Package 'Rata'

October 24, 2019

Type Package

Title Automated Test Assembly

Version 0.0.2

Date 2019-10-22

Author Xiao Luo [aut, cre]

Maintainer Xiao Luo <xluo1986@gmail.com>

Description Automated test assembly of linear and adaptive tests using the mixed-integer programming. The full documentation and tutorials are at <https://github.com/xluo11/Rata>.

License GPL (>= 3)

Depends R (>= 3.6.0)

URL https://github.com/xluo11/Rata

BugReports https://github.com/xluo11/Rata/issues

Imports ggplot2, glpkAPI, lpSolveAPI, Rirt, reshape2, stats

Suggests testthat

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-10-24 05:10:02 UTC

R topics documented:

ata *Automated Test Assembly (ATA)*

Description

ata creates a basic ATA model

ata_relative_objective adds a relative objective to the model

ata_absolute_objective adds an absolute objective to the model

ata_constraint adds a constraint to the model

ata_item_use limits the minimum and maximum usage for items

ata_item_enemy adds an enemy-item constraint to the model

ata_item_fix forces an item to be selected or not selected

ata_solve solves the MIP model

Usage

```
ata(pool, n_forms = 1, test_len = NULL, max_use = NULL, ...)

ata_relative_objective(x, coef, mode = c("max", "min"), tol = NULL,
  negative = FALSE, forms = NULL, collapse = FALSE,
  internal_index = FALSE)

ata_absolute_objective(x, coef, target, equal_tol = FALSE,
  tol_up = NULL, tol_down = NULL, forms = NULL, collapse = FALSE,
  internal_index = FALSE)

ata_constraint(x, coef, min = NA, max = NA, level = NULL,
  forms = NULL, collapse = FALSE, internal_index = FALSE)

ata_item_use(x, min = NA, max = NA, items = NULL)

ata_item_enemy(x, items)

ata_item_fix(x, items, min = NA, max = NA, forms)

ata_solve(x, solver = c("lpsolve", "glpk"), return_format = c("model",
  "form", "simple"), silent = FALSE, time_limit = 10,
  message = FALSE, ...)

## S3 method for class 'ata'
print(x, ...)

## S3 method for class 'ata'
plot(x, ...)
```

Arguments

pool	the item pool(s), a list of '3pl', 'gpcm', and 'grm' items
n_forms	the number of forms to be assembled
test_len	test length of each form
max_use	maximum use of each item
...	options, e.g. group, common_items, overlap_items
x	an ATA object
coef	the coefficients of the objective function
mode	optimization direction: 'max' for maximization and 'min' for minimization
tol	the tolerance paraemter
negative	TRUE when the objective function is expected to be negative
forms	forms where objectives are added. NULL for all forms
collapse	TRUE to collapse into one objective function
internal_index	TRUE to use internal form indices
target	the target values of the objective function
equal_tol	TRUE to force upward and downward tolerance to be equal
tol_up	the range of upward tolerance
tol_down	the range of downward tolerance
min	the lower bound of the constraint
max	the upper bound of the constraint
level	the level of a categorical variable to be constrained
items	a vector of item indices, NULL for all items
solver	use 'lpsolve' for lp_solve 5.5 or 'glpk' for GLPK
return_format	the format of the results: use 'form' to organize results in a list of forms, 'model' to organize results in a list of models, use 'simple' to organize results in data.frame after removing item paraemters.
silent	TRUE to mute solution information
time_limit	the time limit in seconds passed along to solvers
message	TRUE to print messages from solvers

Details

The ATA model stores the definitions of a MIP model. When ata_solve is called, a real MIP object is created from the definitions.

ata_obj_relative: when mode='max', maximize (y-tol), subject to y <= sum(x) <= y+tol; when mode='min', minimize (y+tol), subject to y-tol <= sum(x) <= y. When negative is TRUE, $y < 0$, $tol > 0$. coef can be a numeric vector that has the same length with the pool, or a variable name in the pool, or a numeric vector of theta points. When tol is NULL, it is optimized; when it's FALSE, ignored; when it's a number, fixed; when it's a range, constrained with lower and upper bounds.

ata_obj_absolute minimizes y0+y1 subject to t-y0 <= sum(x) <= t+y1.

When level is NA, it is assumed that the constraint is on a quantitative item property; otherwise, a categorical item property. coef can be a variable name, a constant, or a numeric vector that has the same size as the pool.

ata_solve takes control options in .... For lpsolve, see lpSolveAPI::lp.control.options. For glpk, see glpkAPI::glpkConstants
Once the model is solved, additional data are added to the model. status shows the status of the solution, optimum the optimal value of the objective fucntion found in the solution, obj_vars the values of two critical variables in the objective function, result the assembly results in a binary matrix, and items the assembled items

Value

ata returns a ata object

ata_solve returns a solved ata object

Examples

```
## generate a pool of 100 items
library(Rirt)
n_items <- 100
pool <- with(model_3pl_gendata(1, n_items), data.frame(id=1:n_items, a=a, b=b, c=c))
pool$content <- sample(1:3, n_items, replace=TRUE)
pool$time <- round(rlnorm(n_items, log(60), .2))
pool$group <- sort(sample(1:round(n_items/3), n_items, replace=TRUE))
pool <- list('3pl'=pool)

## ex. 1: four 10-item forms, maximize b parameter
x <- ata(pool, 4, test_len=10, max_use=1)
x <- ata_relative_objective(x, "b", "max")
x <- ata_solve(x, time_limit=2)
with(x$items$'3pl', aggregate(b, by=list(form=form), mean))
with(x$items$'3pl', table(form))


## ex. 2: four 10-item forms, minimize b parameter
x <- ata(pool, 4, test_len=10, max_use=1)
x <- ata_relative_objective(x, "b", "min", negative=TRUE)
x <- ata_solve(x, time_limit=5)
with(x$items$'3pl', aggregate(b, by=list(form=form), mean))
with(x$items$'3pl', table(form))

## ex. 3: two 10-item forms, mean(b)=0, sd(b)=1
## content = (3, 3, 4), avg. time = 55--65 seconds
constr <- data.frame(name='content',level=1:3, min=c(3,3,4), max=c(3,3,4), stringsAsFactors=FALSE)
constr <- rbind(constr, c('time', NA, 55*10, 65*10))
x <- ata(pool, 2, test_len=10, max_use=1)
x <- ata_absolute_objective(x, pool$'3pl'$b, target=0*10)
x <- ata_absolute_objective(x, (pool$'3pl'$b-0)^2, target=1*10)
for(i in 1:nrow(constr))
  x <- with(constr, ata_constraint(x, name[i], min[i], max[i], level=level[i]))
x <- ata_solve(x)
```

```
with(x$items$'3pl', aggregate(b, by=list(form=form), mean))
with(x$items$'3pl', aggregate(b, by=list(form=form), sd))
with(x$items$'3pl', aggregate(time, by=list(form=form), mean))
with(x$items$'3pl', aggregate(content, by=list(form=form), function(x) freq(x, 1:3)$freq))

## ex. 4: two 10-item forms, max TIF over (-1, 1), consider item sets
x <- ata(pool, 2, test_len=10, max_use=1, group="group")
x <- ata_relative_objective(x, seq(-1, 1, .5), 'max')
x <- ata_solve(x, time_limit=5)
plot(x)
```

Index