Package 'Rirt'

January 20, 2025

Type Package

Title Data Analysis and Parameter Estimation Using Item Response
Theory

Version 0.0.2

Date 2019-10-22

Author Xiao Luo [aut, cre]

Maintainer Xiao Luo <xluo1986@gmail.com>

Description Parameter estimation, computation of probability, information, and
(log-)likelihood, and visualization of item/test characteristic curves and
item/test information functions for three uni-dimensional item response theory
models: the 3-parameter-logistic model, generalized partial credit model,
and graded response model. The full documentation and tutorials are at
<https://github.com/xluo11/Rirt>.

License GPL (>= 3)

Depends R (>= 3.6.0)

URL https://github.com/xluo11/Rirt

BugReports https://github.com/xluo11/Rirt/issues

LinkingTo Rcpp

Imports ggplot2, Rcpp, reshape2, stats

Suggests testthat

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-10-23 14:00:02 UTC

Contents

---

estimate_mixed   *Estimation of the Mixed Format Model*

Description

Estimate the mixed format model

Usage

```
model_mixed_eap(u, items, D = 1.702, priors = c(0, 1),
  bounds_t = c(-4, 4))

model_mixed_map(u, items, D = 1.702, priors = c(0, 1),
  bounds_t = c(-4, 4), iter = 30, conv = 0.001)
```

Arguments

u	the response data, 2d marix
items	a list of 3pl, gpcm, grm items
D	the scaling constant
priors	the prior distribution
bounds_t	the lower- and upper-bound of the parameter
iter	the maximum number of newton-raphson iterations
conv	the convergence criterion

Value

model_mixed_eap returns a list of point estimates and standard error of the ability parameters

model_mixed_map returns a list of point estimates of the ability parameters

Examples

```
x <- model_mixed_gendata(200, 30, 5, 5, 3)
y <- model_mixed_eap(x$u, x$items)
c('corr'=cor(x$t, y$t), 'rmse'=rmse(x$t, y$t))
x <- model_mixed_gendata(200, 30, 5, 5, 3)
y <- model_mixed_map(x$u, x$items)
c('corr'=cor(x$t, y$t), 'rmse'=rmse(x$t, y$t))
```

---

model_3pl_prob *3-parameter-logistic model*

Description

Common computations and operations for the 3PL model

Usage

```
model_3pl_prob(t, a, b, c, D = 1.702)

model_3pl_info(t, a, b, c, D = 1.702)

model_3pl_lh(u, t, a, b, c, D = 1.702, log = FALSE)

model_3pl_rescale(t, a, b, c, scale = c("t", "b"), mean = 0, sd = 1)

model_3pl_gendata(n_p, n_i, t = NULL, a = NULL, b = NULL, c = NULL,
  D = 1.702, t_dist = c(0, 1), a_dist = c(-0.1, 0.2), b_dist = c(0,
  0.7), c_dist = c(5, 46), t_bounds = c(-3, 3), a_bounds = c(0.01,
  2.5), b_bounds = c(-3, 3), c_bounds = c(0, 0.5), missing = NULL,
  ...)

model_3pl_plot(a, b, c, D = 1.702, type = c("prob", "info"),
  total = FALSE, xaxis = seq(-4, 4, 0.1))

model_3pl_plot_loglh(u, a, b, c, D = 1.702, xaxis = seq(-4, 4, 0.1),
  verbose = FALSE)
```

Arguments

t	ability parameters, 1d vector
a	discrimination parameters, 1d vector
b	difficulty parameters, 1d vector
c	guessing parameters, 1d vector
D	the scaling constant, default=1.702
u	observed responses, 2d matrix

log	True to return log-likelihood
scale	the scale, 't' for theta or 'b' for b-parameters
mean	the mean of the new scale
sd	the standard deviation of the new scale
n_p	the number of people to be generated
n_i	the number of items to be generated
t_dist	parameters of the normal distribution used to generate t-parameters
a_dist	parameters of the lognormal distribution used to generate a-parameters
b_dist	parameters of the normal distribution used to generate b-parameters
c_dist	parameters of the beta distribution used to generate c-parameters
t_bounds	bounds of the ability parameters
a_bounds	bounds of the discrimination parameters
b_bounds	bounds of the difficulty parameters
c_bounds	bounds of the guessing parameters
missing	the proportion or number of missing responses
...	additional arguments
type	the type of plot: 'prob' for item characteristic curve (ICC) and 'info' for item information function curve (IIFC)
total	TRUE to sum values over items
xaxis	the values of x-axis
verbose	TRUE to print rough maximum likelihood estimates

