Package 'catSurv'

March 16, 2017

Title Computerized Adaptive Testing

Version 0.0.0.9000

Description Provides methods of computerized adaptive testing for survey researchers. Includes functionality for data fit with the Latent Trait, Birnbaum`s Three Parameter, the Graded Response, or the Generalized Partial Credit model. Additionally, includes several ability parameter estimation and item selection routines. During item selection, all calculations are done in compiled C++ code.

Depends R(>= 3.3.2)

License What license is it under?

Imports ltm(>= 1.0.0),
methods(>= 3.3.2),
stats(>= 3.3.2),
Rcpp(>= 0.12.9),

LinkingTo BH(>= 1.60.0.2),
Rcpp(>= 0.12.9),
RcppArmadillo,
RcppGSL(>= 0.3.1)

LazyData true

RoxygenNote 5.0.1

SystemRequirements C++11

Suggests catIrt(>= 0.5.0),
catR(>= 3.10),
testthat(>= 1.0.2)

R topics documented:

Cat-class 2
checkStopRules 4
d2LL 5
dLL 6
estimateSE 8
estimateTheta 9
expectedKL 11
expectedObsInf 12
expectedPV 13
fisherInf 14

fisherTestInfo . . . 15
gpcm-class . . . 16
grm-class . . . 17
likelihood . . . 18
likelihoodKL . . . 20
lookAhead . . . 21
ltm-class . . . 22
makeTree . . . 23
nfc . . . 24
npi . . . 25
obsInf . . . 27
polknow . . . 28
posteriorKL . . . 32
prior . . . 33
probability . . . 35
selectItem . . . 37
storeAnswer . . . 39
tpm-class . . . 40

Index **42**

`Cat-class` *A Computerized Adaptive Testing Survey (catSurv) Object*

Description

Creates an object of class `Cat`. Cat objects are used in administering Computerized Adaptive Testing (CAT) Surveys. These objects contain several pieces of information relevent for CAT surveys, and are used as input in the main functions of the `catSurv` package.

Details

Assume we have a survey battery with `I` questions. An object of the class `Cat` has the following slots:

- `guessing` A vector of length `I` of guessing parameters. Note: guessing parameters are only applicable for `Cat` objects fit with the `ltm` model, using the `ltmCat` function.
- `discrimination` A named vector of length `I` of disrimination parameters.
- `difficulty` A named vector or list of length `I` of difficulty parameters. For binary `Cat` objects, the vector will contain difficulty parameters for each item. For categorical `Cat` objects, a list will constain a vector for each item, and each vector will contain a difficulty parameter for each response option.
- `answers` A vector of length `I` of answers to questions as given by the survey respondent. Unanswered questions have the value `NA`.
- `priorName` A character vector of length one giving the prior distribution to use for the latent trait estimates. The options are `NORMAL` for the normal distirbution, `STUDENT_T` for the student's t distribution, and `UNIFORM` for the uniform distribution. The default value is `NORMAL`.

- priorParams A numeric vector of length two of parameters for the distribution specified in the priorName slot. When priorName is set to NORMAL, the first element of priorParams is the mean, the second element is the standard deviation. When priorName is set to STUDENT_T, the first element of priorParams is mu, a location parameter, the second is degrees of freedom. When priorName is set to UNIFORM, the elements of priorParams are lower and upper bounds, respectively. Note that the uniform distribution is only applicable for the "EAP" estimation method. The default values are $0, 1$.
- lowerBound A numeric indicating the lower bound of the interval of the latent scale used in estimation. The default value is -5.
- upperBound A numeric indicating the upper bound of the interval of the latent scale used in estimation. The default value is 5.
- model A string indicating the model fit to the data. The options are ltm for the latent trait model, tpm for Birnbaum's three parameter model, grm for the graded response model, and gpcm for the generalized partial credit model.
- estimation A string indicating the choice of approach to estimate ability parameters. The options are EAP for the expected a posteriori approach, MAP for the modal a posteriori approach, MLE for the maximum likelihood approach, and WLE for the weighted maximum likelihood approach. The default value is EAP.
- estimationDefault A string indicating the choice of approach to estimate ability parameters when the primary estimation choice indicated in the estimation slot fails to converge. The options are EAP and MAP. The default value is MAP.
- selection A string indicating the choice of approach for selecting the next item. The options are EPV for minimum expected posterior variance, MEI for maximum expected information, MFI for maximum Fisher information, MPWI for maximum posterior weighted information, MLWI for maximum likelihood weighted information, KL for the maximum expected Kullback-Leibler (KL) information, LKL maximum likelihood weighted KL information, PKL maximum posterior weighted KL information, MFII, and RANDOM where the next item is chosen randomly. The default value is EPV.
- z A numeric used in calculating δ. δ is used in determining the bounds of integration for some selectItem methods. Default value is 0.9.
- lengthThreshold A numeric. The number of questions answered must be greater than or equal to this threshold to stop administering items. The default value is NA.
- seThreshold A numeric. The standard error estimate of the latent trait must be less than this threshold to stop administering items. The default value is NA.
- infoThreshold A numeric. The Fisher's information for all remaining items must be less than this threshold to stop administering items. The default value is NA.
- gainThreshold A numeric. The absolute value of the difference between the standard error of the latent trait estimate and the square root of the expected posterior variance for each item must be less than this threshold to stop administering items. The default value is NA.
- lengthOverride A numeric. The number of questions answered must be less than this override to continue administering items. The default value is NA.
- gainOverride A numeric. The absolute value of the difference between the standard error of the latent trait estimate and the square root of the expected posterior variance for each item must be less than this override to continue administering items. The default value is NA.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`estimateTheta` for more information on the estimation procedures

`selectItem` for more information on the item selection procedures

`checkStopRules` for more information on stopping thresholds and overrides

---

`checkStopRules` *Check if Stop and/or Override Rules are Met*

Description

Evaluates the specified stopping and/or override rules to check if respondent should be asked further questions.

Usage

```
checkStopRules(catObj)
```

Arguments

`catObj`	An object of class `Cat`

Details

The stopping rule thresholds are stored in the following Cat object slots: `lengthThreshold`, `seThreshold`, `infoThreshold`, and `gainThreshold`. The override thresholds are stored in the following Cat object slots: `lengthOverride`, `gainOverride`. A value of `NA` indicates the rule will not be used in evaluating if further questions should be administered. A user can specify any combination of stopping rules and/or overrides.

Value

The function `checkStopRules` returns a boolean. `TRUE` indicates all specified stopping rules are met and no specified overrides are met. No further items should be adminstered. `FALSE` indicates at least one specified stopping rule is not met, or if any specified override threshold is met. Additional items should be administered.

Stopping Rules:

`lengthThreshold`: Number of question's answered $\geq a$

`seThreshold`: $SE(\hat{\theta}) < a$

`infoThreshold`: $FI < a \; \forall$ remaining items

`gainThreshold`: $SE(\hat{\theta}) - \sqrt{EPV}| < a \; \forall$ remaining items

Overrides:

`lengthOverride`: Number of question's answered $< a$

`gainOverride`: $|SE(\hat{\theta}) - \sqrt{EPV}| \geq a \; \forall$ remaining items

References

Babcock, Ben, and David J. Weiss. 2009. "Termination Criteria in Computerized Adaptive Tests: Variable-Length CATs are not Biased." Proceedings of the 2009 GMAC Conference on Computerized Adaptive Testing. Vol. 14.

