Package 'Shadow'

June 23, 2019

Type Package

Title Optimal Test Design Approach to Fixed and Adaptive Test Construction

Version 0.2

Date 2019-04-29

Author Seung W. Choi and Sangdon Lim

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

Description Use the optimal test design approach by Birnbaum and van der Linden to constructing fixed and adaptive tests.

License GPL (>= 2)

Depends R (>= 2.10)

Imports Rcpp (>= 1.0.0), methods, Rsymphony, Matrix, Rglpk, lpSolve, Rdpack, foreach, logitnorm, lattice

Suggests gurobi, shiny, shinythemes, shinyWidgets, shinyjs, DT

LinkingTo Rcpp

RoxygenNote 6.1.1

Encoding UTF-8

LazyData true

RdMacros Rdpack

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).

app *Launch Shiny app*

Description

Launch Shiny app locally.

Usage

```
app()
```

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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

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 supplement*.

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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

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 supplement*.

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, plotrange = c(-3, 3))

## S4 method for signature 'ATA.config'
ATA(config, Constraints, plot = FALSE,
  plotrange = c(-3, 3))
```

Arguments

config	An ATA.config object containing configuration options. Use config.ATA for this.
Constraints	A list representing optimization constraints. Use LoadConstraints for this.
plot	Logical. Draws Fisher information plot from the selected items.
plotrange	Numeric. A vector of length 2 containing the lower and upper bounds of plot range. Default is c(-3, 3).

Value

A list containing the following entries:

- `MIP` MIP model.
- `Selected` Solution returned.
- `solver` MIP solver.
- `obj.value` Objective value returned.
- `solve.time` Solver elapsed time.

`ATA.config-class` *config.ATA*

Description

Create an `ATA.config` object for Automated Test Assembly (ATA).

Usage

```
config.ATA(itemSelection = NULL, MIP = NULL)
```

Arguments

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

- `method` The type of criteria. Accepts `MAXINFO, TIF, TCC`.
- `infoType` The type of information. Accepts `FISHER`.
- `targetLocation` A numeric vector containing the locations of target theta points. (e.g. `c(-1, 0, 1)`)
- `targetValue` A numeric vector containing the target values at each theta location. This should have the same length with `targetLocation`. Ignored if method is `MAXINFO`.
- `targetWeight` 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.
- `timeLimit` Time limit in seconds passed onto the solver. Defaults to 60. Used in solvers `SYMPHONY, GUROBI, GLPK`.
- `gapLimit` Termination criteria in relative scale passed onto the solver. Defaults to .05. Used in solvers `SYMPHONY, GUROBI`.
- `gapLimitAbs` Termination criteria in absolute scale passed onto the solver. Defaults to 1. Used in solver `GUROBI`.

Examples

```
conf.1 = config.ATA(list(method = "MAXINFO",
                         infoType = "FISHER",
                         targetLocation = c(-1, 0, 1),
                         targetWeight   = c( 1, 1, 1)))

conf.2 = config.ATA(list(method = "TIF",
                         infoType = "FISHER",
                         targetLocation = c(-1, 0, 1),
                         targetWeight   = c( 1, 1, 1),
                         targetValue    = c( 8,10,12)))

conf.3 = config.ATA(list(method = "TCC",
                         infoType = "FISHER",
                         targetLocation = c(-1, 0, 1),
                         targetWeight   = c( 1, 1, 1),
                         targetValue    = c(10,15,20)))
```

BuildConstraints *Build constraints*

Description

Read constraints from specified files.

Usage

```
BuildConstraints(pool, file.Constraints, file.ItemAttrib,
  file.StAttrib = NULL)
```

Arguments

`pool`	An `item.pool` object. Use `LoadItemPool` for this.
`file.Constraints`	Character. The name of the file containing constraint specifications.
`file.ItemAttrib`	Character. The name of the file containing item attributes.
`file.StAttrib`	(Optional) Character. The name of the file containing set attributes.

Value

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

`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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika monograph supplement*.

Examples

```
## Not run:
item.1 = new("item.1pl", difficulty = 0.5)
d.item.1 = calcDerivative(item.1, seq(-3, 3, 1))

## End(Not run)
## Not run:
item.2 = new("item.2pl", slope = 1.0, difficulty = 0.5)
d.item.2 = calcDerivative(item.2, seq(-3, 3, 1))

