Package: RSCAT (via r-universe)

October 16, 2024

Title Shadow-Test Approach to Computerized Adaptive Testing

Version 1.1.3

Author Bingnan Jiang [aut, cre], ACT, Inc. [cph]

Maintainer Bingnan Jiang <bnjiangece@gmail.com>

BugReports https://github.com/act-org/RSCAT/issues

Description As an advanced approach to computerized adaptive testing (CAT), shadow testing (van der Linden(2005) <doi:10.1007/0-387-29054-0>) dynamically assembles entire shadow tests as a part of selecting items throughout the testing process. Selecting items from shadow tests guarantees the compliance of all content constraints defined by the blueprint. 'RSCAT' is an R package for the shadow-test approach to CAT. The objective of 'RSCAT' is twofold: 1) Enhancing the effectiveness of shadow-test CAT simulation; 2) Contributing to the academic and scientific community for CAT research. RSCAT is currently designed for dichotomous items based on the three-parameter logistic (3PL) model.

Depends R (>= 3.4.0), rJava, shiny, shinycssloaders, shinyjs

License CC BY-NC 4.0

Encoding UTF-8

RoxygenNote 7.1.1

Imports Metrics, ggplot2, gridExtra, grid, methods, stats, utils

Collate 'EAPConfig.R' 'SimResult.R' 'configClasses.R' 'launchApp.R' 'runSim.R' 'scoreMethodConfig.R' 'shinyAppServer.R' 'shinyAppUI.R' 'utilFunctions.R' 'zzz.R'

Suggests testthat

NeedsCompilation no

Repository CRAN

Date/Publication 2021-10-12 16:40:06 UTC

Contents

`CATConfig-class` *CAT configuration*

Description

An S4 Class to represent parameters of the CAT configuration.

Slots

`solverConfig` an instance of the S4 class `SolverConfig` for the MIP solver configuration.

`initialTheta` the initial ability theta value.

`scalingConstant` the constraint to scale a discrimination coefficient. estimated with the logistic model to the normal metric.

`itemSelectionMethod` a character string specifying the item selection method.

`scoreMethodConfig` a rJava `jobjRef` object for CAT scoring method configuration. It is generated by the function `scoreMethodConfig`.

`exposureControlType` a character string specifying the exposure control type. "None" to disable exposure control, "Item" for item-level exposure control, and "Passage" for passage-level exposure control.

`exposureControlRate` an integer value specifying the exposure goal rate.

`lValue` a non-negative integer specifying the nuumber of items to be randomlized.

EAPConfig-class *EAP configuration*

Description

An S4 class to represent expected A posteriori (EAP) scoring algorithm configuration.

Details

An instance of this S4 class can be applied to the generic function `scoreMethodConfig` to create an Java object for scoring method configuration.

Slots

`numQuad` a positive integer specifying the number of quadrature points

`minQuad` a numeric value specifying the minimum quadrature point

`maxQuad` a numeric value specifying the maximum quadrature point

`priorDistType` a character string specifying the prior distribution of ability. "Normal" for Normal distribution and "Uniform" for uniform distribution.

`distParams` a numeric vector specifying parameters of the prior distribution. (mean, sd) for the Normal distribution, (a, b) for the uniform distribution.

launchApp *Launches the shiny app to confiure and run CAT simulations.*

Description

Launches the shiny app to confiure and run CAT simulations.

Usage

```
launchApp()
```

Examples

```
if(interactive()){
launchApp()
}
```

result2CSV *Creates a simulation result CSV file.*

Description

Creates a simulation result CSV file.

Usage

```
result2CSV(simResult, file)
```

Arguments

simResult	an instance of S4 class SimResult.
file	a writable connection or a character string naming the file to write to.

runSim *Run CAT simulations*

Description

runSim runs CAT simulations based on the provided configurations and returns the simulation result.

Usage

```
runSim(catConfig, testConfig, simConfig)
```

Arguments

catConfig	an instance of the S4 class CATConfig for CAT configurations.
testConfig	an instance of the S4 class TestConfig for test specification configuration.
simConfig	an instance of the S4 class SimConfig for test specification configuration.

Details

This function calls the Java helper method org.act.util.RHelper.runSim via rJava to exceute CAT simulation.

Value

the simulation result in the instance of SimResult.

Examples

```
if(interactive()){
## Defines item attributes types
itemNumericColumn <- c(FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, TRUE,
  TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, FALSE, TRUE, TRUE, FALSE, FALSE, FALSE,
  FALSE, TRUE, FALSE, TRUE, FALSE, FALSE,FALSE)

## Specifies the item pool file
itemPoolCSVPath <- system.file("extdata", "itempool10Items.csv",
  package = "RSCAT")

## Specifies the constraint table file
constraintCSVPath <- system.file("extdata", "constraintSet1.csv",
  package = "RSCAT")

## Configures solver parameters
solverConfig <- SolverConfig(absGap = 1e-3, relGap = 1e-3, intTol = 1e-6)

## Configures the EAP estimaition
eapConfig <- EAPConfig(numQuad = 6L, minQuad = -2, maxQuad = 2,
  priorDistType = "Normal", distParams = c(0,1))

## Configures CAT
catConfig <- CATConfig(solverConfig = solverConfig,
  scoreMethodConfig = scoreMethodConfig(eapConfig), lValue = 3L)

## Configures test specifications
testConfig <- TestConfig(testConfigID = "Test1", testLength = 6L,
  itempoolPath = itemPoolCSVPath, constraintPath = constraintCSVPath,
  itemNumericColumn = itemNumericColumn)

## Configures the simulation
simConfig <- SimConfig(simID = "Sim1", numExaminees = 8L)

## Runs CAT simulation
simResult <- runSim(catConfig, testConfig, simConfig)
}
```

`scoreMethodConfig` *Creates a scoring method configuration for CAT simulation*

Description

This is a gneric function to create a scoring method configuration from a speicfic estimation algorithm configuration.

