Package ‘diffIRT’

October 13, 2022

Type Package

Title Diffusion IRT Models for Response and Response Time Data

Version 1.5

Date 2015-08-13

Author Dylan Molenaar

Maintainer Dylan Molenaar <d.molenaar@uva.nl>

Depends statmod

Description Package to fit diffusion-based IRT models to response and response time data. Models are fit using marginal maximum likelihood. Parameter restrictions (fixed value and equality constraints) are possible. In addition, factor scores (person drift rate and person boundary separation) can be estimated. Model fit assessment tools are also available. The traditional diffusion model can be estimated as well.

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-08-14 00:20:21

R topics documented:

anova 2
brightness 3
coef 4
diffIRT 5
extraversion 9
factest 10
QQdiff 11
RespFit 13
rotation 15
simdiff 16
simdiffT 17
summary 19

Index **21**

---

anova *Conduct a likelihood ratio test on two nested `diffIRT` models*

Description

Using this function, a likelihood ratio test is conducted on two nested `diffIRT` model. Results are printed to the screen togheter with the AIC, BIC, sample size adjusted BIC, and DIC comparative fit indices for both models.

Usage

```
## S3 method for class 'diffIRT'
anova(object, object2, ...)
```

Arguments

`object`	a `diffIRT` object that is nested within the model provided in `object2`.
`object2`	a `diffIRT` object.
`...`	additional parameters, currently not used.

Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

See Also

`diffIRT` for fitting diffusion IRT models.

Examples

```
## Not run:

# simulate data according to a D-diffusion model
# with equal a[i] parameters
data=simdiff(100,10,
  ai=rep(.3,10),vi=seq(-1,1,length=10),ter=runif(10,2,3),
  gamma=rlnorm(100,0,.3),theta=rnorm(100,0,.5),
  model="D")

# fit a full D-diffusion model
res=diffIRT(data$rt,data$x,model="D")

# fit a D-diffusion model subject to an
# equality constraint across all a[i] parameters
res2=diffIRT(data$rt,data$x,model="D",constrain=c(rep(1,10),2:21,22,23))

# use the anova function to conduct the likelihood ratio test
```

```
anova(res2,res)

## End(Not run)
```

---

brightness *A Simulated Response Time Dataset according to an Experimental Design*

Description

The data are simulated according to a design similar as that of a real brightness discrimination experiment by Ratcliff & Rouder (1998). In this experiment, the subject had to decide for a number of trials whether the brightness of a stimulus (a randomly generated array of pixels displayed on a computer screen) was either 'high' or 'low'. The true brightness of the stimuli were manipulated into a number of levels and administered with a speed instruction ("respond as fast as possible") and with an accuracy instruction ("respond as accurate as possible"). Present dataset was simulated according to a design with 6 different brightness levels and 2 speed instructions resulting in 12 conditions. In the brightness data matrix, the first 12 columns are the responses and the next 12 columns are the response times. Each trial is assigned to a separate row with the response time of that trial in the corresponding column and NA's on the remaining columns. Similarly for the responses. In addition, the data are arranged in such a way that the first 6 conditions are the speed instructed stimuli and the next 6 conditions are the corresponding accuracy instructed versions of these stimuli. See below for an example how to analyse these data using the **diffIRT** package (taken from Molenaar, Tuerlinckx, & van der Maas, 2015).

References

Molenaar, D., Tuerlinkcx, F., & van der Maas, H.L.J. (2015). Fitting Diffusion Item Response Theory Models for Responses and Response Times Using the R Package diffIRT. *Journal of Statistical Software*, **66(4)**, 1-34. URL http://www.jstatsoft.org/v66/i04/.

Ratcliff, R., & Rouder, J. N. (1998). Modeling response times for two-choice decisions. *Psychological Science*, **9(5)**, 347-356.

Examples

```
data(brightness)
x=brightness[,1:12]
rt=brightness[,13:24]

## Not run:
res = diffIRT(rt,x,model="D",constrain=c(rep(1,6),
rep(2,6),3:8,3:8,rep(9,12),0,10), start=c(rep(NA,36),0,NA))

## End(Not run)
```

coef *Return estimated parameters*

Description

Returns estimated item and population parameters from a diffIRT object

Usage

```
## S3 method for class 'diffIRT'
coef(object, ...)
```

Arguments

object	a diffIRT object from which parameter estimates need to be extracted.
...	additional parameters, currently not used.

