Package ‘catR’

October 21, 2011

Type Package

Title Procedures to generate IRT adaptive tests (CAT)

Version 2.0

Date 2011-10-21

Author David Magis (U Liege, Belgium), Gilles Raiche (UQAM, Canada)

Maintainer David Magis <david.magis@ulg.ac.be>

Depends R (>= 2.8.0), sfsmisc

Description The catR package allows the generation of response patterns under computerized adaptive testing (CAT) framework, with the choice of several starting rules, next item selection routines, stopping rules and ability estimators. Control methods for item exposure and content balancing are also included.

LazyLoad yes

R topics documented:

`createItemBank` *Item bank generation*

Description

This command creates an item bank from a matrix of item parameters. Item information functions are evaluated for all items and a fine grid of ability levels, to be supplied. Subgroups of items can also be specified for content balancing purposes.

Usage

```
createItemBank(items=100, cb=FALSE, model="4PL",
 aPrior=c("norm",1,0.2), bPrior=c("norm",0,1),
 cPrior=c("unif",0,0.25), dPrior=c("unif",0.75,1),
 thMin=-4, thMax=4, step=0.01, seed=1, D=1)
```

Arguments

`items`	either an integer value or a matrix (or data frame) of item parameters (and possibly subgroups of items). See **Details**.
`cb`	logical: should subgroups of items be returned in the item bank for content balancing pruposes? (default is `FALSE`).
`model`	character: the name of the logistic IRT model, with possible values `"1PL"`, `"2PL"`, `"3PL"` or `"4PL"` (default). Ignored if `items` is a matrix or a data frame.
`aPrior`	vector of three components, specifying the prior distribution and ites parameters for generating the item discrimination levels. Ignored if `items` is a matrix or a data frame. See **Details**.
`bPrior`	vector of three components, specifying the prior distribution and ites parameters for generating the item difficulty levels. Ignored if `items` is a matrix or a data frame. See **Details**.
`cPrior`	vector of three components, specifying the prior distribution and ites parameters for generating the item lower asymptote levels. Ignored if `items` is a matrix or a data frame. See **Details**.
`dPrior`	vector of three components, specifying the prior distribution and ites parameters for generating the item upper asymptote levels. Ignored if `items` is a matrix or a data frame. See **Details**.
`thMin`	numeric: the lower bound for the fine grid of ability levels (default is -4). See **Details**.
`thMax`	numeric: the upper bound for the fine grid of ability levels (default is 4). See **Details**.
`step`	numeric: the step value for the fine grid of ability levels (default is 0.01). See **Details**.
`seed`	numeric: the random seed number for the generation of item parameters (default is 1). See `set.seed` for further details. Ignored if `items` is a matrix or a data frame.
`D`	numeric: the metric constant. Default is `D=1` (for logistic metric); `D=1.702` yields approximately the normal metric (Haley, 1952).

Details

If items is a matrix or a data frame, it has the following format: one row per item and four (or five) columns, with respectively the discrimination a_i, the difficulty b_i, the pseudo-guessing c_i and the inattention d_i parameters (Barton and Lord, 1981). A fifth column can be added; in this case it holds the names of the subgroups of items (for content balancing purposes for instance). This column is returned in the output list only if items has five columns and cb argument is set to TRUE (default value is FALSE).

If items is an integer, it corresponds to the number of items to be included in the item bank. Corresponding item parameters are drawn from distributions to be specified by arguments aPrior, bPrior, cPrior and dPrior for respective parameters a_i, b_i, c_i and d_i. Each of these arguments is of length 3, the first component containing the name of the distribution and the last two components coding the distribution parameters. Possible distributions are:

- the *normal distribution* $N(\mu, \sigma^2)$, available for parameters a_i and b_i. It is specified by "norm" as first argument while the latter two arguments contain the values of μ and σ respectively.
- the *log-normal distribution* $\log N(\mu, \sigma^2)$, available for parameter a_i only. It is specified by "lnorm" as first argument while the latter two arguments contain the values of μ and σ respectively.
- the *uniform distribution* $U([a, b])$, available for all parameters. It is specified by "unif" as first argument while the latter two arguments contain the values of a and b respectively. Note that taking a and b equal to a common value, say t, makes all parameters to be equal to t.
- the *Beta distribution* $Beta(\alpha, \beta)$, available for parameters c_i and d_i. It is specified by "beta" as first argument while the latter two arguments contain the values of α and β respectively.

Inattention parameters d_i are fixed to 1 if model is not "4PL"; pseudo-guessing parameters c_i are fixed to zero if model is either "1PL" or "2PL"; and discrimination parameters a_i are fixed to 1 if model="1PL". The random generation of item parameters can be controlled by the seed argument.

Note that currently it is not possible to randomly draw subgroups of items for generated sets of item parameters. Content balancing can thus be performed only with provided item parameters (not generated).

The item bank consists of the(infoTab) matrix, which holds Fisher information functions (Baker, 1992), evaluated for each item in the bank and at each value of a sequence of ability levels. These abilities are ranging from thMin to thMax by steps of step units.

The returned list contains in addition the sequence of ability levels and the matrix of item parameters. If items has five columns and cb argument is set to TRUE, the returned list has the additional vector cbGroup that contains the elements of the fifth column of items (i.e. the subgroup names). Otherwise, the cbGroup element is returned as NULL.

Value

A list of class "itBank" with four arguments:

itemPar	the matrix of item parameters, either provided by items or generated.
theta	a vector with the ability levels of the fine grid, defined by arguments thMin, thMax and step.
infoTab	a matrix of Fisher information functions, evaluated for each ability level (one row per ability level) and each item (one column per item).
cbGroup	either the fifth column of items (if cb was set to TRUE in addition) or NULL.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Baker, F.B. (1992). *Item response theory: parameter estimation techniques*. New York, NY: Marcel Dekker.

Barton, M.A., and Lord, F.M. (1981). An upper asymptote for the three-parameter logistic item-response model. Research Bulletin 81-20. Princeton, NJ: Educational Testing Service.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

See Also

Ii

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Item bank creation with 'tcals' item parameters
 createItemBank(tcals)

 # Changing the fine grid of ability levels
 createItemBank(tcals, thMin=-2, thMax=2, step=0.05)

 # Item bank creation with 500 items
 createItemBank(items=500)

 # Item bank creation with 100 items, 2PL model and log-normal
 # distribution with parameters (0, 0.1225) for discriminations
 createItemBank(items=100, model="2PL", aPrior=c("lnorm",0,0.1225))

 # A completely identical method as for previous example
 createItemBank(items=100, aPrior=c("lnorm",0,0.1225),
  cPrior=c("unif",0,0), dPrior=c("unif",1,1))

 # Item bank creation with 'tcals' item parameters and keeping
 # the subgroups of items
 createItemBank(tcals, cb=TRUE)

## End(Not run)
```

eapEst *EAP ability estimation under the 4PL model*

Description

This command returns the EAP (expected a posteriori) ability estimate for a given matrix of item parameters of the 4PL model and a given response pattern.

Usage

```
eapEst(it, x, D=1, priorDist="norm", priorPar=c(0,1),
 lower=-4, upper=4, nqp=33)
```

Arguments

`it`	numeric: a matrix with one row per item and four columns, with the values of the discrimination, the difficulty, the pseudo-guessing and the inattention parameters (in this order).
`x`	numeric: a vector of dichotomous item responses.
`D`	numeric: the metric constant. Default is `D=1` (for logistic metric); `D=1.702` yields approximately the normal metric (Haley, 1952).
`priorDist`	character: specifies the prior distribution. Possible values are `"norm"` (default), `"unif"` and `"Jeffreys"`.
`priorPar`	numeric: vector of two components specifying the prior parameters (default is `c(0,1)`). Ignored if `priorDist="Jeffreys"`. See **Details**.
`lower`	numeric: the lower bound for numercal integration (default is -4).
`upper`	numeric: the upper bound for numercal integration (default is 4).
`nqp`	numeric: the number of quadrature points (default is 33).

Details

The EAP (expected a posteriori) ability estimator (Bock and Mislevy, 1982) is obtained by computing the average of the posterior distribution of ability, set as the prior distribution times the likelihood function.

Three prior distributions are available: the normal distribution, the uniform distribution and Jeffreys' prior distribution (Jeffreys, 1939, 1946). The prior distribution is specified by the argument `priorPar`, with values `"norm"`, `"unif"` and `"Jeffreys"`, respectively.

The argument `priorPar` determines either the prior mean and standard deviation of the normal prior distribution (if `priorDist="norm"`), or the range for defining the prior uniform distribution (if `priorDist="unif"`). This argument is ignored if `priorDist="Jeffreys"`.

The required integrals are approximated by numerical adaptive quadrature. This is achieved by using the `integrate.xy` function of the package `sfsmisc`. Arguments `lower`, `upper` and `nqp` define respectively the lower and upper bounds for numerical integration, and the number of quadrature points. By default, the numerical integration runs with 33 quadrature points on the range [-4; 4], that is, a sequence of values from -4 to 4 by steps of 0.25.

Value

The estimated EAP ability level.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Bock, R. D., and Mislevy, R. J. (1982). Adaptive EAP estimation of ability in a microcomputer environment. *Applied Psychological Measurement, 6*, 431-444.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

Jeffreys, H. (1939). *Theory of probability*. Oxford, UK: Oxford University Press.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 186*, 453-461.

See Also

thetaEst, integrate.xy

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Selecting item parameters only
 tcals <- as.matrix(tcals[,1:4])

 # Creation of a response pattern (tcals item parameters,
 # true ability level 0)
 set.seed(1)
 x <- rbinom(85, 1, Pi(0, tcals)$Pi)

 # EAP estimation, standard normal prior distribution
 eapEst(tcals, x)

 # EAP estimation, uniform prior distribution upon range [-2,2]
 eapEst(tcals, x, priorDist="unif", priorPar=c(-2,2))

 # EAP estimation, Jeffreys' prior distribution
 eapEst(tcals, x, priorDist="Jeffreys")

 # Changing the integration settings
 eapEst(tcals, x, nqp=100)

## End(Not run)
```

`eapSem` *Standard error of EAP ability estimation under the 4PL model*

Description

This command returns the estimated standard error of the ability estimate, for a given matrix of item parameters of the 4PL model, an ability estimate and a specified estimator.

