Package 'cat.dt'

March 31, 2021

Type Package

Title Computerized Adaptive Testing and Decision Trees

Version 0.3.1

Imports Rglpk, Matrix, ggplot2

Description Implements the Merged Tree-CAT method (Javier Rodriguez-Cuadrado et al., 2020, <doi:10.1016/j.eswa.2019.113066>) to generate Computerized Adaptive Tests (CATs) based on a decision tree. The tree growth is controlled by merging branches with similar trait distributions and estimations. This package has the necessary tools for creating CATs and estimate the subject's ability level.

URL https://github.com/jlaria/cat.dt

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

Suggests knitr, rmarkdown

VignetteBuilder knitr

Author Javier Rodriguez-Cuadrado [aut, cre],
Juan C. Laria [aut],
David Delgado-Gomez [aut]

Maintainer Javier Rodriguez-Cuadrado <javierro@est-econ.uc3m.es>

Repository CRAN

Date/Publication 2021-03-31 12:20:06 UTC

R topics documented:

`ability_density` *Vector of density values of ability level*

Description

Computes the density function values of the evaluated ability levels

Usage

```
ability_density(dens, ...)
```

Arguments

`dens`	density function (e.g. dnorm, dunif, etc.)
`...`	parameters of the density function

Value

A vector of density values

Author(s)

Javier Rodriguez-Cuadrado

allocate_sons *Allocate sons in the CAT decision tree*

Description

Fills the information of the sons of the previous level nodes

Usage

```
allocate_sons(nodes_prev, nodes, level)
```

Arguments

nodes_prev	list of node lists of the nodes from the previous level
nodes	list of node lists of the nodes from the current level
level	level of the CAT decision tree

Value

A list of node lists updated with the information of the sons

Author(s)

Javier Rodríguez-Cuadrado

a_posteriori *Vector of a posteriori density values of ability level*

Description

Computes the a posteriori density function values of the evaluated ability levels given the item response

Usage

```
a_posteriori(apriori, prob)
```

Arguments

apriori	a vector of a priori density function values of the evaluated ability levels
prob	a vector of probability response for every evaluated ability level given the item response

Value

A vector of a posteriori density values

Author(s)

Javier Rodriguez-Cuadrado

`cat.dt` *cat.dt: Computerized Adaptive Testing and Decision Trees*

Description

The cat.dt package implements the Merged Tree-CAT method to generate Computerized Adaptive Tests (CATs) based on a decision tree. The tree growth is controlled by merging branches with similar trait distributions and estimations. This package has the necessary tools for creating CATs and estimate the subject's ability level. The Merged Tree-CAT method is an extension of the Tree-CAT method (see Delgado-Gómez et al., 2019 <doi:10.1016/j.eswa.2018.09.052>).

Main interface function

CAT_DT

Author(s)

Javier Rodríguez-Cuadrado, David Delgado-Gómez, Juan C. Laria

See Also

`CAT_DT`

`CAT_ability_est` *Ability level estimation for an individual using a CAT decision tree*

Description

Computes the test taker's estimated ability level based on the CAT decision tree previously built and the test taker's responses to every item at every tree level

Usage

```
CAT_ability_est(cat.dt, res)
```

Arguments

`cat.dt`	A `cat.dt` object returned by `CAT_DT`.
`res`	Vector containing the test taker's responses to every item

Value

A list containing the following elements:

`$estimation` Estimated ability level after each level of the tree.

`$linf` Lower limit of the final estimation at 95

`$lsup` Upper limit of the final estimation at 95

`$items` Administered item in each level.

`$graphics` Plot object of the evolution of the ability level estimation. It shows the ability level estimation after the individual has answered to every administered item.