Value

Returns a list with two entries:

item	the estimated item parameters, item boundary (ai), item drift (vi), and item non-desicion time (ter).
pop	the estimated population parameters, omega_ap and omega_vp.

Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

See Also

diffIRT for fitting diffusion IRT models.

Examples

```
## Not run:
# simulate data accroding to D-diffusion IRT model
data=simdiff(N=100,nit=10,model="D")

# fit the D-diffusion IRT model
res1=diffIRT(data$rt,data$x)

# extract parameter estimates
coef(res1)

## End(Not run)
```

diffIRT *Fitting diffusion IRT models*

Description

This function fits the D-diffusion or the Q-diffusion IRT model to response and response time data using marginal maximum likelihood. Item parameters that are estimated are item drift rate, v[i] item boundary separation, a[i], and item specific non-decision time, Ter[i]. Population parameters that are estimated are omega[gamma] and omega[theta], which are scale parameters for the person boundary separation and person drift rate respectively. Parameters can be submitted to equality and/or fixed value constraints.

Usage

```
diffIRT(rt, x, model="D", constrain=NULL,
        start=NULL, se=F, control=list())
```

Arguments

`rt` A matrix of size `N` by `nit` containing the response times, where `N` is the number of subjects and `nit` is the number of items. NA's are allowed. If a given element in rt is NA, the corresponding element in `x` is also treated as missing and vice versa.

`x` A matrix of `N` by `nit` containing the responses. A's are allowed. If a given element in `rt` is NA, the corresponding element in `x` is also treated as missing and vice versa.

`model` String; Either "D" to fit the D-diffusion IRT model or "Q" to fit the Q-diffusion IRT model.

`constrain` If `NULL`, the unconstrained model is fitted. Otherwise, one could use a manual setup or a preprogrammed setup. In the manual setup a vector of length `3*nit+2` should be provided. Each element of this vector corresponds to a parameter in the model. Each parameter should be consecutively numbered from '1' onwards. Equality constraints can be imposed by giving two or more parameters the same number. Fixed parameter constraints can be imposed by putting `0`'s at the corresponding elements. The value to which those parameters should be fixed need then to be be specified in the `start` argument. The first 1 to `nit` elements in `constrain` correspond to the item boundary parameters (`ai`), the next `nit` elements correspond to the item drift parameters (`vi[i]`), the next `nit` elements correspond to the item non-decision time parameters (`ter[i]`), and the final 2 elements contain the population scale parameters of the person boundary distribution (`omega[gamma]`) and the person drift distribution (`omega[theta]`). Note that `omega[gamma]` and `omega[theta]` can be fixed to 0 to fit a model without random drift rates and random person boundaries respectively, see **Examples**. In the preprogrammed setups argument `start` should be either equal to `"ai.equal"`, `"vi.equal"`, `"ter.equal"`, or `"scale.equal"` to equate respectively `a[i]`, `v[i]`, or `Ter[i]` across items or to equate `omega[gamma]` and omega[kappa]. See **Details** for more explanation on the parameters in the different models and see **Examples** for examples on constraining parameters.

start
: If NULL the starting values are automatically chosen, see **Details**. Otherwise, a vector of length 3 x nit + 2 should be provided. Each element of this vector corresponds to a parameter from the model, similarly as in the constrain argument. For each parameter a starting value can be provided. If NA, the starting value for that parameter is chosen automatically. If a parameter is submitted to a fixed value constraint in the constrain argument, the value of that parameter should be provided (i.e., an NA is not allowed). See **Examples**.

se
: Logical; Denoting whether standard errors of the parameters should be estimated (this will increase computation time, default is F).

control
: a list of control values for the optimizations

nq Vector of any length containing the number of quadrature points that should be used in successive optimizations. Optimisation will start using the number of quadratures points specified in the first element. When converged, the resulting parameter estimates will be used as starting values in a new optimisation using the number of quadrature points from the second element. This process continues until the end of vector nq is reached. Default is c(7).

method The optimisation method used by optim. Default "BFGS".

eps Precision with which the infinite integral in the likelihood function is approximated. Default is 0.01. See **Details**.

delta Precision used in the finite difference approximation of the gradient. Default is 1e-7.

trace See optim for details and default.

fnscale See optim for details and default.

parscale See optim for details and default.

maxit See optim for details. Default is 1999.

reltol See optim for details and default.

