Package 'shortIRT'

January 27, 2026

Type Package

Title Procedures Based on Item Response Theory Models for the Development of Short Test Forms

Version 1.0.0

Maintainer Ottavia M. Epifania <ottavia.epifania@unitn.it>

Description Implement different Item Response Theory (IRT) based procedures for the development of static short test forms (STFs) from a test. Two main procedures are considered (Epifania, Anselmi & Robusto, 2022 <doi:10.1007/978-3-031-27781-8_7>).
The procedures differ in how the most informative items are selected for the inclusion in the STF, either by considering their item information functions without any reference to any specific latent trait level (benchmark procedure) or by considering their information with respect to specific latent trait levels, denoted as theta targets (theta target procedure).
Three methods are implemented for the definition of the theta targets: (i) as the midpoints of equal intervals on the latent trait,
(ii) as the centroids of the clusters obtained by clustering the latent trait,
and (iii) as user-defined values. Importantly, the number of theta targets defines the number of items included in the STF.
For further details on the procedure, please refer to Epifania, Anselmi & Robusto (2022) <doi:10.1007/978-3-031-27781-8_7>.

License MIT + file LICENSE

Encoding UTF-8

Imports ggplot2

RoxygenNote 7.3.2

Suggests MASS, rmarkdown, sirt, testthat (>= 3.0.0), V8

Config/testthat/edition 3

NeedsCompilation no

Author Ottavia M. Epifania [aut, cre] (ORCID:
<https://orcid.org/0000-0001-8552-568X>),
Pasquale Anselmi [ctb],
Egidio Robusto [ctb],
Livio Finos [ctb]

Repository CRAN

Date/Publication 2026-01-27 22:50:02 UTC

Contents

---

bench *Benchmark Procedure*

Description

Create a Short Test Form (STF) using a benchmark procedure (i.e., the n most informative items are selected, where n is the number of items to include in the STF)

Usage

```
bench(item_par = NULL, iifs = NULL, theta = NULL, num_item = NULL)
```

Arguments

`item_par`	`data.frame`, dataframe with nrows equal to the number of items and 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i.
`iifs`	`data.frame`, dataframe with n-rows equal to the length of the latent trait θ and n-cols equal to the number of items in the full-length test. It contains the item information functions (IIFs) of the items in the full-length test. Cannot use both `ipar` and `iifs`.
`theta`	`numeric`, vector with the latent trait values
`num_item`	`integer`, the number of items to include in the short test form

Details

A short test form composed of N items is constructed from an item bank B by selecting the items with the highest item information values.

Let $I_i(\theta)$ denote the item information function (IIF) for item i, with $i = 1, \ldots, |B|$. The IIFs of the item bank are sorted in decreasing order:

$$\text{iif} = \left(\max_{1 \leq i \leq |B|} I_i(\theta), \ldots, \min_{1 \leq i \leq |B|} I_i(\theta) \right)$$

The first N items in the ordered vector iif, with $N < |B|$, are selected to be included in the short test form.

Value

An object of class `bench` of length 3 with:

- `stf`: dataframe with the items selected for inclusion in the STF (`isel`), their maximum information function (`maxiif`), for a specific latent trait level θ (column `theta`)
- `item_par`: the original dataframe containing the item parameters
- `selected_items`: dataframe with the parameters of the selected items

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500, sd = 2)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
resB <- bench(item_par, theta = theta, num_item = 5)
str(resB)
```

`define_targets` *Define θ targets*

Description

Define θ targets according to two procedures, either by considering the midpoints of equal intervals defined on the latent trait (`equal`) or the centroids obtained by clustering the latent trait (`clusters`)

Usage

```
define_targets(theta, num_targets = NULL, method = c("equal", "clusters"))
```

Arguments

`theta`	`numeric` vector defining the latent trait θ
`num_targets`	`integer` value, define the number of θ targets. The number of θ targets defines the number of items included in the STF.
`method`	`character`, either `equal` (default) or `clusters`

Value

A vector of length `num_targets` with the generated θ targets. The class can be either `equal` or `clusters`, dependening on the method used for the definition of the θ targets

Examples

```
set.seed(123)
theta <- rnorm(1000)
targets <- define_targets(theta, num_targets = 5, method = "clusters")
```

IRT *Compute expected probability for a single item*

Description

Compute the expected probability for an item i according to its IRT parameters. According to the parameters that are specified, the probability according to the 1-PL, 2-PL, 3-PL, or 4-PL models is computed

Usage

