Package ‘infutil’

February 20, 2015

Version 1.0
Date 2013-03-28
Title Information Utility
Author Kristian E. Markon
Maintainer Kristian E. Markon <kristian-markon@uiowa.edu>
Depends ltm
Enhances MASS
Description Calculation of information utility (i.e., Lindley
information) quantities for item response models.
License GPL (>= 2)
URL http://www.r-project.org,
http://www.psychology.uiowa.edu/faculty/markon
NeedsCompilation no
Repository CRAN
Date/Publication 2013-04-02 07:30:11

R topics documented:

infutil-package ........................................ 2
iota ..................................................... 2
iota.c .................................................. 4
iota.l .................................................. 6
iota.u .................................................. 7
Jeffreys ............................................... 8
nmru .................................................. 10
rJeffreys ............................................ 11

Index 13
Description

This package contains functions for calculating information utility (i.e., Lindley information) quantities for item response models. It requires the ltm package.

Functions are available for calculating the Jeffreys prior for IRT models, calculating the information utility for individual response vectors, calculating the expected information utility for a set of response vectors, and estimating the criterion information utility.

Details

Jeffreys calculates the Jeffreys prior for IRT models, and optionally returns the normalizing constant used in calculating the prior.

iota returns the expected information utility for a set of response vectors, together with the individual information utilities, and optionally tests for a change from the prior to posterior probability of a latent trait value.

iota_c returns an estimate of the criterion information utility and its standard error.

rJeffreys randomly generates latent trait values from a Jeffreys prior distribution.

Author(s)

Kristian E. Markon

References


Description

This function calculates Lindley information (i.e., information utility) quantities for item response patterns, including the Lindley information for individual response patterns and the expected information for a set of response patterns.

Usage

iota(ltm.obj, logL.fun, fscore.obj = NULL, data = NULL, prior = NULL, theta0 = NULL, range.int = c(-Inf, Inf), ...)

iota
Arguments

ltm.obj  An object representing an IRT model of a set of items, from the ltm package (e.g., using the ltm, grm, or gpcm functions).

logL.fun  A function returning the log-likelihood for a response vector (see details).

fscore.obj  A factor.scores object from the ltm package.

data  A matrix of response patterns.

prior  A prior density function, taking a quantile and returning its density.

theta0  Optional. A latent trait value, vector of values, ‘"max.prior"’ (see details), or ‘NULL’. ‘theta0 = NULL’ by default.

range.int  The range to integrate over in calculating the marginal likelihood and Lindley information.

range.theta  The range of the latent variable to search over if ‘theta0="max.prior"’ is specified.

Details

This function will calculate Lindley information values for each response pattern in a set, and return the expected Lindley information for the set as a whole. It will also optionally return the log-likelihood value for a hypothesized latent trait value of interest, together with the significance of the change from prior to posterior probability of that trait value.

A set of response patterns must be supplied, using either a factor.scores object or a data matrix. A prior density function must also be supplied. Additionally, either ‘ltm.obj’ or ‘logL.fun’ must be supplied. If a log-likelihood function is supplied directly, it must take three arguments: ‘z’, a vector of latent trait values, ‘dat’, a data matrix, and ‘i’, a row index. ‘logL.fun’ must return the vector of log-likelihoods for the vector of latent trait values, for a single response pattern (i.e., the function must return the vector of log-likelihoods for the trait vector ‘z’, for the response pattern in row ‘i’ of data matrix ‘dat’).

iota will also optionally return a p-value for the null hypothesis of no change from the prior to posterior probability of a specific latent trait value. ‘theta0’ specifies the latent trait value. If ‘theta0’ is a scalar, it is assumed to be the latent trait value of interest. If ‘theta0’ is a vector, it must be equal in length to the number of unique response patterns in ‘fscore.obj’ or ‘data’, where each value in the vector is assumed to correspond to each response pattern. Finally, the latent trait with the greatest prior density (over the range ‘range.theta’) can be specified with ‘theta0="max prior"’. By default, ‘theta0=NULL’, specifying that no log-likelihoods or p-values should be returned.

Note that currently, ‘range.int’ must be within [-10, 10] for grm and gpcm objects. If a grm or gpcm object is supplied and ‘range.int’ is outside this range, the range will be reset to [-10, 10].