Details

diffIRT fits either the D-diffusion or the Q-diffusion IRT model to data by minimizing -2 times the log marginal likelihood function using optim. In the diffusion IRT model the traditional parameters from the diffusion model, boundary separation and drift rate are decomposed into person and item parameters (see van der Maas et al., 2011; Tuerlinckx & De Boeck, 2005). This results in: item boundary parameter $a[i]$, item drift parameter $v[i]$, person boundary $gamma[p]$, and person drift $theta[p]$. The model for the responses in the D-diffusion IRT model is then:

$$logit(x[p,i]) = gamma[p]/a[i]x(theta[i] - v[i]),$$

where $gamma[p] >= 0$ and $ai[i] >= 0$. The Q-diffusion IRT model for the responses is given by:

$$logit(x[p,i]) = gamma[p]/a[i]x(theta[p]/v[i]).$$

where $gamma[p] >= 0$, $a[i] >= 0$, $theta[p] >= 0$, and $theta[p] >= 0$. As discussed in van der Maas et al. (2011), this setup makes the D-diffusion IRT model suitable for personality data and the Q-diffusion IRT model for ability data. The response times follow a distribution according to a Wiener process which includes - in addition to the parameters above - a non-decision parameter for each item, $Ter[i]$. For the random effects, $theta[p]$ and $gamma[p]$ scale parameters are estimated which are respectively, $omega[gamma]$ and $omega[theta]$. The joint distribution of responses and response times is evaluated using the approach by Navarro & Fuss (2009). In this approach, the infinite sum in the density function of the diffusion model is being approximated with a maximum discrepancy of `epsilon`. This discrepancy can be set using the `eps` setting within the `control` argument. To facilitated numerical estimation, the natural logarithm of the parameters that are strictly positive are estimated (i.e., `log(a[i])`, `log(Ter[i])`, `log(omega[gamma])`, `log(omega[theta])`, and `log(v[i])` in the Q-diffusion model). However, in the output the parameters are transformed back to their original scale. In addition, if `se=T`, standard errors for the original parameters are calculated from the standard errors of the transformed parameters using the delta method.

Because gamma[p] and theta[p] in the Q-diffusion model and gamma[p] in the D-diffusion model can only be positive, their population distribution is assumed to be normal on the log-scale. As a consequence, gamma[p] and theta[p] follow a log-normal distribution. Thus, in the Q-diffusion model `gamma[p]` and `theta[p]` are log-normally distributed with scale parameters `omega[gamma]` and `omega[theta]` respectively. In case of the D-diffusion model, `theta[p]` is distributed log-normally with scale parameter `omega[gamma]`, and `theta[p]` is distributed normally with scale parameter `omega[theta]` which equals the standard deviation.

By default, starting values are calculated using the EZ-diffusion model (Wagemakers, van der Maas, & Grasman, 2007).

For more details see Molenaar, Tuerlinkcx, & van der Maas (2013).

Value

An object of class `diffIRT` with values

`start.val`	starting values
`par`	parameter estimates
`par.log`	log-transformed parameters (note that for the D-diffusion model v[i] is not transformed, see **Details**.
`std.err`	if argument `se` equals `T`, it contains the standard errors of the parameters
`totLL`	value of -2 times the log-marginal likelihood at convergence
`npars`	number of parameters in the model
`AIC`	AIC
`BIC`	BIC
`sBIC`	sample size adjusted BIC
`DIC`	DIC
`subjLL`	a vector of size `N` containing the individual contribution of each subject to the marginal likelihood of the model.
`hist`	a matrix containing the history of the optimisation procedure. First row contains the starting values. Next rows contain the successive optimizations using the different number of quadrature points as specified in `control`. Final 4 columns contain `LL` (-2 times the log-marginal likelihood at convergence), `converg` (the convergence status returned by optim, `func` (the number of likelihood evaluations by `optim`), and `gradient` (the number of gradient evaluations by `optim`).

conv	integer convergence status returned by optim
nr_fail	the number of subjects for which the likelihood function was intractable (i.e., for these subjects, the log-likelihood approaches minus infinity). This could be due to the starting values, or due to extremely fast response times.

Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

References

Molenaar, D., Tuerlinkcx, F., & van der Maas, H.L.J. (2015). Fitting Diffusion Item Response Theory Models for Responses and Response Times Using the R Package diffIRT. *Journal of Statistical Software*, **66(4)**, 1-34. URL http://www.jstatsoft.org/v66/i04/.

Navarro, D.J. & Fuss, I.G. (2009). Fast and accurate calculations for first-passagetimes in Wiener diffusion models. *Journal of Mathematical Psychology*, **53**, 222-230.

Tuerlinckx, F., & De Boeck, P. (2005). Two interpretations of the discrimination parameter. *Psychometrika*, **70**, 629-650.

van der Maas, H.L.J., Molenaar, D., Maris, G., Kievit, R.A., & Borsboom, D. (2011). Cognitive psychology meets psychometric theory: On the relation between process models for decision making and latent variable models for individual differences. *Psychological Review*, **118**, 339-356.

Wagenmakers, E. J., Van Der Maas, H. L., & Grasman, R. P. (2007). An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review*, **14**, 3-22.

See Also

simdiff for simulating data according to the D-diffusion or Q-diffusion IRT model. factest for estimation of factor scores (person drift rate and person boundary separation). QQdiff and RespFit for model fit assessment. summary.diffIRT for a overview of the model estimation results, including model fit statistics. anova.diffIRT to conduct a likelihood ratio test between two nested diffIRT models. coef.diffIRT to extract parameter estimates.

Examples

```
## Not run:
 # open extraversion data
data(extraversion)
x=extraversion[,1:10]
rt=extraversion[,11:20]

# fit an unconstrained D-diffusion model
res1=diffIRT(rt,x,model="D")

# fit a model with equal item boundaries, a[i] using the manual setup
res2=diffIRT(rt,x,model="D",
  constrain=c(rep(1,10),2:11,12:21,22,23))

# fit a model with equal item boundaries, a[i] using the preprogrammed setup
res2=diffIRT(rt,x,model="D",
```

```
  constrain="ai.equal")

# fit a model where all item drift parameters,vi, are fixed to 0.
res3=diffIRT(rt,x,model="D",
  constrain=c(1:10,rep(0,10),11:20,21,22),
  start=c(rep(NA,10),rep(0,10),rep(NA,10),NA,NA))

# fit a model without random person boundary parameters.
res3=diffIRT(rt,x,model="D",
  constrain=c(1:30,0,31),
  start=c(rep(NA,30),0,NA))


## End(Not run)
```

`extraversion` *Responses and Response Times to 10 Extraversion Items*

Description

These data comprise responses (first 10 columns labelled 'X[]') and response times (next 10 columns labelled 'T[]') of 146 subjects to 10 extraversion items with a binary answer scale. Each item consists of a particular habit, e.g., 'active' and 'noisy'. Subjects were asked to indicate whether (yes/no) these habits are applicable to their personalities. Response times are in seconds.

Format

The specific habits for the 10 items are (translated from Dutch):

item 1 'active'

item 2 'noisy'

item 3 'energetic'

item 4 'enthusiastic'

item 5 'impulsive'

item 6 'jovial'

item 7 'viable'

item 8 'eupeptic'

item 9 'communicative'

item 10 'spontaneous'

References

Molenaar, D., Tuerlinkcx, F., & van der Maas, H.L.J. (2015). Fitting Diffusion Item Response Theory Models for Responses and Response Times Using the R Package diffIRT. *Journal of Statistical Software*, **66(4)**, 1-34. URL http://www.jstatsoft.org/v66/i04/.

Examples

```
data(extraversion)
x=extraversion[,1:10]            # responses, 0 for 'no', 1 for 'yes'
rt=extraversion[,11:20]          # response times in seconds
```

`factest` *Estimating factor scores for diffIRT models*

Description

This function estimates the person drift rate and person boundary separation for `diffIRT` objects.