Value

model_3pl_prob returns the resulting probabilities in a matrix

model_3pl_info returns the resulting information in a matrix

model_3pl_lh returns the resulting likelihood in a matrix

model_3pl_rescale returns t, a, b, c parameters on the new scale

model_3pl_gendata returns the generated response matrix and parameters in a list

model_3pl_plot returns a ggplot object

model_3pl_plot_loglh returns a ggplot object

Examples

```
with(model_3pl_gendata(10, 5), model_3pl_prob(t, a, b, c))
with(model_3pl_gendata(10, 5), model_3pl_info(t, a, b, c))
with(model_3pl_gendata(10, 5), model_3pl_lh(u, t, a, b, c))
model_3pl_gendata(10, 5)
model_3pl_gendata(10, 5, a=1, c=0, missing=.1)
with(model_3pl_gendata(10, 5), model_3pl_plot(a, b, c, type="prob"))
with(model_3pl_gendata(10, 5), model_3pl_plot(a, b, c, type="info", total=TRUE))
with(model_3pl_gendata(5, 50), model_3pl_plot_loglh(u, a, b, c))
```

---

model_gpcm   *Generalized Partial Credit Model*

Description

Common computations and operatoins for the GPCM

Usage

```
model_gpcm_prob(t, a, b, d, D = 1.702, d0 = NULL)

model_gpcm_info(t, a, b, d, D = 1.702, d0 = NULL)

model_gpcm_lh(u, t, a, b, d, D = 1.702, d0 = NULL, log = FALSE)

model_gpcm_gendata(n_p, n_i, n_c, t = NULL, a = NULL, b = NULL,
  d = NULL, D = 1.702, sort_d = FALSE, t_dist = c(0, 1),
  a_dist = c(-0.1, 0.2), b_dist = c(0, 0.8), d_dist = c(0, 1),
  t_bounds = c(-3, 3), a_bounds = c(0.01, 2.5), b_bounds = c(-3, 3),
  d_bounds = c(-3, 3), missing = NULL, ...)

model_gpcm_rescale(t, a, b, d, scale = c("t", "b"), mean = 0, sd = 1)

model_gpcm_plot(a, b, d, D = 1.702, d0 = NULL, type = c("prob",
  "info"), item_level = FALSE, total = FALSE, xaxis = seq(-6, 6,
  0.1))

model_gpcm_plot_loglh(u, a, b, d, D = 1.702, d0 = NULL,
  xaxis = seq(-6, 6, 0.1), verbose = FALSE)
```

Arguments

t	ability parameters, 1d vector
a	discrimination parameters, 1d vector
b	item location parameters, 1d vector
d	item category parameters, 2d vector
D	the scaling constant, default=1.702
d0	insert an initial category value
u	observed scores (starting from 0), 2d matrix
log	TRUE to return log-likelihood
n_p	the number of people to be generated
n_i	the number of items to be generated
n_c	the number of score categories
sort_d	TRUE to sort d parameters for each item

t_dist	parameters of the normal distribution used to generate t-parameters
a_dist	parameters of the lognormal distribution parameters of a-parameters
b_dist	parameters of the normal distribution used to generate b-parameters
d_dist	parameters of the normal distribution used to generate d-parameters
t_bounds	the bounds of the ability parameters
a_bounds	the bounds of the discrimination parameters
b_bounds	the bounds of the difficulty parameters
d_bounds	the bounds of the category parameters
missing	the proportion or number of missing responses
...	additional arguments
scale	the scale, 't' for theta or 'b' for b-parameters
mean	the mean of the new scale
sd	the standard deviation of the new scale
type	the type of plot, prob for ICC and info for IIFC
item_level	TRUE to add item level data
total	TRUE to sum values over items
xaxis	the values of x-axis
verbose	TRUE to print rough maximum likelihood values

Details

Use NA to represent unused category.