Value

A list with the following elements:

I  The expected Lindley information over the set of response patterns.
I.x  The vector of Lindley information values for each response pattern.
p.x  The marginal probabilities for each response pattern.
Additionally, if ‘theta0’ is not ‘NULL’:

- \( \logL0 \) The vector of log-likelihoods for the latent trait values specified by ‘theta0’.
- \( \logNL0.pval \) The p-values for the null hypothesis test of no change from prior to posterior probability of the latent trait values specified by ‘theta0’.

**Author(s)**

Kristian E. Markon

**References**


**See Also**

`Jeffreys`, `ltm`, `grm`, `gpcm`, `factor.scores`

**Examples**

```r
ltm.lsat <- ltm(LSAT~z1, IRT=FALSE)
Nlsat = nrow(unique(LSAT))

iota(ltm.lsat, data=LSAT, prior=Jeffreys(ltm.lsat))
iota(ltm.lsat, data=LSAT, prior=Jeffreys(ltm.lsat), theta0=c("max.prior"))
iota(ltm.lsat, data=LSAT, prior=Jeffreys(ltm.lsat), theta0=0)
iota(ltm.lsat, data=LSAT, prior=Jeffreys(ltm.lsat), theta0=rnorm(Nu.lsat))
```

**Description**

This function provides Monte Carlo estimates of the criterion information utility.

**Usage**

```r
iota.c(ltm.obj, M = NULL, prior = NULL, logL.fun = NULL, rirm = NULL, range.int = c(-Inf, Inf))
```

**Arguments**

- `ltm.obj` An object representing an IRT model of a set of items, from the `ltm` package (e.g., using the `ltm`, `grm`, or `gpcm` functions).
- `M` The number of Monte Carlo replications.
- `prior` A reference prior density function, taking a quantile and returning its density under the reference prior.
Details

This function will estimate the criterion information utility for a set of items under an IRT model, using Monte Carlo simulation. It will also return the standard error of the estimate, as well as the individual information utilities and marginal likelihoods for each simulated response pattern.

Either `ltm.obj` or `logl.fun` must be supplied. If a log-likelihood function is supplied directly, it must take three arguments: `z`, a vector of latent trait values, `dat`, a data matrix, and `i`, a row index. `logl.fun` must return the vector of log-likelihoods for the vector of latent trait values, for a single response pattern (i.e., the function must return the vector of log-likelihoods for the trait vector `z`, for the response pattern in row `i` of data matrix `dat`).

The number of Monte Carlo replications `M` must be supplied.

If an `ltm.obj` object is not specified, the reference prior in the form of a density function must also be supplied. In that situation, a function that randomly generates a matrix of response patterns must also be supplied to `rirm`. The `rirm` function must take as its first argument a number of response patterns `M`, and as the second argument a vector of latent trait values `m.theta`, returning a matrix of response patterns.

If an `ltm.obj` object is specified, the reference prior is assumed to be the Jeffreys prior, and `rirm` is assumed to be the IRT model used to create `ltm.obj`.

Note that currently, `range.int` must be within [-10, 10] for grm and gpcm objects. If a grm or gpcm object is supplied and `range.int` is outside this range, the range will be reset to [-10, 10].

Value

A list with the following components:

- `I`: The estimated criterion information utility.
- `se.I`: The standard error of the estimated criterion information utility.
- `I.x`: The vector of Lindley information values for each simulated response pattern.
- `p.x`: The marginal probabilities for each simulated response pattern.

Author(s)

Kristian E. Markon

References

Markon, K. E. (2013). Information utility: Quantifying the total psychometric information provided by a measure. Psychological Methods, 18, 15-35. doi: 10.1037/a0030638..

See Also

iota, rJeffreys, Jeffreys, ltm, grm, gpcm
Examples

```r
ltm.lsat <- ltm(LSAT~1, IRT=FALSE)
iota.c.lsat <- iota.c(ltm.lsat, M=100) # typically more Monte Carlo replications would be used
iota.c.lsat
```

---

**iota.l**

*Lindley Information (i.e., Information Utility) Lower Bound*

Description

This function calculates the lower bound to the Lindley information (i.e., information utility) given a Jeffreys prior normalizing constant or ltm object.

Usage

```r
iota.l(x)
```

Arguments

- `x`  
  Either an `ltm` object or a Jeffreys prior normalizing constant. `x` can be a scalar or vector.

