Package 'mudfold'

December 10, 2018

Type Package

Title Multiple UniDimensional unFOLDing

Version 1.1.1

Date 2018-12-7

Description Nonparametric unfolding item response theory (IRT) model for dichotomous data (see W.H. Van Schuur (1984). Structure in Political Beliefs: A New Model for Stochastic Unfolding with Application to European Party Activists, and W.J.Post (1992). Nonparametric Unfolding Models: A Latent Structure Approach). The package implements MUDFOLD (Multiple UniDimensional unFOLDing), an iterative item selection algorithm that constructs unfolding scales from dichotomous preferential-choice data without explicitly assuming a parametric form of the item response functions. Scale diagnostics from Post(1992) and estimates for the person locations proposed by Johnson(2006) and Van Schuur(1984) are also available. This model can be seen as the unfolding variant of Mokken(1971) scaling method.

Depends R (>= 3.3.3)

Imports gtools, boot, zoo, ggplot2, reshape2, stats, utils

License GPL (>= 2)

URL https://github.com/cran/mudfold

Maintainer Spyros Balafas <s.balafas@rug.nl>

Repository CRAN

NeedsCompilation no

Author Spyros Balafas [aut, cre],
Wim Krijnen [aut],
Wendy Post [ctb],
Ernst Wit [aut]

Date/Publication 2018-12-10 08:50:03 UTC

R topics documented:

`mudfold-package` *MUDFOLD : A nonparametric unfolding item response theory model for dichotomous preferential-choice data.*

Description

This package can be used for the purpose of finding unfolding structures from selected items in tests or questionnaires. Such structures, represent the underlying ordering on a latent scale of those items. The main function of this package is called `mudfold` and fits the Van Schuur's scaling method to binary valued preference items. The method is called Multiple UniDimensional unFOLDing (MUDFOLD) and is an item selection algorithm belonging in the class of Nonparametric Item Response Theory (IRT) models.

Details

MUDFOLD is a nonparametric probabilistic model for unidimensional unfolding. Originally developed by W. Van Schuur (1984) and further extended following ideas by W.J. Post (1992) who derived testable properties for the model fit. This method can be used to analyse the categorical (binary) responses of individuals to a set of questionnaire items pressumably generated from a non-monotonic (unimodal) Item Response Function (IRF). The package incorporates the main function `mudfold` which is used to estimate the MUDFOLD scale from binary valued unfolding items. The output of the main function is a list of S3 class `"mdf"`, for which `print()`, `summary()` and `plot()` generic functions are available to the user. The package provides the user also with the function `mudfoldsim` that simulates unfolding scales using an item response function (IRF) with flexible parametrization.

The data must be given in an $n \times N$ binary `matrix` or `data.frame` with n respondents in the rows and N items in the columns. Each row of the data corresponds to the selections of the i-th individual on a set of N items. Missing values (NA) are not allowed.

Ultimate goal for MUDFOLD is to determine a unidimensional rank order of a (sub)set of items such that, they constitute an appropriate scale for measuring a common latent trait of the respondents. The estimation of the item order is done through an heuristic item selection algorithm, which tests iteratively the item fit to the scale with the use of scalability coefficients.

MUDFOLD's H coefficients of scalability are based to Loevinger's coefficient of homogeneity. In MUDFOLD, H coefficients utilize a scalability measure that is used in several criteria in the item selection algorithm. This coefficient in MUDFOLD can be calculated for triples of items,

individual items, and the total scale. Diagnostic statistics are used to assess how well the unfolding scale conforms to the assumptions of unfolding response processeses. Uncertainty estimates for the scalability measures and the diagnostic statistics both at the item and scale level are obtained by exploiting nonparametric ordinary bootstrap. A bootstrap estimate of the unfolding scale is also available.

After an unfolding scale is obtained, it can be used to estimate item locations. Two estimators are available to the user of the **mudfold** package who can choose between an estimator proposed by Van Schuur and an estimator derived by Johnson.