Usage

```
eapSem(thEst, it, x, D=1, priorDist="norm", priorPar=c(0,1),
 lower=-4, upper=4, nqp=33)
```

Arguments

`thEst`	numeric: the EAP ability estimate.
`it`	numeric: a matrix with one row per item and four columns, with the values of the discrimination, the difficulty, the pseudo-guessing and the inattention parameters (in this order).
`x`	numeric: a vector of dichotomous item responses.
`D`	numeric: the metric constant. Default is `D=1` (for logistic metric); `D=1.702` yields approximately the normal metric (Haley, 1952).
`priorDist`	character: specifies the prior distribution. Possible values are `"norm"` (default), `"unif"` and `"Jeffreys"`.
`priorPar`	numeric: vector of two components specifying the prior parameters (default is `c(0,1)`). Ignored if `priorDist="Jeffreys"`. See **Details**.
`lower`	numeric: the lower bound for numercal integration (default is -4).
`upper`	numeric: the upper bound for numercal integration (default is 4).
`nqp`	numeric: the number of quadrature points (default is 33).

Details

This command computes the standard error of the EAP (expected a posteriori) ability estimator (Bock and Mislevy, 1982).

Three prior distributions are available: the normal distribution, the uniform distribution and Jeffreys' prior distribution (Jeffreys, 1939, 1946). The prior distribution is specified by the argument `priorPar`, with values `"norm"`, `"unif"` and `"Jeffreys"`, respectively.

The argument `priorPar` determines either the prior mean and standard deviation of the normal prior distribution (if `priorDist="norm"`), or the range for defining the prior uniform distribution (if `priorDist="unif"`). This argument is ignored if `priorDist="Jeffreys"`.

The required integrals are approximated by numerical adaptive quadrature. This is achieved by using the `integrate.xy` function of the package `sfsmisc`. Arguments `lower`, `upper` and `nqp` define respectively the lower and upper bounds for numerical integration, and the number of quadrature points. By default, the numerical integration runs with 33 quadrature points on the range [-4; 4], that is, a sequence of values from -4 to 4 by steps of 0.25.

Note that in the current version, the EAP ability estimate must be specified through the `thEst` argument.

Value

The estimated standard error of the EAP ability level.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Bock, R. D., and Mislevy, R. J. (1982). Adaptive EAP estimation of ability in a microcomputer environment. *Applied Psychological Measurement, 6*, 431-444.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

Jeffreys, H. (1939). *Theory of probability*. Oxford, UK: Oxford University Press.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 186*, 453-461.

See Also

thetaEst, integrate.xy

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Selecting item parameters only
 tcals <- as.matrix(tcals[,1:4])

 # Creation of a response pattern (tcals item parameters,
 # true ability level 0)
 set.seed(1)
 x <- rbinom(85, 1, Pi(0, tcals)$Pi)

 # EAP estimation, standard normal prior distribution
 th <- eapEst(tcals, x)
 c(th, eapSem(th, tcals, x))

 # EAP estimation, uniform prior distribution upon range [-2,2]
 th <- eapEst(tcals, x, priorDist="unif", priorPar=c(-2,2))
 c(th, eapSem(th, tcals, x, priorDist="unif", priorPar=c(-2,2)))

 # EAP estimation, Jeffreys' prior distribution
 th <- eapEst(tcals, x, priorDist="Jeffreys")
 c(th, eapSem(th, tcals, x, priorDist="Jeffreys"))

## End(Not run)
```

EPV *Expected Posterior Variance (EPV)*

Description

This command returns the expected posterior variance (EPV) for a given item, as used for Minimum Expected Posterior Variance (MEPV) criterion.

Usage

```
EPV(itemBank, item, x, theta, it, priorDist="norm",
 priorPar=c(0,1), D=1, parInt=c(-4,4,33))
```

Arguments

itemBank	an item bank, i.e. a list of class `itBank`, typically an output of the function `createItemBank`.
item	numeric: the item (referred to as its rank in the item bank) for which the maximum information must be computed.
x	binary: a vector of item responses, coded as 0 or 1 only.
theta	numeric: the provisional ability estimate.
it	numeric: a matrix with one row per item and four columns, with the values of the discrimination, the difficulty, the pseudo-guessing and the inattention parameters (in this order). The number of rows of `it` must be equal to the length of `x`.
priorDist	character: specifies the prior distribution. Possible values are `"norm"` (default) and `"unif"`.
priorPar	numeric: vector of two components specifying the prior parameters (default is `c(0,1)`) of the prior ability distribution.
D	numeric: the metric constant. Default is `D=1` (for logistic metric); `D=1.702` yields approximately the normal metric (Haley, 1952).
parInt	numeric: vector of three components, defining the sequence of ability values for computing the posterior variance. See **Details**.

Details

The EPV can be used as a rule for selecting the next item in the CAT process (Choi and Swartz, 2009; Owen, 1975; van der Linden, 1998). This command serves as a subroutine for the `nextItem` function.

Let *k* be the number of administered items, and set $x_1, ..., x_k$ as the provisional response pattern. Set $\hat{\theta}_k$ as the provisional ability estimate (with the first *k* responses) and let *j* be the item of interest (not previously administered). Set also $P_j(\theta)$ as the probability of answering item *j* correctly for a given ability level θ, and set $Q_j(\theta) = 1 - P_j(\theta)$. Finally, set $Var(\theta|x_1, ..., x_k, 0)$ and $Var(\theta|x_1, ..., x_k, 1)$ as the posterior variances of θ, given the provisional response pattern (updated by response 0 and 1 respectively). Then, the EPV for item *j* equals

$$EPV_j = P_j(\hat{\theta}_k)\, Var(\theta|x_1, ..., x_k, 1) + Q_j(\hat{\theta}_k)\, Var(\theta|x_1, ..., x_k, 0)$$

.

The posterior variance $Var(\theta|x_1, ..., x_k, x_j)$ (where x_j takes value 0 or 1) is computed as the squared standard error of the EAP estimate of ability, using the response pattern $(x_1, ..., x_k, x_j)$. This is done by a joint use of the eapEst and eapSem functions.

The prior distribution is set up by the arguments priorDist and priorPar, with the by-default standard normal distribution. The range of integration is defined by the parInt argument, with by default, the sequence from -4 to 4 and of length 33 (or, by steps of 0.25). See the function eapEst for further details.

The item bank is provided through the argument itemBank. The provisional response pattern and the related item parameters are provided by the arguments x and it respectively. The target item (for which the maximum information computed) is given by its number in the item bank, through the item argument.

Value

The expected posterior variance for the selected item.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Choi, S. W., and Swartz, R. J. (2009). Comparison of CAT item selection criteria for polytomous items. *Applied PScyhological Measurement, 32*, 419-440.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

Owen, R. J. (1975). A Bayesian sequential procedure for quantal response in the context of adaptive mental testing. *Journal of the American Statistical Association, 70*, 351-356.

van der Linden, W. J. (1998). Bayesian item selection criteria for adaptive testing. *Psychometrika, 63*, 201-216.

See Also

nextItem, eapEst, eapSem

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Selecting item parameters only
 tcals <- as.matrix(tcals[,1:4])

 # Item bank creation with 'tcals' item parameters
 bank <- createItemBank(tcals)
```

```
 # Selection of two arbitrary items (15 and 20) of the
 # 'tcals' data set
 it <- bank$itemPar[c(15,20),]

 # Creation of a response pattern
 x <- c(0,1)

 # MEI for item 1, provisional ability level 0
 EPV(bank, 1, x, 0, it)

 # With prior standard deviation 2
 EPV(bank, 1, x, 0, it, priorPar=c(0,2))

## End(Not run)
```

`Ii` *Item information functions, first and second derivatives (4PL)*

Description

This command returns the Fisher information functions for a given matrix of item parameters of the 4PL model and a given ability value. Numerical values of the first and second derivatives of the item information functions are also returned.

Usage

```
Ii(th, it, D=1)
```

Arguments

`th`	numeric: the ability value.
`it`	numeric: a matrix with one row per item and four columns, with the values of the discrimination, the difficulty, the pseudo-guessing and the inattention parameters (in this order).
`D`	numeric: the metric constant. Default is `D=1` (for logistic metric); `D=1.702` yields approximately the normal metric (Haley, 1952).

Details

The first and second derivatives are computed algebraically from the four-parameter logistic (4PL) model (Barton and Lord, 1981). These derivatives are necessary for both the estimation of ability and the computation of related standard errors.

Value

A list with three arguments:

`Ii`	the vector with item informations (one value per item)
`dIi`	the vector with first derivatives of the item information functions (one value per item)
`d2Ii`	the vector with second derivatives of the item information functions (one value per item)

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Barton, M.A., and Lord, F.M. (1981). An upper asymptote for the three-parameter logistic item-response model. Research Bulletin 81-20. Princeton, NJ: Educational Testing Service.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

See Also

Pi, thetaEst

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Selecting item parameters only
 tcals <- as.matrix(tcals[,1:4])

 # Item information functions and derivatives
 # (various th and D values)
 Ii(th=0, tcals)
 Ii(th=0, tcals, D=1.702)
 Ii(th=1, tcals)

## End(Not run)
```

MEI *(Maximum) Expected Information (MEI)*

Description

This command returns the expected information (EI) for a given item, as used for Maximum Expected Information (MEI) criterion.

Usage

