Package 'TestDesign'

September 12, 2019

Type Package

Title Optimal Test Design Approach to Fixed and Adaptive Test Construction

Version 0.2.5

Date 2019-09-11

Maintainer Seung W. Choi <schoi@austin.utexas.edu>

Description Use the optimal test design approach by Birnbaum (1968, ISBN:9781593119348) and van der Linden (2018) <doi:10.1201/9781315117430> in constructing fixed and adaptive tests. Supports the following mixed-integer programming (MIP) solver packages: 'Rsymphony', 'gurobi', 'lpSolve', and 'Rglpk'. The 'gurobi' package is not available from CRAN; see <https://www.gurobi.com/downloads>. See vignette for installing 'Rsymphony' package on Mac systems.

URL https://github.com/choi-phd/TestDesign

BugReports https://github.com/choi-phd/TestDesign/issues

License GPL (>= 2)

Depends R (>= 2.10)

Imports Rcpp (>= 1.0.0), methods, Matrix, lpSolve, foreach, logitnorm, Rdpack, crayon

Suggests gurobi, Rsymphony, Rglpk, shiny, shinythemes, shinyWidgets, shinyjs, DT, knitr, rmarkdown, kableExtra, testthat (>= 2.1.0)

LinkingTo Rcpp

RoxygenNote 6.1.1

Encoding UTF-8

LazyData true

RdMacros Rdpack

VignetteBuilder knitr

Collate 'import.R' 'RcppExports.R' 'item_class.R' 'item_functions.R' 'loading_functions.R' 'ATA_class.R' 'ATA_functions.R' 'solver_functions.R' 'shadow_class.R' 'shadow_functions.R' 'datasets.R' 'runshiny.R'

NeedsCompilation yes

Author Seung W. Choi [aut, cre] (<https://orcid.org/0000-0003-4777-5420>), Sangdon Lim [aut] (<https://orcid.org/0000-0002-2988-014X>)

Repository CRAN

Date/Publication 2019-09-12 05:30:14 UTC

R topics documented:

`addTrans` *Add transparancy to color*

Description

Add transparancy to color.

Usage

```
addTrans(color, alpha)
```

Arguments

`color`	A vector of color names or RGB color codes.
`alpha`	A vector of integers between 0 and 255 (0 = fully transparent, 255 = fully visible).

array_info_1pl *Calculate Fisher information at multiple thetas (1PL)*

Description

Calculate the Fisher information at theta values according to the 1PL model.

Usage

```
array_info_1pl(x, b)
```

Arguments

x	Numeric. A vector of theta values.
b	Numeric. A difficulty parameter value.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

array_info_2pl *Calculate Fisher information at multiple thetas (2PL)*

Description

Calculate the Fisher information at theta values according to the 2PL model.

Usage

```
array_info_2pl(x, a, b)
```

Arguments

x	Numeric. A vector of theta values.
a	Numeric. A slope parameter value.
b	Numeric. A difficulty parameter value.

References

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

`array_info_3pl` *Calculate Fisher information at multiple thetas (3PL)*

Description

Calculate the Fisher information at theta values according to the 3PL model.

Usage

```
array_info_3pl(x, a, b, c)
```

Arguments

`x`	Numeric. A vector of theta values.
`a`	Numeric. A slope parameter value.
`b`	Numeric. A difficulty parameter value.
`c`	Numeric. A guessing parameter value.

References

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

array_info_gpc *Calculate Fisher information at multiple thetas (GPC)*

Description

Calculate the Fisher information at theta values according to the generalized partial credit model.

Usage

```
array_info_gpc(x, a, b)
```

Arguments

x	Numeric. A vector of theta values.
a	Numeric. A slope parameter value.
b	Numeric. A vector of threshold parameter values.

References

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

array_info_gr *Calculate Fisher information at multiple thetas (GR)*

Description

Calculate the Fisher information at theta values according to the graded response model.

Usage

```
array_info_gr(x, a, b)
```

Arguments

x	Numeric. A vector of theta values.
a	Numeric. A slope parameter value.
b	Numeric. A vector of category boundary parameter values.

References

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

array_info_pc *Calculate Fisher information at multiple thetas (PC)*

Description

Calculate the Fisher information at theta values according to the partial credit model.

Usage

```
array_info_pc(x, b)
```

Arguments

x	Numeric. A vector of theta values.
b	Numeric. A vector of threshold parameter values.

References

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

array_p_1pl *Calculate probability at multiple thetas (1PL)*

Description

Calculate the probability of correct response at theta values, under the 1PL model.

Usage

```
array_p_1pl(x, b)
```

Arguments

x	Numeric. A vector of theta values.
b	Numeric. A difficulty parameter value.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

array_p_2pl *Calculate probability at multiple thetas (2PL)*

Description

Calculate the probability of correct response at theta values, under the 2PL model.

Usage

```
array_p_2pl(x, a, b)
```

Arguments

x	Numeric. A vector of theta values.
a	Numeric. A slope parameter value.
b	Numeric. A difficulty parameter value.

References

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

array_p_3pl *Calculate probability at multiple thetas (3PL)*

Description

Calculate the probability of correct response at theta values, under the 3PL model.

Usage

```
array_p_3pl(x, a, b, c)
```

Arguments

x	Numeric. A vector of theta values.
a	Numeric. A slope parameter value.
b	Numeric. A difficulty parameter value.
c	Numeric. A guessing parameter value.

References

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

`array_p_gpc` *Calculate probability at multiple thetas (GPC)*

Description

Calculate the probability of correct response at theta values, under the generalized partial credit model.

Usage

```
array_p_gpc(x, a, b)
```

Arguments

`x`	Numeric. A vector of theta values.
`a`	Numeric. A slope parameter value.
`b`	Numeric. A vector of threshold parameter values.

References

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

`array_p_gr` *Calculate probability at multiple thetas (GR)*

Description

Calculate the probability of correct response at theta values, under the graded response model.

Usage

```
array_p_gr(x, a, b)
```

Arguments

`x`	Numeric. A vector of theta values.
`a`	Numeric. A slope parameter value.
`b`	Numeric. A vector of category boundary parameter values.

References

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

`array_p_pc` *Calculate probability at multiple thetas (PC)*

Description

Calculate the probability of correct response at theta values, under the partial credit model.

Usage

```
array_p_pc(x, b)
```

Arguments

`x`	Numeric. A vector of theta values.
`b`	Numeric. A vector of threshold parameter values.

References

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

`ATA` *Run Automated Test Assembly*

Description

Perform Automated Test Assembly with specified configurations.

Usage

```
ATA(config, constraints, plot = FALSE, plot_range = c(-3, 3))

## S4 method for signature 'config_ATA'
ATA(config, constraints, plot = FALSE,
  plot_range = c(-3, 3))
```

Arguments

config	An config_ATA object containing configuration options. Use createStaticTestConfig for this.
constraints	A list representing optimization constraints. Use loadConstraints for this.
plot	Logical. If TRUE, draw Fisher information plot from the selected items.
plot_range	Numeric. A vector of length 2 containing the lower and upper bounds of plot range. The default is c(-3,3).

Value

A list containing the following entries:

- MIP A list containing the result from MIP solver.
  - solution Solution vector. Each value represents an item. A value of 1 indicates the item was selected.
  - objval Objective value of the solution.
  - status Status value indicating whether an optimal solution was found.
- selected The attributes of the selected items.
- solver The name of the MIP solver used in the assembly.
- obj_value Objective value of the solution. Identical to the one above.
- solve_time The elapsed time in running the solver.

References

van der Linden WJ (2005). *Linear Models for Optimal Test Design*. Springer Science & Business Media.

Examples

```
config_science <- createStaticTestConfig(
  list(
    method = "MAXINFO",
    target_location = c(-1, 1)
  )
)
solution <- ATA(config_science, constraints_science, plot = TRUE)
```

buildConstraints *Build constraints*

Description

Read constraints from specified files.

Usage

```
buildConstraints(pool, file_constraints, file_item_attrib,
  file_st_attrib = NULL)
```

Arguments

pool	An item_pool object. Use loadItemPool for this.
file_constraints	Character. The name of the file containing constraint specifications.
file_item_attrib	Character. The name of the file containing item attributes.
file_st_attrib	(Optional) Character. The name of the file containing set attributes.

Value

A list containing the parsed constraints, to be used in ATA and Shadow.

Examples

```
## Write to tempdir() and clean afterwards
f1 <- file.path(tempdir(), "constraints_science.csv")
write.csv(constraints_science_raw, f1, row.names = FALSE)
f2 <- file.path(tempdir(), "itemattrib_science.csv")
write.csv(itemattrib_science_raw, f2, row.names = FALSE)

constraints <- buildConstraints(itempool_science, f1, f2)

file.remove(f1)
file.remove(f2)
```

`calcDerivative` *Calculate first derivative*

Description

An S4 generic and its methods to calculate the first derivative of the probability function.

Usage

```
calcDerivative(object, theta)

## S4 method for signature 'item_1PL,numeric'
calcDerivative(object, theta)

## S4 method for signature 'item_2PL,numeric'
calcDerivative(object, theta)

## S4 method for signature 'item_3PL,numeric'
calcDerivative(object, theta)

## S4 method for signature 'item_PC,numeric'
calcDerivative(object, theta)

## S4 method for signature 'item_GPC,numeric'
calcDerivative(object, theta)

## S4 method for signature 'item_GR,numeric'
calcDerivative(object, theta)

## S4 method for signature 'item_pool,numeric'
calcDerivative(object, theta)

## S4 method for signature 'pool_cluster,numeric'
calcDerivative(object, theta)
```

Arguments

`object`	An instance of an item class.
`theta`	A vector of theta values.

Value

First derivative values.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

Examples

```
item_1   <- new("item_1PL", difficulty = 0.5)
d.item_1 <- calcDerivative(item_1, seq(-3, 3, 1))
item_2   <- new("item_2PL", slope = 1.0, difficulty = 0.5)
d.item_2 <- calcDerivative(item_2, seq(-3, 3, 1))
item_3   <- new("item_3PL", slope = 1.0, difficulty = 0.5, guessing = 0.2)
d.item_3 <- calcDerivative(item_3, seq(-3, 3, 1))
item_4   <- new("item_PC", threshold = c(-1, 0, 1), ncat = 4)
d.item_4 <- calcDerivative(item_4, seq(-3, 3, 1))
item_5   <- new("item_GPC", slope = 1.2, threshold = c(-0.8, -1.0, 0.5), ncat = 4)
d.item_5 <- calcDerivative(item_5, seq(-3, 3, 1))
item_6   <- new("item_GR", slope = 0.9, category = c(-1, 0, 1), ncat = 4)
d.item_6 <- calcDerivative(item_6, seq(-3, 3, 1))
d_itempool <- calcDerivative(itempool_science, seq(-3, 3, 1))
```

`calcDerivative2` *Calculate second derivative*

Description

An S4 generic and its methods to calculate the second derivative of the probability function.