```
IRT(theta, b = 0, a = 1, c = 0, e = 1)
```

Arguments

`theta`	`numeric` latent trait level of person p. It can be a single value or a vector of values.
`b`	`numeric` location of item i. Default is 0.
`a`	`numeric` discrimination parameter for item i. Default is 1.
`c`	`numeric` pesudoguessing parameter of item i. Default is 0.
`e`	`numeric` upper asymptote of item i. Default is 1.

Details

The probability of a correct response under the four-parameter logistic (4PL) model is defined as:

$$P(X = 1 \mid \theta_p) = c_i + \frac{e_i - c_i}{1 + \exp\left[-a_i(\theta_p - b_i)\right]}$$

where a is the discrimination parameter, b is the difficulty parameter, c is the lower asymptote (guessing), and e is the upper asymptote (inattention/slip). By constraining $e_i = 1$, $c_i = 0$, and $a_i = 1\ \forall i$, the probability is computed according to the 3-PL, 2-PL and 1-PL, respectively

Value

a single value, that is the probability of the correct response for item i given the specified parameters

Examples

```
IRT(theta = 0, b = 0,  a = 1, c = 0, e = 1)
# compute probability for a vector of thetas for the same item
IRT(theta = c(-1, 0, 1), b = 0,  a = 1, c = 0, e = 1)
```

---

irt_estimate *Estimate of theta*

Description

Maximum Likelihood estimation of theta

Usage

```
irt_estimate(item_par, responses = NULL, theta, lower = -3, upper = abs(lower))
```

Arguments

item_par	data.frame, dataframe with nrows equal to the number of items and 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i.
responses	matrix, $p \times i$ matrix with the dichotomous responses of each respondent p on each item i. Default is NULL.
theta	numeric latent trait level of person p, it can be a single value or a vector of values.
lower	integer lower value of θ to be considered for the estimation
upper	integer upper value of θ to be considered for the estimation

Value

A numeric vector of length equal to the length of theta with the ML estimation of the latent trait

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
```

```
)
# estimate theta
theta_hat <- irt_estimate(item_par, theta = theta)
plot(theta, theta_hat)
```

`item_info` *Item Information Functions (multiple items, IIFs)*

Description

Computes the item information functions for multiple items

Usage

```
item_info(item_par, theta = seq(-5, 5, length.out = 1000))
```

Arguments

`item_par`	`data.frame`, dataframe with nrows equal to the number of items and 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i.
`theta`	`numeric` latent trait level of person p, it can be a single value or a vector of values.

Value

A `matrix` of class `iifs` with nrows equal to the length of `theta` and ncols equal to the number of items in `item_par`

Examples

```
set.seed(123)
parameters <- data.frame(b = c(-3,-2,0, 2, 3),
a = runif(5, 1.2, 1.9),
c = rep(0,5),
e= rep(1, 5))
infos <- item_info(parameters)
head(infos)
```

`i_info` *Item Information Function (single item, IIF)*

Description

Compute the item information function for a single item. See `Details`.

Usage

```
i_info(b, a = 1, c = 0, e = 1, theta = seq(-5, 5, length.out = 1000))
```

Arguments

`b`	`numeric`, difficulty/location parameter b_i
`a`	`numeric`, discrimination parameter a_i. Default is 1.
`c`	`numeric`, pesudoguessing parameter c_i. Default is 0.
`e`	`numeric`, upper asymptote e_i. Default is 1.
`theta`	`numeric` latent trait level of person p, it can be a single value or a vector of values. Default is a vector of 1 thousand values rangin from -5 to +5

Details

Let $P(\theta)$ denote the probability of a correct response under the four-parameter logistic (4PL) model:

$$P(\theta) = c + \frac{e - c}{1 + \exp\left[-a(\theta - b)\right]}$$

and let $Q(\theta) = 1 - P(\theta)$.

The item information function is computed as:

$$I(\theta) = \frac{a^2 \left[P(\theta) - c\right]^2 \left[e - P(\theta)\right]^2}{(e - c)^2 \, P(\theta) \, Q(\theta)}$$

Value

A numeric vector of length equal to `theta`, which contains the item information function for a single item with respect to the values specified in theta

Examples

```
# IIF of an item with b = 0
i_info(b = 0, theta = c(-3,-1,0,1,3))
```

logLik_theta *Log-likelihood estiamtion of theta*

Description

Log-likelihood estiamtion of theta

Usage

```
logLik_theta(theta, x, item_par)
```

Arguments

theta	numeric vector with true values of θ
x	integer vector of 0s and 1s, response pattern of each respondent
item_par	data.frame, dataframe with nrows equal to the number of items and 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i.