```
MEI(itemBank, item, x, theta, it, method="BM", priorDist="norm",
  priorPar=c(0,1), D=1, range=c(-4,4), parInt=c(-4,4,33),
  infoType="observed")
```

Arguments

`itemBank`	an item bank, i.e. a list of class `itBank`, typically an output of the function `createItemBank`.
`item`	numeric: the item (referred to as its rank in the item bank) for which the maximum information must be computed.
`x`	binary: a vector of item responses, coded as 0 or 1 only.
`theta`	numeric: the provisional ability estimate.
`it`	numeric: a matrix with one row per item and four columns, with the values of the discrimination, the difficulty, the pseudo-guessing and the inattention parameters (in this order). The number of rows of `it` must be equal to the length of `x`.
`method`	character: the ability estimator. Possible values are `"BM"` (default), `"ML"` and `"WL"`. See **Details**.
`priorDist`	character: specifies the prior distribution. Possible values are `"norm"` (default), `"unif"` and `"Jeffreys"`. Ignored if `method` is neither `"BM"` nor `"EAP"`. See **Details**.
`priorPar`	numeric: vector of two components specifying the prior parameters (default is `c(0,1)`) of the prior ability distribution. Ignored if `method` is neither `"BM"` nor `"EAP"`, or if `priorDist="Jeffreys"`. See **Details**.
`D`	numeric: the metric constant. Default is `D=1` (for logistic metric); `D=1.702` yields approximately the normal metric (Haley, 1952).
`range`	numeric: vector of two components specifying the range wherein the ability estimate must be looked for (default is `c(-4,4)`). Ignored if `method=="EAP"`.
`parInt`	numeric: vector of three components, holding respectively the values of the arguments `lower`, `upper` and `nqp` of the `eapEst` command. Default vector is (-4, 4, 33). Ignored if `method` is not `"EAP"`.
`infoType`	character: the type of information function to be used. Possible values are `"observed"` (default) for observed information function, and `"Fisher"` for Fisher information function.

Details

The MEI (van der Linden, 1998; van der Linden and Pashley, 2000) can be used as a rule for selecting the next item in the CAT process (see also Choi and Swartz, 2009). This command serves as a subroutine for the `nextItem` function.

Let *k* be the number of administered items, and set $x_1, ..., x_k$ as the provisional response pattern. Set $\hat{\theta}_k$ as the provisional ability estimate (with the first *k* responses) and let *j* be the item of interest (not previously administered). Set also $P_j(\theta)$ as the probability of answering item *j* correctly for a given ability level θ, and set $Q_j(\theta) = 1 - P_j(\theta)$. Finally, set $\hat{\theta}^0_{k+1}$ and $\hat{\theta}^1_{k+1}$ as the ability estimates computed under the condition that the response to item *j* is 0 or 1 respectively (that is, if the response pattern is updated by 0 or 1 for item *j*). Then, the MEI for item *j* equals

$$MEI_j = P_j(\hat{\theta}_k)\, I_j(\hat{\theta}^1_{k+1}) + Q_j(\hat{\theta}_k)\, I_j(\hat{\theta}^0_{k+1})$$

where $I_j(\theta)$ is the information function for item *j*.

Two types of information functions are available. The first one is the observed information function, defined as

$$I_j(\theta) = -\frac{\partial^2}{\partial\,\theta^2} \log P_j(\theta).$$

(van der Linden, 1998). The second one is Fisher information function:

$$I_j(\theta) = -E\left[\frac{\partial^2}{\partial\,\theta^2}\,\log P_j(\theta)\right].$$

Under the 1PL and the 2PL models, these functions are identical (Veerkamp, 1996).

The observed and Fisher information functions are specified by the `infoType` argument, with respective values `"observed"` and `"Fisher"`. By default, the observed information function is considered (Choi and Swartz, 2009; van der Linden, 1998).

The estimator of provisional ability is defined by means of the arguments `method`, `priorDist`, `priorPar`, `D`, `range` and `parInt` of the `thetaEst` function. See the corresponding help file for further details.

The item bank is provided through the argument `itemBank`. The provisional response pattern and the related item parameters are provided by the arguments `x` and `it` respectively. The target item (for which the maximum information computed) is given by its number in the item bank, through the `item` argument.

Value

The required maximum expected information for the selected item.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
`<david.magis@ulg.ac.be>`

References

Choi, S. W., and Swartz, R. J. (2009). Comparison of CAT item selection criteria for polytomous items. *Applied PScyhological Measurement, 32*, 419-440.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

van der Linden, W. J. (1998). Bayesian item selection criteria for adaptive testing. *Psychometrika, 63*, 201-216.

van der Linden, W. J., and Pashley, P. J. (2000). Item selection and ability estimlation in adaptive testing. In W. J. van der Linden and C. A. W. Glas (Eds.), *Computerized adaptive testing. Theory and practice* (pp. 1-25). Boston, MA: Kluwer.

Veerkamp, W. J. J. (1996). *Statistical inference for adaptive testing*. Internal report. Enschede, The Netherlands: University of Twente.

See Also

`Ii`, `OIi`, `nextItem`, `integrate.xy`, `thetaEst`

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Selecting item parameters only
 tcals <- as.matrix(tcals[,1:4])

 # Item bank creation with 'tcals' item parameters
 bank <- createItemBank(tcals)

 # Selection of two arbitrary items (15 and 20) of the
 # 'tcals' data set
 it <- bank$itemPar[c(15,20),]

 # Creation of a response pattern
 x <- c(0,1)

 # MEI for item 1, provisional ability level 0
 MEI(bank, 1, x, 0, it)

 # With Fisher information instead
 MEI(bank, 1, x, 0, it, infoType="Fisher")

 # With WL estimator instead
 MEI(bank, 1, x, 0, it, method="WL")

## End(Not run)
```

MWI *Maximum likelihood weighted information (MLWI) and maximum posterior weighted information (MPWI)*

Description

This command returns the maximum likelihood (MLWI) or the maximum posterior (MPWI) weighted information for a given item and an item bank.

Usage

```
MWI(itemBank, item, x, it, lower=-4, upper=4, nqp=33,
  type="MLWI", priorDist="norm", priorPar=c(0,1))
```

Arguments

`itemBank`	an item bank, i.e. a list of class `itBank`, typically an output of the function `createItemBank`.
`item`	numeric: the item (referred to as its rank in the item bank) for which the maximum information must be computed.

`x`	binary: a vector of item responses, coded as 0 or 1 only.
`it`	numeric: a matrix with one row per item and four columns, with the values of the discrimination, the difficulty, the pseudo-guessing and the inattention parameters (in this order). The number of rows of `it` must be equal to the length of `x`.
`lower`	numeric: the lower bound for numercal integration (default is -4).
`upper`	numeric: the upper bound for numercal integration (default is 4).
`nqp`	numeric: the number of quadrature points (default is 33).
`type`	character: the type of information to be computed. Possible values are `"MLWI"` (default) and `"MPWI"`. See **Details**.
`priorDist`	character: the prior ability distribution. Possible values are `"norm"` (default) for the normal distribution, and `"unif"` for the uniform distribution. Ignored if `type` is not `"MPWI"`.
`priorPar`	numeric: a vector of two components with the prior parameters. If `priorDist` is `"norm"`, then `priorPar` contains the mean and the standard deviation of the normal distribution. If `priorDist` is `"unif"`, then `priorPar` contains the bounds of the uniform distribution. The default values are 0 and 1 respectively. Ignored if `type` is not `"MPWI"`.

Details

Both the MLWI (Veerkamp and Berger, 1997) and the MPWI (van der Linden, 1998; van der Linden and Pashley, 2000) can be used as rules for selecting the next item in the CAT process (see also Choi and Swartz, 2009). This command serves as a subroutine for the `nextItem` function.

Let *k* be the number of administered items, and set $x_1, ..., x_k$ as the binary responses to the first *k* administered items. Set also $I_j(\theta)$ as the information function of item *j* evaluated at θ, and set $L(\theta|x_1, ..., x_k)$ as the likelihood function evaluated at θ, given the provisional response pattern. Then, the MLWI for item *j* is given by

$$MLWI_j = \int I_j(\theta) L(\theta|x_1, ..., x_k) d\theta$$

and the MPWI by

$$MPWI_j = \int I_j(\theta) \pi(\theta) L(\theta|x_1, ..., x_k) d\theta$$

where $\pi(\theta)$ is the prior distribution of the ability level.

These integrals are approximated by the `integrate.xy` function from the package `sfsmisc`. The range of integration is set up by the arguments `lower`, `upper` and `nqp`, giving respectively the lower bound, the upper bound and the number of quadrature points. The default range goes from -4 to 4 with length 33 (that is, by steps of 0.25).

The argument `type` defines the type of information to be computed. The default value, `"MLWI"`, computes the MLWI value, while the MPWI value is obtained with `type="MPWI"`. For the latter, the `priorDist` and `priorPar` arguments fix the prior ability distribution. The normal distribution is set up by `priorDist="norm"` and then, `priorPar` contains the mean and the standard deviation of the normal distribution. If `priorDist` is `"unif"`, then the uniform distribution is considered, and `priorPar` fixes the lower and upper bounds of that uniform distribution. By default, the standard normal prior distribution is assumed. This argument is ignored whenever `method` is not `"MPWI"`.

The item bank is provided through the argument `itemBank`. The provisional response pattern and the related item parameters are provided by the arguments `x` and `it` respectively. The target item (for which the maximum information computed) is given by its number in the item bank, through the `item` argument.

Value

The required maximum information for the selected item.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Choi, S. W., and Swartz, R. J. (2009). Comparison of CAT item selection criteria for polytomous items. *Applied PScyhological Measurement, 32*, 419-440.

van der Linden, W. J. (1998). Bayesian item selection criteria for adaptive testing. *Psychometrika, 63*, 201-216.

van der Linden, W. J., and Pashley, P. J. (2000). Item selection and ability estimlation in adaptive testing. In W. J. van der Linden and C. A. W. Glas (Eds.), *Computerized adaptive testing. Theory and practice* (pp. 1-25). Boston, MA: Kluwer.

Veerkamp, W. J. J., and Berger, M. P. F. (1997). Some new item selection criteria for adaptive testing. *Journal of Educational and Behavioral Statistics, 22*, 203-226.

See Also

`Ii`, `nextItem`, `integrate.xy`

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Selecting item parameters only
 tcals <- as.matrix(tcals[,1:4])

 # Item bank creation with 'tcals' item parameters
 bank <- createItemBank(tcals)

 # Selection of two arbitrary items (15 and 20) of the
 # 'tcals' data set
 it <- bank$itemPar[c(15,20),]

 # Creation of a response pattern
 x <- c(0,1)

 # MLWI for item 1
 MWI(bank, 1, x, it)

 # MPWI for item 1
 MWI(bank, 1, x, it, type="MPWI")

 # MLWI for item 1, different integration range
 MWI(bank, 1, x, it, lower=-2, upper=2, nqp=20)
```

```
# MPWI for item 1, uniform prior distribution on the range [-2,2]
MWI(bank, 1, x, it, type="MPWI", priorDist="unif", priorPar=c(-2,2))

## End(Not run)
```

---

nextItem *Selection of the next item*

Description

This command selects the next item to be administered, given the list of previously administered items and the current ability estimate, with several possible criteria. Item exposure and content balancing can also be controlled.

Usage