## End(Not run)
## Not run:
item.3 = new("item.3pl", slope = 1.0, difficulty = 0.5, guessing = 0.2)
d.item.3 = calcDerivative(item.3, seq(-3, 3, 1))

## End(Not run)
## Not run:
item.4 = new("item.pc", threshold = c(-1, 0, 1), ncat = 4)
d.item.4 = calcDerivative(item.4, seq(-3, 3, 1))

## End(Not run)
## Not run:
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))

## End(Not run)
## Not run:
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))

## End(Not run)
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
d.itemPool.1 = calcDerivative(itemPool.1, seq(-3, 3, 1))

## End(Not run)
```

`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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika monograph supplement*.

Examples

```
## Not run:
item.1 = new("item.1pl", difficulty = 0.5)
dd.item.1 = calcDerivative2(item.1, seq(-3, 3, 1))

## End(Not run)
## Not run:
item.2 = new("item.2pl", slope = 1.0, difficulty = 0.5)
dd.item.2 = calcDerivative2(item.2, seq(-3, 3, 1))

## End(Not run)
## Not run:
item.3 = new("item.3pl", slope = 1.0, difficulty = 0.5, guessing = 0.2)
dd.item.3 = calcDerivative2(item.3, seq(-3, 3, 1))

## End(Not run)
## Not run:
item.4 = new("item.pc", threshold = c(-1, 0, 1), ncat = 4)
dd.item.4 = calcDerivative2(item.4, seq(-3, 3, 1))

## End(Not run)
## Not run:
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))

## End(Not run)
## Not run:
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))

## End(Not run)
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
dd.itemPool.1 = calcDerivative2(itemPool.1, seq(-3, 3, 1))

## End(Not run)
```

`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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika monograph supplement*.

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))
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
TCC.itemPool.1 = calcEscore(itemPool.1, seq(-3, 3, 1))

## End(Not run)
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
#this needs to be updated for pool.cluster
TCC.itemPool.1 = calcEscore(itemPools, seq(-3, 3, 1))

## End(Not run)
```

`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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika monograph supplement*.

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))
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
info.itemPool.1 = calcFisher(itemPool.1, seq(-3, 3, 1))

## End(Not run)
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
#this needs to be updated for item.pool
info.itemPool.1 = calcFisher(itemPools, seq(-3, 3, 1))

## End(Not run)
```

---

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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika monograph supplement*.

Examples

```
## Not run:
item.1 = new("item.1pl", difficulty = 0.5)
h.item.1 = calcHessian(item.1, seq(-3, 3, 1), 0)

## End(Not run)
## Not run:
item.2 = new("item.2pl", slope = 1.0, difficulty = 0.5)
h.item.2 = calcHessian(item.2, seq(-3, 3, 1), 0)

## End(Not run)
## Not run:
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)

## End(Not run)
## Not run:
item.4 = new("item.pc", threshold = c(-1, 0, 1), ncat = 4)
h.item.4 = calcHessian(item.4, seq(-3, 3, 1), 0)

## End(Not run)
## Not run:
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)

## End(Not run)
## Not run:
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)

## End(Not run)
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
h.itemPool.1 = calcHessian(itemPool.1, seq(-3, 3, 1), 0)

## End(Not run)
```

`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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika monograph supplement*.

Examples

```
## Not run:
item.1 = new("item.1pl", difficulty = 0.5)
j.item.1 = calcJacobian(item.1, seq(-3, 3, 1), 0)

## End(Not run)
## Not run:
item.2 = new("item.2pl", slope = 1.0, difficulty = 0.5)
j.item.2 = calcJacobian(item.2, seq(-3, 3, 1), 0)

## End(Not run)
## Not run:
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)

## End(Not run)
## Not run:
item.4 = new("item.pc", threshold = c(-1, 0, 1), ncat = 4)
j.item.4 = calcJacobian(item.4, seq(-3, 3, 1), 0)

## End(Not run)
## Not run:
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)

## End(Not run)
## Not run:
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)

## End(Not run)
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
j.itemPool.1 = calcJacobian(itemPool.1, seq(-3, 3, 1), 0)

## End(Not run)
```

`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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika monograph supplement*.

Examples