Author(s)

Spyros E. Balafas (auth.), Wim P. Krijnen (auth.), Wendy J. Post (contr.), Ernst C. Wit (auth.)

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

References

W.H. Van Schuur.(1984). *Structure in Political Beliefs: A New Model for Stochastic Unfolding with Application to European Party Activists*. CT Press.

W.J. Post. (1992). *Nonparametric Unfolding Models: A Latent Structure Approach*. M & T series. DSWO Press.

W.J. Post. and T.AB. Snijders (1993). *Nonparametric unfolding models for dichotomous data*. Methodika.

M.S. Johnson. (2006). Nonparametric Estimation of Item and Respondent Locations from Unfolding-type Items. Psychometrica

Examples

```
## Not run:
# Install the R package mudfold
install.packages("mudfold")

# Load the R package mudfold
library(mudfold)


## End(Not run)
```

ANDRICH *Andrich's attitude scale towards capital punishment*

Description

D. Andrich's (1988) scale designed to measure the attitude from a sample of students towards capital punishment. The data set contains the dichotomous responses of 54 students on 8 statements concerning capital punishment.

Usage

```
data(ANDRICH)
```

Format

A data frame with 54 observations on the following 8 variables.

`HIDEOUS` a column vector containing the binary responses on the statement:
"Capital punishment is one of the most hideous practices of our time"

`LIFESACRED` a column vector containing the binary responses on the statement:
"The state cannot teach the sacredness of human life by destroying it"

`INEFFECTIV` a column vector containing the binary responses on the statement:
"Capital punishment is not an effective deterrent to crime"

`DONTBELIEV` a column vector containing the binary responses on the statement:
"I do not believe in capital punishment but i am not sure it is not necessary"

`WISHNOTNEC` a column vector containing the binary responses on the statement:
"I think capital punishment is necessary but i wish it were not"

`MUSTHAVEIT` a column vector containing the binary responses on the statement:
"Until we find a more civilized way to prevent crime we must have capital punishment"

`DETERRENT` a column vector containing the binary responses on the statement:
"Capital punishment is justified because it does act as a deterrent to crime"

`CRIMDESERV` a column vector containing the binary responses on the statement:
"Capital punishment gives the criminal what he deserves"

Details

The persons who responded to the statements for the analysis were 54 graduate students taking an introductory course in educational measurement and statistics. They responded simply by agreeing (1) or disagreeing (0) with each statement, with no restrictions placed on how many statements should receive an *Agree* response.

Source

D. Andrich. (1988). *The Application of an Unfolding Model of the PIRT Type to the Measurement of Attitude*. Applied psychological measurement 12.1: 33-51.

References

D. Andrich. (1988). *The Application of an Unfolding Model of the PIRT Type to the Measurement of Attitude*. Applied psychological measurement 12.1 (1988): 33-51.

W.J. Post. (1992). *Nonparametric Unfolding Models: A Latent Structure Approach*. M & T series. DSWO Press.

W.J. Post. and T.AB. Snijders. (1993). *Nonparametric unfolding models for dichotomous data*. Methodika.

Examples

```
## Not run:
data(ANDRICH)
str(ANDRICH)

## End(Not run)
```

`as.mudfold` *Function for calculating MUDFOLD statistics for a given unfolding scale.*

Description

This function calculates the MUDFOLD statistics for data whose columns are assumed to be ranked to the order they are provided. The resulting object from the `as.mudfold` function is an object of S3 class `"mdf"`, for which generic functions `print`, `summary`, and `plot` are available.

Usage

```
as.mudfold(data,estimation="rank")
```

Arguments

`data`	: A binary `matrix` or `data.frame` containing the responses of `nrow(data)` persons to `ncol(data)` items. Missing values in `data` are not allowed.
`estimation`	: This argument controls the nonparametric estimation method for person locations. By deafult this argument equals to `"rank"` and implies that Van Schuur's estimator will be used in order to infer the person parameters. The user can set this argument to `"quantile"` and then an estimator proposed by Johnson is applied.