Value

The log-likelihood

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
obs_response <- obsirt(mpirt(item_par, theta))
# LogLikelihood of theta
logLik_theta(theta, obs_response, item_par)
```

mpirt *Compute expected probability for multiple items*

Description

Compute expected probability for multiple items

Usage

```
mpirt(item_par, theta)
```

Arguments

item_par	`data.frame`, dataframe with nrows equal to the number of items and 4 columns, one for each of the item parameters. The columns must be named "a", "b", "c", "e" and must contain the respective IRT parameters, namely discrimination a_i, location b_i, pseudo-guessing c_i, and upper asymptote e_i.
theta	`numeric` latent trait level of person p, it can be a single value or a vector of values.

Value

A $p \times i$ matrix of class `mpirt` with the expected probability of observing a correct response for respondent p on item i

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
expected_prob <- mpirt(item_par, theta)
```

obsirt *Simulate responses according to IRT probabilities*

Description

Simulate responses according to IRT probabilities

Usage

```
obsirt(myp)
```

Arguments

myp	Object of class mpirt containing the expected IRT probabilities obtained with function mpirt()

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
expected_prob <- mpirt(item_par, theta)
simulated_responses <- obsirt(expected_prob)
```

---

plot.bench *Method for plotting the TIF of the STF*

Description

The STF is obtained with the benchmark procedure implemented with function bench()

Usage

```
## S3 method for class 'bench'
plot(
  x,
  fun = "sum",
  theta = seq(-5, 5, length.out = 1000),
  show_both = TRUE,
  ...
)
```

Arguments

x	Object of class bench
fun	character, whether to consider the mean or the sum for the computation of the TIF
theta	numeric, latent trait for the graphical representation
show_both	logical, default is TRUE. Whether to show or not the TIF obtained from the full-length test
...	other arguments

Value

A ggplot showing the TIFs of both the STF and the full-length test

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500, sd = 2)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
resB <- bench(item_par, theta = theta, num_item = 5)
plot(resB)
# plot only the TIF of the STF
plot(resB, show_both = FALSE)
```

---

plot.iifs *Method for plotting the IIFs*

Description

Plot the IIFs

Usage

```
## S3 method for class 'iifs'
plot(x, single_panels = TRUE, items = NULL, ...)
```

Arguments

x	data.frame of class iifs obtained with the function item_info()
single_panels	logical, default is TRUE. Whether to show the IIFs of each item on different panels
items	default is NULç (shows all items). Allows for selecting specific items for the plot
...	other arguments

Value

A ggplot

Examples

```
set.seed(123)
parameters <- data.frame(b = c(-3,-2,0, 2, 3),
a = runif(5, 1.2, 1.9),
c = rep(0,5),
e= rep(1, 5))
infos <- item_info(parameters)
plot(infos)
# plot only items 1 and 3 on a single panel
plot(infos, items = c(1,3), single_panels = FALSE)
```

`plot.theta_target` *Method for plotting the TIF of the STF*

Description

The STF is obtained with the theta target procedure

Usage

```
## S3 method for class 'theta_target'
plot(
  x,
  fun = "sum",
  theta = seq(-5, 5, length.out = 1000),
  show_targets = TRUE,
  show_both = TRUE,
  ...
)
```

Arguments

`x`	Object of class `theta_target` obtained with function `theta_target()`
`fun`	`character`, whether to consider the mean or the sum for the computation of the TIF
`theta`	`numeric`, latent trait for the graphical representation
`show_targets`	`logical`, default is TRUE. Whether to show or not the theta targets
`show_both`	`logical`, default is `TRUE`. Whether to show or not the TIF obtained from the full-length test
`...`	other arguments

Value

A `ggplot` showing the TIFs of both the STF and the full-length test

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
targets <- define_targets(theta, num_targets = 4)
resT <- theta_target(targets, item_par)
plot(resT)
# plot without showing the theta targets
plot(resT, show_targets = FALSE)
```

`plot.tif` *Plot TIF*

Description

Plot TIF

Usage

```
## S3 method for class 'tif'
plot(x, ...)
```

Arguments

`x`	object of class `tif` obtained with the `tif()` function
`...`	other arguments

Value

A `ggplot` displaying the TIF

Examples