Usage

```
factest(object,start=NULL,se=F, control=list())
```

Arguments

`object`	A `diffIRT` object for which factor scores need to be estimated.
`start`	If `NULL` starting values are automatically chosen. Otherwise, `start` should be a vector of size 2 x `N`, where `N` denotes the number of subjects. The first `N` elements correspond to the starting values for person boundary separation (`ap`), the next `N` elements correspond to the starting values for person drift rate (`vp`. `NA` are allowed.
`se`	Logical; Denoting whether standard errors of the parameters should be estimated (can be time consuming, therefore, default is `F`).
`control`	a list of control values for the optimisation **method** The optimisation method used by `optim`. Default "BFGS". **eps** See `optim` for details and default. **delta** See `optim` for details and default. **trace** See `optim` for details and default. **fnscale** See `optim` for details and default. **parscale** See `optim` for details and default. **maxit** See `optim` for details. Default is 1999. **reltol** See `optim` for details and default.

Details

`factest` returns empirical Bayes estimates of the person drift rate and the person boundary separation. See diffIRT for more explanation concerning the parameters in the D-diffusion and Q-diffusion IRT model.

Value

Function `factest` returns a matrix of parameter estimates and - if se=T - standard errors.

Author(s)

Dylan Molenaar `<d.molenaar@uva.nl>`

References

Navarro, D.J. & Fuss, I.G. (2009). Fast and accurate calculations for first-passagetimes in Wiener diffusion models. *Journal of mathematical psychology*, **53**, 222-230.

Tuerlinckx, F., & De Boeck, P. (2005). Two interpretations of the discrimination parameter. *Psychometrika*, **70**, 629-650.

van der Maas, H.L.J., Molenaar, D., Maris, G., Kievit, R.A., & Borsboom, D. (2011). Cognitive Psychology Meets Psychometric Theory: On the Relation Between Process Models for Decision Making and Latent Variable Models for Individual Differences. *Psychological Review*, **118**, 339-356.

See Also

`diffIRT` for fitting diffusion IRT models. `simdiff` for simulating data according to the D-diffusion or Q-diffusion IRT model. `QQdiff` and RespFit for model fit assesment.

Examples

```
## Not run:
 # simulate data accroding to D-diffusion model
data=simdiff(N=100,nit=10,model="D")

# fit an unconstrained model
res1=diffIRT(data$rt,data$x,model="D")

# estimate factor scores
fs=factest(res1)

## End(Not run)
```

`QQdiff` *Assessing diffIRT model fit for the response times using QQ-plots*

Description

This function plots the observed response times against the predicted response times for a `diffIRT` object.

Usage

```
QQdiff(object, items, plot=2, breaks=15, quant=NULL, maxRT=NULL)
```

Arguments

`object`	A `diffIRT` object for which the QQ-plots need to be created.
`items`	A vector denoting for which items the QQ-plots need to be created.
`plot`	Integer; 1: only QQ-plots. 2: both a QQ-plot and a histogram containing the predicted and observed distribution.
`breaks`	Number of breaks to be used in hist when `plot=2`.
`quant`	The number of quantiles to be used. If `NULL`, the number of quantiles will equal the number of subjects.
`maxRT`	The maximum response time used in finding the quantiles of the theoretical distribution. If `NULL`, twice the maximum observed response time is used for each item. Increasing `maxRT` will increase computation time and should only be used when uniroot produces errors.

Details

`QQdiff` calculates the predicted quantiles in the marginal response time distribution of the given model (D-diffusion or Q-diffusion).

Value

Function `QQdiff` returns a list with entries:

`qexp`	a vector with predicted quantiles.
`qobs`	a vector with observed quantiles.

Author(s)

Dylan Molenaar `<d.molenaar@uva.nl>`

References

Navarro, D.J. & Fuss, I.G. (2009). Fast and accurate calculations for first-passagetimes in Wiener diffusion models. *Journal of mathematical psychology*, **53**, 222-230.

Tuerlinckx, F., & De Boeck, P. (2005). Two interpretations of the discrimination parameter. *Psychometrika*, **70**, 629-650.

van der Maas, H.L.J., Molenaar, D., Maris, G., Kievit, R.A., & Borsboom, D. (2011). Cognitive Psychology Meets Psychometric Theory: On the Relation Between Process Models for Decision Making and Latent Variable Models for Individual Differences. *Psychological Review*, **118**, 339-356.

