

Package ‘simCAT’

February 25, 2024

Title Implements Computerized Adaptive Testing Simulations

Version 1.0.0

Author Alexandre Jaloto <alexandrejaloto@gmail.com> [aut, cre] (<<https://orcid.org/0000-0002-5291-1768>>)

Ricardo Primi [ths] (<<https://orcid.org/0000-0003-4227-6745>>)

Maintainer Alexandre Jaloto <alexandrejaloto@gmail.com>

Description Computerized Adaptive Testing simulations with dichotomous and polytomous items. Selects items with Maximum Fisher Information method or randomly, with or without constraints (content balancing and item exposure control). Evaluates simulation in terms of precision, item exposure and test length.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Imports dplyr,
mirt,
mirtCAT,
shiny,
shinycssloaders

URL <https://github.com/alexandrejaloto/simCAT>

R topics documented:

calc.info	2
calc.prob	2
cat.evaluation	3
content.balancing	5
eap	6
exposure.rate	6
gen.resp	7
rmse	7
select.item	8
sim.shiny	10
simCAT	10
stop.cat	13

Index**15**

calc.info	<i>Compute item information</i>
-----------	---------------------------------

Description

Calculate information of each item in the bank for a theta

Usage

```
calc.info(bank, theta, model = "3PL")
```

Arguments

bank	matrix with item parameters (a, b, c)
theta	current theta
model	may be 3PL or graded

Value

A vector with the information of each item

Author(s)

Alexandre Jaloto

calc.prob	<i>Compute probability</i>
-----------	----------------------------

Description

Calculate probability of observing certain answer to a dichotomous item, given a theta

Usage

```
calc.prob(theta, bank, u = 1)
```

Arguments

theta	theta
bank	matrix with item parameters (a, b, c)
u	1 for correct, 0 for wrong

Value

A vector with the probability of seeing determined response in each item

Author(s)

Alexandre Jaloto

cat.evaluation	<i>CAT Evaluation</i>
----------------	-----------------------

Description

Evaluate a CAT simulation

Usage

```
cat.evaluation(results, true.scores, item.name, rmax)
```

Arguments

results	list with results of a CAT simulation from simCAT
true.scores	true scores
item.name	vector with the name of all items in the bank
rmax	item maximum exposure rate

Value

a list with two elements.

evaluate is a data.frame. Each line corresponds to a replication, and the columns are the following variables:

- rmse root mean square error between true and estimated score
- se standard error of measurement
- correlation correlation between true and estimated score
- bias bias between true and estimated score
- overlap overlap rate
- min_exp minimum exposure rate
- max_exp maximum exposure rate
- n_exp0 number of items not administered
- n_exp_rmax number of items with exposure rate higher than rmax
- length_mean average mean of test length
- length_sd standard deviation of test length
- length_median average median of test length
- min_length minimum test length
- max_length maximum test length

conditional is a data.frame with the same variables (except for length_sd and length_median) conditioned to the true scores. The colnames are the thetas in each decile, that is, `quantile(true.scores, probs = seq(.1, 1, length.out = 10))`. Each line corresponds to the mean of the investigated variables for each decile. If there are replications, values are the replication means for each decile.

Author(s)

Alexandre Jaloto

Examples

```
## Not run:
set.seed(1)
n.items <- 50
pars <- data.frame(
  a = rlnorm(n.items),
  b = rnorm(n.items),
  c = rbeta(n.items, 5, 17),
  d = 1)

# thetas
theta <- rnorm(100)

# simulate responses
resps <- gen.resp(theta, pars[,1:3])

results <- simCAT(resps = resps,
  bank = pars[,1:3],
  start.theta = 0,
  sel.method = 'MFI',
  cat.type = 'variable',
  threshold = .3,
  stop = list(se = .3, max.items = 10))

eval <- cat.evaluation(
  results = results,
  true.scores = theta,
  item.name = paste0('I', 1:nrow(pars)),
  rmax = 1)

#### 3 replications
replications <- 3

# simulate responses
set.seed(1)
resps <- list()
for(i in 1:replications)
  resps[[i]] <- gen.resp(theta, pars[,1:3])

# CAT
results <- list()
for (rep in 1:replications)
{
  print(paste0('replication: ', rep, '/', replications))
  results[[rep]] <- simCAT(
    resps = resps[[rep]],
    bank = pars[,1:3],
    start.theta = 0,
    sel.method = 'MFI',
    cat.type = 'variable',
    threshold = .3,
    stop = list(se = .5, max.items = 10))
}

eval <- cat.evaluation(
  results = results,
```

```

true.scores = theta,
item.name = paste0('I', 1:nrow(pars)),
rmax = 1)

## End(Not run)