```
## Not run:
item.1 = new("item.1pl", difficulty = 0.5)
theta.item.1 = calcLocation(item.1)

## End(Not run)
## Not run:
item.2 = new("item.2pl", slope = 1.0, difficulty = 0.5)
theta.item.2 = calcLocation(item.2)

## End(Not run)
## Not run:
item.3 = new("item.3pl", slope = 1.0, difficulty = 0.5, guessing = 0.2)
theta.item.3 = calcLocation(item.3)

## End(Not run)
## Not run:
item.4 = new("item.pc", threshold = c(-1, 0, 1), ncat = 4)
theta.item.4 = calcLocation(item.4)

## End(Not run)
## Not run:
item.5 = new("item.gpc", slope = 1.2, threshold = c(-0.8, -1.0, 0.5), ncat = 4)
theta.item.5 = calcLocation(item.5)

## End(Not run)
## Not run:
item.6 = new("item.gr", slope = 0.9, category = c(-1, 0 , 1), ncat = 4)
theta.item.6 = calcLocation(item.6)

## End(Not run)
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
theta.itemPool.1 = calcLocation(itemPool.1)

## End(Not run)
```

`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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika monograph supplement*.

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))
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
prob.itemPool.1 = calcProb(itemPool.1, seq(-3, 3, 1))

## End(Not run)
## Not run:
itemPool.1 = LoadItemPool("C:/item_par_1.csv")
itemPool.2 = LoadItemPool("C:/item_par_2.csv")
itemPools = vector(mode = "list", 2)
itemPools@pools[[1]] = itemPool.1
itemPools@pools[[2]] = itemPool.2
prob.itemPools = calcProb(itemPools, seq(-3, 3, 1))

## End(Not run)
```

`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, maxIter = 100, conv = 1e-04,
  startTheta = 0)
```

Arguments

`object`	An `item.pool` object.
`rp`	A response probability value.
`maxIter`	A maximum number of iterations.
`conv`	A convergence criterion.
`startTheta`	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, usageMatrix, trueTheta = NULL)
```

Arguments

constraints	A list constraints generated by LoadConstraints.
usageMatrix	A matrix of item usage data from Shadow.
trueTheta	A vector of true theta values.

constraint-class *constraint*

Description

Represents a set of constriants.

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 this object.

EAP *Generate expected a posteriori estimates of theta*

Description

Generate expected a posteriori estimates of theta.

Usage

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

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

## S4 method for signature 'test.cluster'
EAP(object, prior, select = NULL,
  resetPrior = 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.
resetPrior	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

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

Description

Find the segment to which each theta value belongs.

Usage

```
findSegment(segment, x)
```

Arguments

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

guiShadow *guiShadow*

Description

Launches a Shiny app locally.

Usage

```
guiShadow()
```

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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

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 supplement*.

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, nSample = 500)
```

Arguments

pool	An item.pool object.
nSample	An integer as the number of sampled parameters.

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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

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 supplement*.

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.

`maxCat` 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.

Examples

```
## Not run:
itemPool.1 = LoadItemPool("C:/itemPool.csv")

## End(Not run)
```

item.pool.operators *Item pool and pool cluster operators*

Description

`pool1 + pool2` combines two `item.pool` objects.

`pool1 - pool2` combines the second from the first. Thw 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.

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.

LoadConstraints *Load constraints*

Description

Read constraints from specified file.

Usage

```
LoadConstraints(file.csv, pool, ItemAttrib, StAttrib = NULL)
```

Arguments

file.csv	Character. The name of the file containing specifications for constraints.
pool	An item.pool object.
ItemAttrib	A data.frame containing item attributes. Use LoadItemAttrib for this.
StAttrib	(Optional) A data.frame containing stimulus attributes. Use LoadStAttrib for this.

Value

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

LoadItemAttrib *Load item attributes*

Description

Read item attributes from specified file.

Usage

```
LoadItemAttrib(file.csv, pool)
```

Arguments

file.csv	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.

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.csv, ipar = NULL, se.file.csv = NULL)
```

Arguments

file.csv	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.csv	File path of a .csv file containing standard errors.

Value

An item.pool object.

