Package ‘Rmst’

November 12, 2019

Type Package

Title Computerized Adaptive Multistage Testing

Version 0.0.3

Date 2019-11-12

Author Xiao Luo [aut, cre]

Maintainer Xiao Luo <xluo1986@gmail.com>


License GPL (>= 3)

Depends R (>= 3.6.0)

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

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

Imports ggplot2, Rata, reshape2, Rirt, stats

Suggests testthat

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2019-11-12 17:30:15 UTC

R topics documented:

  assembly .............................................................. 2
  print.mst_sim ...................................................... 4

Index 6
assembly

Assemble Computerized Adaptive Multistage Testing

Description

mst creates a multistage (MST) assembly model
mst_route adds/removes a route to/from the assembly model
mst_objective adds an objective to the assembly model
mst_constraint adds constraints to the assembly model
mst_stage_length sets length limits on stages
mst_rdp anchors the routing decision point (rdp) between adjacent modules
mst_module_info sets the information requirements for modules
mst_assemble tries to solve the assembly model
mst_get_items retrieves items from the assembly results

Usage

mst(pool, design, n_panels = 1, method = c("topdown", "bottomup"),
    test_len = NULL, max_use = NULL, ...)
mst_route(x, route, op = c("+", "-"))
mst_objective(x, coef, mode = c("max", "min"), indices = NULL,
    target = NULL, method = NULL, ...)
mst_constraint(x, coef, min = NA, max = NA, level = NULL,
    indices = NULL, method = NULL)
mst_stage_length(x, stages, min = NA, max = NA)
mst_rdp(x, theta, indices, tol = 0.5)
mst_module_info(x, theta, min = NA, max = NA, indices)
mst_assemble(x, solver = c("lp_solve", "glpk"), silent = FALSE,
    time_limit = 30, message = FALSE, ...)
mst_get_items(x, panel_ix = NULL, stage_ix = NULL, module_ix = NULL,
    route_ix = NULL)

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

## S3 method for class 'mst'
plot(x, ...)
Arguments

- **pool**: the item pool (a list of '3pl', 'gpcm', and 'grm' items)
- **design**: the MST design (string): e.g., "1-3", "1-2-2", "1-2-3"
- **n_panels**: the number of panels (integer)
- **method**: the design method (string): 'topdown' or 'bottomup'
- **test_len**: the module/route length (integer)
- **max_use**: the maximum selection of items (integer)
- **x**: the MST object
- **route**: a MST route (a vector of module index)
- **op**: "+" to add a route and "-" to remove a route
- **coef**: the coefficients (numeric vector or string)
- **mode**: the optimization direction: "max" or "min"
- **indices**: the indices of the route (topdown) or the module (bottomup) where the objective is added
- **target**: the target values of the absolute objectives, NULL for the relative objective
- **min**: the lower bound of the constraint
- **max**: the upper bound of the constraint
- **level**: the constrained level of categorical item attribute, NULL for continuous item attributes
- **stages**: the stage indices
- **theta**: the theta point where TIF is controlled
- **tol**: tolerance parameter (numeric)
- **solver**: the MIP solver: "lpSolve" or "glpk"
- **silent**: TRUE to mute solving status
- **time_limit**: the time limit for solving the model in seconds
- **message**: TRUE to print messages from the solver
- **panel_ix**: the panel index (int vector)
- **stage_ix**: the stage index (int vector)
- **module_ix**: the module index (int vector)
- **route_ix**: the route index (int vector)

Details

A `mst` object stores the definitions of the MST. When `mst_assemble` is called, definitions are converted to a real mixed integer programming model for assembly. If the model is solved, assembled items are appended to the original object.

The bottom-up approach adds objectives and constraints on individual modules, and the top-down approach adds objectives and constraints on routes.
coef in \texttt{mst.objective} can be a vector of theta points where TIFs are optimized, or a continuous variable in the pool where the item attribute is optimized, or a numeric value with the same length of the pool at either item or group level.