```

content.balancing	<i>Content balancing</i>
-------------------	--------------------------

Description

Constricts the selection with content balancing (CCAT or MCCAT)

Usage

```

content.balancing(
  bank,
  administered = NULL,
  content.names,
  content.props,
  content.items,
  met.content = "MCCAT"
)

```

Arguments

bank	matrix with item parameters (a, b, c)
administered	vector with administered items, NULL if it is the first item (default)
content.names	vector with the contents of the test
content.props	desirable proportion of each content in test, in the same order of content.names
content.items	vector indicating the content of each item
met.content	content balancing method <ul style="list-style-type: none"> • MCCAT (default): the function picks all subgroups with proportions most distant from desirable. • CCAT: if there is any subgroup without administered item, the function will randomly pick one. If all subgroups has at least one applied item, the function randomly picks one from those with the proportions most distant from desirable. • MMM: based on the desired proportions of content, the algorithm builds a sum-one cumulative distribution. Then, a random number with uniform distribution between zero and one is drawn. This number corresponds to an area in the cumulative distribution. It is from the content located in this area that the content will be selected.

Value

A numeric vector with the items that will be excluded for selection. That is, it returns the unavailable items. If all items are available, it returns NULL.

Author(s)

Alexandre Jaloto

eap	<i>EAP estimation</i>
-----	-----------------------

Description

Estimates theta with Expected a Posteriori

Usage

eap(pattern, bank)

Arguments

pattern	response pattern (0 and 1) with the number of columns corresponding to the number of items
bank	data.frame with item parameters (a, b, c)

Details

40 quadrature points, ranging from -4 to 4. Priori with normal distribution (mean = 0, sd = 1).

Value

data.frame with estimated theta and SE.

Author(s)

Alexandre Jaloto

exposure.rate	<i>Compute exposure rates</i>
---------------	-------------------------------

Description

Calculate exposure rate of items in a bank

Usage

exposure.rate(previous, item.name)

Arguments

previous	list with previous responses. Each element corresponds to a person and has the names of the applied items.
item.name	vector with the name of all items in the bank

Value

data.frame with

- items name of the items
- Freq exposure rate

Author(s)

Alexandre Jaloto

gen.resp	<i>Generate response pattern</i>
----------	----------------------------------

Description

Generate response pattern based on probability of answering correct a dichotomous item, given a theta and an item bank

Usage

```
gen.resp(theta, bank)
```

Arguments

theta	theta
bank	matrix with item parameters (a, b, c)

Value

A vector with the probability of seeing determined response in each item

Author(s)

Alexandre Jaloto

rmse	<i>Root Mean square Error</i>
------	-------------------------------

Description

Calculate the root mean square error

Usage

```
rmse(true, estimated)
```

Arguments

true	true values
estimated	estimated values

Value

A numeric vector

Author(s)

Alexandre Jaloto

select.item

Select next item

Description

Select next item to be administered

Usage

```
select.item(
  bank,
  model = "3PL",
  theta,
  administered = NULL,
  sel.method = "MFI",
  cat.type = "variable",
  threshold = 0.3,
  SE,
  acceleration = 1,
  met.weight = "mcclarty",
  max.items = 45,
  content.names = NULL,
  content.props = NULL,
  content.items = NULL,
  met.content = "MCCAT"
)
```

Arguments

bank	matrix with item parameters (a, b, c)
model	may be 3PL or graded
theta	current theta
administered	vector with administered items, NULL if it is the first item
sel.method	item selection method: may be MFI, progressive or random
cat.type	CAT with variable or fixed length. Necessary only for progressive method.
threshold	threshold for cat.type. Necessary only for progressive method.
SE	current standard error. Necessary only for progressive method, with cat.type = "variable"
acceleration	acceleration parameter. Necessary only for progressive method.
met.weight	the procedure to calculate the progressive's weight in variable-length CAT. It can be "magis" or "mcclarty" (default). See details.

max.items	maximum number of items to be administered. Necessary only for progressive method, with cat.type = "variable"
content.names	vector with the contents of the test
content.props	desirable proportion of each content in test, in the same order of content.names
content.items	vector indicating the content of each item
met.content	content balancing method: MCCAT (default), CCAT or MMM. See content.balancing for more information.