```
nextItem(itemBank, theta, out=NULL, x=NULL, criterion="MFI",
 method="BM", priorDist="norm", priorPar=c(0,1), D=1,
 range=c(-4,4), parInt=c(-4,4,33), infoType="observed",
 randomesque=1, cbControl=NULL)
```

Arguments

itemBank	an item bank of class itBank as output of the function createItemBank.
theta	numeric: the current value of the ability estimate (default is 0).
out	either a vector of integer values specifying the items previously administered, or NULL (default).
x	numeric: the provisional response pattern, with the same length as out (and NULL by default). Ignored if method is either "MFI" or "Owen". See **Details**.
criterion	character: the method for next item selection. Possible values are "MFI" (default), "Urry" "MLWI", "MPWI", "MEI", "MEPV" and random. See **Details**.
method	character: the ability estimator. Possible values are "BM" (default), "ML" and "WL". See **Details**.
priorDist	character: the prior ability distribution. Possible values are "norm" (default) for the normal distribution, and "unif" for the uniform distribution. Ignored if type is not "MPWI".
priorPar	numeric: a vector of two components with the prior parameters. If priorDist is "norm", then priorPar contains the mean and the standard deviation of the normal distribution. If priorDist is "unif", then priorPar contains the bounds of the uniform distribution. The default values are 0 and 1 respectively. Ignored if type is not "MPWI".
D	numeric: the metric constant. Default is D=1 (for logistic metric); D=1.702 yields approximately the normal metric (Haley, 1952).

`range`	numeric: vector of two components specifying the range wherein the ability estimate must be looked for (default is `c(-4,4)`). Ignored if `method=="EAP"`.
`parInt`	numeric: a vector of three numeric values, specifying respectively the lower bound, the upper bound and the number of quadrature points for numerical integration (default is `c(-4,4,33)`). Ignored if `method` is either `"MFI"` or `"Owen"`. See **Details**.
`infoType`	character: the type of information function to be used. Possible values are `"observed"` (default) for observed information function, and `"Fisher"` for Fisher information function. Ignored if `criterion` is not `"MEI"`.
`randomesque`	integer: the number of items to be chosen from the next item selection rule, among those the next item to be administered will be randomly picked up. Default value is 1 and leads to usual selection of the optimal item for the specified `criterion`. See **Details**.
`cbControl`	either a list of accurate format to control for content balancing, or `NULL`. See **Details**.

Details

Currently seven methods are available for selecting the next item to be administered in the adaptive test. For a given current ability estimate, the next item is selected (among the available items) by using: the maximum Fisher information (MFI) criterion, the maximum likelihood weighted information (MLWI) (Veerkamp and Berger, 1997), the maximum posterior weighted information (MPWI) (van der Linden, 1998), Urry's procedure (Urry, 1970), the maximum expected information (MEI) criterion (van der Linden, 1998), the minimum expected posterior variance (MEPV) or by selecting the next item completely randomly among the available items.

The MFI criterion selects the next item as the one which maximizes the item information function (Baker, 1992). The most informative item is selected from the table of item informations provided by the bank of items specified with `itemBank`. Urry's procedure consists in selecting as next the item whose difficulty level is closest to the current ability estimate. Under the 1PL model, both Urry and MFI methods are equivalent. The MLWI and MPWI criteria select the next item as the one with maximal information, weighted either by the likelihood function or the posterior distribution. See the function `MWI` for further details. Finally, the MEI criterion selects the item with maximum expected information, computed with the `MEI` function.

The method for next item selection is specified by the `criterion` argument. Possible values are `"MFI"` for maximum Fisher information criterion, `"Urry"` for Owen's method, `"MLWI"` for maximum likelihood weighted information criterion, `"MPWI"` for the maximum posterior weighted information criterion, `"MEI"` for the maximum expected information criterion, `"MEPV"` for minimum expected posterior variance, and `"random"` for random selection. Other values return an error message.

For MFI, MEI and Urry criteria, the provisional ability estimate must be supplied throught the `theta` argument (by default, it is equal to zero). For MLWI and MPWI criteria, this argument is ignored.

The available items are those that are not specified in the `out` argument. By default, `out` is `NULL`, which means that all items are available.

For MEI, MEPV, MLWI and MPWI methods, the provisional response pattern must be provided through the `x` argument. It must be of 0/1 entries and of the same length as the `out` argument. It is ignored with MFI and Urry criteria. Moreover, the range of integration (or posterior variance computation) is specified by the triplet `parInt`, where the first, second, and third value correspond to the arguments `lower`, `upper` and `nqp` of the `MWI` function, respectively.

The method, priorDist, priorPar, D, range and intPar arguments fix the ability estimator. See the thetaEst function for further details.

Finally, for MEI criterion, the type of information function must be supplied through the infoType argument. It is equal to "observed" by default, which refers to the observed information function, and the other possible value is "Fisher" for Fisher information function. See the MEI funtion for further details. This argumpent is ignored if criterion is not "MEI".

Item exposure can be controlled by using the so-called *randomesque* approach (Kingsbury an Zara, 1989), which consists in selecting more than one item as the best items to be administered (according to the specified criterion). The final item that is administered is randomly chosen among this set of optimal items. The argument randomesque controls for the number of optimal items to be selected. The default value is 1, which corresponds to the usual framework of selecting the optimal item for next administration. Note that, for compatibility issues, if the number of remaining items is smaller than randomesque, the latter is replaced by this number of remaining items.

Control for content balancing is also possible, given two conditions: (a) the item bank set by itemBank holds as element $cbGroup the names of the subgroups of items for content balancing, and (b) the argument cbControl is a correctly specified list. The correct format for cbControl is a list with two elements. The first one is called names and holds the names of the subgroups of items (in the order that is prespecified by the user). The second element is called props and contains the (theoretical) proportions of items to be administered from eazch subgroup for content balancing. These proportions must be strictly positive but may not sum to one; in this case they are internally normalized to sum to one. Note that if cbControl is misspecified, then the test.cbList will return a warning message and the nextItem function will stop.

Under content balancing, the selection of the next item is done in several steps.

1. If no item was administered yet, one subgroup is randomly picked up and the optimal item from this subgroup is selected.
2. If at least one subgroup wasn't targeted yet by item selection, one of these subgroups is randomly picked up and the optimal item from this subgroup is selected.
3. If at least one item per subgroup was already administered, the empirical relative proportions of items administered per subgroup are computed, and the subgroup(s) whose difference between empirical and theoretical (i.e. given by cbControl$props) proportions is (are) selected. The optimal item is then selected from this subgroup for next administration (in case of several such groups, one group is randomly picked up first).

See Kingsbury and Zara (1989) for further details.

Three vectors of proportions are returned in the output list: $prior.prop contains the empirical relative proportions for items already administered (i.e. passed through the out argument); $post.prop contains the same empirical relative proportions but including the optimal item that was just selected; and $th.prop contains the theoretical proportions (i.e. those from cbControl$props or the normalized values).

Value

A list with eight arguments:

item	the selected item (identified by its number in the item bank).
par	the vector of item parameters of the selected item.
info	the value of the MFI, Fisher's information, the MLWI, the MPWI, the MEI, the EPV,or NA (for "random" criterion) for the selected item and the current ability estimate.
criterion	the value of the criterion argument.

`randomesque`	the value of the `randomesque` argument.
`prior.prop`	a vector with empirical proportions of items previously administered for each subgroup of items set by `cbControl`.
`post.prop`	a vector with empirical proportions of items previously administered, together with the one currently selected, for each subgroup of items set by `cbControl`.
`th.prop`	a vector with theoretical proportions given by `cbControl$props`.

Note

van der Linden and Pashley (2000) also introduced the Maximum Expected Posterior Weighted Information (MEPWI) criterion, as a mix of both MEI and MPWI methods. However, Choi and Swartz (2009) established that this method is completely equivalent to MPWI. For this reason, MEPWI was not implemented here.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Baker, F.B. (1992). *Item response theory: parameter estimation techniques*. New York, NY: Marcel Dekker.

Choi, S. W., and Swartz, R. J. (2009). Comparison of CAT item selection criteria for polytomous items. *Applied PScyhological Measurement, 32*, 419-440.

Kingsbury, G. G., and Zara, A. R. (1989). Procedures for selecting items for computerized adaptive tests. *Applied Measurement in Education, 2*, 359-375.

Urry, V. W. (1970). *A Monte Carlo investigation of logistic test models.* Unpublished doctoral dissertation. West Lafayette, IN: Purdue University.

van der Linden, W. J. (1998). Bayesian item selection criteria for adaptive testing. *Psychometrika, 63*, 201-216.

van der Linden, W. J., and Pashley, P. J. (2000). Item selection and ability estimlation in adaptive testing. In W. J. van der Linden and C. A. W. Glas (Eds.), *Computerized adaptive testing. Theory and practice* (pp. 1-25). Boston, MA: Kluwer.

Veerkamp, W. J. J., and Berger, M. P. F. (1997). Some new item selection criteria for adaptive testing. *Journal of Educational and Behavioral Statistics, 22*, 203-226.

See Also

`createItemBank`, `MWI`, `MEI`, `thetaEst`, `test.cbList`, `randomCAT`

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Item bank creation with 'tcals' item parameters
```

```
bank <- createItemBank(tcals)

## MFI criterion

# Selecting the next item, current ability estimate is 0
nextItem(bank, 0) # item 63 is selected

# Selecting the next item, current ability estimate is 0
# and item 63 is removed
nextItem(bank, 0, out=63) # item 10 is selected

# Selecting the next item, current ability estimate is 0
# and items 63 and 10 are removed
nextItem(bank, 0, out=c(63,10)) # item 62 is selected

# Item exposure control by selecting three items
# (selected item will be either 10, 62 or 63)
nextItem(bank, 0, randomesque = 3)

## Urry's method

# Selecting the next item, current ability estimate is 0
nextItem(bank, 0, criterion="Urry") # item 24 is selected

# Selecting the next item, current ability estimate is 0
# and item 24 is removed
nextItem(bank, 0, out=24, criterion="Urry")

## MLWI and MPWI methods

# Selecting the next item, current response pattern is 0
# and item 63 was administered first
nextItem(bank, x=0, out=63, criterion="MLWI")
nextItem(bank, x=0, out=63, criterion="MPWI")

# Selecting the next item, current response pattern is
# (0,1) and item 19 is removed
nextItem(bank, x=c(0,1), out=c(63, 19), criterion="MLWI")
nextItem(bank, x=c(0,1), out=c(63, 19), criterion="MPWI")

## MEI method

# Selecting the next item, current response pattern is 0
# and item 63 was administered first
nextItem(bank, x=0, out=63, criterion="MEI")

# With Fisher information
nextItem(bank, x=0, out=63, criterion="MEI", infoType="Fisher")

## MEPV method

# Selecting the next item, current response pattern is 0
# and item 63 was administered first
nextItem(bank, x=0, out=63, criterion="MEPV")

## Random method
```

```
# Selecting the next item, item 63 was administered first
nextItem(bank, out=63, criterion="random")
nextItem(bank, out=63, criterion="random")  # may produce a
                                            # different result

# Item bank creation for content balancing
bank2 <- createItemBank(tcals, cb=TRUE)

# Creation of the 'cbList' list with arbitrary proportions
cbList <- list(names=c("Audio1","Audio2","Written1","Written2",
               "Written3"), props=c(0.1,0.2,0.2,0.2,0.3))

# Selecting the next item, MFI criterion, current ability
# estimate is 0, items 12,33,46 and 63 previously administered
nextItem(bank2, 0, out=c(12,33,46,63), cbControl=cbList)
                                       # item 70 is selected


## End(Not run)
```

OIi *Observed information function (4PL)*

Description

This command returns the observed information functions for a given matrix of item parameters of the 4PL model and a given ability value.