See Also

`diffIRT` for fitting diffusion IRT models. `factest` for estimation of factor scores (person drift rate and person boundary separation). `simdiff` for simulating data according to the D-diffusion or Q-diffusion IRT model.

Examples

```
## Not run:
# open rotation data
data(rotation)
x=rotation[,1:10]
rt=rotation[,11:20]

# fit an unconstrained Q-diffusion model
res1=diffIRT(rt,x,model="Q")

# make QQ-plots and histograms for items 1 to 4.
QQdiff(res1, items=1:4, plot=2, maxRT=rep(50,4))

## End(Not run)
```

RespFit — *Assessing diffIRT model fit for the responses using limited information goodness-of-fit testing.*

Description

This function uses the procedure by Maydeu-Olivares & Joe (2005) to asses the goodness-of-fit of the responses from a `diffIRT` object.

Usage

```
RespFit(object, order=2)
```

Arguments

`object`	A `diffIRT` object for which the Maydeu-Olivares & Joe test needs to be conducted.
`order`	Integer; The order of the moments to be compared see **details**.

Details

`RespFit` is an implementation of the method outlined in Maydeu-Olivares & Joe (2005). The traditional Pearson chi-square method are sub optimal in this case because in common IRT settings, contingency tables tend to be sparse. This causes the asymptotic distribution of the traditional test statistic to depart from its theoretical distribution. In the method proposed by Maydeu-Olivares & Joe, this problem is overcome by focussing on the first r moments (specified in `order`) of the observed and predicted response distributions. Choosing `order` to be equal to the number of items will result in the traditional chi-square test statistic. Commonly `order` is chosen to be small (e.g., 1 or 2).

Value

Returns an object of class `RespFit` with entries:

`Z`	A matrix with predicted statistics, observed statistics, and Z-values
`Mr`	The test statistic.
`df`	Degrees of freedom.
`order`	Order of the test statistic.

Warning

For large numbers of items, this test becomes computationally infeasible.

Note

The degrees of freedom for the test statistic differ between the D-diffusion and Q-diffusion model as for the Q-diffusion model, $a[i]$ and $v[i]$ are not simultaneously identified in response data only. See Molenaar, Tuerlinckx, & van der Maas (2013) for more details.

Author(s)

Dylan Molenaar `<d.molenaar@uva.nl>`

References

Maydeu-Olivares, A., & Joe, H. (2005). Limited and full information estimation and testing in 2n contingency tables: A unified framework. *Journal of the American Statistical Association*, **100**, 1009-1020.

Molenaar, D., Tuerlinkcx, F., & van der Maas, H.L.J. (2015). Fitting Diffusion Item Response Theory Models for Responses and Response Times Using the R Package diffIRT. *Journal of Statistical Software*, **66(4)**, 1-34. URL http://www.jstatsoft.org/v66/i04/.

See Also

`QQdiff` for model fit assessment of the response times. `diffIRT` for fitting diffusion IRT models. `factest` for estimation of factor scores (person drift rate and person boundary separation). `simdiff` for simulating data according to the D-diffusion or Q-diffusion IRT model.

Examples

```
## Not run:
 # open extraversion data
data(extraversion)
x=extraversion[,1:10]
rt=extraversion[,11:20]

# fit an unconstrained D-diffusion model
res1=diffIRT(rt,x,model="D")

# Conduct the limited-information test
```

```
RespFit(res1, 2)

## End(Not run)
```

`rotation` *Responses and Response Times to 10 Mental Rotation Items*

Description

These data comprise responses (first 10 columns labelled 'X[]') and response times (next 10 columns labelled 'T[]') of 121 subjects to 10 binary mental rotation items. The data are part of the paper by van der Maas et al (2011). These data are taken from a larger database published in Kievit (2010; see also Borst, Kievit, Thompson, & Kosslyn, 2011). Each item consists of a graphical display of two 3-dimensional objects. The second object was either a rotated version of the first object, or a rotated version of a different object. Subjects were asked whether the second object was the same as the first object (yes/no). The degree of rotation of the second object was either 50, 100, or 150 degrees. Answers are coded to be correct (1) or false (0). Response times were recorded in seconds.