Author(s)

Javier Rodr?guez-Cuadrado

Examples

```
data("itemBank")
# Build the cat.dt
nodes = CAT_DT(bank = itemBank, model = "GRM", crit = "MEPV",
               C = 0.3, stop = c(3,0.5), limit = 200, inters = 0.98,
               p = 0.9, dens = dnorm, 0, 1)

# Estimate the ability level of a subject with responses res
estimation = CAT_ability_est(nodes, res = itemRes[1, ])

#plot the estimations
plot(estimation$graphics)
```

CAT_ability_est_group *Ability level estimation for a group using a CAT decision tree*

Description

Computes the test takers' estimated ability level based on the CAT decision tree previously built and the test takers' responses to every item at every tree level

Usage

```
CAT_ability_est_group(cat.dt, res)
```

Arguments

`cat.dt`	A `cat.dt` object returned by `CAT_DT`.
`res`	Matrix containing the test takers' responses to every item. Rows represent each individual and columns represent the responses given to each item

Value

A list of lists containing the following elements for each individual:

`$estimation` Estimated ability level after each level of the tree.

`$linf` Lower limit of the final estimation at 95

`$lsup` Upper limit of the final estimation at 95

`$items` Administered item in each level.

`$graphics` Plot object of the evolution of the ability level estimation. It shows the ability level estimation after the individual has answered to every administered item.

Author(s)

Javier Rodríguez-Cuadrado

Examples

```
# data("itemBank")
# Build the cat.dt
# nodes = CAT_DT(bank = itemBank, model = "GRM", crit = "MEPV",
#                C = 0.3, stop = c(2, 0.5), limit = 100, inters = 0.8,
#                p = 0.8, dens = dnorm, 0, 1)

# Estimate the ability level of a subject with responses res
# CAT_ability_est_group(nodes, res = itemRes)
```

`CAT_DT` *CAT decision tree*

Description

Generates a `cat.dt` object containing the CAT decision tree. This object has all the necessary information to build the tree.

Usage

```
CAT_DT(
  bank,
  model = "GRM",
  crit = "MEPV",
  C = 0.3,
  stop = c(6, 0),
  limit = 200,
  inters = 0.98,
  p = 0.9,
  dens,
  ...
)
```

Arguments

bank	data.frame or matrix of the item bank. Rows represent items, and columns represent parameters. If the model is "GRM", the first column represents the alpha parameters and the next columns represent the beta parameters. If the model is "NRM", odd columns represent the alpha parameters and even columns represent beta parameters
model	polytomous IRT model. Options: "GRM" for Graded Response Model and "NRM" for Nominal Response Model
crit	item selection criterion. Options: "MEPV" for Minimum Expected Posterior Variance and "MFI" for Maximum Fisher Information
C	vector of maximum item exposures. If it is an integer, this value is replicated for every item
stop	vector of two components that represent the decision tree stopping criterion. The first component represents the maximum level of the decision tree, and the second represents the minimum standard error of the ability level (if it is 0, this second criterion is not applied)
limit	maximum number of level nodes
inters	minimum common area between density functions in the nodes of the evaluated pair in order to join them
p	a-priori probability that controls the tolerance to join similar nodes
dens	density function (e.g. dnorm, dunif, etc.)
...	parameters of the density function

Value

An object of class cat.dt

Author(s)

Javier Rodr?guez-Cuadrado

Examples

```
data("itemBank")
# Build the cat.dt
nodes = CAT_DT(bank = itemBank, model = "GRM", crit = "MEPV",
               C = 0.3, stop = c(3,0.05), limit = 100, inters = 0.9,
               p = 0.9, dens = dnorm, 0, 1)

# Estimate the ability level of a subject with responses res
CAT_ability_est(nodes, res = itemRes[1, ])
# or
nodes$predict(res = itemRes[1, ])
# or
predict(nodes, itemRes[1, ])
```

create_E_MEPV *MSE of every item for an specified node*

Description

Computes a vector of the mean squared error of every item allocated to the specified level node in the CAT decision tree. Every MSE is computed using the ability level density function in the specified node and the ability level estimations given the item responses