Examples

```
## Not run:
## Fitting ltm Cat object
data(npi)
ltm_cat <- ltmCat(npi)

## Storing example answers
setAnswers(ltm_cat) <- c(1,0,1,0,1,0,0,0,1,1, rep(NA, 30))

## Stop administering items if standard error of ability
## estimate is low enough
setSeThreshold(ltm_cat) <- .5
checkStopRules(ltm_cat)

## Now stop if standard error is low enough, but only if respondent has
## answered 11 questions
setLengthOverride(ltm_cat) <- 11
checkStopRules(ltm_cat)

## When respondent has answered 11 questions and standard error
## of ability estimate is below .5, stop administering items
setAnswers(ltm_cat) <- c(1,0,1,0,1,0,0,0,1,1,0, rep(NA, 29))
checkStopRules(ltm_cat)

## End(Not run)
```

---

d2LL *The Second Derivative of the Log-Likelihood*

Description

Calculates either the second derivative of the log-likelihood or the second derivative of the log-posterior evaluated at point θ.

Usage

```
d2LL(catObj, theta, use_prior)
```

Arguments

catObj	An object of class Cat
theta	A numeric or an integer indicating the value for θ
use_prior	A logical indicating whether to calculate baseded on the log-likelihood or log-posterior

Details

When usePrior = FALSE, the function d2LL evaluates the second derivative of the log-likelihood at point θ.

When usePrior = TRUE, the function d2LL evaluates the second derivative of the log-posterior at point θ.

The function dLL2 is only available when using the normal prior distribution when use_prior=TRUE.

Value

The function d2LL returns a numeric of the second derivative of the log-likelihood (or log-posterior) given a respondent's answer profile.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

Cat, prior, dLL

Examples

```
## Not run:
## Create Cat object, store example answers, and calculate
## second derivative of log-likelihood at theta = 1

## d2LL for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
d2LL(ltm_cat, theta = 1)

## d2LL for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
setAnswers(tpm_cat) <- c(1,0,1,0, rep(NA, 35))
d2LL(tpm_cat, theta = 1)

## d2LL for Cat object of the grm model
data(nfc)
grm_cat <- grmCat(nfc)
setAnswers(grm_cat) <- c(1,3,4,5, rep(NA, 13))
d2LL(grm_cat, theta = 1)

## End(Not run)
```

---

dLL *The First Derivative of the Log-Likelihood*

Description

Calculates either the first derivative of the log-likelihood or the first derivative of the log-posterior evaluated at point θ.

Usage

```
dLL(catObj, theta, use_prior)
```

Arguments

catObj	An object of class Cat
theta	A numeric or an integer indicating the value for θ_j
use_prior	A logical indicating whether to calculate baseded on the log-likelihood or log-posterior

Details

When usePrior = FALSE, the function dLL evaluates the first derivative of the log-likelihood at point θ.

When usePrior = TRUE, the function dLL evaluates the first derivative of the log-posterior at point θ.

The function dLL is only available when using the normal prior distribution when use_prior=TRUE.

Value

The function dLL returns a numeric of the derivative of the log-likelihood (or log-posterior) given a respondent's answer profile.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

Cat, prior

Examples

```
## Not run:
## Create Cat object, store example answers, and calculate
## first derivative of log-likelihood at theta = 1

## dLL for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
dLL(ltm_cat, theta = 1)

## dLL for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
setAnswers(tpm_cat) <- c(1,0,1,0, rep(NA, 35))
```

```
dLL(tpm_cat, theta = 1)

## dLL for Cat object of the grm model
data(nfc)
grm_cat <- grmCat(nfc)
setAnswers(grm_cat) <- c(1,3,4,5, rep(NA, 13))
dLL(grm_cat, theta = 1)

## End(Not run)
```

`estimateSE` *Standard Error of Ability Parameter Estimate*

Description

Estimates the standard error for a respondent's ability parameter estimate.

Usage

```
estimateSE(catObj)
```

Arguments

`catObj`	An object of class `Cat`

Details

The EAP estimator: The MAP estimator (This is implemented only for the normal prior.) The MLE estimator (When MLE can't be calculated... estimationDefault) The WLE estimator (Brent method)

Value

The function `estimateSE` returns a numeric for the standard error for θ.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`estimateTheta` for estimation of θ

Examples

```
## Not run:
## Prior calculation using Cat object of the ltm model
## specifying different distributions
data(npi)
cat <- ltmCat(npi)

## End(Not run)
```

`estimateTheta` *Estimate of the Respondent's Ability Parameter*

Description

Estimates the expected value of the ability parameter θ, conditioned on the observed answers, prior, and the item parameters.

Usage

```
estimateTheta(catObj)
```

Arguments

`catObj`	An object of class `Cat`

Details

Estimation approach is specified in `estimation` slot of `Cat` object.

The expected a posteriori approach is used when `estimation` slot is "EAP".

The modal a posteriori approach is used when `estimation` slot is "MAP". This method is only available using the normal prior distribution.

The maximum likelihood approach is used when `estimation` slot is "MLE". When the likelihood is undefined, the MAP or EAP method will be used, determined by what is specified in the `estimationDefault` slot in `Cat` object.

The weighted maximum likelihood approach is used when `estimation` slot is "WLE". Estimating θ requires root finding with the "Brent" method in the `gsl` library.

Value

The function `estimateTheta` returns a numeric consisting of the expected value of the respondent's ability parameter.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

Cat

Examples

```
## Not run:
## Create Cat object, store example answers, and estimate ability parameter
## using different estimation procedures

## theta estimates for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
setEstimation(ltm_cat) <- "EAP"
estimateTheta(ltm_cat)

setEstimation(ltm_cat) <- "MAP"
estimateTheta(ltm_cat)

setEstimation(ltm_cat) <- "MLE"
estimateTheta(ltm_cat)

setEstimation(ltm_cat) <- "WLE"
estimateTheta(ltm_cat)

## theta estimates for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
setAnswers(tpm_cat) <- c(1,0,1,0, rep(NA, 35))
setEstimation(tpm_cat) <- "EAP"
estimateTheta(tpm_cat)

setEstimation(tpm_cat) <- "MAP"
estimateTheta(tpm_cat)

setEstimation(tpm_cat) <- "MLE"
estimateTheta(tpm_cat)

setEstimation(tpm_cat) <- "WLE"
estimateTheta(tpm_cat)

## theta estimates for Cat object of the grm model
data(nfc)
grm_cat <- grmCat(nfc)
setAnswers(grm_cat) <- c(1,3,4,5, rep(NA, 13))
setEstimation(grm_cat) <- "EAP"
estimateTheta(grm_cat)

setEstimation(grm_cat) <- "MAP"
estimateTheta(grm_cat)

setEstimation(grm_cat) <- "MLE"
estimateTheta(grm_cat)

setEstimation(grm_cat) <- "WLE"
estimateTheta(grm_cat)
```

```
## End(Not run)
```

`expectedKL` *Expected Kullback-Leibeler information*

Description

Calculates the expected Kullback-Leibeler information for an individual question item

Usage

```
expectedKL(catObj, item)
```

Arguments

`catObj`	An object of class `Cat`
`item`	An integer indicating the index of the question item

Details

Binary details (Due to the conditional independence assumption, we only need to calculate the expected value for potential new items.)

Value

The function returns a numeric indicating the KL information for the desired item, given the current answer profile and ability parameter estimate

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`likelihoodKL` and/or `posteriorKL` for alternative KL methods

Examples

```
## Not run:
## Prior calculation using Cat object of the ltm model
## specifying different distributions
data(npi)
cat <- ltmCat(npi)

## End(Not run)
```

`expectedObsInf` *Expected Observed Information*

Description

Calculates the expected information, which is the observed information attained from a specific response set times the probability of that profile occurring.