Format

The specific rotation angles of the different items are:

item 1 '150'
item 2 '50'
item 3 '100'
item 4 '150'
item 5 '50'
item 6 '100'
item 7 '150'
item 8 '50'
item 9 '150'
item 10 '100'

References

Borst, G., Kievit, R. A., Thompson, W. L., & Kosslyn, S. M. (2011). Mental rotation is not easily cognitively penetrable. *Journal of Cognitive Psychology*, **23**, 60-75.

Kievit, R. A. (2010). *Representational inertia: The influence of associative knowledge on 3D mental transformations.* Unpublished manuscript, Department of Psychology, University of Amsterdam, The Netherlands.

Molenaar, D., Tuerlinkcx, F., & van der Maas, H.L.J. (2015). Fitting Diffusion Item Response Theory Models for Responses and Response Times Using the R Package diffIRT. *Journal of Statistical Software*, **66(4)**, 1-34. URL http://www.jstatsoft.org/v66/i04/.

van der Maas, H.L.J., Molenaar, D., Maris, G., Kievit, R.A., & Borsboom, D. (2011). Cognitive psychology meets psychometric theory: On the relation between process models for decision making and latent variable models for individual differences. *Psychological Review*, **118**, 339-356.

Examples

```
data(rotation)
x=rotation[,1:10]          # responses, 0 for 'false', 1 for 'correct'
rt=rotation[,11:20]        # response times in seconds
```

---

simdiff — *Simulate data according to the D-diffusion or Q-diffusion IRT model.*

Description

This function simulates responses and response time data according to the D-diffusion or Q-diffusion IRT model.

Usage

```
simdiff(N,nit,ai=NULL,vi=NULL,gamma=NULL,theta=NULL,ter=NULL,
        model="D",max.iter=19999,eps=1e-15)
```

Arguments

`N`	number of subjects.
`nit`	number of items.
`ai`	a vector of length `nit` containing the true values for the item boundary separation, a[i].
`vi`	a vector of length `nit` containing the true values for the item drift rate, v[i].
`gamma`	a vector of length `N` containing the true values for the person boundary separation, gamma[p].
`theta`	a vector of length `N` containing the true values for the person drift rate, theta[p].
`ter`	a vector of length `nit` containing the true values for the item non-decision time, ter[i].
`model`	string; Either "D" to fit the D-diffusion IRT model or "Q" to fit the Q-diffusion IRT model.
`max.iter`	maximum number of iterations for the rejection algorithm. See **Details**.
`eps`	convergence criterion for the rejection algorithm. See **Details**

Details

Function `simdiff` is an extension of the rejection algorithm outlined in Tuerlinckx et al. (2001). In this algorithm, a proposal response time is sampled from an exponential distribution. This proposal is accepted as actual response time when a specific condition is satisfied (see Eq. 16 in Tuerlinckx, 2001). As this condition requires the approximation of an infinite sum, a convergence criterion needs to be specified (see the argument `eps`). When the condition is not satisfied, a new proposal response time is sampled. This is repeated until the proposal response time is accepted or when `max.iter` has been reached.

Value

Returns a list with the following entries:

rt	the simulated matrix of response times
x	the simulated matrix of responses
ai	true values for ai[i]
vi	true values for vi[i]
gamma	true values for gamma[p]
theta	true values for theta[p]
ter	true values for ter[i]

Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

References

Tuerlinckx, F., Maris, E., Ratcliff, R., & De Boeck, P. (2001). A comparison of four methods for simulating the diffusion process. *Behavior Research Methods, Instruments & Computers*, **33**, 443-456.

See Also

diffIRT for fitting diffusion IRT models.

Examples

```
## Not run:
# simulate data accroding to D-diffusion model
data=simdiff(N=100,nit=10,model="D")


## End(Not run)
```

---

simdiffT — *Simulate data according to the traditional diffusion model.*

Description

This function simulates responses and response time data according to the traditional diffusion model for a single subject on a given number of trails. The parameters of the traditional diffusion model include: boundary separation, mean drift rate, standard deviation of drift rate, variance of the process, and ter.