Usage

```
OIi(th, it, x, D=1)
```

Arguments

`th`	numeric: the ability value.
`it`	numeric: a matrix with one row per item and four columns, with the values of the discrimination, the difficulty, the pseudo-guessing and the inattention parameters (in this order).
`x`	numeric: the item response (coded as 0 or 1). Can be either a single value or a vector of the same length of the number of items.
`D`	numeric: the metric constant. Default is `D=1` (for logistic metric); `D=1.702` yields approximately the normal metric (Haley, 1952).

Details

The observed information function for item *j* is given by

$$-\frac{\partial^2}{\partial\theta^2}\,\log L(\theta|x_j)$$

where θ is the ability level, L is the likelihood function and x_j is the item response. For dichotomous item response models with success probability $P_j(\theta)$, it takes the following form:

$$-\frac{\partial^2}{\partial\theta^2} \log L(\theta|x_j) = \frac{P_j\, Q_j\, P_j'^2 - (x_j - P_j)\,[P_j\, Q_j\, P_j'' + P_j{}^2\,(P_j - Q_j]}{P_j{}^2\, Q_j{}^2}$$

where $P_j = P_j(\theta)$, $Q_j = 1 - P_j$ and P_j' and P_j'' are the first and second derivatives of P_j respectively.

Under the 2PL model, the observed information function is exactly equal to Fisher's information function

$$-E\left[\frac{\partial^2}{\partial\theta^2} \log L(\theta|x_j)\right] = \frac{P_j'^2}{P_j Q_j}$$

(van der Linden, 1998; Veerkamp, 1996).

The observed information function is used to compute some item selection criteria, such as the Maximum Expected Information (MEI). See `nextItem` for further details.

Value

A vector with the observed item informations (one per item).

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Barton, M.A., and Lord, F.M. (1981). An upper asymptote for the three-parameter logistic item-response model. Research Bulletin 81-20. Princeton, NJ: Educational Testing Service.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

van der Linden, W. (1998). Bayesian item selection criteria for adaptive testing. *Psychometrika, 63*, 201-216.

Veerkamp, W. J. J. (1996). *Statistical inference for adaptive testing*. Internal report. Enschede, The Netherlands: University of Twente.

See Also

`createItemBank`, `nextItem`

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Selecting item parameters only
 tcals <- as.matrix(tcals[,1:4])
```

```
 # Observed information functions
 # (various th, x and D values)
 OIi(th=0, tcals, x=0)
 OIi(th=0, tcals, x=0, D=1.702)
 OIi(th=0, tcals, x=1)
 OIi(th=1, tcals, x=1)

## End(Not run)
```

`Pi` *Item response probabilities, first, second and third derivatives (4PL)*

Description

This command returns the item reponse probabilities for a given matrix of item parameters of the 4PL model and a given ability value. Numerical values of the first, second and third derivatives of the response probabilities are also returned.

Usage

```
Pi(th, it, D=1)
```

Arguments

`th`	numeric: the ability value.
`it`	numeric: a matrix with one row per item and four columns, with the values of the discrimination, the difficulty, the pseudo-guessing and the inattention parameters (in this order).
`D`	numeric: the metric constant. Default is `D=1` (for logistic metric); `D=1.702` yields approximately the normal metric (Haley, 1952).

Details

The first, second and third derivatives are computed algebraically from the four-parameter logistic (4PL) model (Barton and Lord, 1981). These derivatives are necessary for both the estimation of ability and the computation of related standard errors.

Value

A list with four arguments:

`Pi`	the vector with response probabilities (one value per item)
`dPi`	the vector with first derivatives of the response probabilities (one value per item)
`d2Pi`	the vector with second derivatives of the response probabilities (one value per item)
`d3Pi`	the vector with third derivatives of the response probabilities (one value per item)

Note

Response probabilites exactly equal to zero are returned as `1e-10` values, as well as probabilities exactly equal to one which are returned as `1-1e-10` values. This is to permit the computation of ability estimates (with the `thetaEst` function) in such extreme cases. Many thanks to Pan Tong (University of Texas MD Anderson Cancer Center, USA) who noticed this problem.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Barton, M.A., and Lord, F.M. (1981). An upper asymptote for the three-parameter logistic item-response model. Research Bulletin 81-20. Princeton, NJ: Educational Testing Service.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

See Also

`Ii`, `thetaEst`

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Selecting item parameters only
 tcals <- as.matrix(tcals[,1:4])

 # Response probabilities and derivatives (various th and D values)
 Pi(th=0, tcals)
 Pi(th=0, tcals, D=1.702)
 Pi(th=1, tcals)

## End(Not run)
```

`randomCAT` *Random generation of adaptive tests*

Description

This command generates a response pattern to an adaptive test, for agiven item bank, a true ability level, and several lists of CAT parameters (starting items, stopping rule, provisional and final ability estimators).

Usage

```
 randomCAT(trueTheta, itemBank, maxItems=50, cbControl=NULL,
  start=list(fixItems=NULL, seed=NULL, nrItems=1, theta=0,
  halfRange=2, startSelect="bOpt"), test=list(method="BM",
  priorDist="norm", priorPar=c(0,1), range=c(-4,4), D=1,
  parInt=c(-4,4,33), itemSelect="MFI", infoType="observed",
  randomesque=1), stop=list(rule="length", thr=20,
  alpha=0.05), final=list(method="BM", priorDist="norm",
  priorPar=c(0,1), range=c(-4,4),D=1, parInt=c(-4,4,33),
  alpha=0.05))
 ## S3 method for class 'cat':
print(x, ...)
 ## S3 method for class 'cat':
plot(x, ci=FALSE, alpha=0.05, trueTh=TRUE, classThr=NULL, ...)
```

Arguments

`trueTheta`	numeric: the value of the true ability level.
`itemBank`	an item bank, i.e. a list of class `itBank`, typically an output of the function `createItemBank`.
`maxItems`	numeric: the maximal number of items to be administered (default is 50).
`cbControl`	either a list of accurate format to control for content balancing, or `NULL`. See **Details**.
`start`	a list with the options for starting the adaptive test. See **Details**.
`test`	a list with the options for provisional ability estimation and next item selection. See **Details**.
`stop`	a list with the options of the stopping rule. See **Details**.
`final`	a list with the options for final ability estimation. See **Details**.
`x`	an object of class "cat", typically an output of `randomCAT` function.
`ci`	logical: should the confidence intervals be plotted for each provisional ability estimate? (default is `TRUE`).
`alpha`	numeric: the significance level for provisional confidence intervals (default is 0.05). Ignored if `ci` is `FALSE`.
`trueTh`	logical: should the true ability level be drawn by a horizontal line? (default is TRUE).
`classThr`	either a numeric value giving the classification threshold to be displayed, or `NULL`.
`...`	other generic arguments to be passed to `print` and `plot` functions.

Details

The `randomCAT` function generates an adaptive test using an item bank specified by argument `itemBank`, and for a given true ability level specified by argument `trueTheta`. The maximal length of the test can be fixed through the `maxItems` argument, with a default value of 50 items.

Content balancing can be controlled by the `cbControl` argument. See the `nextItem` function for further details on how to specify `cbControl` properly and under which conditions it is operational. By default, content balancing is not controlled (see Kingsbury and Zara, 1989, for further details on content balancing).

The test specification is made by means of four lists of options: one list for the selection of the starting items, one list with the options for provisional ability estimation, one list to define the stopping rule, and one list with the options for final ability estimation. These lists are specified respectively by the arguments `start`, `test`, `stop` and `final`.

The `start` list can contain one or several of the following arguments:

- `fixItems`: either a vector of integer values, setting the items to be administered as first items, or `NULL` (default) to let the function select the items.
- `seed`: either a numeric value to fix the random seed for item selection, or `NULL` (default) to select the items on the basis of their difficulty level. Ignored if `fixItems` is not `NULL`.
- `nrItems`: the number of first items to be selected (default is 1). Ignored if `fixItems` is not `NULL`.
- `theta`: the central initial ability value, used to define the range of ability levels for selecting the first items (default is 0). Ignored if either `fixItems` or `seed` is not `NULL`. See `startItems` for further details.
- `halfRange`: the half range of starting ability levels for selecting the first items (default is 2). Ignored if either `fixItems` or `seed` is not `NULL`. See `startItems` for further details.
- `startSelect`: the method for selecting the first items of the test, with possible values `"bOpt"` (default) and `"MFI"`. Ignored if either `fixItems` or `seed` is not `NULL`. See `startItems` for further details.

These arguments are passed to the function `startItems` to select the first items of the test.

The `test` list can contain one or several of the following arguments:

- `method`: a character string to specify the method for ability estimation. Possible values are: `"BM"` (default) for Bayesian modal estimation (Birnbaum, 1969), `"ML"` for maximum likelihood estimation (lord, 1980), `"EAP"` for expected a posteriori (EAP) estimation (Bock and Mislevy, 1982), and `"WL"` for weighted likelihood estimation (Warm, 1989).
- `priorDist`: a character string which sets the prior distribution. Possible values are: `"norm"` (default) for normal distribution, `"unif"` for uniform distribution, and `"Jeffreys"` for Jeffreys' noninformative prior distribution (Jeffreys, 1939, 1946). ignored if `method` is neither `"BM"` nor `"EAP"`.
- `priorPar`: a vector of two numeric components, which sets the parameters of the prior distribution. If (`method="BM"` or `method=="EAP"`) and `priorDist="norm"`, the components of `priorPar` are respectively the mean and the standard deviation of the prior normal density. If (`method="BM"` or `method=="EAP"`) and `priorDist="unif"`, the components of `priorPar` are respectively the lower and upper bound of the prior uniform density. Ignored in all other cases. By default, `priorPar` takes the parameters of the prior standard normal distribution (i.e., `priorPar=c(0,1)`). In addition, `priorPar` also provides the prior parameters for the comoutation of MLWI and MPWI values for next item selection (see `nextItem` for further details).
- `range`: the maximal range of ability levels, set as a vector of two numeric components. The ability estimate will always lie to this interval (set by default to [-4, 4]). Ignored if `method=="EAP"`.
- `D`: the value of the metric constant. Default is `D=1` for logistic metric. Setting `D=1.702` yields approximately the normal metric (Haley, 1952).
- `parInt`: a numeric vector of three components, holding respectively the values of the arguments `lower`, `upper` and `nqp` of the `eapEst`, `eapSem` and `MWI` commands. It specifies the range of quadrature points for numerical integration, and is used for computing the EAP estimate, its standard error, and the MLWI and MPWI values for next item selection. Default

vector is (-4, 4, 33), thus setting the range from -4 to 4 by steps of 0.25. Ignored if `method` is not `"EAP"` and if `itemSelect` is neither `"MLWI"` nor `"MPWI"`.