Usage

```
calcDerivative2(object, theta)

## S4 method for signature 'item_1PL,numeric'
calcDerivative2(object, theta)

## S4 method for signature 'item_2PL,numeric'
calcDerivative2(object, theta)

## S4 method for signature 'item_3PL,numeric'
calcDerivative2(object, theta)

## S4 method for signature 'item_PC,numeric'
calcDerivative2(object, theta)

## S4 method for signature 'item_GPC,numeric'
calcDerivative2(object, theta)

## S4 method for signature 'item_GR,numeric'
calcDerivative2(object, theta)

## S4 method for signature 'item_pool,numeric'
calcDerivative2(object, theta)

## S4 method for signature 'pool_cluster,numeric'
calcDerivative2(object, theta)
```

Arguments

`object`	An instance of an item class.
`theta`	A vector of theta values.

Value

Second derivative values.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

Examples

```
item_1    <- new("item_1PL", difficulty = 0.5)
dd_item_1 <- calcDerivative2(item_1, seq(-3, 3, 1))
item_2    <- new("item_2PL", slope = 1.0, difficulty = 0.5)
dd_item_2 <- calcDerivative2(item_2, seq(-3, 3, 1))
item_3    <- new("item_3PL", slope = 1.0, difficulty = 0.5, guessing = 0.2)
dd_item_3 <- calcDerivative2(item_3, seq(-3, 3, 1))
item_4    <- new("item_PC", threshold = c(-1, 0, 1), ncat = 4)
dd_item_4 <- calcDerivative2(item_4, seq(-3, 3, 1))
item_5    <- new("item_GPC", slope = 1.2, threshold = c(-0.8, -1.0, 0.5), ncat = 4)
dd_item_5 <- calcDerivative2(item_5, seq(-3, 3, 1))
item_6    <- new("item_GR", slope = 0.9, category = c(-1, 0, 1), ncat = 4)
dd_item_6 <- calcDerivative2(item_6, seq(-3, 3, 1))
dd_itempool <- calcDerivative2(itempool_science, seq(-3, 3, 1))
```

---

calcEscore *Calculate expected scores*

Description

An S4 generic and its methods to calculate expected scores given a vector of thetas for different item classes.

Usage

```
calcEscore(object, theta)

## S4 method for signature 'item_1PL,numeric'
calcEscore(object, theta)

## S4 method for signature 'item_2PL,numeric'
calcEscore(object, theta)
```

```
## S4 method for signature 'item_3PL,numeric'
calcEscore(object, theta)

## S4 method for signature 'item_PC,numeric'
calcEscore(object, theta)

## S4 method for signature 'item_GPC,numeric'
calcEscore(object, theta)

## S4 method for signature 'item_GR,numeric'
calcEscore(object, theta)

## S4 method for signature 'item_pool,numeric'
calcEscore(object, theta)

## S4 method for signature 'pool_cluster,numeric'
calcEscore(object, theta)
```

Arguments

`object`	An instance of an item class.
`theta`	A vector of theta values.

Value

A vector of expected scores of length nq (the number of values on theta grid).

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

Examples

```
item_1     <- new("item_1PL", difficulty = 0.5)
ICC_item_1 <- calcEscore(item_1, seq(-3, 3, 1))
item_2     <- new("item_2PL", slope = 1.0, difficulty = 0.5)
ICC_item_2 <- calcEscore(item_2, seq(-3, 3, 1))
item_3     <- new("item_3PL", slope = 1.0, difficulty = 0.5, guessing = 0.2)
ICC_item_3 <- calcEscore(item_3, seq(-3, 3, 1))
item_4     <- new("item_PC", threshold = c(-1, 0, 1), ncat = 4)
ICC_item_4 <- calcEscore(item_4, seq(-3, 3, 1))
item_5     <- new("item_GPC", slope = 1.2, threshold = c(-0.8, -1.0, 0.5), ncat = 4)
ICC_item_5 <- calcEscore(item_5, seq(-3, 3, 1))
item_6     <- new("item_GR", slope = 0.9, category = c(-1, 0, 1), ncat = 4)
ICC_item_6 <- calcEscore(item_6, seq(-3, 3, 1))
TCC_itempool <- calcEscore(itempool_science, seq(-3, 3, 1))
```

---

calcFisher *Calculate Fisher information*

Description

An S4 generic and its methods to calculate Fisher information given a vector of thetas for different item classes.

Usage

```
calcFisher(object, theta)

## S4 method for signature 'item_1PL,numeric'
calcFisher(object, theta)

## S4 method for signature 'item_2PL,numeric'
calcFisher(object, theta)

## S4 method for signature 'item_3PL,numeric'
calcFisher(object, theta)

## S4 method for signature 'item_PC,numeric'
calcFisher(object, theta)

## S4 method for signature 'item_GPC,numeric'
calcFisher(object, theta)

## S4 method for signature 'item_GR,numeric'
calcFisher(object, theta)
```

```
## S4 method for signature 'item_pool,numeric'
calcFisher(object, theta)

## S4 method for signature 'pool_cluster,numeric'
calcFisher(object, theta)
```

Arguments

object	An instance of an item class.
theta	A vector of theta values.

Value

A vector of Fisher information values over theta (nq values) for a single item or a matrix of dimension (nq, ni) for an "item_pool".

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

Examples

```
item_1      <- new("item_1PL", difficulty = 0.5)
info_item_1 <- calcFisher(item_1, seq(-3, 3, 1))
item_2      <- new("item_2PL", slope = 1.0, difficulty = 0.5)
info_item_2 <- calcFisher(item_2, seq(-3, 3, 1))
item_3      <- new("item_3PL", slope = 1.0, difficulty = 0.5, guessing = 0.2)
```

```
info_item_3 <- calcFisher(item_3, seq(-3, 3, 1))
item_5      <- new("item_GPC", slope = 1.2, threshold = c(-0.8, -1.0, 0.5), ncat = 4)
info_item_5 <- calcFisher(item_5, seq(-3, 3, 1))
item_6      <- new("item_GR", slope = 0.9, category = c(-1, 0, 1), ncat = 4)
info_item_6 <- calcFisher(item_6, seq(-3, 3, 1))
info_itempool <- calcFisher(itempool_science, seq(-3, 3, 1))
```

`calcHessian` *Calculate second derivative of log-likelihood*

Description

An S4 generic and its methods to calculate the second derivative of the log-likelihood function.

Usage

```
calcHessian(object, theta, resp)

## S4 method for signature 'item_1PL,numeric,numeric'
calcHessian(object, theta, resp)

## S4 method for signature 'item_2PL,numeric,numeric'
calcHessian(object, theta, resp)

## S4 method for signature 'item_3PL,numeric,numeric'
calcHessian(object, theta, resp)

## S4 method for signature 'item_PC,numeric,numeric'
calcHessian(object, theta, resp)

## S4 method for signature 'item_GPC,numeric,numeric'
calcHessian(object, theta, resp)

## S4 method for signature 'item_GR,numeric,numeric'
calcHessian(object, theta, resp)

## S4 method for signature 'item_pool,numeric,numeric'
calcHessian(object, theta, resp)

## S4 method for signature 'pool_cluster,numeric,list'
calcHessian(object, theta, resp)
```

Arguments

`object`	An instance of an item class.
`theta`	A vector of theta values.
`resp`	Response data.

Value

Second derivative values of log-likelihoods.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

Examples

```
item_1   <- new("item_1PL", difficulty = 0.5)
h_item_1 <- calcHessian(item_1, seq(-3, 3, 1), 0)
item_2   <- new("item_2PL", slope = 1.0, difficulty = 0.5)
h_item_2 <- calcHessian(item_2, seq(-3, 3, 1), 0)
item_3   <- new("item_3PL", slope = 1.0, difficulty = 0.5, guessing = 0.2)
h_item_3 <- calcHessian(item_3, seq(-3, 3, 1), 0)
item_4   <- new("item_PC", threshold = c(-1, 0, 1), ncat = 4)
h_item_4 <- calcHessian(item_4, seq(-3, 3, 1), 0)
item_5   <- new("item_GPC", slope = 1.2, threshold = c(-0.8, -1.0, 0.5), ncat = 4)
h_item_5 <- calcHessian(item_5, seq(-3, 3, 1), 0)
item_6   <- new("item_GR", slope = 0.9, category = c(-1, 0, 1), ncat = 4)
h_item_6 <- calcHessian(item_6, seq(-3, 3, 1), 0)
h_itempool <- calcHessian(itempool_science, seq(-3, 3, 1), 0)
```

`calcJacobian` *Calculate first derivative of log-likelihood*

Description

An S4 generic and its methods to calculate the first derivative of the log-likelihood function.

Usage

```
calcJacobian(object, theta, resp)

## S4 method for signature 'item_1PL,numeric,numeric'
calcJacobian(object, theta, resp)

## S4 method for signature 'item_2PL,numeric,numeric'
calcJacobian(object, theta, resp)

## S4 method for signature 'item_3PL,numeric,numeric'
calcJacobian(object, theta, resp)

## S4 method for signature 'item_PC,numeric,numeric'
calcJacobian(object, theta, resp)

## S4 method for signature 'item_GPC,numeric,numeric'
calcJacobian(object, theta, resp)

## S4 method for signature 'item_GR,numeric,numeric'
calcJacobian(object, theta, resp)

## S4 method for signature 'item_pool,numeric,numeric'
calcJacobian(object, theta, resp)

## S4 method for signature 'pool_cluster,numeric,list'
calcJacobian(object, theta, resp)
```

Arguments

`object`	An instance of an item class.
`theta`	A vector of theta values.
`resp`	Response data.

Value

First derivative values of log-likelihoods.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

Examples

```
item_1   <- new("item_1PL", difficulty = 0.5)
j_item_1 <- calcJacobian(item_1, seq(-3, 3, 1), 0)
item_2   <- new("item_2PL", slope = 1.0, difficulty = 0.5)
j_item_2 <- calcJacobian(item_2, seq(-3, 3, 1), 0)
item_3   <- new("item_3PL", slope = 1.0, difficulty = 0.5, guessing = 0.2)
j_item_3 <- calcJacobian(item_3, seq(-3, 3, 1), 0)
item_4   <- new("item_PC", threshold = c(-1, 0, 1), ncat = 4)
j_item_4 <- calcJacobian(item_4, seq(-3, 3, 1), 0)
item_5   <- new("item_GPC", slope = 1.2, threshold = c(-0.8, -1.0, 0.5), ncat = 4)
j_item_5 <- calcJacobian(item_5, seq(-3, 3, 1), 0)
item_6   <- new("item_GR", slope = 0.9, category = c(-1, 0, 1), ncat = 4)
j_item_6 <- calcJacobian(item_6, seq(-3, 3, 1), 0)
j_itempool <- calcJacobian(itempool_science, seq(-3, 3, 1), 0)
```

`calcLocation` *Calculate item location*

Description

An S4 generic and its methods to calculate item location.