Usage

```
expectedObsInf(catObj, item)
```

Arguments

`catObj`	An object of class `Cat`
`item`	An integer indicating the index of the question item

Value

The function returns a numeric value of the expected information

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`estimateTheta` for calculation of θ and `obsInf` for observed information calculation

Examples

```
## Not run:
## Prior calculation using Cat object of the ltm model
## specifying different distributions
data(npi)
ltm_cat <- ltmCat(npi)
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
expectedObsInf(ltm_cat, item = 10)

## End(Not run)
```

`expectedPV` *Expected Posterior Variance*

Description

Estimates the expected posterior variance for a respondent's estimated ability parameter for an item yet to be answered based on a respondent's ability parameter estimate from the already-answered items

Usage

```
expectedPV(catObj, item)
```

Arguments

`catObj`	An object of class `Cat`
`item`	An integer indicating the index of the question item

Value

A numeric value indicating a respondent's expected posterior variance for a yet to be asked question item

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`probability` for probability of responses to individual question item

`estimateTheta` for estimation of θ

Examples

```
## Not run:
## Prior calculation using Cat object of the ltm model
## specifying different distributions
data(npi)
cat <- ltmCat(npi)

## End(Not run)
```

`fisherInf` *Fisher's Information*

Description

Calculates the expected value of the observed information of the likelihood evaluated at the input value θ

Usage

```
fisherInf(catObj, theta, item)
```

Arguments

`catObj`	An object of class `Cat`
`theta`	A numeric or an integer indicating the potential value for θ_j
`item`	An integer indicating the index of the question item

Details

For the dichotomous case, this is equivalent to the observed information.

Value

The function returns a numeric of the expected value of the observed information

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`obsInf` for observed information calculation

`fisherTestInfo` for further application of Fisher's information

Examples

```
## Not run:
## EI using Cat object of the ltm model
data(npi)
cat <- ltmCat(npi)

## End(Not run)
```

`fisherTestInfo` *Fisher's Test Information*

Description

Calculates the total information gained for a respondent j for all answered items, conditioned on $theta$.

Usage

```
fisherTestInfo(catObj)
```

Arguments

`catObj`	An object of class `Cat`

Value

The total information gained for a respondent, given a specific answer set and a value of $theta$.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`fisherInf` for calculation of Fisher's information for an individual question item

Examples

```
## Not run:
## Prior calculation using Cat object of the ltm model
## specifying different distributions
data(npi)
cat <- ltmCat(npi)

## End(Not run)
```

gpcm-class *Computerized Adaptive Testing Generalized Partial Credit Model*

Description

This function fits the Generalized Partial Credit model for ordinal polytomous data and populates the fitted values for discimination and difficulty parameters to an object of class Cat.

Arguments

data	A data.frame of manifest variables or an object of class gpcm.
quadraturePoints	A numeric to be passed into the gpcm function indicating the number of Gauss-Hermite quadrature points. Only applicable when data is a data.frame. Default value is 21.
...	arguments to be passed to methods. For more details about the arguments, see gpcm in the ltm package.

Details

The data argument of the function gpcmCat is either a data.frame or an object of class gpcm from the ltm package. If it is a data.frame each row represents a respondent and each column represents a question item. If it is an object of the class gpcm, it is output from the gpcm function in the ltm package.

The quadraturePoints argument of the function gpcmCat is used only when the data argument is of class data.frame. quadraturePoints is then passed to the gpcm function from the ltm package when fitting the Generalized Partial Credit model to the data and is used when approximating the value of integrals.

Value

The function gpcmCat returns an object of class Cat with changes to the following slots:

- difficulty A list of difficulty parameters, where each element in the list corresponds to the difficulty parameters for an item.
- discrimination A vector consisting of of disrimination parameters for each item.
- model The string "gpcm", indicating this Cat object corresponds to a Generalized Partial Credit model.

Note

In case the Hessian matrix at convergence is not positive definite try to use start.val = "random".

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

References

Baker, Frank B. and Seock-Ho Kim. 2004. Item Response Theory: Parameter Estimation Techniques. New York: Marcel Dekker.

Muraki, Eiji. 1992. “A generalized partial credit model: Application of an EM algorithm." ETS Research Report Series 1992(1):1-30.

Rizopoulos, Dimitris. 2006. “ltm: An R Package for Latent Variable Modeling and Item Response Theory Analyses." Journal of Statistical Software 17(5):1-25.

See Also

`Cat` for information on all `Cat` slots and their default values

`grmCat` for an alternative model fit to ordinal polytomous data

---

grm-class *Computerized Adaptive Testing Graded Response Model*

Description

This function fits the Graded Response model for ordinal polytomous data and populates the fitted values for discimination and difficulty parameters to an object of class `Cat`.

Arguments

`data`	A `data.frame` of manifest variables or an object of class `grm`.
`quadraturePoints`	A numeric to be passed into the `grm` function indicating the number of Gauss-Hermite quadrature points. Only applicable when `data` is a `data.frame`. Default value is `21`.
`...`	arguments to be passed to methods. For more details about the arguments, see `grm` in the `ltm` package.

Details

The `data` argument of the function `grmCat` is either a `data.frame` or an object of class `grm` from the `ltm` package. If it is a `data.frame` each row represents a respondent and each column represents a question item. If it is an object of the class `grm`, it is output from the `grm` function in the `ltm` package.

The `quadraturePoints` argument of the function `grmCat` is used only when the `data` argument is of class `data.frame`. `quadraturePoints` is then passed to the `grm` function from the `ltm` package when fitting the Graded Response model to the data and is used when approximating the value of integrals.

Value

The function `grmCat` returns an object of class `Cat` with changes to the following slots:

- `difficulty` A list of difficulty parameters, where each element in the list corresponds to the difficulty parameters for an item.
- `discrimination` A vector consisting of disrimination parameters for each item.
- `model` The string `"grm"`, indicating this `Cat` object corresponds to a Graded Response model.

Note

In case the Hessian matrix at convergence is not positive definite try to use `start.val = "random"`.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

References

Baker, Frank B. and Seock-Ho Kim. 2004. Item Response Theory: Parameter Estimation Techniques. New York: Marcel Dekker.

Samejima, Fumiko. 1969. "Estimation of Latent Ability Using a Response Pattern of Graded Scores." Psychometrika monograph supplement 34(4):100-114.

Rizopoulos, Dimitris. 2006. "ltm: An R Package for Latent Variable Modeling and Item Response Theory Analyses." Journal of Statistical Software 17(5):1-25.

See Also

`Cat` for information on all `Cat` slots and their default values

`gpcmCat` for an alternative model fit to ordinal polytomous data

Examples

```
## Not run:
## Creating Cat object with raw data
data(nfc)
grm_cat1 <- grmCat(nfc, quadraturePoints = 100)

## Creating Cat object with fitted object of class grm
grm_fit <- grm(nfc, control = list(GHk = 100)) ## from ltm package
class(grm_fit)
grm_cat2 <- grmCat(grm_fit)

## Note the two Cat objects are identical
identical(grm_cat1, grm_cat2)

## Note the slots that have changed from default values
grm_cat1@model
grm_cat1@difficulty
grm_cat1@discrimination

## End(Not run)
```

---

likelihood *Likelihood of the Specified Response Set*

Description

Calculates the likelihood of a respondent, with ability parameter θ, having offered the specific set of responses stored in the `Cat` objects `answers` slot. All calculations are conditional on the item-level parameters stored in the `Cat` object.