- `itemSelect`: the rule for next item selecion, with possible values `"MFI"` (default) for maximum Fisher information criterion; `"Urry"` for Urry's procedure; `"MLWI"` and `"MPWI"` for respectively maximum likelihood and posterior weighted information criterion; `"MEPV"` for minimum expected posterior variance; `"MEI"` for maximum expected information; and `"random"` for random selection. For further details, see `nextItem`.
- `infoType`: character: the type of information function to be used for next item selection. Possible values are `"observed"` (default) for observed information function, and `"Fisher"` for Fisher information function. Ignored if `itemselect` is not `"MEI"`.
- randomesqueinteger: the number of items to be chosen from the next item selection rule, among those the next item to be administered will be randomly picked up. Default value is 1 and leads to usual selection of the optimal item (Kingsbury and Zara, 1989).

These arguments are passed to the functions `thetaEst` (or `eapEst`) and `semTheta` (or `eapSem`) to estimate the ability level and the standard error of this estimate. In addition, some arguments are passed to `nextItem` to select the next item appropriately.

The `stop` list can contain one or several of the following arguments:

- `rule`: a character string specifying the type of stopping rule. Possible values are: `"length"` (default), to stop the test after a pre-specified number of items administered; `"precision"`, to stop the test when the provisional standard error of ability becomes less than or equal to the pre-specified value; and `"classification"`, for which the test ends whenever the provisional confidence interval (set by the `alpha` argument) does not hold the classification threshold anymore.
- `thr`: a numeric value fixing the threshold of the stopping rule. If `rule="length"`, `thr` is the maximal number of items to be administered (in practice, it is replaced by the value of the `maxItems` argument if the latter is smaller than `thr`). If `rule="precision"`, `thr` is the precision level (i.e. the standard error) to be reached before stopping. Finally, if `rule="classification"`, `thr` corresponds to the ability level which serves as a classification rule (i.e. which must not be covered by the provisional confidence interval).
- `alpha`: the significance (or α) level for computing the priovisional confidence interval of ability. Ignored if `rule` is not `"classification"`.

Eventually, the `final` list can contain one or several arguments of the `test` list (with possiblly different values), as well as the additional `alpha` argument. The latter specifies the α level of the final confidence interval of ability, which is computed as

$$[\hat{\theta} - z_{1-\alpha/2}\; se(\hat{\theta}); \hat{\theta} + z_{1-\alpha/2}\; se(\hat{\theta})]$$

where $\hat{\theta}$ and $se(\hat{\theta})$ are respectively the ability estimate and its standard error. Note that the argument `itemSelect` of the `test` list is not used for final estimation of the ability level, and is therefore not allowed into the `final` list.

If some arguments of these lists are missing, they are automatically set to their default value. The contents of the lists is checked with the `testList` function, and the adaptive test is generated only if the lists are adequately defined. Othgerwise, a message error is printed.

The function `plot.cat` represents the set of provisional and final ability estimates througghout the test. Corresponding confidence intervals (with confidence level defined by the argument `alpha`) are also drawn if `ci=TRUE` (which is not the default value). The true ability level can be drawn by a horizontal solid line by specifying `trueTh=TRUE` (which is the default value); setting it to `FALSE` will undo the drawing. Finally, any classification threshold can be additionally displayed by specifying a numeric value to the argument `classThr`. The default value `NULL` does not display any threshold.

Value

The function `randomCAT` returns a list of class "cat" with the following arguments:

- `trueTheta` the value of the `trueTheta` argument.
- `maxItems` the value of the `maxItems` argument.
- `testItems` a vector with the items that were administered during the test.
- `itemPar` a matrix with the parameters of the items administered during the test.
- `pattern` the generated response pattern (as vector of 0 and 1 entries).
- `thetaProv` a vector with the provisional ability estimates.
- `seprov` a vector with the standard errors of the provisional ability estimates.
- `thFinal` the final ability estimate.
- `seFinal` the standrad error of the final ability estimate.
- `ciFinal` the confidence interval of the final ability estimate.
- `startFixItems` the value of the `start$fixItems` argument (or its default value if missing).
- `startSeed` the value of the `start$seed` argument (or its default value if missing).
- `startNrItems` the value of the `start$nrItems` argument (or its default value if missing).
- `startTheta` the value of the `start$theta` argument (or its default value if missing).
- `startHalfRange` the value of the `start$halfRange` argument (or its default value if missing).
- `startThStart` the starting ability values used for selecting the first items of the test.
- `startSelect` the value of the `start$startSelect` argument (or its default value if missing).
- `provMethod` the value of the `test$method` argument (or its default value if missing).
- `provDist` the value of the `test$priorDist` argument (or its default value if missing).
- `provPar` the value of the `test$priorPar` argument (or its default value if missing).
- `provRange` the value of the `test$range` argument (or its default value if missing).
- `provD` the value of the `test$D` argument (or its default value if missing).
- `itemSelect` the value of the `test$itemSelect` argument (or its default value if missing).
- `infoType` the value of the `test$infoType` argument (or its default value if missing).
- `randomesque` the value of the `test$randomesque` argument (or its default value if missing).
- `cbControl` the value of the `cbControl` argument (or its default value if missing).
- `cbGroup` the value of the `itemBank$cbGroup` element of the item bank `itemBank`.
- `stopRule` the value of the `stop$rule` argument (or its default value if missing).
- `stopThr` the value of the `stop$thr` argument (or its default value if missing).
- `stopAlpha` the value of the `stop$alpha` argument (or its default value if missing).
- `endWarning` a logical indactor indicating whether the adaptive test stopped because the stopping rule was satisfied or not.
- `finalMethod` the value of the `final$method` argument (or its default value if missing).
- `finalDist` the value of the `final$priorDist` argument (or its default value if missing).
- `finalPar` the value of the `final$priorPar` argument (or its default value if missing).
- `finalRange` the value of the `final$range` argument (or its default value if missing).
- `finalD` the value of the `final$D` argument (or its default value if missing).
- `finalAlpha` the value of the `final$alpha` argument (or its default value if missing).

The function `print.cat` returns similar (but differently organized) results.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Birnbaum, A. (1969). Statistical theory for logistic mental test models with a prior distribution of ability. *Journal of Mathematical Psychology, 6*, 258-276.

Bock, R. D., and Mislevy, R. J. (1982). Adaptive EAP estimation of ability in a microcomputer environment. *Applied Psychological Measurement, 6*, 431-444.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

Jeffreys, H. (1939). *Theory of probability*. Oxford, UK: Oxford University Press.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 186*, 453-461.

Kingsbury, G. G., and Zara, A. R. (1989). Procedures for selecting items for computerized adaptive tests. *Applied Measurement in Education, 2*, 359-375.

Lord, F.M. (1980). *Applications of item response theory to practical testing problems*. Hillsdale, NJ: Lawrence Erlbaum.

Urry, V. W. (1970). *A Monte Carlo investigation of logistic test models.* Unpublished doctoral dissertation. West Lafayette, IN: Purdue University.

van der Linden, W. J. (1998). Bayesian item selection criteria for adaptive testing. *Psychometrika, 63*, 201-216.

Veerkamp, W. J. J., and Berger, M. P. F. (1997). Some new item selection criteria for adaptive testing. *Journal of Educational and Behavioral Statistics, 22*, 203-226.

Warm, T.A. (1989). Weighted likelihood estimation of ability in item response models. *Psychometrika, 54*, 427-450.

See Also

testList, startItems, nextItem, thetaEst, semTheta, eapEst, eapSem, MWI, MEI

Examples

```
## Not run:
 # Loading the 'tcals' parameters
 data(tcals)

 # Item bank creation with 'tcals' item parameters
 bank <- createItemBank(tcals)

 # Creation of a starting list: 5 items, initial theta 0, bw 2
 start <- list(nrItems=5, theta=0, halfRange=2)

 # Creation of 'test' list: weighted likelihood
 # estimation of provisional ability, and MEI criterion
```

```
# for next item selection
test <- list(method="WL", itemSelect="MEI")

# Creation of 'final' list: EAP estimation of final
# ability
final <- list(method="EAP")

# Creation of a stopping rule: precision criterion, standard
# error to be reached 0.3
stop <- list(rule="precision", thr=0.3)

# CAT test
res <- randomCAT(0, bank, start=start, test=test, stop=stop,
 final=final)

# New 'test' and 'final' rules (BM and EAP estimation
# with Jeffreys' prior, randomesque value 5)
test2 <- list(method="BM", priorDist="Jeffreys", randomesque=5)
final2 <- list(method="EAP", priorDist="Jeffreys")

# New stopping rule: classification criterion, with
# classification threshold 0 and alpha level 0.05
stop2 <- list(rule="classification", thr=0, alpha=0.05)

# CAT test with new 'test', 'stop' and 'final' rules
res2 <- randomCAT(0, bank, start=start, test=test2, stop=stop2,
 final=final2)

# New stopping rule: classification criterion, with
# classification threshold 0.5 and alpha level 0.05
stop3 <- list(rule="classification", thr=0.5, alpha=0.05)

# CAT test with new 'stop' rule
res3 <- randomCAT(0, bank, start=start, test=test2, stop=stop3,
 final=final2)

# new 'test' and 'stop' rule for next item selection
test3 <- list(method="WL", itemSelect="MLWI")
stop4 <- list(rule="length",thr=10)
res4 <- randomCAT(0, bank, start=start, test=test3, stop=stop4,
 final=final2)

# Item bank creation for content balancing
bank2 <- createItemBank(tcals, cb=TRUE)

# Creation of the 'cbList' list with arbitrary proportions
cbList <- list(names=c("Audio1","Audio2","Written1","Written2",
               "Written3"), props=c(0.1,0.2,0.2,0.2,0.3))

# CAT test with 'start', 'test2', 'stop4' and 'final2' lists
# and content balancing using 'cbList'
res5 <- randomCAT(0, bank2, start=start, test=test2, stop=stop4,
 final=final2, cbControl=cbList)

# Plotting results
plot(res)
plot(res, ci=TRUE)
```

```
 plot(res, ci=TRUE, trueTh=FALSE)
 plot(res, ci=TRUE, classThr=1)

 # With mistake
 plot(res, ci=0.05)
 plot(res, classThr=TRUE)

## End(Not run)
```

---

semTheta *Standard error of ability estimation under the 4PL model*

Description

This command returns the estimated standard error of the ability estimate, for a given matrix of item parameters of the 4PL model, an ability estimate and a specified estimator.

