Package ‘cacIRT’

February 15, 2013

Type Package

Title Classification accuracy and consistency under Item Response Theory

Version 1.1

Date 2012-01-26

Author Quinn Lathrop

Maintainer <qlathrop@nd.edu>

Description Computes classification accuracy and consistency under Item Response Theory by the approach proposed by Lee, Hanson & Brennen (2002) and Lee (2010) or the approach proposed by Rudner (2001, 2005). For dichotomous IRT models.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2012-01-27 20:03:22

NeedsCompilation no

R topics documented:

cacIRT-package ................................................................. 2
class.Lee ................................................................. 3
class.Rud ................................................................. 4
recursive.raw ............................................................. 6
TTable.F ................................................................. 7
Useful IRT Functions ....................................................... 8

Index 10
Description

Computes classification accuracy and consistency under Item Response Theory by the approach proposed by Lee, Hanson & Brennen (2002) and Lee (2010) or the approach proposed by Rudner (2001, 2005).

Details

- **Package**: cacIRT
- **Type**: Package
- **Version**: 1.1
- **Date**: 2011-01-26
- **License**: GPL (>= 2)
- **LazyLoad**: yes

This package computes classification accuracy and consistency with two recent approaches proposed by Lee, Hanson & Brennen (2002) and Lee (2010) or by Rudner (2001, 2005), for dichotomous IRT models. The two functions `class.Lee` and `class.Rud` are the wrapper functions for the respective approaches. They can accept a range of inputs: ability estimates, quadrature points, or response data matrix and item parameters. Marginal indices are computed with either the D or P method (see Lee (2010)).

Author(s)

Quinn Lathrop
Maintainer: <qlathrop @ nd.edu>

References


class.Lee

Computes classification accuracy and consistency with Lee's approach.

Description

Computes classification accuracy and consistency with Lee’s approach. Basically, the probability of each possible total score conditional on ability is found with a recursive algorithm (see recursive.raw). Those probabilities are grouped according to the cut scores and used to estimate the indices. See references for details.

Usage

```r
class.Lee(cutscore, ip, ability = NULL, rdm = NULL, quadrature = NULL)
Lee.D(cutscore, quadrature, ip)
Lee.P(cutscore, theta, ip)
```

Arguments

- **cutscore**: A scalar or vector of cut scores on the True Score scale. If you have cut scores on the theta scale, you can transform them with `irf` (See example for `irf`). Should not include 0 or the max total score, the function will include them.
- **ip**: Matrix of item parameters, columns are discrimination, difficulty, guessing. For 1PL and 2PL, still give a Jx3 matrix, with ip[,1] = 1 and ip[,3] = 0 for example.
- **ability, theta**: Ability estimates for each subject.
- **rdm**: The response data matrix with rows as subjects and columns as items.
- **quadrature**: A list containing 1) The quadrature points and 2) Their corresponding weights.

Details

Must give only one ability, rdm, or quadrature. If ability is given, those scores are used for the P method. If rdm is given, ability is estimated with MLE (perfect response patterns given a -4 or 4) and used for the P method. If quadrature, the D method is used. `class.Lee` calls `Lee.D` or `Lee.P`.

Value

- **Marginal**: A matrix with two columns of marginal accuracy and consistency per cut score and/or simultaneous.
- **Conditional**: A matrix of conditional accuracy and conditional consistency returned in the order of subjects if ability or rdm is given.

Note

To expand the code to other IRT models or to estimate classification indices for weighted total score, the code within `Lee.P` or `Lee.D` needs to be modified. Also the recursive algorithm `recursive.raw()` will need to be adapted for polytomous IRT models.
Author(s)

Quinn Lathrop

References


Examples

```r
##from rdm, item parameters denote 4 item 1PL test, cut score at x=2
##only return marginal indices
params<-matrix(c(1,1,1,1,-2,1,0,0,0,0,0,0,4,3),4,3)
rdm<-sim(params, rnorm(100))
class.Lee(2, params, rdm = rdm)$Marginal

##or from 40 quadrature points and weights, 2 cut scores
quad <- normal.qu(40)
class.Lee(c(2,3), params, quadrature = quad)$Marginal
```

class.Rud | Computes classification accuracy and consistency with Rudner’s approach.