Usage

```
calcLocation(object)

## S4 method for signature 'item_1PL'
calcLocation(object)

## S4 method for signature 'item_2PL'
calcLocation(object)

## S4 method for signature 'item_3PL'
calcLocation(object)

## S4 method for signature 'item_PC'
calcLocation(object)

## S4 method for signature 'item_GPC'
calcLocation(object)

## S4 method for signature 'item_GR'
calcLocation(object)

## S4 method for signature 'item_pool'
calcLocation(object)

## S4 method for signature 'pool_cluster'
calcLocation(object)
```

Arguments

`object`	An instance of an item class.

Value

Item location values.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

Examples

```
item_1       <- new("item_1PL", difficulty = 0.5)
theta_item_1 <- calcLocation(item_1)
item_2       <- new("item_2PL", slope = 1.0, difficulty = 0.5)
theta_item_2 <- calcLocation(item_2)
item_3       <- new("item_3PL", slope = 1.0, difficulty = 0.5, guessing = 0.2)
theta_item_3 <- calcLocation(item_3)
item_4       <- new("item_PC", threshold = c(-1, 0, 1), ncat = 4)
theta_item_4 <- calcLocation(item_4)
item_5       <- new("item_GPC", slope = 1.2, threshold = c(-0.8, -1.0, 0.5), ncat = 4)
theta_item_5 <- calcLocation(item_5)
item_6       <- new("item_GR", slope = 0.9, category = c(-1, 0, 1), ncat = 4)
theta_item_6 <- calcLocation(item_6)
theta_itempool <- calcLocation(itempool_science)
```

---

calcProb *Calculate item response probabilities*

Description

An S4 generic and its methods to calculate item response probabilities for different item classes

Usage

```
calcProb(object, theta)

## S4 method for signature 'item_1PL,numeric'
calcProb(object, theta)

## S4 method for signature 'item_2PL,numeric'
calcProb(object, theta)

## S4 method for signature 'item_3PL,numeric'
calcProb(object, theta)
```

```
## S4 method for signature 'item_PC,numeric'
calcProb(object, theta)

## S4 method for signature 'item_GPC,numeric'
calcProb(object, theta)

## S4 method for signature 'item_GR,numeric'
calcProb(object, theta)

## S4 method for signature 'item_pool,numeric'
calcProb(object, theta)

## S4 method for signature 'pool_cluster,numeric'
calcProb(object, theta)
```

Arguments

object	An instance of an item class.
theta	A vector of theta values.

Value

A matrix of probability values with a dimension (nq, ncat) for a single item or a list of matrices for an instance of "item_pool".

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

Examples

```
item_1      <- new("item_1PL", difficulty = 0.5)
prob_item_1 <- calcProb(item_1, seq(-3, 3, 1))
item_2      <- new("item_2PL", slope = 1.0, difficulty = 0.5)
prob_item_2 <- calcProb(item_2, seq(-3, 3, 1))
item_3      <- new("item_3PL", slope = 1.0, difficulty = 0.5, guessing = 0.2)
prob_item_3 <- calcProb(item_3, seq(-3, 3, 1))
item_4      <- new("item_PC", threshold = c(-1, 0, 1), ncat = 4)
prob_item_4 <- calcProb(item_4, seq(-3, 3, 1))
item_5      <- new("item_GPC", slope = 1.2, threshold = c(-0.8, -1.0, 0.5), ncat = 4)
prob_item_5 <- calcProb(item_5, seq(-3, 3, 1))
item_6      <- new("item_GR", slope = 0.9, category = c(-1, 0, 1), ncat = 4)
prob_item_6 <- calcProb(item_6, seq(-3, 3, 1))
prob_itempool <- calcProb(itempool_science, seq(-3, 3, 1))
```

---

calcRP *Find matching theta to supplied probability*

Description

Find theta corresponding to a response probability value for each item.

Usage

```
calcRP(object, rp = 0.5, max_iter = 100, conv = 1e-04,
  start_theta = 0)
```

Arguments

object	An item_pool object.
rp	A response probability value.
max_iter	A maximum number of iterations.
conv	A convergence criterion.
start_theta	A starting theta value.

---

calc_info *Calculate the Fisher information matrix for a single theta value and a set of items, potentially with a mixture of different models*

Description

Calculate the Fisher information matrix for a single theta value and a set of items, potentially with a mixture of different models

Usage

```
calc_info(x, item_parm, ncat, model)
```

Arguments

x	Numeric. A single theta value.
item_parm	A matrix of item parameters.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).

---

calc_info_EB *Calculate the Fisher information using empirical Bayes*

Description

Calculate the Fisher information using empirical Bayes.

Usage

```
calc_info_EB(x, item_parm, ncat, model)
```

Arguments

x	A numeric vector of MCMC sampled theta values.
item_parm	A numeric matrix of item parameters.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).

---

calc_info_FB *Calculate the Fisher information using full Bayesian*

Description

Calculate the Fisher information using full Bayesian.

Usage

```
calc_info_FB(x, items_list, ncat, model, useEAP = FALSE)
```

Arguments

x	A numeric vector of MCMC sampled theta values.
items_list	A list of item parameter matrices.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
useEAP	TRUE to use the mean of MCMC theta draws.

---

calc_info_matrix *Calculate the Fisher information matrix for a vector of theta values and a set of items, potentially with a mixture of different models*

Description

Calculate the Fisher information matrix for a vector of theta values and a set of items, potentially with a mixture of different models

Usage

```
calc_info_matrix(x, item_parm, ncat, model)
```

Arguments

x	Numeric. A vector of theta values.
item_parm	A matrix of item parameters.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).

---

calc_likelihood *Calculate a likelihood value of theta*

Description

Calculate a likelihood value of theta.

Usage

```
calc_likelihood(x, item_parm, resp, ncat, model)
```

Arguments

x	Numeric. A single theta value.
item_parm	A numeric matrix of item parameters.
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).

---

calc_likelihood_function

Calculate a likelihood function of theta

Description

Calculate a likelihood function of theta.

Usage

```
calc_likelihood_function(theta_grid, item_parm, resp, ncat, model)
```

Arguments

theta_grid	An equi-spaced grid of theta values.
item_parm	A numeric matrix of item parameters.
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).

---

calc_log_likelihood *Calculate a log-likelihood value of theta*

Description

Calculate a log-likelihood value of theta.

Usage

```
calc_log_likelihood(x, item_parm, resp, ncat, model, prior, prior_parm)
```

Arguments

x	A length-one numeric vector for a theta value.
item_parm	A numeric matrix of item parameters.
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

---

calc_log_likelihood_function

Calculate a log-likelihood function of theta

Description

Calculate a log-likelihood function of theta.

Usage

```
calc_log_likelihood_function(theta_grid, item_parm, resp, ncat, model,
  prior, prior_parm)
```

Arguments

theta_grid	An equi-spaced grid of theta values.
item_parm	A numeric matrix of item parameters.
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

calc_MI_FB *Calculate the mutual information using full Bayesian*

Description

Calculate the mutual information using full Bayesian.

Usage

```
calc_MI_FB(x, items_list, ncat, model)
```

Arguments

x	A numeric vector of MCMC sampled theta values.
items_list	A list of item parameter matrices.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).

calc_posterior *Calculate a posterior value of theta*

Description

Calculate a posterior value of theta.

Usage

```
calc_posterior(x, item_parm, resp, ncat, model, prior, prior_parm)
```

Arguments

x	A length-one numeric vector for a theta value.
item_parm	A numeric matrix of item parameters.
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

calc_posterior_function *Calculate a posterior distribution of theta*

Description

Calculate a posterior distribution of theta.

Usage

```
calc_posterior_function(theta_grid, item_parm, resp, ncat, model, prior,
  prior_parm)
```

Arguments

theta_grid	An equi-spaced grid of theta values.
item_parm	A numeric matrix of item parameters.
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

calc_posterior_single *Calculate a posterior value of theta for a single item*

Description

Calculate a posterior value of theta for a single item.

Usage

```
calc_posterior_single(x, item_parm, resp, ncat, model, prior, prior_parm)
```

Arguments

x	A length-one numeric vector for a theta value.
item_parm	A numeric vector of item parameters (for one item).
resp	A length-one numeric vector of item responses.
ncat	A length-one numeric vector of the number of response categories by item.
model	A length-one numeric vector of the IRT model by item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

---

checkConstraints *Check the consistency of constraints and item usage*

Description

Check the consistency of constraints and item usage.

Usage

```
checkConstraints(constraints, usage_matrix, true_theta = NULL)
```

Arguments

constraints	A list constraints generated by loadConstraints.
usage_matrix	A matrix of item usage data from Shadow.
true_theta	A vector of true theta values.

---

config_ATA-class *createStaticTestConfig*

Description

Create an config_ATA object for Automated Test Assembly (ATA).

Usage

```
createStaticTestConfig(item_selection = NULL, MIP = NULL)
```

Arguments

`item_selection` A list containing item selection criteria. This should have the following entries:

- `method` The type of criteria. Accepts `MAXINFO`,`TIF`,`TCC`.
- `info_type` The type of information. Accepts `FISHER`.
- `target_location` A numeric vector containing the locations of target theta points. (e.g. `c(-1,0,1)`)
- `target_value` A numeric vector containing the target values at each theta location. This should have the same length with `target_location`. Ignored if method is `MAXINFO`.
- `target_weight` A numeric vector containing the weights for each theta location. This should have the same length with `targetlocation`. Defaults to a vector of 1s.

`MIP` A list containing solver options. This should have the following entries:

- `solver` The type of solver. Accepts `SYMPHONY`,`GUROBI`,`GLPK`,`LPSOLVE`.
- `verbosity` Verbosity level of the solver. Defaults to -2.
- `time_limit` Time limit in seconds passed onto the solver. Defaults to 60. Used in solvers `SYMPHONY`,`GUROBI`,`GLPK`.
- `gap_limit` Termination criterion. Gap limit in relative scale passed onto the solver. Defaults to .05. Used in solver `GUROBI`.
- `gap_limit_abs` Termination criterion. Gap limit in absolute scale passed onto the solver. Defaults to .05. Used in solver `SYMPHONY`.
- `obj_tol` Termination criterion. Tolerance on target objective value in absolute difference scale. Defaults to .05. Ignored if method is `MAXINFO`.