Usage

```
create_E_MEPV(bank, dens_vec, nres, prob_array, C)
```

Arguments

bank	matrix of the item bank. Rows represent items, and columns represent parameters. If the model is "GRM", the first column represents the alpha parameters and the next columns represent the beta parameters. If the model is "NRM", odd columns represent the alpha parameters and even columns represent beta parameters
dens_vec	vector of the density function values in the specified node of the evaluated ability levels
nres	vector of number of possible responses for every item
prob_array	3-D array of probability responses. Dim 1 represent items, dim 2 represent evaluated ability levels and dim 3 represent possible responses
C	vector of item capacities

Value

A vector of all item MSE for the specified node

Author(s)

Javier Rodr?guez-Cuadrado

create_E_MFI *Fisher Information of every item for an specified node*

Description

Computes a vector of the Fisher Information of every item allocated to the specified level node in the CAT decision tree. Every FI is computed using the estimated ability level in the specified node

Usage

```
create_E_MFI(bank, theta_est, nres, C)
```

Arguments

bank	matrix of the item bank. Rows represent items, and columns represent parameters. If the model is "GRM", the first column represents the alpha parameters and the next columns represent the beta parameters. If the model is "NRM", odd columns represent the alpha parameters and even columns represent beta parameters
theta_est	estimated ability level
nres	vector of number of possible responses for every item
C	vector of item capacities

Value

A vector of all item Fisher Information for the specified node

Author(s)

Javier Rodríguez-Cuadrado

---

create_last_level *CAT decision tree last level generator*

Description

Generates a list of node lists for the last level of the CAT decision tree

Usage

```
create_last_level(nodes_prev, nres, level, prob_array, SE)
```

Arguments

nodes_prev	list of node lists of the nodes from the previous level
nres	vector of number of possible responses for every item
level	last-level number (equals the length of the test plus one)
prob_array	3-D array of probability responses. Dim 1 represent items, dim 2 represent evaluated ability levels and dim 3 represent possible responses
SE	minimum standard error of the ability level

Value

A list of lists. Each of these lists represent a node of the last level of the decision tree

Author(s)

Javier Rodríguez-Cuadrado

`create_levels` *CAT decision tree level generator*

Description

Generates a list of node lists for a specific level of the CAT decision tree

Usage

```
create_levels(
  nodes_prev,
  bank,
  crit,
  C,
  nres,
  level,
  prob_array,
  limit,
  tol,
  inters,
  SE
)
```

Arguments

`nodes_prev`	list of node lists of the nodes from the previous level
`bank`	matrix of the item bank. Rows represent items, and columns represent parameters. If the model is `"GRM"`, the first column represents the `alpha` parameters and the next columns represent the `beta` parameters. If the model is `"NRM"`, odd columns represent the `alpha` parameters and even columns represent `beta` parameters
`crit`	item selection criterion. Options: "MEPV" for Minimum Expected Posterior Variance and "MFI" for Maximum Fisher Information
`C`	vector of item capacities
`nres`	vector of number of possible responses for every item
`level`	level number
`prob_array`	3-D array of probability responses. Dim 1 represent items, dim 2 represent evaluated ability levels and dim 3 represent possible responses
`limit`	maximum number of level nodes
`tol`	maximum distance between estimated ability levels in the nodes of the evaluated pair in order to consider whether to join them

inters	minimum common area between density functions in the nodes of the evaluated pair in order to join them
SE	minimum standard error of the ability level

Value

A list of lists and a scalar. Each of the lists represent a node of the specified level of the decision tree, and the scalar represents if the created level is the last (1) or not (0) due to the SE stopping criterion

Author(s)

Javier Rodr?guez-Cuadrado

---

create_level_1 *Level 1 CAT decision tree generator*

Description

Generates a list of nodes lists for the first level of the CAT decision tree

Usage

```
create_level_1(bank, crit, dens_vec, C, nres, prob_array)
```

Arguments

bank	matrix of the item bank. Rows represent items, and columns represent parameters. If the model is "GRM", the first column represents the alpha parameters and the next columns represent the beta parameters. If the model is "NRM", odd columns represent the alpha parameters and even columns represent beta parameters
crit	item selection criterion. Options: "MEPV" for Minimum Expected Posterior Variance and "MFI" for Maximum Fisher Information
dens_vec	vector of the a priori density function values of the evaluated ability levels
C	vector of item capacities
nres	vector of number of possible responses for every item
prob_array	3-D array of probability responses. Dim 1 represent items, dim 2 represent evaluated ability levels and dim 3 represent possible responses