Examples

```
## Not run:
itemPool.1 = LoadItemPool("C:/sample.csv")
ipar.1 = read.csv("C:/sample.csv")
itemPool.1 = LoadItemPool(ipar = ipar.1)

## End(Not run)
```

LoadStAttrib *Load set/stimulus/passage attributes*

Description

Read set attributes from specified file.

Usage

```
LoadStAttrib(file.csv, ItemAttrib)
```

Arguments

file.csv	Character. The name of the file containing item attributes.
ItemAttrib	A data.frame containing item attributes. Use LoadItemAttrib for this.

Value

A data.frame containing stimulus attributes.

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.

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.

MakeTest *Generate a test object*

Description

Generate a test object

Usage

```
MakeTest(object, theta, infoType = "FISHER", trueTheta = NULL)

## S4 method for signature 'item.pool'
MakeTest(object, theta, infoType = "FISHER",
  trueTheta = NULL)
```

Arguments

object	An item.pool object.
theta	A grid of theta values.
infoType	An information type.
trueTheta	An optional vector of true theta values to simulate response data.

MakeTestCluster *Generate a test cluster object*

Description

Generate a test.cluster object

Usage

```
MakeTestCluster(object, theta, trueTheta)

## S4 method for signature 'pool.cluster,numeric,numeric'
MakeTestCluster(object, theta,
  trueTheta)

## S4 method for signature 'pool.cluster,numeric,list'
MakeTestCluster(object, theta,
  trueTheta)
```

Arguments

object	An pool.cluster object
theta	A grid of theta values
trueTheta	An optional vector of true theta values to simulate response data

---

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

Description

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

Usage

```
maxinfoplot(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.

MLE *Generate maximum likelihood estimates of theta*

Description

Generate maximum likelihood estimates of theta.

Usage

```
MLE(object, startTheta = NULL, maxIter = 100, crit = 0.001,
  select = NULL, thetaRange = c(-4, 4), truncate = FALSE,
  maxChange = 1, FisherScoring = TRUE)

## S4 method for signature 'test'
MLE(object, startTheta = NULL, maxIter = 100,
  crit = 0.001, select = NULL, thetaRange = c(-4, 4),
  truncate = FALSE, maxChange = 1, FisherScoring = TRUE)

## S4 method for signature 'test.cluster'
MLE(object, startTheta = NULL, maxIter = 100,
  crit = 0.001, select = NULL)
```

Arguments

object	A test object.
startTheta	An optional vector of start theta values.
maxIter	Maximum number of iterations.
crit	Convergence criterion.
select	A vector of indices identifying the items to subset.
thetaRange	A range of theta values, c(minTheta, maxTheta).
truncate	TRUE to bound MLE to thetaRange.
maxChange	Maximum change between iterations.
FisherScoring	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, startTheta = NULL, maxIter = 100, crit = 0.001,
  select = NULL, thetaRange = c(-4, 4), truncate = FALSE,
  maxChange = 1, FisherScoring = TRUE)

## S4 method for signature 'item.pool'
mle(object, resp, startTheta = NULL,
  maxIter = 50, crit = 0.005, select = NULL, thetaRange = c(-4, 4),
  truncate = FALSE, maxChange = 1, FisherScoring = TRUE)
```

Arguments

object	A test object.
resp	A vector (or matrix) of item responses.
startTheta	An optional vector of start theta values.
maxIter	Maximum number of iterations.
crit	Convergence criterion.
select	A vector of indices identifying the items to subset.
thetaRange	A range of theta values.
truncate	TRUE to bound MLE to thetaRange, c(minTheta, maxTheta).
maxChange	Maximum change between iterations.
FisherScoring	TRUE to use Fisher's method of scoring.

---

par_fatigue *par_fatigue*

Description

An item-based example pool with item contents.

Details

This pool is associated with the following objects:

- par_fatigue Item parameters.
- item_attrib_fatigue Item attributes.
- item_content_fatigue Item contents.
- constraints_fatigue A constraint set (111 constraints).
- raw_fatigue Raw response data.

Examples

```
## Not run:
write.csv(par_fatigue, "par_fatigue.csv", row.names = F)
itempool.fatigue = LoadItemPool("par_fatigue.csv")

write.csv(item_attrib_fatigue, "item_attrib_fatigue.csv", row.names = F)
itemattrib.fatigue = LoadItemAttrib("item_attrib_fatigue.csv", itempool.fatigue)

write.csv(constraints_fatigue, "constraints_fatigue.csv", row.names = F)
constraints.fatigue = LoadConstraints("constraints_fatigue.csv",
          itempool.fatigue, itemattrib.fatigue)

## End(Not run)
```

---

par_reading *par_reading*

Description

Stimulus-based example item pool.