Usage

```
simdiffT(N,a,mv,sv,ter,vp,max.iter=19999,eps=1e-15)
```

Arguments

N	number of trails.
a	boundary separation.
mv	mean of the normally distributed drift rates across trails.
sv	standard deviation of the normally distributed drift rate across trails.
ter	non-decision time.
vp	variance of the process, which is a scaling parameter. Default equals 1.
max.iter	Maximum number of iterations for the rejection algorithm. See **Details**.
eps	Convergence criterion for the rejection algorithm. See **Details**

Details

Function simdiffT is an application of the rejection algorithm outlined in Tuerlinckx et al. (2001) subject to normally distributed inter-trail variability in drift. In this algorithm, a proposal response time is sampled from an exponential distribution. This proposal is accepted as actual response time when a specific condition is satisfied (see Eq. 16 in Tuerlinckx, 2001). As this condition requires the approximation of an infinite sum, a convergence criterion needs to be specified (see the argument eps). When the condition is not satisfied, a new proposal response time is sampled. This is repeated until the proposal response time is accepted or when max.iter has been reached.

Value

Returns a list with the following entries:

rt	the simulated matrix of response times
x	the simulated matrix of responses

Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

References

Molenaar, D., Tuerlinkcx, F., & van der Maas, H.L.J. (2015). Fitting Diffusion Item Response Theory Models for Responses and Response Times Using the R Package diffIRT. *Journal of Statistical Software*, **66(4)**, 1-34. URL http://www.jstatsoft.org/v66/i04/.

Tuerlinckx, F., Maris, E., Ratcliff, R., & De Boeck, P. (2001). A comparison of four methods for simulating the diffusion process. *Behavior Research Methods, Instruments & Computers*, **33**, 443-456.

See Also

diffIRT for fitting diffusion IRT models.

Examples

```
## Not run:

# simulate data accroding to the traditional diffusion model
set.seed(1310)
a=2
v=1
ter=2
sdv=.3
N=10000

data=simdiffT(N,a,v,sdv,ter)
rt=data$rt
x=data$x

# fit the traditional diffusion model (i.e., a restricted D-diffusion model,
# see application 3 of the paper by Molenaar et al., 2013)

diffIRT(rt,x,model="D",constrain=c(1,2,3,0,4),start=c(rep(NA,3),0,NA))

# this constrained model is a traditional diffusion model
# please note that the estimated a[i] value = 1/a
# and that the estimated v[i] value = -v


## End(Not run)
```

`summary` *Summary function for diffIRT objects*

Description

Summarizes the modeling results of a diffIRT object.

Usage

```
## S3 method for class 'diffIRT'
summary(object, digits=3, ...)
```

Arguments

`object`	a `diffIRT` object from which a summary of the modeling results is wanted.
`digits`	integer; number of decimal places the output is rounded to.
`...`	additional parameters, currently not used.

Details

Parameter estimates are displayed, including the standard errors if these have been estimated (see the se argument of diffIRT. In addition, -2 times the log-likelihood function, AIC, BIC, sample size adjusted BIC, and DIC, are provided.

Author(s)

Dylan Molenaar <d.molenaar@uva.nl>

See Also

diffIRT for fitting diffusion IRT models.

Examples

```
## Not run:

# simulate data according to D-diffusion model
data=simdiff(N=100,nit=10,model="D")

# fit a D-diffusion model
res=diffIRT(data$rt,data$x,model="D")

# use the summary function to obtain modeling results
summary(res)

## End(Not run)
```

Index

∗ **datasets**
- brightness, 3
- extraversion, 9
- rotation, 15

∗ **models**
- anova, 2
- coef, 4
- diffIRT, 5
- factest, 10
- QQdiff, 11
- RespFit, 13
- simdiff, 16
- simdiffT, 17
- summary, 19

∗ **multivariate**
- anova, 2
- coef, 4
- diffIRT, 5
- factest, 10
- QQdiff, 11
- RespFit, 13
- simdiff, 16
- simdiffT, 17
- summary, 19

anova, 2
anova.diffIRT, *8*

brightness, 3

coef, 4
coef.diffIRT, *8*

diffIRT, *2*, *4*, 5, *10–12*, *14*, *17*, *18*, *20*

extraversion, 9

factest, *8*, 10, *12*, *14*

hist, *12*

optim, *6*, *8*, *10*

QQdiff, *8*, *11*, 11, *14*

RespFit, *8*, *11*, 13
rotation, 15

simdiff, *8*, *11*, *12*, *14*, 16
simdiffT, 17
summary, 19
summary.diffIRT, *8*

uniroot, *12*