Usage

```
semTheta(thEst, it, x=NULL, D=1, method="BM", priorDist="norm",
 priorPar=c(0,1), parInt=c(-4,4,33))
```

Arguments

`thEst`	numeric: the ability estimate.
`it`	numeric: a matrix with one row per item and four columns, with the values of the discrimination, the difficulty, the pseudo-guessing and the inattention parameters (in this order).
`x`	numeric: a vector of dichotomous item responses (default is `NULL`). Ignored if `method` is not `"EAP"`.
`D`	numeric: the metric constant. Default is `D=1` (for logistic metric); `D=1.702` yields approximately the normal metric (Haley, 1952).
`method`	character: the ability estimator. Possible values are `"BM"` (default), `"ML"`, `"WL"` and `"EAP"`. See **Details**.
`priorDist`	character: specifies the prior distribution. Possible values are `"norm"` (default), `"unif"` and `"Jeffreys"`. Ignored if `method` is neither `"BM"` nor `"EAP"`. See **Details**.
`priorPar`	numeric: vector of two components specifying the prior parameters (default is `c(0,1)`) of the prior ability distribution. Ignored if `method` is neither `"BM"` nor `"EAP"`, or if `priorDist="Jeffreys"`. See **Details**.
`parInt`	numeric: vector of three components, holding respectively the values of the arguments `lower`, `upper` and `nqp` of the `eapEst` command. Default vector is (-4, 4, 33). Ignored if `method` is not `"EAP"`.

Details

Four ability estimators are available: the maximum likelihood (ML) estimator (Lord, 1980), the Bayes modal (BM) estimator (Birnbaum, 1969), the expected a posteriori (EAP) estimator (Bock and Mislevy, 1982)and the weighted likelihood (WL) estimator (Warm, 1989). The selected estimator is specified by the `method` argument, with values `"ML"`, `"BM"`, `"EAP"` and `"WL"` respectively.

For the BM and EAP estimators, three prior distributions are available: the normal distribution, the uniform distribution and the Jeffreys' prior distribution (Jeffreys, 1939, 1946). The prior distribution is specified by the argument `priorPar`, with values `"norm"`, `"unif"` and `"Jeffreys"`, respectively. The `priorPar` argument is ignored if `method="ML"` or `method="WL"`.

The argument `priorPar` determines either: the prior mean and standard deviation of the normal prior distribution (if `priorDist="norm"`), or the range for defining the prior uniform distribution (if `priorDist="unif"`). This argument is ignored if `priorDist="Jeffreys"`.

The `eapPar` argument sets the range and the number of quadrature points for numerical integration in the EAP process. By default, it takes the vector value (-4, 4, 33), that is, 33 quadrature points on the range [-4; 4] (or, by steps of 0.25). See `eapEst` for further details.

Note that in the current version, the ability estimate must be specified through the `thEst` argument. Moreover, the response pattern must be specified through the `x` argument to compute the standard error of the EAP estimate. For the other estimation methods, thi is not necessary, and `x` is set to `NULL` by default for this purpose.

Value

The estimated standard error of the ability level.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Barton, M.A., and Lord, F.M. (1981). An upper asymptote for the three-parameter logistic item-response model. Research Bulletin 81-20. Princeton, NJ: Educational Testing Service.

Birnbaum, A. (1969). Statistical theory for logistic mental test models with a prior distribution of ability. *Journal of Mathematical Psychology, 6*, 258-276.

Bock, R. D., and Mislevy, R. J. (1982). Adaptive EAP estimation of ability in a microcomputer environment. *Applied Psychological Measurement, 6*, 431-444.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

Jeffreys, H. (1939). *Theory of probability*. Oxford, UK: Oxford University Press.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 186*, 453-461.

Lord, F.M. (1980). *Applications of item response theory to practical testing problems*. Hillsdale, NJ: Lawrence Erlbaum.

Warm, T.A. (1989). Weighted likelihood estimation of ability in item response models. *Psychometrika, 54*, 427-450.

See Also

eapSem, thetaEst

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Selecting item parameters only
 tcals <- as.matrix(tcals[,1:4])

 # Creation of a response pattern (tcals item parameters,
 # true ability level 0)
 set.seed(1)
 x <- rbinom(85, 1, Pi(0, tcals)$Pi)

 # ML estimation
 th <- thetaEst(tcals, x, method="ML")
 c(th, semTheta(th, tcals, method="ML"))

 # BM estimation, standard normal prior distribution
 th <- thetaEst(tcals, x)
 c(th, semTheta(th, tcals))

 # BM estimation, uniform prior distribution upon range [-2,2]
 th <- thetaEst(tcals, x, method="BM", priorDist="unif",
                priorPar=c(-2,2))
 c(th, semTheta(th, tcals, method="BM", priorDist="unif",
    priorPar=c(-2,2)))

 # BM estimation, Jeffreys' prior distribution
 th <- thetaEst(tcals, x, method="BM", priorDist="Jeffreys")
 c(th, semTheta(th, tcals, method="BM", priorDist="Jeffreys"))

 # EAP estimation, standard normal prior distribution
 th <- thetaEst(tcals, x, method="EAP")
 c(th, semTheta(th, tcals, x, method="EAP"))

 # EAP estimation, uniform prior distribution upon range [-2,2]
 th <- thetaEst(tcals, x, method="EAP", priorDist="unif",
                priorPar=c(-2,2))
 c(th, semTheta(th, tcals, x, method="EAP", priorDist="unif",
    priorPar=c(-2,2)))

 # EAP estimation, Jeffreys' prior distribution
 th <- thetaEst(tcals, x, method="EAP", priorDist="Jeffreys")
 c(th, semTheta(th, tcals, x, method="EAP", priorDist="Jeffreys"))

 # WL estimation
 th <- thetaEst(tcals, x, method="WL")
 c(th, semTheta(th, tcals, method="WL"))

## End(Not run)
```

`startItems` *Selection of the first items*

Description

This command selects the first items of the adaptive test, either randomly or on the basis of their difficulty level.

Usage

```
startItems(itemBank, fixItems=NULL, seed=NULL, nrItems=1,
  theta=0, halfRange=2, startSelect="bOpt")
```

Arguments

`itemBank`	an item bank of class `itBank` as output of the function `createItemBank`.
`fixItems`	either a vector of integer values or `NULL` (default). See **Details**.
`seed`	either a numeric value or `NULL` (default).Ignored if `fixItems` is not `NULL`. See **Details**.
`nrItems`	numeric: the number of starting items to be selected (default is 1). Ignored if `fixItems` is not `NULL`.
`theta`	numeric: the initial ability level for selecting the first items (default is 0). Ignored if either `fixItems` or `seed` is not `NULL`. See **Details**.
`halfRange`	numeric: the half of the range of initial ability values (default is 2). Ignored if either `fixItems` or `seed` is not `NULL`. See **Details**.
`startSelect`	character: the criterion for selecting the first items. Possible values are `"bOpt"` (default) and `"MFI"`. See **Details**.

Details

This function permits to select the first items of the test. The number of starting items is given by the `nrItems` argument, with default value 1.

The first item(s) of the adaptive test can be selected by one of the following methods.

1. By specifying the item(s) to be administered. The argument `fixItems` then holds the item number(s) as listed in the item bank. Setting `fixItems` to `NULL` (default value) disables this method.
2. By selecting it (them) randomly into the item bank. The argument `seed` permits to fix the random selection by specifying the random seed number. Setting `seed` to `NULL` (default value) disables this method.
3. By selecting the item(s) according to an initial sequence of ability values (see below). In this case, two criteria can be used: either one selects the item(s) whose difficulty level is as close as possible to the inital ability value(s), or one selects the most informative item(s) for the given initial ability value(s). The criterion is specified by the `startSelect` argument, with values `"bOpt"` (default) for the 'difficulty' criterion, and `"MFI"` for the 'information' criterion.

The third method above will be used if and only if both `fixItems` and `seed` arguments are fixed to `NULL`. Otherwise, one of the first two methods will be used (see also `testList` for details about debugging misspecifications of the starting arguments).

The sequence of initial ability estimates is specified by the triplet of arguments (`nrItems`, `theta`, `halfRange`). As mentioned above, `nrItems` is the number of items to select, and thus the length of the sequence. The `theta` value is the central ability value, and `halfRange` sets half of the range of the ability values. These three arguments altogether permit to define any type of (equidistant) ability values. For instance,

- the set (-1, 1) can be obtained by specifying the triplet to (2, 0, 1);
- the set (-1, 0, 1) can be obtained by specifying the triplet to (3, 0, 1);
- the set (-1, 0, 1, 2) can be obtained by specifying the triplet to (4, 0.5, 1.5);
- etc.

Value

A list with four arguments:

`items`	the selected items (identified by their number in the item bank).
`par`	the matrix of item parameters of the selected items (one row per item).
`thStart`	the sequence of starting ability values used for selecting the items.
`startSelect`	the value of the `startSelect` argument.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

See Also

`createItemBank`, `testList`

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Item bank creation with 'tcals' item parameters
 bank <- createItemBank(tcals)

 # Random selection of 4 starting items
 startItems(bank, seed=1, nrItems=4)

 # Selection of the first 5 starting items
 startItems(bank, fixItems=1:5)

 # Selecting 1 starting item, initial ability estimate is 0
 startItems(bank)
```

```
 # Selecting 3 starting items, initial ability estimate is 1
 # and half range is 2
 startItems(bank, nrItems=3, theta=1, halfRange=2)

 # Idem but with 'information' criterion
 startItems(bank, nrItems=3, theta=1, halfRange=2, startSelect="MFI")

 # Selecting 5 starting items, initial ability estimate is 2
 # and half range is 3
 startItems(bank, nrItems=5, theta=2, halfRange=3)

## End(Not run)
```

`tcals` *Items parameters of the TCALS 1998 data set and subgroups of items*

Description

The TCALS (*Test d'Aptitude en Anglais Langue Seconde*) is an aptitude test of English language as a second language in the French speaking college of Outaouais (Gatineau, QC, Canada). The test consists of 85 items and is administered every year to newly incoming students. The item parameters of the year 1998 have been estimated under the 3PL model. Inattention parameters are therefore fixed to one. Subgroups of items are also included for content balancing purposes.

Format

A matrix with 85 rows and five columns, respectively holding the discrimination, difficulty, pseudo-guessing and inattention parameters as calibrated on the results of the 1998 application of the TCALS questionnaire. The fifth column holds the name of the subgroups of items:

- *Audio1*: listening comprehension of sentences (items **1** to **12**).
- *Audio2*: listening comprehension of dialogs and short texts(items **13** to **33**).
- *Written1*: written vocabulary exercises(items **34** to **46**).
- *Written2*: written grammar exercises(items **47** to **63**).
- *Written3*: written exercises of other types: reading and mistake detection (items **64** to **85**).

Source

The TCALS test was originally developed by Laurier, Froio, Pearo and Fournier (1998) and item parameters were obtained from Raiche (2002).