Description

Computes classification accuracy and consistency with Rudner’s approach. For each examinee, a normal distribution is created with mean at the ability estimate and standard deviation equal to the standard error of the ability estimate. Rudner’s method assumes the standard error is conditionally normally distributed. The area under this normal curve between cut scores is used to estimate the indices. See references.

Usage

```r
class.Rud(cutscore, ip, ability = NULL, se = NULL, rdm = NULL, quadrature = NULL)
Rud.D(cutscore, quadrature, sem)
Rud.P(cutscore, os, sem)
```
class.Rud

Arguments

cutscore  A scalar or vector of cut scores on the theta scale. Should not include +- inf, the function will include them.

ip  Matrix of item parameters, columns are discrimination, difficulty, guessing. For 1PL and 2PL, still give a Jx3 matrix, with ip[,1] = 1 and ip[,3] = 0 for example.

ability, os  Ability estimates for each subject.

se, sem  Standard errors of ability estimates

rdm  The response data matrix with rows as subjects and columns as items

quadrature  A list containing 1) The quadrature points and 2) Their corresponding weights

Details

Must give only ability and se, rdm, or quadrature. If ability and se are given, those scores are used for the P method. If rdm is given, ability and se are estimated with MLE (perfect response patterns given a -4 or 4) and used for the P method. If quadrature, the D method is used.

Value

Marginal  A matrix with two columns of marginal accuracy and consistency per cut score and/or simultaneous

Conditional  A matrix of conditional accuracy and conditional consistency returned in the order of subjects if ability or rdm is given

Note

class.Rud is a wrapper for Rud.P and Rud.D.

To expand the code to other IRT models for example, the code within Rud.P or Rud.D may need to be modified. If you can supply the theta estimates and their se's (which are conditionally normal), then the function should work as is.

Author(s)

Quinn Lathrop

References


Examples

```r
## from rdm, item parameters denote 4 item 1PL test, cut score at theta=.5
## only return marginal indices

params <- matrix(c(1,1,1,1,-2,1,0,0,0,0), 4, 3)
rdm <- sim(params, rnorm(100))

class.Rud(.5, params, rdm = rdm)$Marginal

## or from 40 quadrature points and weights, 2 cut scores
quad <- normal.qu(40)

class.Rud(c(-.5,1.5), params, quadrature = quad)$Marginal
```

---

**recursive.raw**  
*Recursive computation of conditional total score*

**Description**

Returns probabilities of each possible total score conditional on theta

**Usage**

```r
recursive.raw(theta, ip)
```

**Arguments**

- `theta` Vector of ability estimates or points to condition on.
- `ip` Jx3 matrix of item parameters, columns are discrimination, difficulty, and guessing.

**Value**

A matrix of theta points by possible score 0,1,\ldots,J.

**Note**

As described in Huynh 1990

**Author(s)**

Quinn Lathrop

**Examples**

```r
params <- matrix(c(1,1,1,1,-2,1,0,0,0,0), 4, 3)
rec.mat <- recursive.raw(c(-1,0,1), params)
```
TOtable.F

A general classification table for computing True accuracy or consistency

Description

Creates a 2x2 table (or larger for multiple cutscores), to compute the proportion of correctly classified, or consistently classified, subjects. Can easily be modified to return kappa if needed.

Usage

TOtable.F(theta, os, theta.cutoff, os.cutoff)

Arguments

theta a vector of scores, could be True scores, or simply the first test’s scores
os a vector of scores with elements corresponding to those given for theta. Could be observed scores, or the second test’s scores
theta.cutoff the cut score on the same scale as the scores given in theta
os.cutoff the cut score on the same scale as the scores given in os

Details

The cut scores need to be on the same scale as their subjects’ scores, but do not need to match each other. For example, one test scored with theta.hat and a theta scaled cutoff, and the second scored on total score and a number correct cutoff. See example.