Details

This function calculates the lower bound to the Lindley information (i.e., information utility) given a Jeffreys prior normalizing constant or `ltm` object.

- `x` can be an object from the `ltm` package describing a set of IRT parameters (e.g., as returned by `ltm`, `grm`, or `gpcm`).
- If `x` is a vector, the function will return a vector of lower bounds.

Value

A lower bound to the Lindley information, or a vector of lower bounds for different sets of items.

Author(s)

Kristian E. Markon

References


See Also

`Jeffreys, iota.c, nmru, iota, ltm, grm, gpcm`
Examples

```r
ltm.lsat <- ltm(LSAT~z1, IRT=FALSE)
iota.l(ltm.lsat)
iota.l(Jeffreys(ltm.lsat, return="nc"))
```

### Description

This function calculates the upper bound to the Lindley information (i.e., information utility) given a prior.

### Usage

```r
iota.u(prior, range.int = c(-Inf, Inf))
```

### Arguments

- `prior`: a prior in the form of a function.
- `range.int`: the range of integration; defaults to c(-Inf, Inf).

### Details

This function calculates the upper bound to the Lindley information (i.e., information utility) given a prior. It is identical to the entropy of the prior.

Note that the range of integration may have to be changed; the integration function sometimes throws an error with infinite limits.

### Value

The upper bound to the Lindley information; the entropy of the prior.

### Author(s)

Kristian E. Markon

### References

Markon, K. E. (2013). Information utility: Quantifying the total psychometric information provided by a measure. Psychological Methods, 18, 15-35. doi: 10.1037/a0030638..

### See Also

`Jeffreys`, `nmru`, `iota.l`
Jeffreys Prior and Normalizing Constant

Examples

```r
ltm.lsat <- ltm(LSAT~1, IRT=FALSE)
iota.u(Jeffreys(ltm.lsat))
iota.u(Jeffreys(ltm.lsat), range.int=c(-20,20))
```

Description

This function returns the Jeffreys prior for an IRT model in the form of a density function, and optionally the normalizing constant of that prior.

Usage

```
Jeffreys(ltm.obj = NULL, inf.mat = NULL, inf.func = NULL, return = "prior", spl.method = "natural", range.int)
```

Arguments

- `ltm.obj`: An object representing an IRT model of a set of items, from the ltm package (e.g., using the `ltm`, `grm`, or `gpcm` functions).
- `inf.mat`: A two-column matrix representing the information function evaluated at a set of latent trait values, with the first column being the latent trait values and the second column the information at each value (e.g., as returned by the `ltm` plot function).
- `inf.func`: An information function; a function taking a latent trait value and returning the information at that value.
- `return`: The value(s) to be returned. ‘"prior"’ returns the prior as a function; ‘"nc"’ returns the normalizing constant; ‘"both"’ returns both as a list.
- `spl.method`: A `splinefun` method to use in constructing an information function; only used if ‘`inf.mat`’ is supplied.
- `range.int`: The range to integrate over in calculating the normalizing constant.

Details

Only one of ‘`ltm.obj`’, ‘`inf.mat`’, or ‘`inf.function`’ should be supplied. An ‘`inf.mat`’ matrix can be obtained using the plot function in the ltm package, with the options ‘type="IIC"’, ‘item=0’, and ‘plot=F’. As the information function must be interpolated when ‘`inf.mat`’ is supplied (using `splinefun`), supplying an `ltm` object or information function directly will generally be more accurate. If ‘`inf.mat`’ is supplied, the information should be evaluated at a large number of points over a wide range, to maintain accuracy.

Note that currently, ‘`range.int`’ must be within (-10, 10) for `grm` and `gpcm` objects. If a `grm` or `gpcm` object is supplied and ‘`range.int`’ is outside this range, the range will be reset.
**Value**

If `return = "prior"`, the default, a function taking a latent trait value and returning the Jeffreys prior density at that point. If `return = "nc"`, the normalizing constant used to calculate the Jeffreys prior and also the lower bound to the Lindley information. If `return = "both"`, a list having the following structure:

```plaintext
prior          The Jeffreys prior density as a function.
nc             The normalizing constant.
```

**Author(s)**

Kristian E. Markon

**References**


**See Also**

- `rJeffreys`, which randomly generates latent trait values distributed according to a Jeffreys prior density, and `iota` and `iota.c`, which calculate Lindley information quantities, possibly using the Jeffreys prior. Also see `splinefun`, `ltm`, `grm`, and `gpcm`.