References

Laurier, M., Froio, L., Pearo C., and Fournier, M. (1998). Test de classement d'anglais langue seconde au collegial. Montreal, Canada: College de Maisonneuve.

Raiche, G. (2002). Le depistage du sous-classement aux tests de classement en anglais, langue seconde, au collegial [The detection of under classification to the collegial English as a second language placement tests]. Gatineau, QC: College de l'Outaouais.

test.cbList *Testing the format of the list for content balancing*

Description

This command tests whether format of the list to be assigned to `cbControl` argument of function `nextItem` is appropriate for content balancing, and returns a warning message otherwise.

Usage

```
test.cbList(list, itemBank)
```

Arguments

`list`	a list of arguments to be tested. See **Details**.
`itemBank`	an item bank of class `itBank` as output of the function `createItemBank`, holding subgroups of items (i.e. such that `itemBank$cbGroup` is not `NULL`.

Details

The `test.cbList` function checks whether the list provided in the `cbControl` argument of the `nextItem` and `randomCAT` functions, is accurate for controlling for ontent balancing. It mainly serves as an initial check for the `randomCAT` function.

The function returns an `"ok"` message if the arguments of `list` match the requirement of the list `cbControl` for content balancing. Otherwise, a message is returned with information about list - type mismatch. This will be the case if:

- `list` is not a list or has not exactly two elements,
- at least one of the argument names is incorrect,
- the lengths of the arguments are different, or different from the number of subgroups of items,
- the 'names' element does not match with the subgroups' names,
- the 'props' element is not numeric or holds negative values.

Each mismatch yields a different output message to help in debugging the problem.

Value

A list with two arguments:

`test`	a logical value indicating whether the format of the list is accurate (`TRUE`) or not (`FALSE`).
`message`	either a message to indicate the type of misspecification, or `"ok"` if the format is accurate.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

See Also

nextItem, randomCAT

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Item bank creation for content balancing
 bank <- createItemBank(tcals, cb=TRUE)

 # Creation of a correct list with arbitrary proportions
 cbList <- list(names=c("Audio1","Audio2","Written1","Written2",
                "Written3"), props=c(0.1,0.2,0.2,0.2,0.3))

 # Testing 'cbList'
 test.cbList(cbList, bank)

 # Creation of an incorrect list (mismatch in first name)
 cbList <- list(names=c("audio1","Audio2","Written1","Written2",
                "Written3"), props=c(0.1,0.2,0.2,0.2,0.3))
 test.cbList(cbList, bank)

 # Creation of an incorrect list (mismatch in name of second
 # element)
 cbList <- list(names=c("Audio1","Audio2","Written1","Written2",
                "Written3"), prop=c(0.1,0.2,0.2,0.2,0.3))
 test.cbList(cbList, bank)

 # Creation of an incorrect list (second element shorter than
 # first element)
 cbList <- list(names=c("Audio1","Audio2","Written1","Written2",
                "Written3"), props=c(0.1,0.2,0.2,0.2))
 test.cbList(cbList, bank)

 # Creation of an incorrect list (adding a third element)
 cbList <- list(names=c("Audio1","Audio2","Written1","Written2",
                "Written3"), props=c(0.1,0.2,0.2,0.2), third="hi")
 test.cbList(cbList, bank)

## End(Not run)
```

---

testList *Testing the format of the input lists*

Description

This command tests whether format of the input lists for the random generation of adaptive tests is convenient, and returns a warning message otherwise.

Usage

```
testList(list, type="start")
```

Arguments

`list`	a list of arguments to be tested. See **Details**.
`type`	character: the type of list for checking. Possible values are `"start"` (default), `"test"`, `"stop"` and `"final"`. See **Details**.

Details

The `testList` function checks whether the list provided in the `list` argument is accurate for the selected `type`. It mainly serves as an initial check for the `randomCAT` function.

The four types of lists are: `"start"` with the parameters for selecting the first items; `"test"` with the options of the adaptive test (i.e. method for next item selection, provisional ability estimator and related information); `"stop"` with the options setting the stopping rule; and `"final"` with the options for final ability estimation. See the help file of `randomCAT` for further details about the different lists, their allowed arguments and their contents.

The function returns an `"ok"` message if the arguments of `list` match the requirement of the corresponding `type`. Otherwise, a message is returned with information about list - type mismatch. This will be the case if:

- `list` is not a list, or has no argument names,
- `list` has too many arguments for the `type` specified,
- at least one of the argument names is incorrect,
- the content of at least one argument is not adequate (e.g. character instead of numeric).

Each mismatch yields a different output message to help in debugging the problem.

Value

A list with two arguments:

`test`	a logical value indicating whether the format of the list is accurate (`TRUE`) or not (`FALSE`).
`message`	either a message to indicate the type of misspecification, or `"ok"` if the format is accurate.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
`<david.magis@ulg.ac.be>`

See Also

`randomCAT`

Examples

```
## Not run:

 # Creation and test of a 'start' list
 start <- list(nrItems=3, theta=0, halfRange=2)
 testList(start, type="start")

 # Modification of the list to introduce a mistake
 names(start)[1] <- "nrItem"
 testList(start, type="start")

 # Creation and test of a 'test' list
 test <- list(method="WL", itemSelect="Urry")
 testList(test, type="test")

# Creation and test of a 'stop' list
 stop <- list(method="WL")
 testList(stop, type="test")

# Creation and test of a 'final' list (with mistake)
 final <- list(method="MAP")
 testList(final, type="final")

## End(Not run)
```

`thetaEst` *Ability estimation under the 4PL model*

Description

This command returns the ability estimate for a given matrix of item parameters of the 4PL model and a given response pattern. Available estimators are maximum likelihood, Bayes modal, expected a posteriori (EAP) and weighted likelihood.

Usage

```
 thetaEst(it, x, D=1, method="BM", priorDist="norm",
  priorPar=c(0,1), range=c(-4,4),
  parInt=c(-4,4,33))
```

Arguments

`it`	numeric: a matrix with one row per item and four columns, with the values of the discrimination, the difficulty, the pseudo-guessing and the inattention parameters (in this order).
`x`	numeric: a vector of dichotomous item responses.
`D`	numeric: the metric constant. Default is `D=1` (for logistic metric); `D=1.702` yields approximately the normal metric (Haley, 1952).
`method`	character: the ability estimator. Possible values are `"BM"` (default), `"ML"`, `"WL"` and `"EAP"`. See **Details**.

priorDist	character: specifies the prior distribution. Possible values are "norm" (default), "unif" and "Jeffreys". Ignored if method is neither "BM" nor "EAP". See **Details**.
priorPar	numeric: vector of two components specifying the prior parameters (default is c(0,1)) of the prior ability distribution. Ignored if method is neither "BM" nor "EAP", or if priorDist="Jeffreys". See **Details**.
range	numeric: vector of two components specifying the range wherein the ability estimate must be looked for (default is c(-4,4)). Ignored if method=="EAP".
parInt	numeric: vector of three components, holding respectively the values of the arguments lower, upper and nqp of the eapEst command. Default vector is (-4, 4, 33). Ignored if method is not "EAP".

Details

Four ability estimators are available: the maximum likelihood (ML) estimator (Lord, 1980), the Bayes modal (BM) estimator (Birnbaum, 1969), the expected a posteriori (EAP) estimator (Bock and Mislevy, 1982) and the weighted likelihood (WL) estimator (Warm, 1989). The selected estimator is specified by the method argument, with values "ML", "BM", "EAP" and "WL" respectively.

For the BM and EAP estimators, three prior distributions are available: the normal distribution, the uniform distribution and Jeffreys' prior distribution (Jeffreys, 1939, 1946). The prior distribution is specified by the argument priorPar, with values "norm", "unif" and "Jeffreys", respectively. The priorPar argument is ignored if method="ML" or method="WL".

The argument priorPar determines either the prior mean and standard deviation of the normal prior distribution (if priorDist="norm"), or the range for defining the prior uniform distribution (if priorDist="unif"). This argument is ignored if priorDist="Jeffreys".

The eapPar argument sets the range and the number of quadrature points for numerical integration in the EAP process. By default, it takes the vector value (-4, 4, 33), that is, 33 quadrature points on the range [-4; 4] (or, by steps of 0.25). See eapEst for further details.

The range argument permits to limit the interval of investigation for the ML, BM and WL ability estimates (in particular, to avoid infinite ability estimates). The default range is [-4, 4].

Value

The estimated ability level.

Author(s)

David Magis
Post-doctoral researcher, FNRS (Fonds National de la Recherche Scientifique)
Department of Mathematics, University of Liege, Belgium
<david.magis@ulg.ac.be>

References

Birnbaum, A. (1969). Statistical theory for logistic mental test models with a prior distribution of ability. *Journal of Mathematical Psychology, 6*, 258-276.

Bock, R. D., and Mislevy, R. J. (1982). Adaptive EAP estimation of ability in a microcomputer environment. *Applied Psychological Measurement, 6*, 431-444.

Haley, D.C. (1952). Estimation of the dosage mortality relationship when the dose is subject to error. Technical report no 15. Palo Alto, CA: Applied Mathematics and Statistics Laboratory, Stanford University.

Jeffreys, H. (1939). *Theory of probability*. Oxford, UK: Oxford University Press.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 186*, 453-461.

Lord, F.M. (1980). *Applications of item response theory to practical testing problems*. Hillsdale, NJ: Lawrence Erlbaum.

Warm, T.A. (1989). Weighted likelihood estimation of ability in item response models. *Psychometrika, 54*, 427-450.

See Also

eapEst, semTheta

Examples

```
## Not run:

 # Loading the 'tcals' parameters
 data(tcals)

 # Selecting item parameters only
 tcals <- as.matrix(tcals[,1:4])

 # Creation of a response pattern (tcals item parameters,
 # true ability level 0)
 set.seed(1)
 x <- rbinom(85, 1, Pi(0, tcals)$Pi)

 # ML estimation
 thetaEst(tcals, x, method="ML")

 # BM estimation, standard normal prior distribution
 thetaEst(tcals, x)

 # BM estimation, uniform prior distribution upon range [-2,2]
 thetaEst(tcals, x, method="BM", priorDist="unif", priorPar=c(-2,2))

 # BM estimation, Jeffreys' prior distribution
 thetaEst(tcals, x, method="BM", priorDist="Jeffreys")

 # EAP estimation, standard normal prior distribution
 thetaEst(tcals, x, method="EAP")

 # EAP estimation, uniform prior distribution upon range [-2,2]
 thetaEst(tcals, x, method="EAP", priorDist="unif", priorPar=c(-2,2))

 # EAP estimation, Jeffreys' prior distribution
 thetaEst(tcals, x, method="EAP", priorDist="Jeffreys")

 # WL estimation
 thetaEst(tcals, x, method="WL")

## End(Not run)
```

Index