Usage

```
likelihood(catObj, theta)
```

Arguments

catObj	An object of class Cat
theta	A numeric or an integer indicating the value for θ_j

Value

The function likelihood returns a numeric value of the likelihood of the respondent having offered the provided response profile.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

probability, Cat

Examples

```
## Not run:
## Create Cat object, store example answers, and calculate
## likelihood at theta = 1

## Likelihood for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
likelihood(ltm_cat, theta = 1)

## Likelihood for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
setAnswers(tpm_cat) <- c(1,0,1,0, rep(NA, 35))
likelihood(tpm_cat, theta = 1)

## Likelihood for Cat object of the grm model
data(nfc)
grm_cat <- grmCat(nfc)
setAnswers(grm_cat) <- c(1,3,4,5, rep(NA, 13))
likelihood(grm_cat, theta = 1)

## End(Not run)
```

`likelihoodKL` *Expected Kullback-Leibeler information, weighted by the likelihood*

Description

Calculate the expected Kullback-Leibeler information, weighted by the likelihood

Usage

```
likelihoodKL(catObj, item)
```

Arguments

`catObj`	An object of class `Cat`
`item`	An integer indicating the index of the question item

Details

The LKL calculation follows the same procedure as `expectedKL`, except it requires weighting the different potential values of θ_0 by the likelihood. Thus, the equation is

Binary details:

Due to the conditional independence assumption, we only need to calculate the expected value for potential new items.

Value

A value indicating the LKL information for the desired item, given the current answer profile and ability estimate.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`expectedKL` and/or `posteriorKL` for alternative KL methods

Examples

```
## Not run:
## Prior calculation using Cat object of the ltm model
## specifying different distributions
data(npi)
cat <- ltmCat(npi)

## End(Not run)
```

`lookAhead` *Look Ahead to Select Next Item*

Description

Selects the next item that would be asked for all possible response options to the question the respondent is currently answering

Usage

```
lookAhead(catObj, item)
```

Arguments

`catObj`	An object of class `Cat`
`item`	A numeric indicating the question item the respondent is currently answering.

Value

A vector of values indicating the possible subsequent questions

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`selectItem` for selection method information

Examples

```
## Not run:
## Prior calculation using Cat object of the ltm model
## specifying different distributions
data(npi)
cat <- ltmCat(npi)

## End(Not run)
```

`ltm-class` *Computerized Adaptive Testing Latent Trait Model*

Description

This function fits the latent trait model for binary data and populates the fitted values for discimination and difficulty parameters to an object of class `Cat`.

Arguments

`data`	A `data.frame` of manifest variables or an object of class `ltm`.
`quadraturePoints`	A numeric to be passed into the `ltm` function indicating the number of Gauss-Hermite quadrature points. Only applicable when `data` is a `data.frame`. Default value is `21`.
`...`	arguments to be passed to methods. For more details about the arguments, see `ltm` in the `ltm` package.

Details

The `data` argument of the function `ltmCat` is either a `data.frame` or an object of class `ltm` from the `ltm` package. If it is a `data.frame` each row represents a respondent and each column represents a question item. If it is an object of the class `ltm`, it is output from the `ltm` function in the `ltm` package.

The `quadraturePoints` argument of the function `ltmCat` is used only when the `data` argument is of class `data.frame`. `quadraturePoints` is then passed to the `ltm` function from the `ltm` package when fitting the latent trait model to the data and is used when approximating the value of integrals.

Value

The function `ltmCat` returns an object of class `Cat` with changes to the following slots:

- `difficulty` A vector consisting of difficulty parameters for each item.
- `discrimination` A vector consisting of disrimination parameters for each item.
- `model` The string `"ltm"`, indicating this `Cat` object corresponds to a latent trait model.

Note

In case the Hessian matrix at convergence is not positive definite try to use `start.val = "random"`.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

References

Baker, Frank B. and Seock-Ho Kim. 2004. Item Response Theory: Parameter Estimation Techniques. New York: Marcel Dekker.

Rizopoulos, Dimitris. 2006. “ltm: An R Package for Latent Variable Modeling and Item Response Theory Analyses." Journal of Statistical Software 17(5):1-25.

See Also

Cat for information on all Cat slots and their default values

tpmCat for an alternative model fit to binary data

Examples

```
## Not run:
## Creating Cat object with raw data
data(npi)
ltm_cat1 <- ltmCat(npi, quadraturePoints = 100)

## Creating Cat object with fitted object of class tpm
ltm_fit <- grm(npi, control = list(GHk = 100)) ## from ltm package
class(ltm_fit)
ltm_cat2 <- ltmCat(ltm_fit)

## Note the two Cat objects are identical
identical(ltm_cat1, ltm_cat2)

## Note the slots that have changed from default values
ltm_cat1@model
ltm_cat1@difficulty
ltm_cat1@discrimination

## End(Not run)
```

---

makeTree *Make tree of possible question combinations*

Description

Creates a tree of possible questions asked and stores it as a list of lists (flat = FALSE) or a table of values (flat = TRUE).

Usage

```
makeTree(cat, flat = FALSE)
```

Arguments

cat	an object of class Cat
flat	a logical; TRUE returns tree as a table; FALSE returns tree as a list

Details

Function takes a Cat object and generate a tree of all possible question combinations, condtional on previous answers and current θ estimates. The tree is stored as a list of lists, iteratively generated by filling in a possible answer, calculating the next question via selectItem, filling in a possible answer for that question, and so forth. The length of the tree is dictated by the lengthThreshold slot within the Cat object.

Value

An object of class `list`

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`selectItem` for specific item selection and selection methods

`nfc` *Need For Cognition dataset from TAPS and AMT*

Description

Data of survey respondents' respond to 18 NFC questions, which is a reduced version of NFC. For each question, respondents could choose one of five response options: Agree strongly, agree somewhat, neither agree nor disagree, disagree somewhat, disagree strongly. In addition, missingness is optional. Sample variable indicates where the dataset come from: 1 = The American Panel Survey, 2 = Amazon's Mechanical Turk (fall 2014), 3 = Amazon's Mechanical Turk (spring 2015)

Usage

```
data("nfc")
```

Format

A data frame with 4043 observations on the following 20 variables.

`NFC1` I really enjoy a task that involves coming up with new solutions to problems

`NFC4` I would prefer a task that is intellectual, difficult, and important to one that is somewhat important but does not require much thought

`NFC10` Learning new ways to think doesnt excite me very much

`NFC12` I usually end up deliberating about issues even when they do not affect me personally

`NFC15` The idea of relying on thought to make my way to the top appeals to me

`NFC16` The notion of thinking abstractly is appealing to me

`NFC19` I only think as hard as I have to

`NFC21` I think tasks that require little thought once I've learned them

`NFC22` I prefer to think about small, daily projects to long-term ones

`NFC23` I would rather do something that requires little thought than something that is sure to challenge my thinking abilities

`NFC24` I find satisfaction in deliberating hard and for long hours

`NFC29` I like to have the responsibility of handing a situation that requires a lot of thinkings

`NFC31` I feel relief rather than satisfaction after completing a task that required a lot of mental effort

`NFC32` Thinking is not my idea of fun

`NFC33` I try to anticipate and avoid situations where there is likely a chance I will have to think in depth about something

`NFC39` I prefer my life to be filed with puzzles that I must solve

`NFC40` I would prefer complex to simple problems

`NFC43` Its enough for me that something gets the job done; I dont care how or why it works

`sample` a numeric vector (1=TAPS, 2=AMT(2014), 3=AMT(2015))

Examples

```
data(nfc)
## maybe str(nfc) ; plot(nfc) ...
```

---

npi *Narcissistic Personality Inventory data*

Description

Data from the test with the 40 paired items on one page. Respondents had to choose the one that fit them the best. Missingness is optional. Respondents' demographics were entered on the next page. In addition, they were also asked to affirm that their answers were acurate and suitable for research, those that did not (9%) are not included in this dataset.