Details

This pool is associated with the following objects:

- par_reading Item parameters.
- item_attrib_reading Item attributes.
- stimulus_attrib_reading Stimulus attributes.
- constraints_reading A constraint set (18 constraints).

Examples

```
## Not run:
write.csv(par_reading, "par_reading.csv", row.names = F)
itempool.reading = LoadItemPool("par_reading.csv")

write.csv(item_attrib_reading, "item_attrib_reading.csv", row.names = F)
itemattrib.reading = LoadItemAttrib("item_attrib_reading.csv", itempool.reading)

write.csv(stimulus_attrib_reading, "stimulus_attrib_reading.csv", row.names = F)
stimattrib.reading = LoadStAttrib("stimulus_attrib_reading.csv", itemattrib.reading)

write.csv(constraints_reading, "constraints_reading.csv", row.names = F)
constraints.reading = LoadConstraints("constraints_reading.csv",
          itempool.reading, itemattrib.reading, stimattrib.reading)

## End(Not run)
```

par_science *par_science*

Description

Item-based example item pool (1000 items).

Details

This pool is associated with the following objects:

- `par_science` Item parameters.
- `item_attrib_science` Item attributes.
- `constraints_science` A constraint set (36 constraints).

Examples

```
## Not run:
write.csv(par_science, "par_science.csv", row.names = F)
itempool.science = LoadItemPool("par_science.csv")

write.csv(item_attrib_science, "item_attrib_science.csv", row.names = F)
itemattrib.science = LoadItemAttrib("item_attrib_science.csv", itempool.science)

write.csv(constraints_science, "constraints_science.csv", row.names = F)
constraints.science = LoadConstraints("constraints_science.csv",
          itempool.science, itemattrib.science)

## End(Not run)
```

plotCAT *Draw an audit trail plot*

Description

Draw an audit trail plot.

Usage

```
plotCAT(object, minTheta = -5, maxTheta = 5, minScore = 0,
  maxScore = 1, zCI = 1.96, PDF = NULL)

## S4 method for signature 'Shadow.output'
plotCAT(object, minTheta = -5, maxTheta = 5,
  minScore = 0, maxScore = 1, zCI = 1.96, PDF = NULL)

## S4 method for signature 'list'
plotCAT(object, minTheta = -5, maxTheta = 5,
  minScore = 0, maxScore = 1, zCI = 1.96, PDF = NULL)
```

Arguments

object	An output object generated by STA.
minTheta	A lower bound of theta.
maxTheta	An upper bound of theta.
minScore	A minimum item score.
maxScore	A maximum item score.
zCI	A quantile of the normal distribution for confidence intervals.
PDF	If supplied a filename, save as a PDF file.

---

plotEligibilityStats *Draw item eligibility statistics plots*

Description

Draw item eligibility statistics plots.

Usage

```
plotEligibilityStats(config, object = NULL, objectNoFading = NULL,
  file = NULL, fileNoFading = NULL, segment = 1, items = c(1),
  PDF = NULL, maxRate = 0.25, discardFirst = NULL)
```

Arguments

config	A Shadow.config object.
object	An object containing eligibility statistics generated by Shadow.
objectNoFading	An object containing eligibility statistics generated without fading.
file	The filename of an object containing eligibility statistics generated by Shadow.
fileNoFading	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.
PDF	If supplied a filename, save as a PDF file.
maxRate	A target item exposure rate.
discardFirst	A integer identifying the first x simulees to discard as burn-in.

plotExposureRateBySegment

Draw exposure rate plots by theta segment

Description

Draw exposure rate plots by theta segment.