Details

The function `as.mudfold` calculates MUDFOLD statistics for a given scale. Descriptive statistics, observed errors, expected errors, scalability coefficients, iso statistic values, are calculated for items and the scale. The user can obtain a summary table for the given scale with the `summary` function which is designed for `"mdf"` class objects.

Value

The function `as.mudfold` returns a list with the same components as the `mudfold` function except the information that concerns the item selection algorithm. The list contains the following:

`CALL`	A list where its components provide information for the function call.
`CHECK`	A list where its components provide information from the data checking step.
`DESCRIPTIVES`	A list with descriptive statistics for the `data`.

`MUDFOLD_INFO` A list with three main components. The first component is called `triple_stats` and is a list where in each element contains the observed errors, expected errors, and scalability coefficients for each item triple. The second element is called `first_step` and informs the user that the first step of the item selection algorithm is not applied in the `as.mudfold` function. The third element of this list is called `second_step` and is also a list with the MUDFOLD statistics and parameter estimates for the given scale.

Author(s)

Spyros E. Balafas (auth.), Wim P. Krijnen (auth.), Wendy J. Post (contr.), Ernst C. Wit (auth.)

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

References

W.H. Van Schuur.(1984). *Structure in Political Beliefs: A New Model for Stochastic Unfolding with Application to European Party Activists*. CT Press.

W.J. Post. (1992). *Nonparametric Unfolding Models: A Latent Structure Approach*. M & T series. DSWO Press.

W.J. Post. and T.AB. Snijders. (1993).Nonparametric unfolding models for dichotomous data. Methodika.

M.S. Johnson. (2006). Nonparametric Estimation of Item and Respondent Locations from Unfolding-type Items. Psychometrica

See Also

`mudfold`

Examples

```
## Not run:
## pick a number for setting the seed
n.seed <- 11

## Simulate an unfolding scale
simulation <- mudfoldsim(N=6, n=100, seed=n.seed)

## get the data
dat <- simulation$dat

## true order
true_order <- simulation$true_ord

## check MUDFOLD statistics for the random simulated rank order
mud_stats1 <- as.mudfold(dat)

# get the summary
summary(mud_stats1)

## check MUDFOLD statistics for the true item rank order
```

```
mud_stats2 <- as.mudfold(dat[,true_order])

# get the summary for the true item rank order
summary(mud_stats2)

## End(Not run)
```

EURPAR2 *Preferences of European party activists.*

Description

European party activists preferences for two political parties in the European parliament in 1980. A sample consisted of 1786 individuals are asked to pick 2 out of 6 political parties from the European parliament.

Usage

```
data("EURPAR2")
```

Format

A data frame with 1786 observations (responses) on the following 6 binary valued items.

`communists` Communistic political party;

`socdemocr` Social Democratic political party;

`demprogres` Progressive Democratic political party;

`liberals` Liberal Democratic political party;

`christians` Christian Democratic political party;

`conservat` Conservative political party;

Details

The data have been first studied by Van Schuur (1984) and further by W. J. Post (1992).

Source

W.H. Van Schuur.(1984). *Structure in Political Beliefs: A New Model for Stochastic Unfolding with Application to European Party Activists*. CT Press.

References

W.H. Van Schuur.(1984). *Structure in Political Beliefs: A New Model for Stochastic Unfolding with Application to European Party Activists*. CT Press.

W.J. Post. (1992). *Nonparametric Unfolding Models: A Latent Structure Approach*. M & T series. DSWO Press.

Examples

```
data(EURPAR2)
str(EURPAR2)
```

Loneliness *De Jong-Gierveld loneliness scale*

Description

De Jong-Gierveld loneliness scale that consists of eleven ordinal items. Five of these items are positively formulated and six are negatively formulated. Each of the items has three possible response categories.