Examples

```
cfg1 <- createStaticTestConfig(
  list(
    method = "MAXINFO",
    info_type = "FISHER",
    target_location = c(-1, 0, 1),
    target_weight = c(1, 1, 1)
  )
)

cfg2 <- createStaticTestConfig(
  list(
    method = "TIF",
    info_type = "FISHER",
    target_location = c(-1, 0, 1),
    target_weight = c(1, 1, 1),
    target_value = c(8, 10, 12)
  )
)

cfg3 <- createStaticTestConfig(
  list(
    method = "TCC",
```

```
    info_type = "FISHER",
    target_location = c(-1, 0, 1),
    target_weight = c(1, 1, 1),
    target_value = c(10, 15, 20)
  )
)
```

---

config_Shadow-class *createShadowTestConfig*

Description

Create a config_Shadow object for Shadow Test Assembly (STA).

Usage

```
createShadowTestConfig(item_selection = NULL, content_balancing = NULL,
  MIP = NULL, MCMC = NULL, refresh_policy = NULL,
  exposure_control = NULL, stopping_criterion = NULL,
  interim_theta = NULL, final_theta = NULL, theta_grid = seq(-4, 4,
  0.1), audit_trail = F)
```

Arguments

`item_selection` A list containing item selection criteria.

- `method` The type of criteria. Accepts one of `MFI`,`MPWI`,`FB`,`EB`.
- `info_type` The type of information. Accepts `FISHER`.
- `initial_theta` Initial theta value(s) for the first item selection.
- `fixed_theta` Fixed theta value(s) to optimize for all items to select.

`content_balancing`
A list containing content balancing options.

- `method` The type of balancing method. Accepts one of `NONE`,`STA`.

`MIP` A list containing solver options.

- `solver` The type of solver. Accepts one of `SYMPHONY`,`GUROBI`,`GLPK`,`LPSOLVE`.
- `verbosity` Verbosity level.
- `time_limit` Time limit to be passed onto solver. Used in solvers `SYMPHONY`,`GUROBI`,`GLPK`.
- `gap_limit` Gap limit (relative) to be passed onto solver. Used in solver `GUROBI`. Uses the solver default when `NULL`.
- `gap_limit_abs` Gap limit (absolute) to be passed onto solver. Used in solver `SYMPHONY`. Uses the solver default when `NULL`.

`MCMC` A list containing Markov-chain Monte Carlo configurations.

- `burn_in` Numeric. The number of chains from the start to discard.
- `post_burn_in` Numeric. The number of chains to use after discarding the first `burn_in` chains.

- thin Numeric. Thinning interval.
- jumpfactor Numeric. Jump factor.

refresh_policy A list containing refresh policy for obtaining a new shadow test.

- method The type of policy. Accepts one of ALWAYS, POSITION, INTERVAL, THRESHOLD, INTERVAL-TH
- interval Integer. Set to 1 to refresh at each position, 2 to refresh at every two positions, and so on.
- threshold Numeric. The shadow test is refreshed when the absolute change in theta estimate is greater than this value.
- position Numeric. Position(s) at which refresh to occur.

exposure_control

A list containing exposure control settings.

- method Accepts one of "NONE", "ELIGIBILITY", "BIGM", "BIGM-BAYESIAN".
- M Big M constant.
- max_exposure_rate Maximum target exposure rate.
- acceleration_factor Acceleration factor.
- n_segment Number of theta segments.
- first_segment Theta segment assumed at the begining of test.
- segment_cut A numeric vector of segment cuts.
- initial_eligibility_stats A list of eligibility statistics from a previous run.
- fading_factor Fading factor.
- diagnostic_stats TRUE to generate diagnostic statistics.

stopping_criterion

A list containing stopping criterion.

- method Accepts one of "FIXED".
- test_length Test length.
- min_ni Maximum number of items to administer.
- max_ni Minumum number of items to administer.
- se_threshold Standard error threshold for stopping.

interim_theta A list containing interim theta estimation options.

- method The type of estimation. Accepts one of EAP, EB, FB.
- shrinkage_correction Set TRUE to correct for shrinkage in EAP
- prior_dist The type of prior distribution. Accepts one of NORMAL, UNIF.
- prior_par Distributional parameters for the prior.
- bound_ML Theta bound for MLE.
- truncate_ML Set TRUE to truncate MLE within bound_ML
- max_iter Maximum number of Newton-Raphson iterations.
- crit Convergence criterion.
- max_change Maximum change in ML estimates between iterations.
- do_fisher Set TRUE to use Fisher's method of scoring.

final_theta A list containing final theta estimation options.

- method The type of estimation. Accepts one of EAP, EB, FB.

- shrinkage_correction Set TRUE to correct for shrinkage in EAP.
- prior_dist The type of prior distribution. Accepts one of NORMAL,UNIF.
- prior_par Distributional parameters for the prior.
- bound_ML Theta bound for MLE.
- truncate_ML Set TRUE to truncate MLE within bound_ML
- max_iter Maximum number of Newton-Raphson iterations.
- crit Convergence criterion.
- max_change Maximum change in ML estimates between iterations.
- do_fisher Set TRUE to use Fisher's method of scoring.

theta_grid A numeric vector. Theta values to represent the continuum.

audit_trail Set TRUE to generate audit trails.

Examples

```
cfg1 <- createShadowTestConfig(refresh_policy = list(
  method = "STIMULUS"
))
cfg2 <- createShadowTestConfig(refresh_policy = list(
  method = "POSITION",
  position = c(1, 5, 9)
))
```

---

constraint-class *An S4 class to represent a set of constraints*

Description

An S4 class to represent a set of constraints.

Slots

constraint Character. The index of the constraint set.

mat A matrix representing the left-hand side weights. Has nc rows.

dir A vector of length nc. Each entry represents a logical operator relating the left-hand side to the right-hand side.

rhs A vector of length nc. Each entry represents the right-hand side of the constraint.

nc Numeric. The number of constraints represented in the constraint set.

suspend TRUE if the constraint is to be turned off.

dataset_fatigue *Fatigue dataset*

Description

Item-based example pool with item contents (95 items).

Details

This pool is associated with the following objects:

- itempool_fatigue An item_pool object.
- itemattrib_fatigue A data frame containing item attributes.
- constraints_fatigue A list containing 111 constraints.

Also, the following datasets are intended for illustrating expected data structures. See examples below.

- itempool_fatigue_raw Item parameters.
- itemattrib_fatigue_raw Item attributes.
- itemcontent_fatigue_raw Item contents.
- constraints_fatigue_raw Constraints.
- resp_fatigue_raw Raw response data.

Examples

```
## Write to tempdir() and clean afterwards
f <- file.path(tempdir(), "itempool_fatigue.csv")
write.csv(itempool_fatigue_raw, f, row.names = FALSE)
itempool_fatigue <- loadItemPool(f)
file.remove(f)

f <- file.path(tempdir(), "itemattrib_fatigue.csv")
write.csv(itemattrib_fatigue_raw, f, row.names = FALSE)
itemattrib_fatigue <- loadItemAttrib(f, itempool_fatigue)
file.remove(f)

f <- file.path(tempdir(), "constraints_fatigue.csv")
write.csv(constraints_fatigue_raw, f, row.names = FALSE)
constraints_fatigue <- loadConstraints(f,
  itempool_fatigue, itemattrib_fatigue)
file.remove(f)

## Item contents for use in shiny app
f <- file.path(tempdir(), "itemcontent_fatigue.csv")
write.csv(itemcontent_fatigue_raw, f, row.names = FALSE)
file.remove(f)
```

```
## Raw item responses for reference
f <- file.path(tempdir(), "resp_fatigue.csv")
write.table(resp_fatigue_raw, f, row.names = FALSE, col.names = FALSE, sep = ",")
file.remove(f)
```

`dataset_reading` *Reading dataset*

Description

Stimulus-based example item pool (303 items).

Details

This pool is associated with the following objects:

- `itempool_reading` An `item_pool` object.
- `itemattrib_reading` A data frame containing item attributes.
- `stimattrib_reading` A data frame containing stimulus attributes.
- `constraints_reading` A list containing 18 constraints.

Also, the following datasets are intended for illustrating expected data structures. See examples below.

- `itempool_reading_raw` Item parameters.
- `itemattrib_reading_raw` Item attributes.
- `stimattrib_reading_raw` Item attributes.
- `constraints_reading_raw` Constraints.

Examples

```
## Write to tempdir() and clean afterwards
f <- file.path(tempdir(), "itempool_reading.csv")
write.csv(itempool_reading_raw, f, row.names = FALSE)
itempool_reading <- loadItemPool(f)
file.remove(f)

f <- file.path(tempdir(), "itemattrib_reading.csv")
write.csv(itemattrib_reading_raw, f, row.names = FALSE)
itemattrib_reading <- loadItemAttrib(f, itempool_reading)
file.remove(f)

f <- file.path(tempdir(), "stimattrib_reading.csv")
write.csv(stimattrib_reading_raw, f, row.names = FALSE)
stimattrib_reading <- loadStAttrib(f, itemattrib_reading)
file.remove(f)
```

```
f <- file.path(tempdir(), "constraints_reading.csv")
write.csv(constraints_reading_raw, f, row.names = FALSE)
constraints_reading <- loadConstraints(f,
  itempool_reading, itemattrib_reading, stimattrib_reading)
file.remove(f)
```

`dataset_science` *Science dataset*

Description

Item-based example item pool (1000 items).

Details

This pool is associated with the following objects:

- `itempool_science` An `item_pool` object.
- `itemattrib_science` A data frame containing item attributes.
- `constraints_science` A list containing 36 constraints.

Also, the following datasets are intended for illustrating expected data structures. See examples below.

- `itempool_science_raw` Item parameters.
- `itemattrib_science_raw` Item attributes.
- `constraints_science_raw` Constraints.

Examples

```
## Write to tempdir() and clean afterwards
f <- file.path(tempdir(), "itempool_science.csv")
write.csv(itempool_science_raw, f, row.names = FALSE)
itempool_science <- loadItemPool(f)
file.remove(f)

f <- file.path(tempdir(), "itemattrib_science.csv")
write.csv(itemattrib_science_raw, f, row.names = FALSE)
itemattrib_science <- loadItemAttrib(f, itempool_science)
file.remove(f)

f <- file.path(tempdir(), "constraints_science.csv")
write.csv(constraints_science_raw, f, row.names = FALSE)
constraints_science <- loadConstraints(f,
  itempool_science, itemattrib_science)
file.remove(f)
```

EAP *Generate expected a posteriori estimates of theta*

Description

Generate expected a posteriori estimates of theta.