Usage

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

Arguments

object	An output object generated by Shadow.
config	A Shadow.config object.
maxRate	A target item exposure rate.
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, maxRate = 0.25,
  theta = "Estimated", segmentCut = NULL, color = "red",
  PDF = NULL, width = 7, height = 6, mfrow = c(2, 4), burnIn = 0,
  retain = NULL)
```

Arguments

object	An output object generated by Shadow.
config	A Shadow.config object.
maxRate	A target item exposure rate.
theta	By which theta to generate the plots, either "Estimated" or "True".
segmentCut	A vector of cut values defining theta segments.
color	A vector of colors.
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).
burnIn	An integer identifying the first x simulees to discard as burn-in.
retain	An optional vector of indices identifying the simulees to retain.

---

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),
  flagCriterior = 0.4, 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.
flagCriterior	A flagging criterion.
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, infoType = "FISHER", select = NULL,
  PDF = NULL, color = "blue", width = 7, height = 6, mfrow = c(2,
  4))
```

Arguments

object	An item.pool object.
theta	A theta grid.
infoType	Type of information.
select	A vector of indices identifying the items to subset.
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).

---

plotInfoOverlay *Overlay item information plots*

Description

Overlay item information plots.

Usage

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

Arguments

object	An item.pool object.
theta	A theta grid.
infoType	Type of information.
select	A vector of indices identifying the items to subset.
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.

plotRMSE *Draw RMSE plots*

Description

Draw RMSE plots.

Usage

```
plotRMSE(..., title = NULL, legendTitle = NULL, legendLabels = NULL,
  ltySet = NULL, colSet = NULL, theta = seq(-2, 2, 1))
```

Arguments

...	A series of RMSE values.
title	A plot title.
legendTitle	A legend title.
legendLabels	A vector of labels for the series.
ltySet	A vector of line types for the series.
colSet	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, sortByDifficulty = FALSE, PDF = NULL)

## S4 method for signature 'Shadow.output'
plotShadow(object, Constraints,
  sortByDifficulty = FALSE, PDF = NULL)

## S4 method for signature 'list'
plotShadow(object, Constraints,
  sortByDifficulty = FALSE, PDF = NULL)
```

Arguments

object	An output from Shadow function.
Constraints	The constraint object used in obtaining the output.
sortByDifficulty	Sort the items by difficulty.
PDF	If supplied a filename, save as a PDF file.

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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

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 supplement*.

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(objectList, file = NULL)
```

Arguments

objectList	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, trueTheta = NULL, Constraints = NULL,
  prior = NULL, priorPar = NULL, Data = NULL, session = NULL)

## S4 method for signature 'item.pool'
Shadow(object, config, trueTheta = NULL,
  Constraints = NULL, prior = NULL, priorPar = NULL, Data = NULL,
  session = NULL)
```

Arguments

object	An item.pool object. Use LoadItemPool for this.
config	A Shadow.config object.
trueTheta	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.
priorPar	Numeric. A vector of parameters for prior distribution.
Data	Numeric. A matrix containing item response data.
session	Used to communicate with a Shiny session.

Shadow.config-class *config.Shadow*

Description

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

Usage