Usage

```
scoreMethodConfig(object)
```

```
## S4 method for signature 'EAPConfig'
scoreMethodConfig(object)
```

Arguments

object	an S4 object for the estimation algorithm configuration

Value

the object of scoring method configuration which is an instance of org/act/rscat/cat/ScoringMethodConfig

Examples

```
if(interactive()){
eapConfig <- EAPConfig(numQuad = 6L, minQuad = -2, maxQuad = 2,
  priorDistType = "Normal", distParams = c(0,1))
scoreMethodConfig <- scoreMethodConfig(eapConfig)
}
```

---

shinyAppServer *Defines server logic to configure and run CAT simulations.*

Description

Defines server logic to configure and run CAT simulations.

Usage

```
shinyAppServer(input, output)
```

Arguments

input	an object that stores the current values of all of the widgets in the app.
output	an object that stores instructions for building the R objects in the app.

shinyAppUI *Defines UI for CAT simulations.*

Description

Defines UI for CAT simulations.

Usage

```
shinyAppUI
```

Format

An object of class `shiny.tag.list` (inherits from `list`) of length 3.

SimConfig-class *CAT simulation configuration*

Description

An S4 class to represent CAT simulation configuraiton.

Slots

`simID` a character string as the identifier of the CAT simulation.

`numExaminees` a positive integer specifying the number of simulated examinees.

`trueThetaDistType` a character string specifying the distribution of true ability of simulated examinees. "Normal" for the Normal distribution and "Uniform" for the uniform distribution.

`trueThetaDistParams` a numeric vector spcifying parameters of the prior distribuution. (mean, sd) for the Normal distribution, (a, b) for the uniform distribution.

SimResult-class *CAT simulation result*

Description

An S4 class to represent CAT simulation results.

Slots

`numExaminees` a positive integer representing the number of simulated examinees.

`trueThetas` a numeric vector representing the true theta values of simulated examinees.

`finalThetas` a numeric vector representing the final theta estimates of simulated examinees.

`finalThetaSEs` a numeric vector representing the final theta estimate standard errors (SEs) of simulated examinees.

`estThetas` a list of length `numExaminees`. Each element of the list is a numeric vector representing theta estimate at adaptive stages for the simulated examinee.

`estThetaSEs` a list of length `numExaminees`. Each element of the list is a numeric vector representing theta estimate standard error (SE) at adaptive stages for the simulated examinee.

`scores` a list of length `numExaminees`. Each element of the list is a numeric vector representing scores at adaptive stages for the simulated examinee. 0 for an incorrect repsonse and 1 for a correct respone.

`itemsAdministered` a list of length `numExaminees`. Each element of the list is a character vector representing identifiers of adminsitered items at adaptive stages for the simulated examinee.

`shadowTests` a list of length `numExaminees`. Each element of the list is also a list representing the shadow test assembled at each adaptive stage.

`engineTime` a list of length `numExaminees`. Each element of the list is a numeric vector representing the engine time at each adaptive step. the engine time includes time consumed by CAT algorithms and shadow test assembly.

SolverConfig-class *MIP solver configuration*

Description

An S4 Class to represent parameters of the MIP solver configuration.

Slots

`absGap` the absolute gap target to terminate the MIP solving.

`relGap` the relative gap target to terminate the MIP solving.

`intTol` the integer tolerance for the MIP solving. if the solution x is between -intTol and intTol, x <= 0 is true if the value of x is at most intTol. x > 0 is fulfilled if x > intTol.

summary,SimResult-method *Generates CAT simulation summary*

Description

Generates CAT simulation summary

Usage

```
## S4 method for signature 'SimResult'
summary(object)
```

Arguments

object	an object of SimResult. Generates the summary report of CAT simulation.

TestConfig-class *Test specification configuration*

Description

An S4 class to represent test blueprint and specification.

Slots

testConfigID a character string as the identifier of the test configuration.

testLength a positive integer specifying the test length.

itempoolPath a character string specifying the location of the item pool csv file.

passagepoolPath a character string specifying the location of the passage pool csv file.

constraintPath a character string specifying the location of the constraint csv file.

itemNumericColumn a boolean vector indicating whether item attribute columns in the item pool table are numeric or not.

passageNumericColumn a boolean vector indicating whether passage attribute columns in the passage pool table are numeric or not.

enableEnemyItem a boolean indicator to specify if enemy item constraints defined by in the item pool is enabled or not.

numPassageLB an integer specifying the minimum number of passages in the test.

numPassageUB an integer specifying the maximum number of passages in the test.

numItemPerPassageLB an integer specifying the minimum number of items in a passages in the test.

numItemPerPassageUB an integer specifying the maximum number of items in a passages in the test.

Index