Usage

```
data(Loneliness)
```

Format

A data frame with 3987 observations on the following 11 variables.

`A` : a column vector containing the ordinal responses on the statement:
"There is always someone I can talk to about my day to day problems (+)"

`B` a column vector containing the ordinal responses on the statement:
"I miss having a really close friend (-)"

`C` a column vector containing the ordinal responses on the statement:
"I experience a general sense of emptiness (-)"

`D` a column vector containing the ordinal responses on the statement:
"There are plenty of people I can lean on in case of trouble (+)"

`E` a column vector containing the ordinal responses on the statement:
"I miss the pleasure of company of others (-)"

`F` a column vector containing the ordinal responses on the statement:
"I find my circle of friends and acquaintances too limited (-)"

`G` a column vector containing the ordinal responses on the statement:
"There are many people that I can count on completely (+)"

`H` a column vector containing the ordinal responses on the statement:
"There are enough people that I feel close to (+)"

`I` a column vector containing the ordinal responses on the statement:
"I miss having people around (-)"

`J` a column vector containing the ordinal responses on the statement:
"Often I feel rejected (-)"

`K` a column vector containing the ordinal responses on the statement:
"I can call on my friends whenever I need them (+)"

Details

Each item in the scale has three possible levels of response, i.e., "no" (=1), "more or less" (=2), "yes" (=3). The data is a subset of the NESTOR study (see C. P. Knipscheer, J. d. Jong-Gierveld, T. G. van Tilburg, P. A. Dykstra, et al. (1995))

Source

G. J. De Jong and T. van Tilburg (1999). Manual of the loneliness scale. Amsterdam: VU University Amsterdam.

References

C. P. Knipscheer, J. d. Jong-Gierveld, T. G. van Tilburg, P. A. Dykstra, et al. (1995). Living arrange-ments and social networks of older adults.Amsterdam: VU University Amsterdam.

J. de Jong-Gierveld and F. Kamphuls (1985). The development of a rasch-type loneliness scale.Applied psychological measurement, 9(3):289-299.

G. J. De Jong and T. van Tilburg (1999). Manual of the loneliness scale. Amsterdam: VU University Amsterdam.

W. J. Post, M. A. van Duijn, and B. van Baarsen (2001). Single-peaked or monotone tracelines? onthe choice of an irt model for scaling data. InEssays on item response theory, pages 391-414.Springer.

Examples

```
## Not run:
data(Loneliness)
str(Loneliness)

## End(Not run)
```

`mudfold` *MUDFOLD: Van Schuur's nonparametric IRT model for dichotomous responses that have been generated by an unfolding process.*

Description

This function is used to fit a unidimensional unfolding scale to the responses of individuals on a set of categorically scored attitudinal items. Fitting is done through Van Schuur's scaling algorithm that determines if a set of items are indicators of the same unobserved latent contstruct such as preference, attitude, ideology etc. Core in this model are the scalability coefficients that are used to assess the fit of the scale and the items to the data.

Diagnostic statistics that are used to test the model assumptions are borrowed from the nonparametric unfolding model of Post(1992). Uncertainty estimates for the scalability coefficients and the diagnostic statistics both for the scale and the individual items are obtained using nonparametric ordinary bootstrap. A bootstrap estimate of the scale is obtained as the most frequently observed scale in R bootstrap iterations.