Usage

```
EAP(object, prior, select = NULL, reset_prior = FALSE)

## S4 method for signature 'test'
EAP(object, prior, select = NULL, reset_prior = FALSE)

## S4 method for signature 'test_cluster'
EAP(object, prior, select = NULL,
  reset_prior = FALSE)
```

Arguments

object	A test or a test_cluster object.
prior	A prior distribution, a numeric vector for a common prior or a matrix for individualized priors.
select	A vector of indices identifying the items to subset.
reset_prior	Set TRUE to reset the prior distribution for each test when object is of class test_cluster.

eap *Generate expected a posteriori estimates of theta*

Description

Generate expected a posteriori estimates of theta.

Usage

```
eap(object, theta, prior, resp, select = NULL)

## S4 method for signature 'item_pool'
eap(object, theta, prior, resp, select = NULL)
```

Arguments

object	An item_pool object.
theta	A theta grid.
prior	A prior distribution, a numeric vector for a common prior or a matrix for individualized priors.
resp	A numeric matrix of item responses, one row per examinee.
select	A vector of indices identifying the items to subset.

---

extract-methods *Extract*

Description

Extract

Usage

```
## S4 method for signature 'test,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'item_pool,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]
```

Arguments

x	x
i	i
j	j
...	...
drop	drop

---

find_segment *Find the segment to which each theta value belongs*

Description

Find the segment to which each theta value belongs.

Usage

```
find_segment(segment, x)
```

Arguments

segment	A numeric vector of segment cuts.
x	A numeric vector of theta values.

info_1pl *Calculate Fisher information at a single theta (1PL)*

Description

Calculate the Fisher information at a theta value according to the 1PL model.

Usage

```
info_1pl(x, b)
```

Arguments

x	Numeric. A single theta value.
b	Numeric. A difficulty parameter value.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

info_2pl *Calculate Fisher information at a single theta (2PL)*

Description

Calculate the Fisher information at a theta value according to the 2PL model.

Usage

```
info_2pl(x, a, b)
```

Arguments

x	Numeric. A single theta value.
a	Numeric. A slope parameter value.
b	Numeric. A difficulty parameter value.

References

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

`info_3pl` *Calculate Fisher information at a single theta (3PL)*

Description

Calculate the Fisher information at a theta value according to the 3PL model.

Usage

```
info_3pl(x, a, b, c)
```

Arguments

`x`	Numeric. A single theta value.
`a`	Numeric. A slope parameter value.
`b`	Numeric. A difficulty parameter value.
`c`	Numeric. A guessing parameter value.

References

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

info_gpc *Calculate Fisher information at a single theta (GPC).*

Description

Calculate the Fisher information at a theta value according to the generalizied partial credit model.

Usage

```
info_gpc(x, a, b)
```

Arguments

x	Numeric. A single theta value.
a	Numeric. A slope parameter value.
b	Numeric. A vector of threshold parameter values.

References

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

info_gr *Calculate Fisher information at a single theta (GR).*

Description

Calculate the Fisher information at a theta value according to the graded resposne model.

Usage

```
info_gr(x, a, b)
```

Arguments

x	Numeric. A single theta value.
a	Numeric. A slope parameter value.
b	Numeric. A vector of category boundary parameter values.

References

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

info_pc *Calculate Fisher information at a single theta (PC)*

Description

Calculate the Fisher information at a theta value according to the partial credit model.

Usage

```
info_pc(x, b)
```

Arguments

x	Numeric. A single theta value.
b	Numeric. A vector of threshold parameter values.

References

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

iparPosteriorSample *Sample item parameter estimates from their posterior distributions*

Description

Sample item parameter estimates from their posterior distributions.

Usage

```
iparPosteriorSample(pool, n_sample = 500)
```

Arguments

pool	An item_pool object.
n_sample	An integer as the number of sampled parameters.

Examples

```
ipar <- iparPosteriorSample(itempool_science, 5)
```

item_1PL-class *An S4 class to represent a 1PL item*

Description

An S4 class to represent a 1PL item.

Slots

`difficulty` Numeric. A difficulty parameter value.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Examples

```
item_1 <- new("item_1PL", difficulty = 0.5)
```

item_2PL-class *An S4 class to represent a 2PL item*

Description

An S4 class to represent a 2PL item.

Slots

`slope` Numeric. A slope parameter value.

`difficulty` Numeric. A difficulty parameter value.

References

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Examples

```
item_2 <- new("item_2PL", slope = 1.0, difficulty = 0.5)
```

item_3PL-class *An S4 class to represent a 3PL item*

Description

An S4 class to represent a 3PL item.

Slots

`slope` Numeric. A slope parameter value.

`difficulty` Numeric. A difficulty parameter value.

`guessing` Numeric. A guessing parameter value.

References

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

Examples

```
item_3 <- new("item_3PL", slope = 1.0, difficulty = 0.5, guessing = 0.2)
```

item_GPC-class *An S4 class to represent a generalized partial credit item*

Description

An S4 class to represent a generalized partial credit item.

Slots

`slope` Numeric. A slope parameter value.

`threshold` Numeric. A vector of threshold parameter values.

`ncat` Numeric. The number of response categories.

References

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Examples

```
item_5 <- new("item_GPC", slope = 1.0, threshold = c(-0.5, 0.0, 0.5), ncat = 4)
```

item_GR-class *An S4 class to represent a graded response item*

Description

An S4 class to represent a graded response item.

Slots

`slope` Numeric. A slope parameter value.

`category` Numeric. A vector of category boundary values.

`ncat` Numeric. The number of response categories.

References

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

Examples

```
item_6 <- new("item_GR", slope = 1.0, category = c(-2.0, -1.0, 0, 1.0, 2.0), ncat = 6)
```

item_PC-class *An S4 class to represent a partial credit item*

Description

An S4 class to represent a partial credit item.

Slots

`threshold` Numeric. A vector of threshold parameter values.

`ncat` Numeric. The number of response categories.

References

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

Examples

```
item_4 <- new("item_PC", threshold = c(-0.5, 0.5), ncat = 3)
```

item_pool-class *An S4 class to represent an item pool*

Description

An S4 class to represent an item pool.

Slots

`ni` Numeric. The number of items in the item pool.

`max_cat` Numeric. The maximum number of response categories across all items.

`index` Numeric. A vector of item indices.

`id` Character. A vector of item ids.

`model` Numeric. A vector of item model codes (1: item.1pl, 2: item_2PL, 3: item_3PL, 4: item_PC, 5: item_GPC, 6: item_GR).

`NCAT` Numeric. A vector of the number of response categories for each item.

`parms` A list of item parameters in the pool.

`ipar` A matrix of item parameters in the pool.

`se` A matrix representing standard errors of the item parameters.

item_pool.operators *Item pool and pool cluster operators*

Description

`pool1 + pool2` combines two `item_pool` objects.

`pool1 -pool2` excludes the items in the second item pool from the first. The two `item_pool` objects must overlap for this to be performed.

`pool1 == pool2` tests equality of the two item_pool objects.

`pool_cluster1 == pool_cluster2` tests equality of the two pool_cluster objects.

Usage

```
## S3 method for class 'item_pool'
pool1 + pool2

## S3 method for class 'item_pool'
pool1 - pool2

## S3 method for class 'item_pool'
pool1 == pool2

## S3 method for class 'pool_cluster'
pool_cluster1 == pool_cluster2
```

Arguments

pool1	An item_pool object.
pool2	An item_pool object.
pool_cluster1	A pool_cluster object.
pool_cluster2	A pool_cluster object.

Examples

```
itempool <- itempool_science + itempool_reading

subitempool <- subsetItemPool(itempool_science, 1:500)
itempool <- itempool_science - subitempool

itempool <- subsetItemPool(itempool_science, 1:500)
subitempool1 <- itempool_science - itempool
subitempool2 <- subsetItemPool(itempool_science, 501:1000)
subitempool1 == subitempool2  ## TRUE

cluster1 <- makeItemPoolCluster(c(itempool_science, itempool_reading))
cluster2 <- makeItemPoolCluster(c(cluster1@pools[[1]], cluster1@pools[[2]]))
cluster1 == cluster2  ## TRUE
```

---

lnHyperPars *Calculate hyperparameters for log-normal distribution*

Description

Calculate hyperparameters for log-normal distribution.

Usage

```
lnHyperPars(mean, sd)
```

Arguments

mean	Mean of the distribution.
sd	Standard deviation of the distribution.

Examples

```
lnHyperPars(.5, 1)
```

`loadConstraints` *Load constraints*

Description

Read constraints from specified file.

Usage

```
loadConstraints(file, pool, item_attrib, st_attrib = NULL)
```

Arguments

`file`	Character. The name of the file containing specifications for constraints.
`pool`	An `item_pool` object.
`item_attrib`	A `data.frame` containing item attributes. Use `loadItemAttrib` for this.
`st_attrib`	(Optional) A `data.frame` containing stimulus attributes. Use `loadStAttrib` for this.

Details

Use `vignette("constraints")` for instructions on how to create a constraint set object.

Value

A list containing the parsed constraints, to be used in `ATA` and `Shadow`.

See Also

dataset_science for example usage.

Examples

```
## Write to tempdir() and clean afterwards
f <- file.path(tempdir(), "itempool_science.csv")
write.csv(itempool_science_raw, f, row.names = FALSE)
itempool_science <- loadItemPool(f)
file.remove(f)

f <- file.path(tempdir(), "itemattrib_science.csv")
write.csv(itemattrib_science_raw, f, row.names = FALSE)
itemattrib_science <- loadItemAttrib(f, itempool_science)
file.remove(f)

f <- file.path(tempdir(), "constraints_science.csv")
write.csv(constraints_science_raw, f, row.names = FALSE)
constraints_science <- loadConstraints(f,
  itempool_science, itemattrib_science)
```

```
file.remove(f)
```

`loadItemAttrib` *Load item attributes*

Description

Read item attributes from specified file.

Usage

```
loadItemAttrib(file, pool)
```

Arguments

`file`	Character. The name of the file containing item attributes.
`pool`	An `item_pool` object. Use `loadItemPool` for this.

Value

A `data.frame` containing parsed dataset.

See Also

dataset_science for example usage.

Examples

```
## Write to tempdir() and clean afterwards
f <- file.path(tempdir(), "itempool_science.csv")
write.csv(itempool_science_raw, f, row.names = FALSE)
itempool_science <- loadItemPool(f)
file.remove(f)

f <- file.path(tempdir(), "itemattrib_science.csv")
write.csv(itemattrib_science_raw, f, row.names = FALSE)
itemattrib_science <- loadItemAttrib(f, itempool_science)
file.remove(f)
```

loadItemPool *Load item paramaters*

Description

Read item parameters from a .csv file or a data.frame and create an item_pool class.

Usage

```
loadItemPool(file, ipar = NULL, se_file = NULL)
```

Arguments

file	File path of a .csv file containing item parameters. The file content should not have column names.
ipar	A data.frame created from a .csv file.
se_file	File path of a .csv file containing standard errors.

