alpha = att.J)
res.NPS.sum.post$recovery$plotPCV + geom_hline(yintercept = class.J$PCV[K], color = "firebrick3")
res.NPS.sum.post$recovery$plotPCA + geom_hline(yintercept = class.J$PCA, color = "firebrick3")
```

`cdcat.comp` *Comparison of multiple cdcat objects*

Description

This function compares different `cdcat` objects in terms of classification accuracy (`FIXED.LENGTH == TRUE`) and/or CAT lenght `FIXED.LENGTH == FALSE`).

Usage

```
cdcat.comp(cdcat.obj.l, alpha, label = NULL, ...)
```

Arguments

`cdcat.obj.l`	List of `cdcat` objects to be compared
`alpha`	N x K matrix with the attribute patterns to be compared to the `cdcat` results
`label`	labels for the `cdcat` objects. If NULL (by default), the models are used as label

Value

`cdcat.comp` returns an object of class `cdcat.comp`.

`cdcat.summary` *Summary information for a cdcat.object*

Description

This function provides classification accuracy (`FIXED.LENGTH == TRUE`) and/or CAT length (`FIXED.LENGTH == FALSE`) results for cdcat object.

Usage

```
cdcat.summary(cdcat.obj, alpha, ...)
```

Arguments

`cdcat.obj`	`cdcat` results
`alpha`	N x K matrix with the attribute patterns to be compared to the `cdcat` results

Value

`cdcat.summary` returns an object of class `cdcat.summary`.

gen.itembank *Item bank generation*

Description

This function can be used to generate an item bank. The user can provide a Q-matrix or create one defining the number of times each attribute should be measured and the q-vector complexity (e.g., number of attributes that a q-vector can measure). Item parameters are sampled from a uniform distribution with mean = *IQ* and variance = *VAR*.

Usage

```
gen.itembank(
  Q = NULL,
  minJ.K = NULL,
  complexity = NULL,
  IQ,
  VAR,
  model = NULL,
  ...
)
```

Arguments

Q	Q-matrix
minJ.K	Vector indicating the minimum number of items measuring each attribute
complexity	Vector indicating the maximum number of attributes being measured by an item in each row of Q. At this moment maximum is 4
IQ	Item discrimination (mean for the uniform distribution). IQ = $P(\mathbf{1})$ - $P(\mathbf{0})$ (Sorrel, Abad, Olea, de la Torre, and Barrada, 2017)
VAR	Item discrimination (variance for the uniform distribution)
model	Vector indicating the model-item correspondence (0 = one-attribute item, 1 = DINA, 2 = DINO, 3 = A-CDM

Value

gen.itembank returns an object of class gen.Item.Bank.

References

Kaplan, M., de la Torre, J., & Barrada, J. R. (2015). New item selection methods for cognitive diagnosis computerized adaptive testing. *Applied Psychological Measurement, 39*, 167-188.

Sorrel, M. A., Abad, F. J., Olea, J., de la Torre, J., & Barrada, J. R. (2017). Inferential item-fit evaluation in cognitive diagnosis modeling. *Applied Psychological Measurement, 41*, 614-631.

Examples

```
####################################
#        Example 1.                #
#     Q and model are provided     #
####################################

Q <- sim30GDINA$simQ
model <- rep(1, each = nrow(Q))
IQ <- .70   # P(1), IQ = Low item quality in Kaplan, de la Torre & Barrada (2015)
VAR <- 0.10 # High variance in Kaplan et al. (2015)
bank <- gen.itembank(Q = Q, IQ = IQ, VAR = VAR, model = model)

####################################
#        Example 2.                #
#   Q and model are not provided   #
####################################

minJ.K  <- c(50, 50, 50)
complexity <- c(3, 3, 3)
IQ <- .70   # P(1), IQ = Low item quality in Kaplan, de la Torre & Barrada (2015)
VAR <- 0.10 # High variance in Kaplan et al. (2015)
bank <- gen.itembank(minJ.K = minJ.K, complexity = complexity, IQ = IQ, VAR = VAR)
```

`LR_2step` *Item-level model comparison using 2LR test*

Description

This function evaluates whether the saturated G-DINA model can be replaced by reduced CDMs without significant loss in model data fit for each item using two-step likelihood ratio test (2LR). Sorrel, de la Torre, Abad, and Olea (2017) and Ma & de la Torre (2018) can be consulted for details.

Usage