Usage

```
mudfold(data, estimation="rank", lambda1=0.3, lambda2=0, start=NULL, nboot=NULL, ...)
```

Arguments

`data`	: A binary matrix or data frame containing the responses of `nrow(data)` persons to `ncol(data)` items. Missing values in `data` are not allowed.
`estimation`	: This argument controls the nonparametric estimation method for person locations. By deafult this argument equals to `"rank"` and implies that Van Schuur's estimator will be used in order to estimate person parameters. The user can set this argument to `"quantile"` and then an estimator proposed by Johnson is applied to obtain the person locations.
`lambda1`	: User specified numerical value that is used as a lower boundary for the scalability criterion of the first step of the item selection algorithm, and in the item scalability criterion at the end of the scale expansion. Default value is $\lambda_1 = 0.3$ but it can be any value between $-\infty$ and 1 (i.e., $\lambda_1 \in (-\infty, 1]$). The higher the value of λ_1 the stricter the scalability criteria of the algorithm.
`lambda2`	: User specified numerical value that controls explicitly the first scalability criterion of the scale expansion. In the default settings $\lambda_2 = 0$, however, the user can choose a negative value for λ_2, which leads to less strict scalability criterion in the beginning of the scale expansion.
`start`	: An ordered character vector with item names from `colnames(data)`. The length of this vector should be greater than or equal to 3 and less than or equal to `ncol(data)`. This ordered item set is used as a startset for the scale extension phase of MUDFOLD method. If `start=NULL` the standard MUDFOLD method is fitted to the data.
`nboot`	: Argument that controls the number of bootstrap iterations. If `nboot=NULL` (default) no bootstrap is applied.
`...`	: Any additional arguments that are passed to the `boot` function from the package **boot**. See `?boot::boot`.

Details

This function incorporates a two-step algorithm that determines an unfolding scale from observed binary `data`. In the first step of the algorithm the best minimal scale that consists of three items is determined. In the second step, the minimal scale from the first step is expanded iteratively by adding the best fitting item in each iteration. The first step of the algorithm can be skiped with the argument `start` which can be used for setting manually an item rank order that will be extended in the second step of the item selection algorithm. The resulting scale consists of the best `m` fitting items based on scalability criteria (where $\texttt{m} \leq \texttt{ncol(data)}$).

In `mudfold` function, the user can specify a value λ_1 that will be used as a lower bound in the scalability criteria of the MUDFOLD algorithm. By default, the lower bound for the scalability coefficients is `lambda1=0.3`. The user can choose a second value λ_2 that will be used as a lower bound only for the second step of the algorithm (by default, `lambda2=0`). The parameter λ_2 is used mostly, in order to relax the first scalability criterion of the second step. Generally, values greater than 0.3 for λ_1, and λ_2 lead to very strict criteria while negative values relax these criteria.

Uncertainty estimates of the MUDFOLD statistics can be calculated with the argument nboot of the mudfold function. When nboot is an integer then nboot bootstrap iterations will run to obtain the variance parameter for each MUDFOLD statistic.

Moreover, the user is able to choose between two nonparametric estimation methods in order to obtain person parameters that are estimated using the item ranks from the MUDFOLD algorithm. The default setting (i.e., estimation="rank") uses an estimation proposed by Van Schuur(1984) based on item ranks. Alternatively, an estimation method described by Johnson(2005), which uses item quantiles for estimating person parameters, can be used by setting estimation="quantile".

Value

The function mudfold returns a list of class "mdf" with the following components:

CALL	A list where its components provide information for the function call.
CHECK	A list where its components provide information from the data checking step.
DESCRIPTIVES	A list with descriptive statistics for the data.
MUDFOLD_INFO	A list with three main components. The first component is called triple_stats and is a list where in each element contains the observed errors, expected errors, and scalability coefficients for each item triple. The second element is a list called first_step and contains the results of the first step of the MUDFOLD item selection algorithm. The third element of this list is called second_step and is a list with the MUDFOLD statistics and parameter estimates for the given scale.

If bootstrap is applied, then, an additional component is included in the output. This component is called BOOTSTRAP and is a list that contains the output of nboot bootstrap iterations.

Author(s)

Spyros E. Balafas (auth.), Wim P. Krijnen (auth.), Wendy J. Post (contr.), Ernst C. Wit (auth.)

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

References

W.H. Van Schuur.(1984). *Structure in Political Beliefs: A New Model for Stochastic Unfolding with Application to European Party Activists*. CT Press.