```
config.Shadow(itemSelection = NULL, contentBalancing = NULL,
  MIP = NULL, MCMC = NULL, refreshPolicy = NULL,
  exposureControl = NULL, stoppingCriterion = NULL,
  interimTheta = NULL, finalTheta = NULL, thetaGrid = seq(-4, 4,
  0.1), auditTrail = F)
```

Arguments

`itemSelection` A list containing item selection criteria.

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

`contentBalancing`
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.
- `timeLimit` Time limit to be passed onto solver. Used in solvers `SYMPHONY, GUROBI, GLPK`.
- `gapLimit` Gap limit to be passed onto solver. Used in solvers `SYMPHONY, GUROBI`.

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

- `burnIn` Numeric. The number of chains from the start to discard.
- `postBurnIn` Numeric. The number of chains to use after discarding the first `burnIn` chains.
- `thin` Numeric. Thinning interval.
- `jumpfactor` Numeric. Jump factor.

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

- `method` The type of policy. Accepts one of `ALWAYS, POSITION, INTERVAL, THRESHOLD, INTER`
- `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.

`exposureControl`
A list containing exposure control settings.

- `method` Accepts one of `"NONE", "ELIGIBILITY", "BIGM", "BIGM-BAYESIAN"`.
- `M` Big M constant.
- `maxExposureRate` Maximum target exposure rate.
- `accelerationFactor` Acceleration factor.
- `nSegment` Number of theta segments.
- `firstSegment` Theta segment assumed at the begining of test.
- `segmentCut` A numeric vector of segment cuts.
- `initialEligibilityStats` A list of eligibility statistics from a previous run.
- `fadingFactor` Fading factor.
- `diagnosticStats` TRUE to generate diagnostic statistics.

`stoppingCriterion`
A list containing stopping criterion.

- `method` Accepts one of `"FIXED"`.
- `testLength` Test length.
- `minNI` Maximum number of items to administer.

- maxNI Minumum number of items to administer.
- SeThreshold Standard error threshold for stopping.

interimTheta A list containing interim theta estimation options.

- method The type of estimation. Accepts one of EAP, EB, FB.
- shrinkageCorrection TRUE to correct for shrinkage in EAP
- priorDist The type of prior distribution. Accepts one of NORMAL, UNIF.
- priorPar Distributional parameters for the prior.
- boundML Theta bound for MLE.
- truncateML TRUE to truncate MLE within boundML
- maxIter Maximum number of Newton-Raphson iterations.
- crit Convergence criterion.
- maxChange Maximum change in ML estimates between iterations.
- FisherScoring TRUE to use Fisher's method of scoring.

finalTheta A list containing final theta estimation options.

- method The type of estimation. Accepts one of EAP, EB, FB.
- shrinkageCorrection TRUE to correct for shrinkage in EAP
- priorDist The type of prior distribution. Accepts one of NORMAL, UNIF.
- priorPar Distributional parameters for the prior.
- boundML Theta bound for MLE.
- truncateML TRUE to truncate MLE within boundML
- maxIter Maximum number of Newton-Raphson iterations.
- crit Convergence criterion.
- maxChange Maximum change in ML estimates between iterations.
- FisherScoring TRUE to use Fisher's method of scoring.

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

auditTrail TRUE to generate audit trails.

Shadow.output-class *Shadow.output*

Description

Shadow.output

Slots

simuleeIndex Numeric. The index of the simulee.

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

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

finalThetaEst Numeric. The estimated theta after the last administered item.

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

administeredItemIndex Numeric. A vector of item indices administered at each position.

administeredItemResp Numeric. A vector of responses at each position.

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

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

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

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

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

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

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

`prior` Numeric. A prior distribution.

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

`posterior` Numeric. A posterior distribution.

`posteriorSample` Numeric. A vector containing MCMC samples.

`likelihood` Numeric. A likelihood distribution.

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

`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 No. 7)." *Iowa City, IA: Psychometric Society*, **35**.

Birnbaum A (1957). "Efficient design and use of tests of mental ability for various decision-making problems. School of Aviation Medicine, USAF." Report.

Birnbaum A (1958). "On the estimation of mental ability." *Series Rep*, **15**, 7755–7723.

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." *Statistical theories of mental test scores*.

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." *ETS Research Report Series*, **1992**(1), i–30.

Samejima F (1969). "Estimation of latent ability using a response pattern of graded scores." *Psychometrika monograph supplement*.

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))
## Not run:
itemPool.1 = LoadItemPool("C:/item_par.csv")
sim.itemPool.1 = simResp(itemPool.1, seq(-3, 3, 1))

## End(Not run)
```

STA *STA*

Description

Shadow Test Approach to CAT.

Usage

```
STA(Constraints, objective, solver = "Symphony", xmat = NULL,
  xdir = NULL, xrhs = NULL, maximize = TRUE, mps = FALSE,
  lp = FALSE, verbosity = -2, time_limit = 5, gap_limit = -1, ...)
```

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 passed onto the solver.
`...`	Only used when `solver` is `SYMPHONY`. Additional parameters to be passed onto the solver.

Value

A list containing the optimal solution and pertinent diagnostics.

subsetItemPool *Create a subset of an item pool*

Description

Create a subset of an item pool.

Usage

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

Arguments

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

subsetPool *Create a subset of an item pool object*

Description

Create a subset of an item.pool object.

Usage

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

Arguments

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

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.

test-class *test*

Description

Create a test object.

Slots

pool An item.pool object.

theta A theta grid.

Prob A list of item response probabilities.

Info A matrix of item information values.

trueTheta An optional vector of true theta values.

Data An optional matrix of item responses.

test.cluster-class *test.cluster*

Description

Create a test.cluster object 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 correspondign 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).

Index