Usage

```
data("npi")
```

Format

A data frame with 11243 observations on the following 44 variables.

`score` a numeric vector indicates the score for each question; see details

`Q1` 1=I have a natural talent for influencing people. 2=I am not good at influencing people

`Q2` 1=Modesty doesnt become me. 2=I am essentially a modest person

`Q3` 1=I would do almost anything on a dare. 2=I tend to be a fairly cautious person

`Q4` 1=When people compliment me I sometimes get embarrassed. 2=I know that I am good because everybody keeps telling me so.

`Q5` 1=The thought of ruling the world frightens the hell out of me. 2=If I ruled the world it would be a better place

`Q6` 1=I can usually talk my way out of anything. 2=I try to accept the consequences of my behavior

`Q7` 1=I prefer to blend in with the crowd. 2=I like to be the center of attention

`Q8` 1=I will be a success. 2=I am not too concerned about success

`Q9` 1=I am no better or worse than most people. 2=I think I am a special person

`Q10` 1=I am not sure if I would make a good leader. 2=I see myself as a good leader

`Q11` 1=I am assertive. 2=I wish I were more assertive

`Q12` 1=I like to have authority over other people. 2=I don't mind following orders

`Q13` 1=I find it easy to manipulate people. 2=I don't like it when I find myself manipulating people

`Q14` 1=I insist upon getting the respect that is due me. 2=I usually get the respect that I deserve

`Q15` 1=I don't particularly like to show off my body. 2=I like to show off my body

`Q16` 1=I can read people like a book. 2=People are sometimes hard to understand

`Q17` 1=If I feel competent I am willing to take responsibility for making decisions. 2=I like to take responsibility for making decisions

`Q18` 1=I just want to be reasonably happy. 2=I want to amount to something in the eyes of the world

`Q19` 1=My body is nothing special. 2=I like to look at my body

`Q20` 1=I try not to be a show off. 2=I will usually show off if I get the chance

`Q21` 1=I always know what I am doing. 2=Sometimes I am not sure of what I am doing

`Q22` 1=I sometimes depend on people to get things done. 2=I rarely depend on anyone else to get things done

`Q23` 1=Sometimes I tell good stories. 2=Everybody likes to hear my stories

`Q24` 1=I expect a great deal from other people. 2=I like to do things for other people

`Q25` 1=I will never be satisfied until I get all that I deserve. 2=I take my satisfactions as they come

`Q26` 1=Compliments embarrass me. 2=I like to be complimented

`Q27` 1=I have a strong will to power. 2=Power for its own sake doesn't interest me

`Q28` 1=I don't care about new fads and fashions. 2=I like to start new fads and fashions

`Q29` 1=I like to look at myself in the mirror. 2=I am not particularly interested in looking at myslef in the mirror

`Q30` 1=I really like to be the center of attention. 2=It makes me uncomfortable to be the center of attention

`Q31` 1=I can live my life in any way I want to. 2=People can't always live their lives in terms of what they want

`Q32` 1=Being an authority doesn't mean that much to me. 2=People alwyas seem to recognize my authority

`Q33` 1=I would prefer to be a leader. 2=It makes little difference to me whether I am a leader or not.

`Q34` 1=I am going to be a great person. 2=I hope I am going to be successful

`Q35` 1=People sometimes believe what I tell them. 2=I can make anybody believe anything I want them to

`Q36` 1=I am a born leader. 2=Leadership is a quality that takes a long time to develop

`Q37` 1=I wish somebody would someday write my biography. 2=I don't like people to pry into my life for any reason

`Q38` 1=I get upset when people don't notice how I look when I go out in public. 2=I don't mind blending into the crowd when I go out in public

`Q39` 1=I am more capable than other people. 2=There is a lot that I can learn from other people

`Q40` 1=I am much like everybody else. 2=I am an extraordinary person

`elapse` a numeric vector indicates between time submitted and time loaded of the questions; see details

`gender` a numeric vector indicates the gender of the respondents; see details

`age` a numeric vector indicates the age of the respondents; see details

Examples

```
data(npi)
## maybe str(npi) ; plot(npi) ...
```

obsInf *Observed Information*

Description

Calculates the observed information of the likelihood of a respondent's ability θ for a given `item`.

Usage

```
obsInf(catObj, theta, item)
```

Arguments

`catObj`	An object of class `Cat`
`theta`	A numeric or an integer indicating the value for θ
`item`	An integer indicating the index of the question item

Details

The observed information is equivalent to the negative second derivative of the log-likelihood evaluated at θ. This function should never be called when the respondent has answered no questions as the likelihood is not defined.

Value

The function `obsInf` returns a numeric value of the observed information of the likelihood, given θ, for a given question item.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`estimateTheta` for calculation of θ

`expectedObsInf` for further application of observed information

Examples

```
## Not run:
## Create Cat object, store example answers, and calculate observed information
for an ability parameter of 1 for item 10

## observed information for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
```

```
setAnswers(ltm_cat) <- c(1,0,1,0,1, rep(NA, 35))
obsInf(ltm_cat, theta = 1, item = 10)

## observed information for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
setAnswers(tpm_cat) <- c(1,0,1,0, rep(NA, 35))
obsInf(tpm_cat, theta = 1, item = 10)

## observed information for Cat object of the grm model
data(nfc)
grm_cat <- grmCat(nfc)
setAnswers(grm_cat) <- c(1,3,4,5, rep(NA, 13))
obsInf(grm_cat, theta = 1, item = 10)

## End(Not run)
```

---

polknow *Political Knowledge (circa 2012) of Mechanical Turk Respondents*

Description

Data from Amazon Mechanical Turkers responding to 64 political knowledge questions.

Usage

```
data(polknow)
```

Format

A data frame with 810 observations on the following 64 questions.

`Q1` How long is one term for the President of the United States?

Eight years; Six years; Four years; Two years

`Q2` The FDA is part of the national government primarily responsible for regulating

Food quality; The national parks; Electricity production and energy; Pollution and the environment

`Q3` Who is the Vice President of the United States?

Leon Panetta; William Daley; Hillary Clinton; Joe Biden

`Q4` The federal debt is

Much smaller than it was 20 years ago; The difference between imports and exports with foreign countries; The annual difference between spending and tax revenues; The accumulated borrowing of the federal government that has not been repaid

`Q5` How many times can an individual be elected President of the United States under current laws?

Any number of terms; Three times; Twice; Once

`Q6` What do we call the first ten amendments to the Constitution?

The Articles of Confederation; The inalienable right; The Bill of Rights; The Declaration of Independence

`Q7` Is the U.S. federal budget deficit, the amount by which the governments spending exceeds the amount of money it collects, now bigger, about the same, or smaller than it was during most of the 1990s?

Smaller; About the same; Bigger

`Q8` Who signs bills to become laws?

The President; The Vice President; The Chief Justice of the Supreme Court; The Secretary of State

`Q9` Which party is generally more supportive of creating a way for immigrants who are in the U.S. illegally to eventually become citizens?

The Republican Party; The Democratic Party

`Q10` In what month do we vote for the President?

November; October; February; January

`Q11` What are the two parts of the U.S. Congress?

The Senate and the Supreme Court; The House of Lords and the House of Commons; The House of Representative and the Supreme Court; The Senate and House of Representatives

`Q12` Which party is generally more supportive of restricting access to abortion?

The Republican Party; The Democratic Party

`Q13` Which of these political parties is considered most conservative?

Green Party; Republican Party; Democratic Party

`Q14` The NRA is an organization that advocates for

Clean elections; A cleaner environment; The rights of gun owners; Women's rights

`Q15` Compared to 30 years ago, is the difference in incomes between the top 20% of households and the bottom 20% of households now bigger, smaller, or the same?