Value

An item_pool object.

See Also

dataset_science for example usage.

Examples

```
## Write to tempdir() and clean afterwards
f <- file.path(tempdir(), "itempool_science.csv")
write.csv(itempool_science_raw, f, row.names = FALSE)
itempool_science <- loadItemPool(f)
file.remove(f)
```

loadStAttrib *Load set/stimulus/passage attributes*

Description

Read set, stimulus, or passage attributes from specified file.

Usage

```
loadStAttrib(file, item_attrib)
```

Arguments

file	Character. The name of the file containing item attributes.
item_attrib	A data.frame containing item attributes. Use loadItemAttrib for this.

Value

A data.frame containing stimulus attributes.

See Also

dataset_reading for example usage.

Examples

```
## Write to tempdir() and clean afterwards
f <- file.path(tempdir(), "itempool_reading.csv")
write.csv(itempool_reading_raw, f, row.names = FALSE)
itempool_reading <- loadItemPool(f)
file.remove(f)

f <- file.path(tempdir(), "itemattrib_reading.csv")
write.csv(itemattrib_reading_raw, f, row.names = FALSE)
itemattrib_reading <- loadItemAttrib(f, itempool_reading)
file.remove(f)

f <- file.path(tempdir(), "stimattrib_reading.csv")
write.csv(stimattrib_reading_raw, f, row.names = FALSE)
stimattrib_reading <- loadStAttrib(f, itemattrib_reading)
file.remove(f)

f <- file.path(tempdir(), "constraints_reading.csv")
write.csv(constraints_reading_raw, f, row.names = FALSE)
constraints_reading <- loadConstraints(f,
  itempool_reading, itemattrib_reading, stimattrib_reading)
file.remove(f)
```

---

logitHyperPars *Calculate hyperparameters for logit-normal distribution*

Description

Calculate hyperparameters for logit-normal distribution.

Usage

```
logitHyperPars(mean, sd)
```

Arguments

mean	Mean of the distribution.
sd	Standard deviation of the distribution.

Examples

```
logitHyperPars(.5, 1)
```

---

makeItemPoolCluster *Create an item pool cluster object*

Description

Create a pool_cluster object.

Usage

```
makeItemPoolCluster(pools, names = NULL)
```

Arguments

pools	A list of item_pool objects.
names	An optional vector of item_pool names.

Examples

```
cluster <- makeItemPoolCluster(c(itempool_science, itempool_reading))
```

---

makeTest *Generate a test object*

Description

Generate a test object

Usage

```
makeTest(object, theta = seq(-4, 4, 0.1), info_type = "FISHER",
  true_theta = NULL)

## S4 method for signature 'item_pool'
makeTest(object, theta = seq(-4, 4, 0.1),
  info_type = "FISHER", true_theta = NULL)
```

Arguments

object	An item_pool object.
theta	A grid of theta values.
info_type	An information type.
true_theta	An optional vector of true theta values to simulate response data.

Examples

```
test <- makeTest(itempool_science, seq(-3, 3, 1))
```

---

makeTestCluster *Generate a test cluster object*

Description

Generate a test_cluster object

Usage

```
makeTestCluster(object, theta, true_theta)

## S4 method for signature 'pool_cluster,numeric,numeric'
makeTestCluster(object, theta,
  true_theta)

## S4 method for signature 'pool_cluster,numeric,list'
makeTestCluster(object, theta,
  true_theta)
```

Arguments

object	An pool_cluster object
theta	A grid of theta values
true_theta	An optional vector of true theta values to simulate response data

MLE *Generate maximum likelihood estimates of theta*

Description

Generate maximum likelihood estimates of theta.

Usage

```
MLE(object, start_theta = NULL, max_iter = 100, crit = 0.001,
  select = NULL, theta_range = c(-4, 4), truncate = FALSE,
  max_change = 1, do_Fisher = TRUE)

## S4 method for signature 'test'
MLE(object, start_theta = NULL, max_iter = 100,
  crit = 0.001, select = NULL, theta_range = c(-4, 4),
  truncate = FALSE, max_change = 1, do_Fisher = TRUE)

## S4 method for signature 'test_cluster'
MLE(object, start_theta = NULL,
  max_iter = 100, crit = 0.001, select = NULL)
```

Arguments

object	A test object.
start_theta	An optional vector of start theta values.
max_iter	Maximum number of iterations.
crit	Convergence criterion.
select	A vector of indices identifying the items to subset.
theta_range	A range of theta values: c(minTheta, maxTheta).
truncate	Set TRUE to bound MLE to theta_range.
max_change	Maximum change between iterations.
do_Fisher	Set TRUE to use Fisher's method of scoring.

mle *Generate maximum likelihood estimates of theta*

Description

Generate maximum likelihood estimates of theta.

Usage

```
mle(object, resp, start_theta = NULL, max_iter = 100, crit = 0.001,
  select = NULL, theta_range = c(-4, 4), truncate = FALSE,
  max_change = 1, do_Fisher = TRUE)

## S4 method for signature 'item_pool'
mle(object, resp, start_theta = NULL,
  max_iter = 50, crit = 0.005, select = NULL, theta_range = c(-4,
  4), truncate = FALSE, max_change = 1, do_Fisher = TRUE)
```

Arguments

object	A item_pool object.
resp	A vector (or matrix) of item responses.
start_theta	An optional vector of start theta values.
max_iter	Maximum number of iterations.
crit	Convergence criterion.
select	A vector of indices identifying the items to subset.
theta_range	A range of theta values.
truncate	Set TRUE to bound MLE to theta_range: c(minTheta, maxTheta).
max_change	Maximum change between iterations.
do_Fisher	TRUE to use Fisher's method of scoring.

Examples

```
mle(itempool_fatigue, resp_fatigue_raw[10,])
```

OAT *Launch Shiny app*

Description

Launch Shiny app locally.

Usage

```
OAT()

app()
```

Examples

```
if (interactive()) {
  OAT()
  ## or
  app()
}
```

`output_Shadow-class` *output_Shadow*

Description

output_Shadow

Slots

`simulee_id` Numeric. The index of the simulee.

`true_theta` Numeric or NULL. True theta value of the simulee if supplied in advance.

`true_theta_segment` Numeric or NULL. Which segment the true theta value is in.

`final_theta_est` Numeric. The estimated theta after the last administered item.

`final_se_est` Numeric. The standard error of estimation after the last administered item.

`administered_item_index` Numeric. A vector of item indices administered at each position.

`administered_item_resp` Numeric. A vector of responses at each position.

`administered_stimulus_index` Numeric. A vector of stimulus indices administered at each position.

`shadow_test_refreshed` Logical. A vector of logical values indicating whether the shadow test was refreshed before administering an item at each position.

`shadow_test_feasible` Logical. A vector of logical values indicating whether a feasible solution to the shadow test was available in each position.

`solve_time` Numeric. A vector of values indicating the time taken in obtaining a shadow test.

`interim_theta_est` Numeric. A vector containing estimated thetas at each position.

`interim_se_est` Numeric. A vector containing standard errors at each position.

`theta_segment_index` Numeric. A vector containing which segments the estimated thetas were in at each position.

`prior` Numeric. A prior distribution.

`prior_par` Numeric. The hyper parameters for the prior distribution.

`posterior` Numeric. A posterior distribution.

`posterior_sample` Numeric. A vector containing MCMC samples.

`likelihood` Numeric. A likelihood distribution.

`shadow_test` A list of vectors containing item indices of the shadow test at each position.

plotCAT *Draw an audit trail plot*

Description

Draw an audit trail plot.

Usage

```
plotCAT(object, examinee_id = 1, min_theta = -5, max_theta = 5,
  min_score = 0, max_score = 1, z_ci = 1.96, file_pdf = NULL, ...)

## S4 method for signature 'list'
plotCAT(object, examinee_id = 1, min_theta = -5,
  max_theta = 5, min_score = 0, max_score = 1, z_ci = 1.96,
  file_pdf = NULL, ...)

## S4 method for signature 'output_Shadow'
plotCAT(object, examinee_id = 1,
  min_theta = -5, max_theta = 5, min_score = 0, max_score = 1,
  z_ci = 1.96, file_pdf = NULL, ...)
```

Arguments

object	An output object generated by Shadow.
examinee_id	Numeric ID of the examinee to draw the plot.
min_theta	A lower bound of theta.
max_theta	An upper bound of theta.
min_score	A minimum item score.
max_score	A maximum item score.
z_ci	A quantile of the normal distribution for confidence intervals.
file_pdf	If supplied a filename, save as a PDF file.
...	Additional options to be passed on to pdf().

Examples

```
object <- itempool_science
config <- createShadowTestConfig()
true_theta <- rnorm(1)
solution <- Shadow(itempool_science, config, true_theta, constraints_science)
plotCAT(solution, 1)
```

plotEligibilityStats *Draw item eligibility statistics plots*

Description

Draw item eligibility statistics plots.

Usage

```
plotEligibilityStats(config, object = NULL, object_no_fading = NULL,
  file = NULL, file_no_fading = NULL, segment = 1, items = c(1),
  file_pdf = NULL, max_rate = 0.25, discard_first = NULL)
```

Arguments

config	A config_Shadow object.
object	An object containing eligibility statistics generated by Shadow.
object_no_fading	An object containing eligibility statistics generated without fading.
file	The filename of an object containing eligibility statistics generated by Shadow.
file_no_fading	The filename of an object containing eligibility statistics generated without fading.
segment	A theta segment index.
items	A vector of item indices to generate the plots.
file_pdf	If supplied a filename, save as a PDF file.
max_rate	A target item exposure rate.
discard_first	A integer identifying the first x simulees to discard as burn-in.

plotExposure *Draw an item exposure plot*

Description

Draw a plot of item exposure rates

Usage

```
plotExposure(object, max_rate = 0.25, theta_segment = "Estimated",
  file_pdf = NULL, ...)

## S4 method for signature 'list'
plotExposure(object, max_rate = 0.25,
  theta_segment = "Estimated", file_pdf = NULL, ...)
```

Arguments

object	An output object generated by Shadow.
max_rate	A target exposure rate.
theta_segment	True or Estimated theta used to create segments ("Estimated" or "True").
file_pdf	If supplied a filename, save as a PDF file.
...	Additional options to be passed on to pdf().

Examples

```
true_theta <- runif(10, min = -3.5, max = 3.5)
resp_science <- makeTest(itempool_science, info_type = "FISHER", true_theta = true_theta)@data
constraints_science2 <- updateConstraints(constraints_science, off = c(14:20, 32:36))
config_science <- createShadowTestConfig(
  MIP = list(solver = "LPSOLVE"),
  exposure_control = list(method = "ELIGIBILITY")
)
solution <- Shadow(itempool_science, config_science,
  true_theta, constraints_science2, data = resp_science)
p <- plotExposure(solution)
```

---

plotExposureRateBySegment

Draw exposure rate plots by theta segment

Description

Draw exposure rate plots by theta segment.