Value

model_gpcm_prob returns the resulting probabilities in a 3d array

model_gpcm_info returns the resulting information in a 3d array

model_gpcm_lh returns the resulting likelihood in a matrix

model_gpcm_gendata returns the generated response matrix and parameters

model_gpcm_rescale returns t, a, b, d parameters on the new scale

model_gpcm_plot returns a ggplot object

model_gpcm_plot_loglh returns a ggplot object

Examples

```
with(model_gpcm_gendata(10, 5, 3), model_gpcm_prob(t, a, b, d))
with(model_gpcm_gendata(10, 5, 3), model_gpcm_info(t, a, b, d))
with(model_gpcm_gendata(10, 5, 3), model_gpcm_lh(u, t, a, b, d))
model_gpcm_gendata(10, 5, 3)
model_gpcm_gendata(10, 5, 3, missing=.1)
# Figure 1 in Muraki, 1992 (APM)
b <- matrix(c(-2,0,2,-.5,0,2,-.5,0,2), nrow=3, byrow=TRUE)
model_gpcm_plot(a=c(1,1,.7), b=rowMeans(b), d=rowMeans(b)-b, D=1.0, d0=0)
```

```
# Figure 2 in Muraki, 1992 (APM)
b <- matrix(c(.5,0,NA,0,0,0), nrow=2, byrow=TRUE)
model_gpcm_plot(a=.7, b=rowMeans(b, na.rm=TRUE), d=rowMeans(b, na.rm=TRUE)-b, D=1.0, d0=0)
# Figure 3 in Muraki, 1992 (APM)
b <- matrix(c(1.759,-1.643,3.970,-2.764), nrow=2, byrow=TRUE)
model_gpcm_plot(a=c(.778,.946), b=rowMeans(b), d=rowMeans(b)-b, D=1.0, d0=0)
# Figure 1 in Muraki, 1993 (APM)
b <- matrix(c(0,-2,4,0,-2,2,0,-2,0,0,-2,-2,0,-2,-4), nrow=5, byrow=TRUE)
model_gpcm_plot(a=1, b=rowMeans(b), d=rowMeans(b)-b, D=1.0)
# Figure 2 in Muraki, 1993 (APM)
b <- matrix(c(0,-2,4,0,-2,2,0,-2,0,0,-2,-2,0,-2,-4), nrow=5, byrow=TRUE)
model_gpcm_plot(a=1, b=rowMeans(b), d=rowMeans(b)-b, D=1.0, type='info', item_level=TRUE)
with(model_gpcm_gendata(5, 50, 3), model_gpcm_plot_loglh(u, a, b, d))
```

---

model_grm	*Graded Response Model*

Description

Common computations and operations for the GRM

Usage

```
model_grm_prob(t, a, b, D = 1.702, raw = FALSE)

model_grm_info(t, a, b, D = 1.702)

model_grm_lh(u, t, a, b, D = 1.702, log = FALSE)

model_grm_gendata(n_p, n_i, n_c, t = NULL, a = NULL, b = NULL,
  D = 1.702, t_dist = c(0, 1), a_dist = c(-0.1, 0.2), b_dist = c(0,
  0.8), t_bounds = c(-3, 3), a_bounds = c(0.01, 2.5),
  b_bounds = c(-3, 3), missing = NULL, ...)

model_grm_rescale(t, a, b, scale = c("t", "b"), mean = 0, sd = 1)

model_grm_plot(a, b, D = 1.702, type = c("prob", "info"),
  item_level = FALSE, total = FALSE, xaxis = seq(-6, 6, 0.1),
  raw = FALSE)

model_grm_plot_loglh(u, a, b, D = 1.702, xaxis = seq(-6, 6, 0.1),
  verbose = FALSE)
```

Arguments

t	ability parameters, 1d vector
a	discrimination parameters, 1d vector
b	item location parameters, 2d matrix

D	the scaling constant, default=1.702
raw	TRUE to return P*
u	observed scores (starting from 0), 2d matrix
log	TRUE to return log-likelihood
n_p	the number of people to be generated
n_i	the number of items to be generated
n_c	the number of score categories
t_dist	parameters of the normal distribution used to generate t-parameters
a_dist	parameters of the lognormal distribution used to generate a-parameters
b_dist	parameters of the normal distribution used to generate b-parameters
t_bounds	the bounds of the ability parameters
a_bounds	the bounds of the discrimination parameters
b_bounds	the bounds of the difficulty parameters
missing	the proportion or number of missing responses
...	additional arguments
scale	the scale, 't' for theta or 'b' for b-parameters
mean	the mean of the new scale
sd	the standard deviation of the new scale
type	the type of plot, prob for ICC and info for IIFC
item_level	TRUE to combine categories
total	TRUE to sum values over items
xaxis	the values of x-axis
verbose	TRUE to print rough maximum likelihood values