Details

In the progressive, the administered item is the one that has the highest weight. The weight of the item i is calculated as following:

$$W_i = (1 - s)R_i + sI_i$$

where R is a random number between zero and the maximum information of an item in the bank for the current theta, I is the item information and s is the importance of the component. As the application progresses, the random component loses importance. There are some ways to calculate s . For fixed-length CAT, Barrada et al. (2008) uses

$$s = 0$$

if it is the first item of the test. For the other administering items,

$$s = \frac{\sum_{f=1}^q (f-1)^k}{\sum_{f=1}^Q (f-1)^k}$$

where q is the number of the item position in the test, Q is the test length and k is the acceleration parameter. simCAT package uses these two equations for fixed-length CAT. For variable-length, simCAT package can use "magis" (Magis & Barrada, 2017):

$$s = \max\left[\frac{I(\theta)}{I_{stop}}, \frac{q}{M-1}\right]^k$$

where $I(\theta)$ is the item information for the current theta, I_{stop} is the information corresponding to the stopping error value, and M is the maximum length of the test. simCAT package uses as default "mcclarty" (adapted from McClarty et al., 2006):

$$s = \left(\frac{SE_{stop}}{SE}\right)^k$$

where SE is the standard error for the current theta, SE_{stop} is the stopping error value.

Value

A list with two elements

- item the number of the selected item in item bank
- name name of the selected item (row name)

Author(s)

Alexandre Jaloto

References

- Barrada, J. R., Olea, J., Ponsoda, V., & Abad, F. J. (2008). *Incorporating randomness in the Fisher information for improving item-exposure control in CATs*. *British Journal of Mathematical and Statistical Psychology*, 61(2), 493–513. 10.1348/000711007X230937
- Leroux, A. J., & Dodd, B. G. (2016). *A comparison of exposure control procedures in CATs using the GPC model*. *The Journal of Experimental Education*, 84(4), 666–685. 10.1080/00220973.2015.1099511
- Magis, D., & Barrada, J. R. (2017). *Computerized adaptive testing with R: recent updates of the package catR*. *Journal of Statistical Software*, 76(Code Snippet 1). 10.18637/jss.v076.c01
- McClarty, K. L., Sperling, R. A., & Dodd, B. G. (2006). *A variant of the progressive-restricted item exposure control procedure in computerized adaptive testing*. *Annual Meeting of the American Educational Research Association*, San Francisco

sim.shiny

CAT simulation in Shiny

Description

CAT simulation in a Shiny application.

Usage

`sim.shiny()`

Details

Uses `simCAT` function in a more friendly way. For now, this application only supports simulation with dichotomous items and one replication.

Author(s)

Alexandre Jaloto

simCAT

CAT simulation

Description

A CAT simulation with dichotomous items.

Usage

```

simCAT(
  resps,
  bank,
  model = "3PL",
  start.theta = 0,
  sel.method = "MFI",
  cat.type = "variable",
  acceleration = 1,
  met.weight = "mcclarty",
  threshold = 0.3,
  rmax = 1,
  content.names = NULL,
  content.props = NULL,
  content.items = NULL,
  met.content = "MCCAT",
  stop = list(se = 0.3, hypo = 0.015, hyper = Inf),
  progress = TRUE
)

```

Arguments

<code>resps</code>	a matrix with responses (0 and 1). The number of columns corresponds to the number of items
<code>bank</code>	matrix with item parameters (a, b, c)
<code>model</code>	may be 3PL or graded
<code>start.theta</code>	first theta
<code>sel.method</code>	item selection method: may be MFI, progressive or random
<code>cat.type</code>	CAT with variable or fixed length Necessary only for progressive method.
<code>acceleration</code>	acceleration parameter. Necessary only for progressive method.
<code>met.weight</code>	the procedure to calculate the progressive's weight in variable-length CAT. It can be "magis" or "mcclarty" (default). See details.
<code>threshold</code>	threshold for <code>cat.type</code> . Necessary only for progressive method.
<code>rmax</code>	item maximum exposure rate
<code>content.names</code>	vector with the contents of the test
<code>content.props</code>	desirable proportion of each content in test, in the same order of <code>content.names</code>
<code>content.items</code>	vector indicating the content of each item
<code>met.content</code>	content balancing method: MCCAT (default), CCAT #' or MMM. See <code>content.balancing</code> for more information.
<code>stop</code>	list with stopping rule and thresholds <ul style="list-style-type: none"> • <code>se</code> minimum standard error • <code>delta.theta</code> minimum absolute difference between current and previous theta • <code>hypo</code> minimum standard error reduction • <code>hyper</code> minimum standard error reduction after achieving <code>se</code> • <code>info</code> maximum information of an available item • <code>max.items</code> maximum number of items

	<ul style="list-style-type: none"> • <code>min.items</code> maximum number of items • <code>fixed</code> fixed number of items
<code>progress</code>	shows progress bar

Details

For details about formula of selection methods, see [select.item](#).