The difference is now the same as 30 years ago; The difference is now smaller than 30 years ago; The difference is now bigger than 30 years ago

`Q16` The EPA is part of the national government primarily responsible for regulating

Food quality; The national parks; Electricity production and energy; Pollution and the environment

`Q17` Which party is generally more supportive of reducing the size of the defense budget?

The Republican Party; The Democratic Party

`Q18` Which party is generally more supportive of increasing taxes on higher income people to reduce the federal budget deficit?

The Republican Party; The Democratic Party

`Q19` Which party is generally more supportive of allowing drilling for oil in the Arctic National Wildlife Refuge?

The Republican Party; The Democratic Party

`Q20` Who is the Commander in Chief of the military?

The Attorney General; The President; The Secretary of Defense; The Vice President

`Q21` The First Amendment to the United States Constitution guarantees all of these rights EXCEPT

Right to remain silent; Right to the free exercise of religion; Right to free speech; Right to peaceably assemble

`Q22` Roe v Wade is a case decided by the Supreme Court that relates to

Executive power; Campaign finance; Birth control; Abortion

`Q23` Social Security is

Funded by the personal income tax; Operated by state government; The responsibility of the Department of Defense; The benefit program for senior citizens

Q24 What is Medicare?

A private, non-profit organization that runs free health clinic; A private health insurance plan sold to individuals in all 50 states; A program run by state governments to provide health care to poor people; A program run by the U.S. federal government to pay for old people's health care

Q25 How many senators are elected from each state?

It depends on the population of the state; Four; Two; One

Q26 How many votes are required in Congress to override a presidential veto

A simple majority of both houses of Congress; A simple majority of one house of Congress; A two-thirds majority of both houses of Congress; A two-thirds majority of one house of Congress

Q27 The Secretary of State

Serves a two-year term; Serves the state governments; Is nominated by the president; Heads the armed services

Q28 Near the end of an election campaign, a polls shows that an issue that no candidate has mentioned is of great concern to voters. What is most likely to happen?

Some candidates will drop out of the race; Candidates will start talking about the issue; Newspapers will not report the results of the poll; The election will be postponed

Q29 Liberals are generally said to

Support pro-life policies; Oppose all tax increases; Support military spending; Support government programs to give government assistance the needy

Q30 Which party is generally more supportive of reducing the size and scope of the federal government?

The Republican Party; The Democratic Party

Q31 The ability of a minority of senators to prevent a vote on a bill is known as

Suspension of the rules; Enrollment; A veto; A filibuster

Q32 conservatives are generally said to

Suppport pro-choice policies; Support tax cuts; Oppose military spending; Support government programs to give government assistance to the needy

Q33 Which of these regions of the country is generally considered to be most supportive of Republican candidates

Midwest; South; West Coast; New England

Q34 The presiding officer in the House of Representatives is

The Majority Leader; The Sergeant at Arms; The Vice President of the United States; The Speaker

Q35 Which of these countries is NOT a permanent member of the U.N. Security Council

United Kingdom; France; India; China

Q36 Which part has a majority of seats in the U.S. House of Representatives?

Neither; Democrats; Republicans

Q37 What state holds the first primary election in Presidential primaries?

Florida; Nevada; South Carolina; New Hampshire

Q38 Who is the Speaker of the House of Representatives?

Mitt Romney; Eric Holder; Harry Reid; John Boehner

Q39 Most cases are considered by the Supreme Court

In even-numbered years; At the request of the Congress; Upon order of the president; With the approval of at least four justices

`Q40` How many Justices typicaly serve on the U.S. Supreme Court

Eleven; Nine; Eight; Seven

`Q41` What job or political office is no held by Ben Bernanke?

None of these; Minority Whip of the U.S. House; Chief Justice of the United States Supreme Court; Majority leader of the U.S. Senate, Chairman of the Federal Reserve

`Q42` Whose responsibility is it to nominate judges to the Federal Courts

The state governors; The Supreme Court; Congress; President

`Q43` Who is the Chief Justice of the U.S. Supreme Court?

Larry Thompson; Anthony Kennedy; David Cole; John Roberts

`Q44` The U.S. Senate

Votes to confirm nominees to the U.S. Supreme court chosen by the House of Representatives; Plays no role in choosing the members of the U.S. Supreme Court; Chooses members of the U.S. Supreme Court; Votes to confirm nominees to the U.S. Supreme Court chosen by the President

`Q45` Which party has a majority of seats in the U.S. Senate

Neither; Democrats; Republicans

`Q46` Whic of the statees listed below has the greatest number of electoral college votes in the U.S. Presidential Elections?

Puerto Rico; Nevada; North Dakota; Washington, D.C.

`Q47` Citizens United v the FEC is a case decided by the Supreme Court that relates to

Executive power; Campaign finance; Birth control; Abortion

`Q48` For how many years is a United States Senator elected that is, how many years are there in one full term of office for a U.S. Senator?

None of these; Eight years; Six years; Four years; Two years

`Q49` Who is the Prime Minister of the United Kingdom?

Richard Branson; Tony Hayward; Nick Clegg; David Cameron

`Q50` The president of Afghanistan is named

Bashar al-Assad; Hosni Mubarak; Hamid Karzai; Nouri al-Maliki

`Q51` The House of Representatives has how many voting members?

Four hundrer and forty-one; Four hundred and thirty-five; Two hundred; One hundred

`Q52` The President of the Senate is

THe Majority Leader; The Sergeant at Arms; The Vice President of the United States; The senior senator of the majority party

`Q53` On which of the following federal programs is the most money spent each year?

Medicare; Education; Subsidies to farmers; Aid to foreign countries

`Q54` What do all constitutional governments have?

Separation of church and state; A bill of rights; A President as the head of government; Limits on political power

`Q55` One which of the following does the U.S. federal government spend the least money?

Social Security; National defense; Medicare; Foreign aid

`Q56` The head of the Department of Justice is

Kathleen Sebelius; Eric Holder; Timothy Geithner; Hillary Clinton

`Q57` The president may NOT

Declare war; Pardon criminals without justification; Appoint federal officials when Congress is in recess; Refuse to sign legislation passed by Congress

`Q58` Which of thse is NOT primarily the responsibility of the Federal government in Washington?
Interstate commerce; Negotiating treaties with foreign countries; Education; National defense

`Q59` Who is the current president of Mexico?
Vincente Fox; Hugo Chavez; Dilma Rousseff; Felipe Calderon

`Q60` Which of the following actions does the United States federal government commonly take to finance a budget deficit?
Expanding public-works projects; Borrowing from the public; Imposing import quotas; Printing more money

`Q61` Common Cause is an organization that advocates for
Women's rights; Clean elections; A cleaner environment; The right of gun owners

`Q62` The Byrd Rule is relevant
During the confirmation of cabinet members; For national party conventions; During Congressional debates over non-budgetary policies; For the Reconciliation process

`Q63` The Majority Leader of the House of Representative is
Nancy Pelosi; Kevin McCarthy; Eric Cantor; John Boehner

`Q64` On which of the following does the U.S. federal government spend the most money each year?
Education; Medicare; Interest on the national debt; National defense

Source

https://dataverse.harvard.edu/dataset.xhtml?persistentId=hdl:1902.1/19381

References

Jacob M. Montgomery and Joshua Cutler. 2013. "Computerized Adaptive Testing for Public Opinion Surveys." Political Analysis (Spring 2013) 21 (2): 172-192.