**Examples**

```r
# using an ltm object
ltm.lsat <- ltm(LSAT~z1, IRT=FALSE)

jp.lsat <- Jeffreys(ltm.lsat)
jp.lsat(0)

# using inf.mat
inf.lsat <- plot(ltm.lsat, type="IIC", item=0, plot=FALSE, z=seq(-15, 15, length=10000))

jp.lsat <- Jeffreys(inf.mat=inf.lsat)
jp.lsat(0)

# returning normalizing constant
Jeffreys(ltm.lsat, return="nc")
Jeffreys(inf.mat=inf.lsat, return="nc")
```
Description

This function calculates the normalized minimum reduction in uncertainty (NMRU) for a test, given an item response model.

Usage

nmru(ltm.obj, range.int = c(-Inf, Inf))

Arguments

- `ltm.obj`: An object representing an IRT model of a set of items, from the ltm package (e.g., using the `ltm`, `grm`, or `gpcm` functions).
- `range.int`: the range of integration; defaults to c(-Inf, Inf).

Details

This function calculates the normalized minimum reduction in uncertainty (NMRU) for a test, given an item response model. It is equal to the information utility lower bound divided by the upper bound (which are returned by `iota.l` and `iota.u`, respectively. The NMRU is a normalized mutual information index, where the mutual information between the data and trait estimate is normalized by the entropy of the prior.

Note that the range of integration may have to be changed; the integration function sometimes throws an error with infinite limits.

Value

- `val`: the NMRU value.
- `iota.l`: the information utility lower bound.
- `iota.u`: the information utility upper bound.

Author(s)

Kristian E. Markon

References

Markon, K. E. (2013). Information utility: Quantifying the total psychometric information provided by a measure. Psychological Methods, 18, 15-35. doi: 10.1037/a0030638..

See Also

`iota.l`, `iota.u`, `ltm`, `grm`, `gpcm`
Examples

# five items
ltm.lsat <- ltm(LSAT~z1, IRT=FALSE)
nmru(ltm.lsat)
nmru(ltm.lsat, range.int=c(-20,20))

# seven items
ltm.lsat <- ltm(cbind(LSAT, LSAT[,1:2])~z1, IRT=FALSE)
nmru(ltm.lsat)

# ten items
ltm.lsat <- ltm(cbind(LSAT, LSAT)~z1, IRT=FALSE)
nmru(ltm.lsat)

rJeffreys(n, prior, range.int = c(-Inf, Inf))

Arguments

n The number of values to be generated.
prior A prior density in the form of a function, such as that returned by Jeffreys.
range.int The integration range used in generating random deviates.

Details

This function generates random values distributed according to a Jeffreys prior (e.g., as used to estimate the criterion information utility) using the generalized inverse transformation of random uniform (0,1) values.

The prior must be specified as a function taking a quantile and returning a density, such as is returned by Jeffreys (note that ‘prior’ could be any density function, not just a Jeffreys prior, in which case it would return random deviates distributed according to that density).
Value
A vector of values distributed according to the density specified by the ‘prior’ function.

Author(s)
Kristian E. Markon

References
Markon, K. E. (2013). Information utility: Quantifying the total psychometric information provided by a measure. Psychological Methods, 18, 15-35. doi: 10.1037/a0030638..

See Also
Jeffreys, which can be used to create a prior density function; also, iota.c, which uses rJeffreys to estimate the criterion information utility.

Examples
ltm.lsat <- ltm(LSAT~z1, IRT=FALSE)
rJeffreys(100, Jeffreys(ltm.lsat))
Index

*Topic distributions
   infutil-package, 2
   Jeffreys, 8
   rJeffreys, 11
*Topic models
   infutil-package, 2
   iota, 2
   iota.c, 4
   iota.l, 6
   iota.u, 7
   Jeffreys, 8
   nmru, 10
   rJeffreys, 11

factor.scores, 4

gpcm, 4–6, 9, 10
grm, 4–6, 9, 10

infutil-package, 2
iota, 2, 5, 6, 9
iota.c, 4, 6, 9
iota.l, 6, 7, 10
iota.u, 7, 10

Jeffreys, 4–7, 8

ltm, 4–6, 9, 10

nmru, 6, 7, 10

rJeffreys, 5, 9, 11

splinefun, 9