Value

A list of lists. Each of these lists represent a node of the first level of the decision tree

Author(s)

Javier Rodr?guez-Cuadrado

`create_node` *Node creator*

Description

Generates a list that represents a specific node of the CAT decision tree

Usage

```
create_node(ID, dens_vec, item, item_prev, est, SE, ID_sons, D, as_val)
```

Arguments

`ID`	integer that represents the specified node identification in the form of `10000*level+position`.
`dens_vec`	vector of the density function values in the specified node of the evaluated ability levels
`item`	integer that represents the item of the specified node
`item_prev`	vector of items of the previous nodes
`est`	estimated ability level in the specified node
`SE`	standard error of the estimated ability level
`ID_sons`	data frame containing the information of the sons of the specified node. Rows represent sons and columns represent the ID of the son, the response given to the item of the specified node that leaded to the son and the probability of reaching the son given that response (not equal to one if the son had previously splitted)
`D`	confluency of the specified node
`as_val`	associated value of the specified node. It can be the MSE if the selection criterium is "MEPV" and the FI if the selection criterium is "MFI"

Value

A list that represents a node of the decision tree

Author(s)

Javier Rodr?guez-Cuadrado

create_prob_array *Multidimensional array of response probabilities*

Description

For every item (dim 1) in an item bank and every evaluated ability level (dim 2), computes the probability of picking every possible response (dim 3) given the ability level

Usage

```
create_prob_array(model, bank, nres)
```

Arguments

model	polytomous IRT model. Options: "GRM" for Graded Response Model and "NRM" for Nominal Response Model
bank	matrix of the item bank. Rows represent items, and columns represent parameters. If the model is "GRM", the first column represents the alpha parameters and the next columns represent the beta parameters. If the model is "NRM", odd columns represent the alpha parameters and even columns represent beta parameters
nres	vector of number of possible responses for every item

Value

A 3-dimensional array of probability responses

Author(s)

Javier Rodríguez-Cuadrado

estimate *Ability level estimation*

Description

Computes the estimated ability level given the ability level density function values and its standard error

Usage

```
estimate(dens_vec)
```

Arguments

dens_vec	vector of density function values of the evaluated ability levels

Value

A list containing the expected value of the ability level density function and the standard error of that expectated value

Author(s)

Javier Rodríguez-Cuadrado

`Fisher_GRM` *Fisher Information under GRM*

Description

Computes the item Fisher Information given an ability level based on the GRM model

Usage

```
Fisher_GRM(theta_est, item_par, nres)
```

Arguments

`theta_est`	ability level
`item_par`	vector containing the item parameters. First component is the `alpha` parameter and the next are the `beta` parameters
`nres`	number of possible item responses

Value

An integer that represents the Fisher Information value of the specified item given the ability level

Author(s)

Javier Rodriguez-Cuadrado

Fisher_NRM *Fisher Information under NRM*

Description

Computes the item Fisher Information given an ability level based on the NRM model

Usage

```
Fisher_NRM(theta_est, item_par, nres)
```

Arguments

theta_est	ability level
item_par	vector containing the item parameters. Odd components are the alpha parameters and even are the beta parameters
nres	number of possible item responses

Value

An integer that represents the Fisher Information value of the specified item given the ability level

Author(s)

Javier Rodriguez-Cuadrado

itemBank *Example item bank*

Description

Item bank data generated using genPolyMatrix from catR package.

Usage

```
data(itemBank)
```

Format

An object of class data.frame.

itemRes *Example item responses*

Description

Item responses data to test with `data(itemBank)`. There are 30 subjects and their responses to 100 items.

Usage

```
data(itemRes)
```

Format

An object of class `matrix`.

item_selector *Linear programming solver*

Description

Computes the exposure rate of every item allocated to every level node. If more than one item is allocated to the same level node, the node splits.