`posteriorKL` *Expected Kullback-Leibeler information, weighted by the posterior*

Description

Calculate the expected Kullback-Leibeler information, weighted by the posterior

Usage

```
posteriorKL(catObj, item)
```

Arguments

`catObj`	An object of class `Cat`
`item`	An integer indicating the index of the question item

Details

We will follow the same procedure as `expectedKL`, except we will weight the different potential values of θ_0 by the posterior.

Due to the conditional independence assumption, we only need to calculate the expected value for potential new items.

Value

A value indicating the posterior KL information for the desired item, given the current answer profile and ability estimate.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

`likelihoodKL` and/or `expectedKL` for alternative KL methods

Examples

```
## Not run:
## Prior calculation using Cat object of the ltm model
## specifying different distributions
data(npi)
cat <- ltmCat(npi)

## End(Not run)
```

---

prior *Evaluate the prior density distribution at position* x

---

Description

Calculates the density at x of either the normal, Student's t, or uniform distribution.

Usage

```
prior(x, dist, params)
```

Arguments

x	A numeric value at which to evaluate the prior
dist	A string indicating the distribution (slot priorName of Cat object)
params	A length two numeric vector indicating the parameters of the distribution (slot priorParams of Cat object)

Details

The dist argument needs to be either "UNIFORM", "NORMAL", or "STUDENT_T".

When dist is "NORMAL", the first element of params is the mean, the second element is the standard deviation.

When dist is "STUDENT_T", the first element of params is the non-centrality parameters and the second is degrees of freedom.

When dist is "UNIFORM", the elements of params are the lower and upper bounds, of the interval, respectively. Note that the "UNIFORM" is only applicable for the "EAP" estimation method.

Value

The function prior returns a numeric consisting of prior value, $\pi(x)$, given the value x.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

This function uses Boost C++ source libraries for the uniform and Student's t distributions and calls dnorm4 written in C which is identical to that of dnorm in R.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

Cat for information on priorName and priorParams slots

Examples

```
## Not run:
## Prior calculation using Cat object of the ltm model
## specifying different distributions
data(npi)
cat <- ltmCat(npi)

cat@priorName <- "NORMAL"
cat@priorParams <- c(0, 1) ## Parameters are mean and standard deviation
prior(x = 1, cat@priorName, cat@priorParams)

cat@priorName <- "STUDENT_T"
cat@priorParams <- c(1, 3) ## Parameters are non-centrality param and degrees of freedom
prior(x = 1, cat@priorName, cat@priorParams)

cat@priorName <- "UNIFORM"
cat@priorParams <- c(-1, 1) ## Parameters are lower bound and upper bound of interval
prior(x = 1, cat@priorName, cat@priorParams)

## End(Not run)
```

`probability` *Probability of Responses to a Question Item or the Left-Cumulative Probability of Responses*

Description

Calculates the probability of specific responses or the left-cumulative probability of responses to `item` conditioned on a respondent's ability (θ).

Usage

```
probability(catObj, theta, item)
```

Arguments

`catObj`	An object of class `Cat`
`theta`	A numeric or an integer indicating the value for θ_j
`item`	An integer indicating the index of the question item

Details

For the `ltm` model, the probability of non-zero response for respondent j on item i is

$$Pr(y_{ij} = 1|\theta_j) = \frac{\exp(a_i + b_i\theta_j)}{1 + \exp(a_i + b_i\theta_j)}$$

where θ_j is respondent j 's position on the latent scale of interest, a_i is item i 's discrimination parameter, and b_i is item i 's difficulty parameter.

For the `tpm` model, the probability of non-zero response for respondent j on item i is

$$Pr(y_{ij} = 1|\theta_j) = c_i + (1 - c_i)\frac{\exp(a_i + b_i\theta_j)}{1 + \exp(a_i + b_i\theta_j)}$$

where θ_j is respondent j 's position on the latent scale of interest, a_i is item i 's discrimination parameter, b_i is item i 's difficulty parameter, and c_i is item i 's guessing parameter.

For the `grm` model, the probability of a response in category k **or lower** for respondent j on item i is

$$Pr(y_{ij} < k|\theta_j) = \frac{\exp(\alpha_{ik} - \beta_i\theta_{ij})}{1 + \exp(\alpha_{ik} - \beta_i\theta_{ij})}$$

where θ_j is respondent j 's position on the latent scale of interest, $\alpha_i k$ the k-th element of item i 's difficulty parameter, β_i is discrimination parameter vector for item i. Notice the inequality on the left side and the absence of guessing parameters.

For the `gpcm` model, the probability of a response in category k for respondent j on item i is

$$Pr(y_{ij} = k|\theta_j) = \frac{\exp(\sum_{t=1}^{k} \alpha_i[\theta_j - (\beta_i - \tau_{it})])}{\sum_{r=1}^{K_i} \exp(\sum_{t=1}^{r} \alpha_i[\theta_j - (\beta_i - \tau_{it}))}$$

where θ_j is respondent j 's position on the latent scale of interest, α_i is the discrimination parameter for item i, β_i is the difficulty parameter for item i, and τ_{it} is the category t threshold parameter for item i, with $k = 1, ..., K_i$ response options for item i. For identification purposes $\tau_{i0} = 0$ and $\sum_{t=1}^{1} \alpha_i[\theta_j - (\beta_i - \tau_{it})] = 0$.

Value

When the argument `catObj` is an `ltm` model, the function `probabilty` returns a numeric vector of length one representing the probabilty of observing a non-zero response.

When the argument `catObj` is an `tpm` model, the function `probabilty` returns a numeric vector of length one representing the probabilty of observing a non-zero response.

When the argument `catObj` is a `grm` model, the function `probabilty` returns a numeric vector of length k+1, where k is the number of possible responses. The first element will always be zero and the (k+1)th element will always be one. The middle elements are the cumulative probability of observing response k or lower.

When the argument `catObj` is a `gpcm` model, the function `probabilty` returns a numeric vector of length k, where k is the number of possible responses. Each number represents the probability of observing response k.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

References

Baker, Frank B. and Seock-Ho Kim. 2004. Item Response Theory: Parameter Estimation Techniques. New York: Marcel Dekker.

Choi, Seung W. and Richard J. Swartz. 2009. “Comparison of CAT Item Selection Criteria for Polytomous Items." Applied Psychological Measurement 33(6):419-440.

Muraki, Eiji. 1992. “A generalized partial credit model: Application of an EM algorithm." ETS Research Report Series 1992(1):1-30.

van der Linden, Wim J. 1998. “Bayesian Item Selection Criteria for Adaptive Testing." Psychometrika 63(2):201-216.

See Also

`Cat` for information on the item parameters: discrimination, difficulty, and guessing.

Examples

```
## Not run:
## Probability for Cat object of the ltm model
data(npi)
ltm_cat <- ltmCat(npi)
probability(ltm_cat, theta = 1, item = 1)

## Probability for Cat object of the tpm model
data(polknow)
tpm_cat <- tpmCat(polknow)
probability(tpm_cat, theta = 1, item = 1)

## Probability for Cat object of the grm model
```

```
data(nfc)
grm_cat <- grmCat(nfc)
probability(grm_cat, theta = 1, item = 1)

## End(Not run)
```

`selectItem` *Select the next item in the question set*

Description

Selects the next item in the question set based on the specified method

Usage

```
selectItem(catObj)
```

Arguments

`catObj`	An object of class `Cat`

Details

The EPV method:

This function takes in a `Cat` object from `R` and constructs the `C++` representation. It then calculates the expected posterior variance for each unanswered item.

The function returns a list with the following two elements: `all.estimates`: A data fame with two columns. The first column ("questions") should be the index of the question items and the second column ("EPV") of the expected posterior variance for that corresponding item. There are as many rows in this data frame as there are unsanswered questions in the battery. `next.item`: A numeric vector with the index of the item with the lowest EPV value.