W.J. Post. (1992). *Nonparametric Unfolding Models: A Latent Structure Approach*. M & T series. DSWO Press.

W.J. Post. and T.AB. Snijders. (1993).Nonparametric unfolding models for dichotomous data. Methodika.

M.S. Johnson. (2006). Nonparametric Estimation of Item and Respondent Locations from Unfolding-type Items. Psychometrica

Examples

```
## Not run:
####################################
#### MUDFOLD method on real data ####
####################################



#############################################################################
###### MUDFOLD method on ANDRICH data (see Post and Snijders pp.147) ######
#############################################################################
data(ANDRICH)
## fit MUDFOLD on ANDRICH data ##
fit_andr <- mudfold(ANDRICH)

## generic functions for the S3 class .mdf object fit ##
## print.mdf
print(fit_andr)
## summary.mdf
summary(fit_andr, diagnostics=TRUE)
## plot.mdf
plot(fit_andr)


## fit MUDFOLD on ANDRICH data with bootsrap ##
fit_andr_boot <- mudfold(ANDRICH, nboot=100)

## generic functions for the S3 class .mdf object fit ##
## print.mdf
print(fit_andr_boot)
## summary.mdf
summary(fit_andr_boot, boot=TRUE, diagnostics=FALSE)
## plot.mdf
plot(fit_andr_boot)

##########################################
###### MUDFOLD method on EURPAR2 data ######
##########################################
data("EURPAR2")

## fit MUDFOLD on EURPAR2 data ##
fit_eurp <- mudfold(EURPAR2)

## print
print(fit_eurp,Diagnostics = TRUE)

## summary
summary(fit_eurp)

## plot
plot(fit_eurp)
```

```
##########################################
###### MUDFOLD method on Plato7 data ######
##########################################

data("Plato7")

## transform to binary data
## using as threshold the mean
## per row of Plato7

dat_plato <- pick(Plato7)

## fit MUDFOLD on Plato7 data ##
fit_plato <- mudfold(dat_plato, nboot=1000)

## print
print(fit_plato)

## summary
summary(fit_plato, boot=TRUE)

## plot
plot(fit_plato, plot.type="scale")
plot(fit_plato, plot.type="IRF")
plot(fit_plato, plot.type="persons")


##########################################
#### MUDFOLD method on simulated data ####
##########################################

### Data with the responses of
### n=3000 on p=20 items

simulation1 <- mudfoldsim(N=20, n=3000, gamma1=2, gamma2=-10, zeros=FALSE,seed = 1)
dat_sim1 <- simulation1$dat

## fit MUDFOLD on simulated data ##
fit.sim1 <- mudfold(dat_sim1)

# print
fit.sim1

# summary
summary(fit.sim1)

# plot
plot(fit.sim1)

### Data with the responses of
### n=3000 on N=26 items

simulation2 <- mudfoldsim(N=26, n=3000, gamma1=2, gamma2=-10, zeros=FALSE,seed = 1)
```

```
dat_sim2 <- simulation2$dat

## fit MUDFOLD on simulated data ##
fit.sim2 <- mudfold(dat_sim2)

# print
fit.sim2

# summary
summary(fit.sim2)

# plot
plot(fit.sim2, plot.type="scale")
plot(fit.sim2, plot.type="IRF")
plot(fit.sim2, plot.type="persons")


## End(Not run)
```

`mudfoldsim` *Function for constructing artificial item response data generated under an unfolding response process. Unfolding processes model the proximity (distance) between person and item parameters.*

Description

`mudfoldsim` function simulates unfolding data following a unimodal parametric function with flexible set up. User can control the number of respondents, the number of items and fixed parameters of the Item Response Function (IRF) under which the responses are generated. Moreover, the user of the **mudfold** package can allow (or not) individuals that are endorsing no items.