Value

Table classification table with margins
P P or proportion of subjects on the diagonal, raw accuracy/consistency

Note

This is useful in simulation studies when we have unknowable knowledge, either a second test (for consistency), or True scores or Theta (not theta.hat) (for accuracy).

Author(s)

Quinn Lathrop
Examples

```r
## with simple 4 item 1PL simulated test:
## find True classification accuracy of theta.hat and total score
## use a cut score of theta = 0 (assume true score for total score cut score)

params <- matrix(c(1,1,1,1,-2,1,0,1,0,0,0), 4, 3)
theta <- rnorm(1, 0, 0)
rdm <- sim(params, theta)
theta.hat <- MLE(rdm, params)
total.score <- rowSums(rdm)
t.cut <- 0
x.cut <- sum(irf(params, t.cut)$f)

T0table.F(theta, theta.hat, t.cut, t.cut)$P
T0table.F(theta, total.score, t.cut, x.cut)$P

# for 1PL model total score is sufficient for theta.hat, repeat with 2PL or 3PL items to see effect of classifying on theta.hat instead of total score
```

Useful IRT Functions

A collection of useful IRT functions.

Description

Most are modified from the package irtoys, using a scaling of 1.7 is used (so that logistic models approximate the normal ogive).

Usage

- `iif(ip, x)`
- `irf(ip, x)`
- `MLE(resp, ip)`
- `normal.qu(n = 15, lower = -4, upper = 4, mu = 0, sigma = 1)`
- `SEM(ip, x)`
- `sim(ip, x)`
- `tif(ip, x)`

Arguments

- `ip` A Jx3 matrix of item parameters. Columns are discrimination, difficulty, and guessing
- `x` Vector of theta points
- `resp` Response data matrix, subjects by items
- `n` Number of quadrature points wanted
- `lower, upper` Range of points wanted
- `mu, sigma` The normal distribution from which points and weights are taken
Useful IRT Functions

Details

iif give item information, irf gives item response function, MLE returns maximum likelihood estimates of theta (perfect scores get +4), normal.qu returns a list length 2 of quadrature points and weights, SEM gives the inverse of the test information, sim returns simulated response matrix, tif gives the test information function.

Note

see R package irtoys for further details

Author(s)

Quinn Lathrop

References

see R package irtoys for further details

Examples

params<-matrix(c(1,1,1,-2,1,0,0,0,0),4,3)
rdm<-sim(params, rnorm(100))

theta.hat <- MLE(rdm, params)
theta.se <- SEM(rdm, params)

## transform a cut score of theta = 0 to the expected true score scale

t.cut <- 0
x.cut <- sum(irf(params, t.cut)$f)
Index

*Topic IRT
  Useful IRT Functions, 8
*Topic \textasciitilde Classification
  cacIRT-package, 2
*Topic \textasciitilde IRT
  cacIRT-package, 2
class.Lee, 3
class.Rud, 4
  recursive.raw, 6
*Topic \textasciitilde Item Response Theory
  cacIRT-package, 2
  cacIRT (cacIRT-package), 2
cacIRT-package, 2
class.Lee, 3
class.Rud, 4
  iif (Useful IRT Functions), 8
  irf (Useful IRT Functions), 8
  Lee.D (class.Lee), 3
  Lee.P (class.Lee), 3
  MLE (Useful IRT Functions), 8
  normal.qu (Useful IRT Functions), 8
  recursive.raw, 6
  Rud.D (class.Rud), 4
  Rud.P (class.Rud), 4
  SEM (Useful IRT Functions), 8
  sim (Useful IRT Functions), 8
  tif (Useful IRT Functions), 8
  TOtable.F, 7
  Useful IRT Functions, 8