\texttt{plot.mst} draws module information functions when \texttt{byroute=FALSE} and route information functions when \texttt{byroute=TRUE}. Use \texttt{label=TRUE} to put labels on routes and modules.

\textbf{Value}

\texttt{mst} returns a \texttt{mst} object.

\texttt{mst.get.items} returns the assembled forms in a list of 3pl, gpcm, and grm items

\textbf{Examples}

```r
## generate item pool
set.seed(123456)
items <- Rirt::model_mixed_gendata(1, n_3pl=200)$items

## Ex. 1: 1-2-2 MST, 2 panels, 20 items, topdown
## maximize info. at -1 and 1 for easy and hard routes
x <- mst(items, "1-2-2", n_panels=2, method="topdown", test_len=10, max_use=1)
x <- mst.objective(x, -1, indices=1:2)
x <- mst.objective(x, 1, indices=3:4)
x <- mst.assemble(x, "lp solve", time_limit=30)
plot(x, byroute=TRUE, label=TRUE)

## Ex. 2: 1-2-3 MST, 2 panels, bottomup,
## remove two routes with large theta change: 1-2-6, 1-3-4
## 10 items in each module, content= and 3 items in content area 1 in each module
## maximize info. at -1, 0 and 1 for easy, medium, and hard modules
x <- mst(items, "1-2-3", 1, 'bottomup', len=10, max_use=1)
x <- mst.route(x, c(1, 2, 6), "-")
x <- mst.route(x, c(1, 3, 4), "-")
x <- mst.objective(x, 0, indices=c(1, 5))
x <- mst.objective(x, -1, indices=c(2, 4))
x <- mst.objective(x, 1, indices=c(3, 6))
x <- mst.assemble(x, timeout=30)
plot(x, byroute=FALSE)
plot(x, byroute=TRUE)
```

\textbf{print.mst_sim} \hspace{2cm} \textit{Simulate the Administration of Multistage Tests}

\textbf{Description}

\texttt{mst_sim} simulates the administration of the assembled MST panel(s)
Usage

```r
## S3 method for class 'mst_sim'
print(x, ...)
```

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

```r
mst_sim(x, true, rdp = NULL, estimator = model_mixed_eap, ...)
```

Arguments

- `x`: the assembled MST object
- `...`: additional option/control parameters
- `true`: the true theta parameter (numeric)
- `rdp`: routing decision points (list)
- `estimator`: the estimator of the ability parameter (function)

Details

Use `theta` to set the initial theta, `panel` to select the MST panel, `prior` to set the prior for theta estimation, `bounds` to set the bounds of theta estimation, and `D` to set the scaling constant.

Value

A list of true and estimated ability theta, administered items, and end-of-stage statistics.

Examples

```r
set.seed(123456)
items <- Rirt::model_mixed_gendata(1, n_3pl=150)$items
x <- mst(items, "1-3", 2, "topdown", len=20, max_use=1)
x <- mst_objective(x, -1, indices=1)
x <- mst_objective(x, 0, indices=2)
x <- mst_objective(x, 1, indices=3)
x <- mst_stage_length(x, 1:2, min=5)
x <- mst_assemble(x, "lpsolve", time_limit=30)

sim1 <- mst_sim(x, true=.5)
print(sim1)
plot(sim1)

sim2 <- mst_sim(x, true=-.5, rdp=list("stage2"=c(-.44, .44)))
print(sim2)
plot(sim2)
```
Index

assembly, 2
mst (assembly), 2
mst_assemble (assembly), 2
mst_constraint (assembly), 2
mst_get_items (assembly), 2
mst_module_info (assembly), 2
mst_objective (assembly), 2
mst_rdp (assembly), 2
mst_route (assembly), 2
mst_sim (print.mst_sim), 4
mst_stage_length (assembly), 2
plot.mst (assembly), 2
plot.mst_sim (print.mst_sim), 4
print.mst (assembly), 2
print.mst_sim, 4
sim (print.mst_sim), 4