Usage

```
mudfoldsim(N, n, gamma1=5, gamma2=-10, zeros=FALSE, parameters="normal", seed=NULL)
```

Arguments

`N`	: This argument specifies the number of items (stimuli).
`n`	: Argument which allows the user to specify the number of respondents in the simulated data.
`gamma1`	: Parameter which is used in the IRF under which the data is generated. Default value is 5.
`gamma2`	: Parameter which is used in the IRF under which the data is generated. Default value is -10.
`zeros`	: Logical argument. If `zeros=FALSE` (default), only individuals who endorse at least one item are allowed. Else, if `zeros=TRUE` individuals with no response are allowed.

`parameters` : A character string that controls the distribution of the person parameters. If `parameters="normal"` (default), individual parameters are drawn from a standard normal distribution. If `parameters="uniform"`, the person parameters are uniformly drawn between the minimum and the maximum item parameters respectively.

`seed` : An integer to be used in the `set.seed` function. If `seed=NULL` (default), then the seed is not set.

Details

For simulating the response of an individual i with scale parameter θ_i to an item j with scale parameter β_j we use the function $P(X_j = 1 \mid \theta_i, \beta_j) = \frac{1}{1+e^{-\gamma_1-\gamma_2(\theta_i-\beta_j)^2}}$. The parameters θ_i, β_j can be samples sampled both from a standard normal distribution, i.e., $\theta \sim \mathcal{N}(0,1)$, and $\beta \sim \mathcal{N}(0,1)$ or the the person parameters will be sampled uniformly within the range of the item parameters.

Value

a list with 11 components.

`obs_ord` : A character vector with the items in the simulated order.

`true_ord` : A character vector with the items in the true order in which they constitute an unfolding scale.

`items` : An integer corresponding to the number of the simulated items.

`sample` : An integer corresponding to the number of the simulated respondents.

`gamma1` : A value that corresponds to the parameter γ_1 of the IRF.

`gamma2` : A value that corresponds to the parameter γ_2 of the IRF.

`seed` : An integer that corresponds to the seed number that is going to be used in the `set.seed` function.

`dat` : data frame containing the binary responses of `n` subjects on `K` items under a parametric Item Response Function.

`probs` : A matrix containing the probabilities of positive response from `n` subjects on `K` items under a parametric Item Response Function.

`item.patameters` : The simulated item parameters that have been used for sampling the data.

`subject.parameters` : The simulated subject parameters that have been used for sampling the data.

Author(s)

Spyros E. Balafas (auth.), Wim P. Krijnen (auth.), Wendy J. Post (contr.), Ernst C. Wit (auth.)

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

References

W.H. Van Schuur.(1984). *Structure in Political Beliefs: A New Model for Stochastic Unfolding with Application to European Party Activists*. CT Press.

W.J. Post. (1992). *Non parametric Unfolding Models: A Latent Structure Approach*. M & T series. DSWO Press.

W.J. Post. and T.AB. Snijders. (1993).Non parametric unfolding models for dichotomous data. Methodika.

Examples

```
## Not run:
## Simulate 5 different scenarios

n.seed <- 10

sim1 <- mudfoldsim(N=6, n=100, gamma1=5, gamma2=-10, zeros=FALSE,seed=n.seed)
sim2 <- mudfoldsim(N=10,n=1000,gamma1=10,gamma2=-100,zeros=FALSE,seed=n.seed)
sim3 <- mudfoldsim(N=15,n=2000,gamma1=50,gamma2=-100,zeros=FALSE,seed=n.seed)
sim4 <- mudfoldsim(N=30,n=2000,gamma1=50,gamma2=-100,zeros=FALSE,seed=n.seed)
sim5 <- mudfoldsim(N=50,n=2000,gamma1=50,gamma2=-100,zeros=FALSE,seed=n.seed)


dat1 <- sim1$dat
dat2 <- sim2$dat
dat3 <- sim3$dat
dat4 <- sim4$dat
dat5 <- sim5$dat

fit1 <- mudfold(dat1)
fit1
fit2 <- mudfold(dat2)
fit2
fit3 <- mudfold(dat3)
fit3
fit4 <- mudfold(dat4)
fit4
fit5 <- mudfold(dat5)
fit5

## End(Not run)
```