Value

a list with five elements

- `score` estimated theta
- `convergence` TRUE if the application ended before reaching the maximum test length
- `theta.history` estimated theta after each item administration
- `se.history` standard error after each item administration
- `prev.resps` previous responses (administered items)

Author(s)

Alexandre Jaloto

References

- Barrada, J. R., Olea, J., Ponsoda, V., & Abad, F. J. (2008). *Incorporating randomness in the Fisher information for improving item-exposure control in CATs*. *British Journal of Mathematical and Statistical Psychology*, 61(2), 493–513. 10.1348/000711007X230937
- Leroux, A. J., & Dodd, B. G. (2016). *A comparison of exposure control procedures in CATs using the GPC model*. *The Journal of Experimental Education*, 84(4), 666–685. 10.1080/00220973.2015.1099511
- Magis, D., & Barrada, J. R. (2017). *Computerized adaptive testing with R: recent updates of the package catR*. *Journal of Statistical Software*, 76(Code Snippet 1). 10.18637/jss.v076.c01
- McClarty, K. L., Sperling, R. A., & Dodd, B. G. (2006). *A variant of the progressive-restricted item exposure control procedure in computerized adaptive testing*. Annual Meeting of the American Educational Research Association, San Francisco

Examples

```
## Not run:
set.seed(1)
n.items <- 50
pars <- data.frame(
  a = rlnorm(n.items),
  b = rnorm(n.items),
  c = rbeta(n.items, 5, 17),
  d = 1)

# thetas
theta <- rnorm(100)

# simulate responses
resps <- gen.resp(theta, pars[,1:3])

results <- simCAT(resps = resps,
```

```

bank = pars[,1:3],
start.theta = 0,
sel.method = 'MFI',
cat.type = 'variable',
threshold = .3,
stop = list(se = .3, max.items = 10))

eval <- cat.evaluation(
  results = results,
  true.scores = theta,
  item.name = paste0('I', 1:nrow(pars)),
  rmax = 1)

#### 3 replications
replications <- 3

# simulate responses
set.seed(1)
resps <- list()
for(i in 1:replications)
  resps[[i]] <- gen.resp(theta, pars[,1:3])

# CAT
results <- list()
for (rep in 1:replications)
{
  print(paste0('replication: ', rep, '/', replications))
  results[[rep]] <- simCAT(
    resps = resps[[rep]],
    bank = pars[,1:3],
    start.theta = 0,
    sel.method = 'MFI',
    cat.type = 'variable',
    threshold = .3,
    stop = list(se = .5, max.items = 10))
}

eval <- cat.evaluation(
  results = results,
  true.scores = theta,
  item.name = paste0('I', 1:nrow(pars)),
  rmax = 1)

## End(Not run)

```

stop.cat

Check if the CAT ended

Description

Check if any stopping rule has been achieved

Usage

```
stop.cat(
  rule = list(se = NULL, delta.theta = NULL, hypo = NULL, hyper = NULL, info = NULL,
    max.items = NULL, min.items = NULL, fixed = NULL),
  current = list(se = NULL, delta.theta = NULL, info = NULL, applied = NULL, delta.se =
    NULL)
)
```

Arguments

rule	list with stopping rules <ul style="list-style-type: none"> • se minimum standard error • delta.theta minimum absolute difference between current and previous theta • hypo minimum standard error reduction • hyper minimum standard error reduction after achieving se • info maximum information of an available item • max.items maximum number of items • min.items maximum number of items • fixed fixed number of items
current	list with current values <ul style="list-style-type: none"> • se current standard error • delta.theta absolute difference between current and previous theta • info maximum information of an available item for current theta • applied quantitative of applied items • delta.se standard error reduction

Value

A list with two elements:

- stop TRUE if any stopping rule has been achieved
- convergence logical. FALSE if the CAT stopped because it achieved the maximum number of items. TRUE for any other case.

Author(s)

Alexandre Jaloto

Index

calc.info, [2](#)
calc.prob, [2](#)
cat.evaluation, [3](#)
content.balancing, [5](#)

eap, [6](#)
exposure.rate, [6](#)

gen.resp, [7](#)

rmse, [7](#)

select.item, [8](#), [12](#)
sim.shiny, [10](#)
simCAT, [10](#)
stop.cat, [13](#)