Usage

```
plotExposureRateBySegment(object, config, max_rate = 0.25,
  file_pdf = NULL, width = 7, height = 6, mfrow = c(2, 4))
```

Arguments

object	An output object generated by Shadow.
config	A config_Shadow object.
max_rate	A target item exposure rate.
file_pdf	If supplied a filename, save as a PDF file.
width	Width of the graphics device.
height	Height of the graphics device.
mfrow	Number of multiple figures defined as c(nrow, ncol).

`plotExposureRateFinal` *Draw exposure rate plots by final theta segment*

Description

Draw exposure rate plots by final theta segment.

Usage

```
plotExposureRateFinal(object, config = NULL, max_rate = 0.25,
  theta = "Estimated", segment_cut = NULL, color = "red",
  file_pdf = NULL, width = 7, height = 6, mfrow = c(2, 4),
  burn = 0, retain = NULL)
```

Arguments

`object`	An output object generated by Shadow.
`config`	A config_Shadow object.
`max_rate`	A target item exposure rate.
`theta`	By which theta to generate the plots, either "Estimated" or "True".
`segment_cut`	A vector of cut values defining theta segments.
`color`	A vector of colors.
`file_pdf`	If supplied a filename, save as a PDF file.
`width`	Width of the graphics object.
`height`	Height of the graphics object.
`mfrow`	Number of multiple figures defined as c(nrow, ncol).
`burn`	An integer identifying the first x simulees to discard as burn-in.
`retain`	An optional vector of indices identifying the simulees to retain.

Examples

```
true_theta <- runif(10, min = -3.5, max = 3.5)
resp_science <- makeTest(itempool_science, info_type = "FISHER", true_theta = true_theta)@data
constraints_science2 <- updateConstraints(constraints_science, off = c(14:20, 32:36))
config_science <- createShadowTestConfig(
  MIP = list(solver = "LPSOLVE"),
  exposure_control = list(method = "ELIGIBILITY")
)
solution <- Shadow(itempool_science, config_science,
  true_theta, constraints_science2, data = resp_science)
p <- plotExposureRateFinal(solution, config_science, 0.25)
```

plotExposureRateFinalFlag *Draw item information plots for flagged items by segment*

Description

Draw item information plots for flagged items by segment.

Usage

```
plotExposureRateFinalFlag(object, pool, theta = seq(-3, 3, 0.1),
  flag_from = 0.4, file_pdf = NULL, width = 7, height = 6,
  color = "red", mfrow = c(2, 4))
```

Arguments

object	A list object generated by plotExposureRateFinal.
pool	An item_pool object.
theta	A theta grid.
flag_from	A flagging criterion.
file_pdf	If supplied a filename, save as a PDF file.
width	Width of the graphics device.
height	Height of the graphics device.
color	Plotting color.
mfrow	Number of multiple figures defined as c(nrow, ncol).

plotInfo *Draw item information plots*

Description

Draw item information plots.

Usage

```
plotInfo(object, theta = seq(-3, 3, 0.1), info_type = "FISHER",
  select = NULL, file_pdf = NULL, color = "blue", width = 7,
  height = 6, mfrow = c(2, 4))
```

Arguments

object	An item_pool object.
theta	A theta grid. Default is seq(-3,3,.1).
info_type	Type of information.
select	A vector of indices identifying the items to subset.
file_pdf	If supplied a filename, save as a PDF file.
color	Plotting color.
width	Width of graphics device.
height	Width of graphics device.
mfrow	Number of multiple figures defined as c(nrow, ncol).

Examples

```
subitempool <- subsetItemPool(itempool_science, 1:8)
plotInfo(subitempool)
```

---

plotInfoOverlay *Overlay item information plots*

Description

Overlay item information plots.

Usage

```
plotInfoOverlay(object, theta, info_type = "FISHER", select = NULL,
  file_pdf = NULL, color = "red", width = 7, height = 6)
```

Arguments

object	An item_pool object.
theta	A theta grid.
info_type	Type of information.
select	A vector of indices identifying the items to subset.
file_pdf	If supplied a filename, save as a PDF file.
color	Plotting color.
width	Width of the graphics device.
height	Height of the graphics device.

plotMaxInfo *Draw a plot of maximum attainable information given the imposed constraints*

Description

Draw a plot of maximum attainable information given the imposed constraints.

Usage

```
plotMaxInfo(pool, constraints, theta = seq(-3, 3, 0.5))
```

Arguments

pool	An item_pool object.
constraints	A list constraints generated by loadConstraints.
theta	A theta grid.

Examples

```
p <- plotMaxInfo(itempool_science, constraints_science)
```

plotRMSE *Draw RMSE plots*

Description

Draw RMSE plots.

Usage

```
plotRMSE(..., title = NULL, legend_title = NULL,
  legend_labels = NULL, lty_set = NULL, col_set = NULL,
  theta = seq(-2, 2, 1))
```

Arguments

...	A series of RMSE values.
title	A plot title.
legend_title	A legend title.
legend_labels	A vector of labels for the series.
lty_set	A vector of line types for the series.
col_set	A vector of colors for the series.
theta	A theta grid.

plotShadow *Draw a shadow test chart*

Description

Draw a chart of shadow tests constructed for each simulee. The index of a column represents the position of item administration process, and each column represents the item pool.

Usage

```
plotShadow(object, constraints, examinee_id = 1,
  sort_by_difficulty = FALSE, file_pdf = NULL, ...)

## S4 method for signature 'list'
plotShadow(object, constraints, examinee_id = 1,
  sort_by_difficulty = FALSE, file_pdf = NULL, ...)

## S4 method for signature 'output_Shadow'
plotShadow(object, constraints,
  examinee_id = 1, sort_by_difficulty = FALSE, file_pdf = NULL, ...)
```

Arguments

object	An output from Shadow function.
constraints	The constraint object used in obtaining the output.
examinee_id	Numeric ID of the examinee to draw the plot.
sort_by_difficulty	Sort the items by difficulty.
file_pdf	If supplied a filename, save as a PDF file.
...	Additional options to be passed on to pdf().

Examples

```
object <- itempool_science
config <- createShadowTestConfig()
true_theta <- rnorm(1)
solution <- Shadow(itempool_science, config, true_theta, constraints_science)
plotShadow(solution, constraints_science, 1)
```

pool_cluster-class *An S4 class to represent a cluster of item pools*

Description

An S4 class to represent a cluster of item pools.

Slots

`np` A scalar to indicate the number of item pools in the cluster.

`pools` A list of `item_pool` objects.

`names` A character vector of item pool names of length np.

p_1pl *Calculate probability at a single theta (1PL)*

Description

Calculate the probability of correct response at a theta value, under the 1PL model.

Usage

```
p_1pl(x, b)
```

Arguments

`x`	Numeric. A single theta value.
`b`	Numeric. A difficulty parameter value.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

p_2pl *Calculate probability at a single theta (2PL)*

Description

Calculate the probability of correct response at a theta value, under the 2PL model.

Usage

```
p_2pl(x, a, b)
```

Arguments

x	Numeric. A single theta value.
a	Numeric. A slope parameter value.
b	Numeric. A difficulty parameter value.

References

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

p_3pl *Calculate probability at a single theta (3PL)*

Description

Calculate the probability of correct response at a theta value, under the 3PL model.

Usage

```
p_3pl(x, a, b, c)
```

Arguments

x	Numeric. A single theta value.
a	Numeric. A slope parameter value.
b	Numeric. A difficulty parameter value.
c	Numeric. A guessing parameter value.

References

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

`p_gpc` *Calculate probability at a single theta (GPC)*

Description

Calculate the probability of correct response at a theta value, under the generalized partial credit model.

Usage

```
p_gpc(x, a, b)
```

Arguments

`x`	Numeric. A single theta value.
`a`	Numeric. A slope parameter value.
`b`	Numeric. A vector of threshold parameter values.

References

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

`p_gr` *Calculate probability at a single theta (GR)*

Description

Calculate the probability of correct response at a theta value, under the graded response model.

Usage

```
p_gr(x, a, b)
```

Arguments

`x`	Numeric. A single theta value.
`a`	Numeric. A slope parameter value.
`b`	Numeric. A vector of category boundary parameter values.

References

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

---

p_pc *Calculate probability at a single theta (PC)*

Description

Calculate the probability of correct response at a theta value, under the partial credit model.

Usage

```
p_pc(x, b)
```

Arguments

x	Numeric. A single theta value.
b	Numeric. A vector of threshold parameter values.

References

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

RE *Calculate Relative Errors*

Description

Calculate Relative Errors.

Usage

```
RE(RMSE_foc, RMSE_ref)
```

Arguments

RMSE_foc	A vector of RMSE values for the focal group.
RMSE_ref	A vector of RMSE values for the reference group.

RMSE *Calculate Root Mean Squared Error*

Description

Calculate Root Mean Squared Error.

Usage

```
RMSE(x, y, conditional = TRUE)
```

Arguments

x	A vector of values.
y	A vector of values.
conditional	If TRUE, calculate RMSE conditional on x.

saveOutput *Save or print audit trails*

Description

Save or print audit trails for all simulees.

Usage

```
saveOutput(object_list, file = NULL)
```

Arguments

object_list	A list of output objects generated from STA.
file	An optional file name as a character string to save the output.

Value

None

Shadow *Run computerized adaptive testing with generalized shadow-test approach*

Description

Run computerized adaptive testing with generalized shadow-test approach.

Usage

```
Shadow(object, config, true_theta = NULL, constraints = NULL,
  prior = NULL, prior_par = NULL, data = NULL, session = NULL)

## S4 method for signature 'item_pool'
Shadow(object, config, true_theta = NULL,
  constraints = NULL, prior = NULL, prior_par = NULL, data = NULL,
  session = NULL)
```

Arguments

object	An item_pool object. Use loadItemPool for this.
config	A config_Shadow object.
true_theta	Numeric. A vector of true theta values to be used in simulation.
constraints	A list representing optimization constraints. Use loadConstraints for this.
prior	Numeric. A matrix or a vector containing priors.
prior_par	Numeric. A vector of parameters for prior distribution.
data	Numeric. A matrix containing item response data.
session	Used to communicate with a Shiny session.