See `expectedPV` for mathematical details.

The MFI method:

This function takes a `Cat` object and calculates Fisher's information for each unanswered item. It then finds the one item that maximizes Fisher's information, based on the respondent's position on the latent trait from the answered items.

The function returns a list with the following two elements: `all.estimates`: A data fame with two columns. The first column ("questions") should be the index of the question items and the second column of the expected posterior variance for that corresponding item. There are as many rows in this data frame as there are unsanswered questions in the battery. `next.item`: A numeric vector with the index of the item with the highest Fisher's information value.

See `fisherInf` for mathematical details.

The MLWI method:

This function calculates the likelihood for each value of X at the input value of θ. Evaluates the integral over a measure of the plausibility of possible values of θ by weighting Fisher's information with the likelihood function and selecting the next question according to:

The MPWI method:

This function calculates the likelihood for each value of X at the input value of θ. Evaluates the integral over a measure of the plausibility of possible values of θ by weighting Fisher's information with the likelihood function and selecting the next question according to:

The MEI method:

This function estimates the expected observed information for a respondent<e2><80><99>s estimated position on the latent trait on an unanswered item based on the respondent<e2><80><99>s position on the latent trait calculated from answered items.

The output should be a single numeric value.

Binary details:

Categorical details:

The KL method:

This procedure chooses the next item with the largest KL value.

See `expectedKL`, `likelihoodKL`, and/or `posteriorKL` for mathematical details.

The MFII method:

This approach chooses items based on the Fisher's information in an interval near the current estimate $\hat{\theta}$.

$$FII_i = \int_{\hat{\theta}-\delta}^{\hat{\theta}+\delta} I_i(\theta_0) d\theta_0$$

where

$$\delta = z(I(\hat{\theta}))^{-1/2}$$

, $I(\hat{\theta})$ is the test information for respondent j evaluated at $\hat{\theta}$,

$$I_i(\cdot)$$

is the Fisher's information for item i, and z is a user specified z-value.

The random method:

This routine serves as a baseline for comparison. The routine simply selects an unanswered question at random.

Value

It returns a list with two elements: (1) A dataframe containing a column with the indexes of unasked questions and a column with the values (calculated by the specified selection method) for those items, and (2) a numeric containing the index of the question that should be asked next.

Note

This function is to allow users to access the internal functions of the package. During item selection, all calculations are done in compiled C++ code.

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

See Also

estimateTheta for calculation of θ; obsInf for observed information calculation; fisherTestInfo for Fisher's information calculation; expectedKL for expected Kullback-Leibeler calculation; likelihoodKL for likelihood Kullback-Leibeler calculation; posteriorKL for posterior Kullback-Leibeler calculation;

Examples

```
## Not run:
## Prior calculation using Cat object of the ltm model
## specifying different distributions
data(npi)
cat <- ltmCat(npi)

## End(Not run)
```

---

storeAnswer *Computerized Adaptive Testing Survey Store Answer Function*

Description

This function updates the object of class Cat by storing the answer to item k into questions data frame.

Usage

```
storeAnswer(cat, item, answer)
```

Arguments

cat	An object of class Cat
item	The question for which to estimate the expected posterior variance for a respondent with a latent trait estimate of theta.hat. This should be the name of a row in the "questions" data-frame in the "questions" slot of a Cat object.
answer	The answer to the item k to be stored

Value

An updated object of class Cat containing the answers to k items

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

tpm-class *Computerized Adaptive Testing Birnbaum's Three Parameter Model*

Description

This function fits Birnbaum's Three Parameter model for binary data and populates the fitted values for discimination, difficulty, and guessing parameters to an object of class Cat.

Arguments

data	A data.frame of manifest variables or an object of class tpm.
quadraturePoints	A numeric to be passed into the tpm function indicating the number of Gauss-Hermite quadrature points. Only applicable when data is a data.frame. Default value is 21.
...	arguments to be passed to methods. For more details about the arguments, see tpm in the ltm package.

Details

The data argument of the function tpmCat is either a data.frame or an object of class tpm from the ltm package. If it is a data.frame each row represents a respondent and each column represents a question item. If it is an object of the class tpm, it is output from the tpm function in the ltm package.

The quadraturePoints argument of the function tpmCat is used only when the data argument is of class data.frame. quadraturePoints is then passed to the tpm function from the ltm package when fitting Birnbaum's Three Parameter model to the data and is used when approximating the value of integrals.

Value

The function tpmCat returns an object of class Cat with changes to the following slots:

- difficulty A vector consisting of difficulty parameters for each item.
- discrimination A vector consisting of disrimination parameters for each item.
- model The string "tpm", indicating this Cat object corresponds to Birnbaum's Three Parameter model.

Note

In case the Hessian matrix at convergence is not positive definite try to use start.val = "random".

Author(s)

Haley Acevedo, Ryden Butler, Josh W. Cutler, Matt Malis, Jacob M. Montgomery, Tom Wilkinson, Erin Rossiter, Min Hee Seo, Alex Weil

References

Baker, Frank B. and Seock-Ho Kim. 2004. Item Response Theory: Parameter Estimation Techniques. New York: Marcel Dekker.

Birnbaum, Allan. 1968. Some Latent Trait Models and their Use in Inferring an Examinee's Ability. In F. M. Lord and M. R. Novick (Eds.), Statistical Theories of Mental Test Scores, 397-479. Reading, MA: Addison-Wesley.

Rizopoulos, Dimitris. 2006. "ltm: An R Package for Latent Variable Modeling and Item Response Theory Analyses." Journal of Statistical Software 17(5):1-25.

See Also

`Cat` for information on all `Cat` slots and their default values

`ltmCat` for an alternative model fit to binary data

Examples

```
## Not run:
## Creating Cat object with raw data
data(polknow)
tpm_cat1 <- tpmCat(polknow, quadraturePoints = 100)

## Creating Cat object with fitted object of class tpm
tpm_fit <- grm(polknow, control = list(GHk = 100)) ## from ltm package
class(tpm_fit)
tpm_cat2 <- tpmCat(tpm_fit)

## Note the two Cat objects are identical
identical(tpm_cat1, tpm_cat2)

## Note the slots that have changed from default values
tpm_cat1@model
tpm_cat1@difficulty
tpm_cat1@discrimination

## End(Not run)
```

Index

∗Topic **datasets**
nfc, 24
npi, 25
polknow, 28

Cat, *6, 7*, *10*, *17–19*, *23*, *34*, *36*, *41*
Cat-class, 2
checkStopRules, *4*, 4

d2LL, 5
dLL, *6*, 6

estimateSE, 8
estimateTheta, *4*, *8*, 9, *12, 13*, *27*, *39*
expectedKL, 11, *20*, *33*, *39*
expectedObsInf, 12, *27*
expectedPV, 13

fisherInf, 14, *15*
fisherTestInfo, *14*, 15, *39*

gpcm-class, 16
gpcmCat, *18*
grm-class, 17
grmCat, *17*

initialize,Cat-method *(*Cat-class*)*, 2

likelihood, 18
likelihoodKL, *11*, 20, *33*, *39*
lookAhead, 21
ltm-class, 22
ltmCat, *41*

makeTree, 23

nfc, 24
npi, 25

obsInf, *12*, *14*, 27, *39*

polknow, 28
posteriorKL, *11*, *20*, 32, *39*
prior, *6, 7*, 33
probability, *13*, *19*, 35

selectItem, *4*, *21*, *24*, 37
storeAnswer, 39

tpm-class, 40
tpmCat, *23*