Usage

```
item_selector(E_mat, D, C, minmax)
```

Arguments

`E_mat`	matrix of the associated value of every item allocated to every level node. Rows represent items and columns represent level nodes. The "associated value" can be the MSE if the selection criterium is "MEPV", the FI if the selection criterium is "MFI" and ****
`D`	vector of confluencies of every level node
`C`	vector of item capacities
`minmax`	optimisation direction. Options: `TRUE` to maximise and `FALSE` to minimise

Value

A matrix of exposure rates. Rows represent items and columns represent level nodes. Every item with a positive exposure rate for a level node is allocated to that node

Author(s)

Javier Rodríguez-Cuadrado

join_node *Node joiner*

Description

Given all the nodes from one level, `join_node` evaluates all possible pairs one by one and decides whether or not to join them based on the similarity between the estimated ability levels and the density functions. If a pair of nodes is joined, the density function of the resulting node is the mean of the density functions of the joined nodes and the confluencies are summed.

Usage

```
join_node(nodes, level, limit, tol, inters)
```

Arguments

`nodes`	list of node lists. Every node list must contain the ID of the node, the vector of density function values of the evaluated abiity levels, the vector of previous items, the estimated ability level and the node confluency
`level`	level of the CAT decision tree
`limit`	maximum number of level nodes
`tol`	minimum distance between estimated ability levels to join two nodes
`inters`	minimum common area between density functions in the nodes of the evaluated pair in order to join them

Value

A list of node lists. This list is the input list updated with the joined nodes

Author(s)

Javier Rodr?guez-Cuadrado

plot_tree *CAT decision tree plot*

Description

Generates a plot object to visualize the CAT decision tree

Usage

```
plot_tree(object, levels = 3, tree = 1)
```

Arguments

object	A cat.dt object
levels	Number of levels to plot, starting from the first one.
tree	Index of tree to plot. The total number of trees is given by length(nodes$nodes[[1]]).

Value

A ggplot2 object

Author(s)

Javier Rodr?guez-Cuadrado

---

predict.cat.dt *Predict S3 method for* cat.dt

Description

Predict S3 method for cat.dt

Usage

```
## S3 method for class 'cat.dt'
predict(object, res, ...)
```

Arguments

object	A cat.dt object returned by CAT_DT.
res	Vector containing the test taker's responses to every item
...	Not used

Value

A list containing the following elements:

$estimation Estimated ability level after each level of the tree.

$linf Lower limit of the final estimation at 95

$lsup Upper limit of the final estimation at 95

$items Administered item in each level.

$graphics Plot object of the evolution of the ability level estimation. It shows the ability level estimation after the individual has answered to every administered item.

Author(s)

Javier Rodr?guez-Cuadrado

probab_GRM *Item response GRM probabilities*

Description

Computes the probabilities of picking every possible response of an specified item from the item bank for all evaluated ability levels using the Graded Response Model

Usage

```
probab_GRM(item_par, nres)
```

Arguments

item_par	vector containing the item parameters. First component is the alpha parameter and the next are the beta parameters
nres	number of possible item responses

Value

A matrix of response probabilities. Rows represent evaluated ability levels and columns represent responses

Author(s)

Javier Rodríguez-Cuadrado

probab_NRM *Item response NRM probabilities*

Description

Computes the probabilities of picking every possible response of an specified item from the item bank for all evaluated ability levels using the Nominal Response Model

Usage

```
probab_NRM(item_par, nres)
```

Arguments

item_par	vector containing the item parameters. Odd components are the alpha parameters and even are the beta parameters
nres	number of possible item responses

Value

A matrix of response probabilities. Rows represent evaluated ability levels and columns represent responses

Author(s)

Javier Rodríguez-Cuadrado

`summary.cat.dt` *CAT summary*

Description

Summary of the `cat.dt` object generated

Usage

```
## S3 method for class 'cat.dt'
summary(object, ...)
```

Arguments

`object`	A `cat.dt` object
`...`	not used

Value

A summary of the `cat.dt` object

Author(s)

Javier Rodríguez-Cuadrado

Index