References

van der Linden WJ, Reese LM (1998). "A model for optimal constrained adaptive testing." *Applied Psychological Measurement*, **22**, 259–270.

van der Linden WJ (1998). "Optimal assembly of psychological and educational tests." *Applied Psychological Measurement*, **22**, 195–211.

van der Linden WJ (2000). "Optimal assembly of tests with item sets." *Applied Psychological Measurement*, **24**, 225–240.

van der Linden WJ (2005). *Linear Models for Optimal Test Design*. Springer Science & Business Media.

Examples

```
object <- itempool_science
config <- createShadowTestConfig()
true_theta <- rnorm(1)
solution <- Shadow(itempool_science, config, true_theta, constraints_science)
solution$output
```

---

simResp *Simulate item responses*

Description

An S4 generic and its methods to simulate responses.

Usage

```
simResp(object, theta)

## S4 method for signature 'item_1PL,numeric'
simResp(object, theta)

## S4 method for signature 'item_2PL,numeric'
simResp(object, theta)

## S4 method for signature 'item_3PL,numeric'
simResp(object, theta)

## S4 method for signature 'item_PC,numeric'
simResp(object, theta)

## S4 method for signature 'item_GPC,numeric'
simResp(object, theta)

## S4 method for signature 'item_GR,numeric'
simResp(object, theta)

## S4 method for signature 'item_pool,numeric'
simResp(object, theta)

## S4 method for signature 'pool_cluster,numeric'
simResp(object, theta)

## S4 method for signature 'pool_cluster,list'
simResp(object, theta)

## S4 method for signature 'pool_cluster,list'
simResp(object, theta)
```

Arguments

`object`	An instance of an item class.
`theta`	A vector of theta values.

Value

Simulated responses.

References

Rasch G (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen: Danish Institute for Educational Research.

Lord FM (1952). "A theory of test scores." *Psychometric Monograph*, **7**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems (Series Report No. 58-16. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "On the estimation of mental ability (Series Report No. 15. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1958). "Further considerations of efficiency in tests of a mental ability (Series Report No. 17. Project No. 7755-23)." Randolph Air Force Base, TX: USAF School of Aviation Medicine.

Birnbaum A (1968). "Some latent trait models and their use in inferring an examinee's ability." In Lord FM, Novick MR (eds.), *Statistical Theories of Mental Test Scores*, 397–479. Addison-Wesley, Reading, MA.

Masters GN (1982). "A Rasch model for partial credit scoring." *Psychometrika*, **47**(2), 149–174.

Andrich D (1978). "A rating formulation for ordered response categories." *Psychometrika*, **43**(4), 561–573.

Muraki E (1992). "A generalized partial credit model: Application of an EM algorithm." *Applied Psychological Measurement*, **16**(2), 159–176.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika Monograph*, **17**.

Examples

```
item_1     <- new("item_1PL", difficulty = 0.5)
sim_item_1 <- simResp(item_1, seq(-3, 3, 1))
item_2     <- new("item_2PL", slope = 1.0, difficulty = 0.5)
sim_item_2 <- simResp(item_2, seq(-3, 3, 1))
item_3     <- new("item_3PL", slope = 1.0, difficulty = 0.5, guessing = 0.2)
sim_item_3 <- simResp(item_3, seq(-3, 3, 1))
item_4     <- new("item_PC", threshold = c(-1, 0, 1), ncat = 4)
sim_item_4 <- simResp(item_4, seq(-3, 3, 1))
item_5     <- new("item_GPC", slope = 1.2, threshold = c(-0.8, -1.0, 0.5), ncat = 4)
sim_item_5 <- simResp(item_5, seq(-3, 3, 1))
item_6     <- new("item_GR", slope = 0.9, category = c(-1, 0, 1), ncat = 4)
sim_item_6 <- simResp(item_6, seq(-3, 3, 1))
sim_itempool <- simResp(itempool_science, seq(-3, 3, 1))
```

STA *Perform shadow test assembly*

Description

Perform Shadow Test Assembly (STA) for computerized adaptive testing.

Usage

```
STA(constraints, objective, solver = "Lpsolve", xmat = NULL,
  xdir = NULL, xrhs = NULL, maximize = TRUE, mps = FALSE,
  lp = FALSE, verbosity = -2, time_limit = 5, gap_limit = NULL,
  gap_limit_abs = NULL, ...)
```

Arguments

constraints	A list representing optimization constraints. Use loadConstraints for this.
objective	A vector of objective values.
solver	The type of solver. Accepts SYMPHONY,GUROBI,GLPK,LPSOLVE.
xmat	A matrix of additional constraint coefficients for any previously administered items.
xdir	A character vector with the directions for the constraints in xmat.
xrhs	A vector of right-side values for the constraints in xmat.
maximize	If TRUE, treat as a maximization problem. Otherwise treat as a minimization problem.
mps	Only used when solver is SYMPHONY. If TRUE, print an MPS representation of the problem for debugging purposes.
lp	Only used when solver is SYMPHONY. If TRUE, print an LP representation of the problem for debugging purposes.
verbosity	Verbosity level.
time_limit	Time limit passed onto the solver.
gap_limit	Gap limit (relative) passed onto the solver. Used in solver GUROBI.
gap_limit_abs	Gap limit (absolute) passed onto the solver. Used in solver SYMPHONY.
...	Only used when solver is SYMPHONY. Additional parameters to be passed onto the solver.

Value

A list containing the optimal solution and pertinent diagnostics.

References

van der Linden WJ, Reese LM (1998). "A model for optimal constrained adaptive testing." *Applied Psychological Measurement*, **22**, 259–270.

van der Linden WJ (1998). "Optimal assembly of psychological and educational tests." *Applied Psychological Measurement*, **22**, 195–211.

van der Linden WJ (2000). "Optimal assembly of tests with item sets." *Applied Psychological Measurement*, **24**, 225–240.

van der Linden WJ (2005). *Linear Models for Optimal Test Design*. Springer Science & Business Media.

`subsetItemPool` *Create a subset of an item pool object*

Description

Create a subset of an `item_pool` object.

Usage

```
subsetItemPool(pool, select = NULL)
```

Arguments

`pool`	An `item_pool` object.
`select`	A vector of indices identifying the items to subset.

Examples

```
subitempool <- subsetItemPool(itempool_science, 1:100)
```

`subsetTest` *Create a subset of a test object*

Description

Create a subset of a test object.

Usage

```
subsetTest(test, select = NULL)
```

Arguments

`test`	An `test` object.
`select`	A vector of item indices to subset.

Examples

```
test <- makeTest(itempool_science, seq(-3, 3, 1))
subtest <- subsetTest(test, 1:100)
```

`test-class` *An S4 class to represent a test*

Description

An S4 class to represent a test.

Slots

`pool` An `item_pool` object.

`theta` A theta grid.

`prob` A list of item response probabilities.

`info` A matrix of item information values.

`true_theta` An optional vector of true theta values.

`data` An optional matrix of item responses.

`test_cluster-class` *An S4 class to represent a test cluster*

Description

An S4 class to represent a test cluster from a list of `test` objects.

Slots

`nt` Numeric. A scalar to indicate the number of `test` objects to be clustered.

`tests` A list `test` objects.

`names` Character. A vector of names corresponding to the `test` objects.

theta_EAP — *Calculate an EAP estimate of theta for one examinee*

Description

Calculate an expected a posterior estimate of theta for one examinee.

Usage

```
theta_EAP(theta_grid, item_parm, resp, ncat, model, prior, prior_parm)
```

Arguments

theta_grid	An equi-spaced theta grid.
item_parm	A numeric matrix of item parameters.
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

theta_EAP_matrix — *Calculate EAP estimates of theta for a group of examinees*

Description

Calculate expected a posteriori estimates of theta for a group of examinees.

Usage

```
theta_EAP_matrix(theta_grid, item_parm, Resp, ncat, model, prior,
  prior_parm)
```

Arguments

theta_grid	An equi-spaced theta grid.
item_parm	A numeric matrix of item parameters.
Resp	A numeric matrix of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector of the IRT model by item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).

prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

`theta_EB` *Calculate an empirical Bayes estimate of theta for one examinee*

Description

Calculate an empirical Bayes estimate of theta for one examinee.

Usage

```
theta_EB(nx, theta_init, theta_prop, item_parm, resp, ncat, model, prior,
  prior_parm)
```

Arguments

nx	The number of MCMC draws.
theta_init	A value for initial estimate of theta.
theta_prop	SD of the proposal distribution.
item_parm	A numeric matrix of item parameters.
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

`theta_EB_single` *Calculate an empirical Bayes estimate of theta for a single item*

Description

Calculate an empirical Bayes estimate of theta for a single item.

Usage

```
theta_EB_single(nx, theta_init, theta_prop, item_parm, resp, ncat, model,
  prior, prior_parm)
```

Arguments

nx	The number of MCMC draws.
theta_init	A value for initial estimate of theta.
theta_prop	SD of the proposal distribution.
item_parm	A numeric matrix of item parameters.
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

---

theta_FB *Calculate a fully Bayesian estimate of theta for an examinee*

Description

Calculate a fully Bayesian estimate of theta for an examinee.

Usage

```
theta_FB(nx, theta_init, theta_prop, items_list, item_init, resp, ncat,
  model, prior, prior_parm)
```

Arguments

nx	The number of MCMC draws.
theta_init	A value for initial estimate of theta.
theta_prop	SD of the proposal distribution.
items_list	A list of item_parm matrices.
item_init	A matrix of item parameter estimates (one row per item).
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

theta_FB_single *Calculate a fully Bayesian estimate of theta for a single item*

Description

Calculate a fully Bayesian estimate of theta for a single item.

Usage

```
theta_FB_single(nx, theta_init, theta_prop, item_mcmc, item_init, resp,
  ncat, model, prior, prior_parm)
```

Arguments

nx	The number of MCMC draws.
theta_init	A value for initial estimate of theta.
theta_prop	SD of the proposal distribution.
item_mcmc	A matrix of sampled item parameters for a single item.
item_init	A matrix of item parameter estimates (one row per item).
resp	A numeric vector of item responses.
ncat	A numeric vector of the number of response categories by item.
model	A numeric vector indicating the IRT models of each item (1: 1PL, 2: 2PL, 3: 3PL, 4: PC, 5: GPC, 6: GR).
prior	The type of prior distribution (1: normal, 2: uniform).
prior_parm	A numeric vector of hyperparameters for the prior distribution, c(mu, sigma) or c(ll, ul).

updateConstraints *Update constraints*

Description

Update the onstraints list

Usage

```
updateConstraints(object, on = NULL, off = NULL)
```

Arguments

object	a list object returned from loadConstraints
on	a vector of constraints index to turn on.
off	a vector of constraints index to turn off.

Value

An updated list of constraints to be used in ATA and Shadow.

Examples

```
constraints_science2 <- updateConstraints(constraints_science, off = 32:36)
constraints_science3 <- updateConstraints(constraints_science, on = 32:36)
```

Index