```
LR_2step(fit, p.adjust.method = "holm", alpha.level = 0.05, ...)
```

Arguments

`fit` Calibrated item bank with GDINA or CDM package

`p.adjust.method`
Correction method for p-values. Possible values include "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". See p.adjust function from stats for additional details. Default is holm

`alpha.level` Alpha level for decision. Default is 0.05

Value

`LR2step` returns an object of class `LR2step`

LR2 LR2 statistics

pvalues p-values associated with the wald statistics

adj.pvalues adjusted p-values associated with the wald statistics

df degrees of freedom

models.adj.pvalues Models selected using the rule *largestp* (Ma, Iaconangelo, & de la Torre, 2016). All statistics whose p-values are less than `alpha.level` are rejected. All statistics with p-value larger than `alpha.level` define the set of candidate reduced models. The G-DINA model is retained if all statistics are rejected. Whenever the set includes more than one model, the model with the largest p-value was selected as the best model for that item.

References

Ma, W. & de la Torre, J. (2018). Category-level model selection for the sequential G-DINA model. *Journal of Educational and Behavorial Statistic, 44*, 45-77.

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

Sorrel, M. A., de la Torre, J., Abad, F. J., & Olea, J. (2017). Two-step likelihood ratio test for item-level model comparison in cognitive diagnosis models. *Methodology, 13*, 39-47.

Examples

```
N <- 1000
Q <- sim30GDINA$simQ
J <- nrow(Q)
gs <- data.frame(guess=rep(0.1,J),slip=rep(0.1,J))
sim <- simGDINA(N, Q, gs.parm = gs, model = "DINA")
resGDINA <- GDINA(dat = sim$dat, Q = sim$Q, model = "GDINA",verbose = F)
resCDM <- gdina(data = sim$dat, q.matrix = sim$Q, rule = "GDINA", progress = F)
LR2.GDINA <- LR_2step(fit = resGDINA)
LR2.CDM <- LR_2step(fit = resCDM)
LR2.GDINA$models.adj.pvalues
LR2.CDM$models.adj.pvalues
```

`sim155complex` *Simulated data (155 items, a combination of DINA, DINO, and A-CDM models)*

Description

Simulated data, Q-matrix and item parameters for a 155-item bank with 5 attributes).

Usage

```
sim155complex
```

Format

A list with components:

`simQ` Artificial Q-matrix. Q-matrix structure is complex (items measure up to four attributes and only 5 of them are one-attribute items

`simcatprob.parm` Artificial item parameters (probability of success for each latent group). Items 1-60 are DINA items, items 61-120 are DINO items, and items 121-180 are A-CDM items `simdat.c` Calibration sample dataset. Simulated responses of 500 examinees

`simalpha.c` Calibration sample alpha patterns. Simulated attribute patterns of 500 examinees)

`simdat.v` Validation sample dataset. Simulated responses of 500 examinees

`simalpha.v` Validation sample alpha patterns. Simulated attribute patterns of 500 examinees)

`sim155simple` *Simulated data (155 items, a combination of DINA, DINO, and A-CDM models)*

Description

Simulated data, Q-matrix and item parameters for a 155-item bank with 5 attributes.

Usage

```
sim155simple
```

Format

A list with components:

`simQ` Artificial Q-matrix. Q-matrix structure is simple (items measure up to three attributes and 35 of them are one-attribute items)

`simcatprob.parm` Artificial item parameters (probability of success for each latent group). Items 1-60 are DINA items, items 61-120 are DINO items, and items 121-180 are A-CDM items

`simdat.c` Calibration sample dataset. Simulated responses of 500 examinees

`simalpha.c` Calibration sample alpha patterns. Simulated attribute patterns of 500 examinees)

`simdat.v` Validation sample dataset. Simulated responses of 500 examinees

`simalpha.v` Validation sample alpha patterns. Simulated attribute patterns of 500 examinees)

`sim180DINA` *Simulated data (180 items, DINA model)*

Description

Simulated data, Q-matrix and item parameters for a 180-item bank with 5 attributes.

Usage

```
sim180DINA
```

Format

A list with components:

`simdat` Simulated responses of 500 examinees

`simQ` Artificial Q-matrix

`simcatprob.parm` Artificial item parameters (probability of success for each latent group). All items are DINA items

`simalpha` Simulated attribute patterns of 500 examinees `sim180GDINA` *Simulated data (180 items, G-DINA model)*

Description

Simulated data, Q-matrix and item parameters for a 180-item bank with 5 attributes.

Usage

```
sim180GDINA
```

Format

A list with components:

- `simdat` Simulated responses of 500 examinees
- `simQ` Artificial Q-matrix
- `simcatprob.parm` Artificial item parameters (probability of success for each latent group). All items are G-DINA items
- `simalpha` Simulated attribute patterns of 500 examinees)

Index