Value

model_grm_prob returns the resulting probabilities in a 3d array

model_grm_info returns the resulting information in a 3d array

model_grm_lh returns the resulting likelihood in a matrix

model_grm_gendata returns the generated response data and parameters in a list

model_grm_rescale returns t, a, b parameters on the new scale

model_grm_plot returns a ggplot object

model_grm_plot_loglh returns a ggplot object

Examples

```
with(model_grm_gendata(10, 5, 3), model_grm_prob(t, a, b))
with(model_grm_gendata(10, 5, 3), model_grm_info(t, a, b))
with(model_grm_gendata(10, 5, 3), model_grm_lh(u, t, a, b))
model_grm_gendata(10, 5, 3)
model_grm_gendata(10, 5, 3, missing=.1)
with(model_grm_gendata(10, 5, 3), model_grm_plot(a, b, type='prob'))
with(model_grm_gendata(10, 5, 3), model_grm_plot(a, b, type='info', item_level=TRUE))
with(model_grm_gendata(5, 50, 3), model_grm_plot_loglh(u, a, b))
```

---

model_mixed   *Mixed-format model*

Description

Common computations and operations for the mixed format model

Usage

```
model_mixed_gendata(n_p, n_3pl = 0, n_gpcm = 0, n_grm = 0, n_c, ...)

model_mixed_prob(t, items, D = 1.702)

model_mixed_info(t, items, D = 1.702, combine = TRUE)

model_mixed_lh(u, t, items, D = 1.702, log = FALSE, combine = TRUE)
```

Arguments

n_p	the number of test takers
n_3pl	the number of 3pl items
n_gpcm	the number of gpcm items
n_grm	the number of grm items
n_c	the number of score categories for polytomous items
...	additional arguments
t	ability parameters, a vector
items	a list of '3pl', 'gpcm', and 'grm' items
D	the scaling constant, default=1.702
combine	TRUE to combine results from list to matrix
u	the response data, a 2d matrix
log	TRUE to return log-likelihood

Value

model_mixed_gendata returns a list of generated responses, ability paramters and items

model_mixed_prob returns a list of probabilities for '3pl', 'gpcm', and 'grm' items

model_mixed_info returns a list or matrix of information

Examples

```
# generate 10 3pl items, 5 gpcm items and 5 grm items
model_mixed_gendata(10, n_3pl=10, n_gpcm=5, n_grm=5, n_c=3)
# generate 5 gpcm items and 5 grm items, 4 score categories each
model_mixed_gendata(10, n_gpcm=5, n_grm=5, n_c=4)
# generate 5 people and 4 items of each type
with(model_mixed_gendata(n_p=5, n_3pl=4, n_gpcm=4, n_grm=4, n_c=3),
     model_mixed_prob(t, items))

# generate 10 people and 5 gpcm and 5 grm items
with(model_mixed_gendata(n_p=10, n_gpcm=4, n_grm=4, n_c=3),
     model_mixed_prob(t, items))
with(model_mixed_gendata(10, 4, 4, 4, 3), model_mixed_info(t, items))
with(model_mixed_gendata(10, 0, 4, 4, 3), model_mixed_info(t, items))
with(model_mixed_gendata(10, 4, 4, 4, 3), model_mixed_lh(u, t, items))
```

---

utils   *Utility functions*

Description

rmse computes the root mean squared error (RMSE) of two numeric vectors/matrices

freq computes the frequency counts of a numeric or character vector

cronbach_alpha computes the Cronbach's alpha internal consistency reliability index

spearman_brown predicts the reliability when the current test is extended to n times longer

spearman_brown_reverse computes how many times the current test length needs to be extended in order to reach targeted reliability

quadratic kappa computes the quadratic weighted kappa of two numeric vectors

Usage

```
rmse(x1, x2)

freq(x, vals = NULL, decimal = NULL)

cronbach_alpha(u)

spearman_brown(rho, n_len)

spearman_brown_reverse(rho, target_rho)

quadratic_kappa(x1, x2)
```

Arguments

x1	a numeric vector or matrix
x2	a numeric vector or matrix
x	a numeric or character vector
vals	valid values, NULL to include all values
decimal	round results to n-th decimal places
u	oberved responses, 2d matrix
rho	the reliability of the current test
n_len	extend the test to n times longer
target_rho	the targeted reliability

Value

freq returns the frequency counts and percentages in a data.frame

Examples

```
rmse(rnorm(10), rnorm(10))
freq(round(runif(100, 1, 5)))
cronbach_alpha(model_3pl_gendata(1000, 20)$u)
spearman_brown(.70, 2)
spearman_brown_reverse(.70, .85)
quadratic_kappa(round(runif(100, 1, 5)), round(runif(100, 1, 5)))
```

Index