```
set.seed(123)
n <- 5
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
iifs <- item_info(item_par)
test_tif <- tif(iifs)
```

```
plot(test_tif)
# compute the mean tif
test_tif_mean <- tif(iifs, fun = "mean")
plot(test_tif_mean)
```

`summary.bench` *Method for the summary of the STF*

Description

The STF is obtained with the benchmark procedure implemented in the function `bench()`

Usage

```
## S3 method for class 'bench'
summary(object, ...)
```

Arguments

`object`	Object of class `bench()`
`...`	other arguments

Value

A summary of the STF obtained from the application of the benchmark procedure

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
resB <- bench(item_par, theta = theta, num_item = 5)
summary(resB)
```

summary.theta_target *Method for the summary of the STF*

Description

The STF is obtained with the θ target procedure implemented in the function `theta_target()`

Usage

```
## S3 method for class 'theta_target'
summary(object, ...)
```

Arguments

`object`	Object of class `theta_target`
`...`	other arguments

Value

A summary of the STF obtained from the application of the θ target procedure

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
targets <- define_targets(theta, num_targets = 4)
resT <- theta_target(targets, item_par)
summary(resT)
```

theta_target *Theta target procedure*

Description

Procedure based on the theta targets procedure for the generation of a short test form

Usage

```
theta_target(targets, item_par)
```

Arguments

`targets`	numeric vector with the discrete values of theta for which the information needs to be maximized
`item_par`	dataframe, with nrows equals to the length of the latent trait and four columns, each denoting the IRT item parameters

Details

Let $k = 0, \ldots, K$ denote the iteration index of the procedure, with $K = N - 1$. Let J be the total number of items in the item bank and N the desired length of the short test form.

Define:

- $S^k \subseteq \{1, \ldots, J\}$ as the set of items selected for inclusion in the short test form up to iteration k;
- $Q^k \subseteq \{1, \ldots, N\}$ as the set of ability targets satisfied up to iteration k.

At initialization ($k = 0$), $S^0 = \varnothing$ and $Q^0 = \varnothing$.

The procedure iterates the following steps until $k = K$:

1. Select the item–target pair (i, n) maximizing the item information function:

$$(i, n) = \arg \max_{i \in B \setminus S^k,\, n \in N \setminus Q^k} \mathrm{IIF}(i, n)$$

2. Update the set of selected items:

$$S^{k+1} = S^k \cup \{i\}$$

3. Update the set of satisfied ability targets:

$$Q^{k+1} = Q^k \cup \{n\}$$

At iteration K, the procedure yields $|S^{K+1}| = N$ and $|Q^{K+1}| = N$.

Value

An object of class `theta_target` of length 4 containing:

- **stf**: a data frame containing the items selected for inclusion in the short test form (column `isel`), their maximum item information function (column `maxiif`), and the corresponding theta target (column `theta_target`).
- **item_par**: the original data frame containing the item parameters.
- **selected_items**: a data frame containing the parameters of the selected items.
- **intervals**: a character string indicating how the theta targets were obtained. The value `"clusters"` denotes clustering of the latent trait, `"equal"` denotes equally spaced intervals, and `"unknown"` identifies any other case.

Examples

```
set.seed(123)
n <- 50
theta <- rnorm(500)
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
targets <- define_targets(theta, num_targets = 4)
resT <- theta_target(targets, item_par)
str(resT)
```

---

tif *Test Information Function (TIF)*

Description

Compute the test information function of a test given a matrix of item information functions

Usage

```
tif(iifs, fun = "sum", theta = seq(-5, 5, length.out = 1000))
```

Arguments

`iifs`	object of class `iifs` containg the item information functions
`fun`	`character`, defines the function for the computation of the TIF, either by summing the items (sum) or by computing the mean (mean)
`theta`	`numeric` latent trait level of person p, it can be a single value or a vector of values. Default is a vector of 1 thousand values ranging from -5 to +5

Value

A `data.frame` of class `tif` with two columns: (i) `theta` containing the latent trait values, and (ii) `tif` containing the TIF values computed as either the sum or the mean of the IIFs

Examples

```
set.seed(123)
n <- 5
item_par <- data.frame(
  b = runif(n, -3, 3),
  a = runif(n, 1.2, 1.9),
  c = rep(0, n),
  e = rep(1, n)
)
iifs <- item_info(item_par)
test_tif <- tif(iifs)
```

Index