`pick` *Transform items to preference binary data.*

Description

Function `pick` can be used to transform quantitative or ordinal type of variables, into binary form (i.e., `0`,1). When `byItem=FALSE`, then the underlying idea is that the individual selects those items

with the higher preference. This is done through user provided cut-off values, or by assuming a *pick* k *out of* N response process, where, each continuous response vector takes a 1 at its k higher values. Binarization can be performed row-wise (default) or column-wise.

Usage

```
pick(data , k=NULL, cutoff=NULL, byItem=FALSE)
```

Arguments

data	: A matrix or data frame containing the continuous or discrete responses of nrow(data) persons/judges to ncol(data) items. Missing values in data are not allowed.
k	: An integer ($1 \leq$ k $\leq$ ncol(data)) that restricts the number of items a person can pick (default k=NULL). This argument, is used if one wants to transform the data into *pick* k *out of* N form. If k is provided by the user, cutoff should be NULL and vice verca. By default, this process is applied to the matrix data rowise. The user can restrict the number
cutoff	:The value(s) that will be used as thresholds. The length of this argument should be equal to 1 (the same threshold for all rows (or columns) of data) or equal to K where K=nrow(data) or K=ncol(data) when byItem=TRUE.
byItem	: logical argument. If byItem=TRUE, the binarization is performed columnwise. In the default byItem=FALSE, the function determines the ones rowise.

Details

Binary transformation of continuous or discrete variables with $\rho \geq 3$ number of levels. Two different methods are available for the transformation.

The first method uses the argument k in the pick function, and assumes a *pick* k *out of* N response process. Such type of response processes are met in surveys and questionnaires, in which respondents are asked to pick exactly the k most preferred items. The value for k is an integer between 1 and ncol(data). By choosing an integer for k, this function "picks" the k higher values in each row (if byItem=FALSE) of data. The k higher values in each row become 1 and the rest ncol(data)-k elements are set to 0. Obviously, if k=ncol(data), then the resulting matrix will only consists of 1's and no 0's.

The second method is based on thresholding in order to binarize the data. For this method, the user should provide threshold(s) with the parameter cutoff in the pick function (default cutoff=NULL). If one value is provided in the cutoff parameter, i.e., cutoff=α, then α is used as threshold in each row i (if byItem=FALSE) of the data matrix data such that, any value greater than or equal to cutoff in row i becomes 1 and 0 else. Additionally, the user can provide row (or column) specific cut off values, i.e., cutoff=α with $\alpha = (\alpha_1, ..., \alpha_K)$ where α_i is the cut-off value for the row or column i. In this case, if $x_{ij} \geq \alpha_i$ then $x_{ij} = 1$ and $x_{ij} = 0$ else.

The two methods cannot be used simultaneously. Only one of the parameters k and cutoff can be different than NULL each time. If both parameters are equal NULL (default), then a row specific cut off is determined automatically for each row i of data, such that, $\alpha_i = \bar{data}_i$. The binarization is performed by row of data, except the case, byItem=TRUE.

When the argument k is used, it can be the case that more than k values can be picked (i.e., ties). In this case, the choice on which item will be picked is being made after we add a small amount of noise in each observation of row or column i. This is done with the function `jitter`.

Value

Binary valued (i.e., 0-1) data with the same dimensions as the input.

Author(s)

Spyros E. Balafas (auth.), Wim P. Krijnen (auth.), Wendy J. Post (contr.), Ernst C. Wit (auth.)

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

Examples

```
## Not run:
### simulate some data with 3 discrete variables with three levels
### and 1 variable with 4 levels
d1 <- cbind(sample(1:3,20,replace = TRUE),
            sample(1:3,20,replace = TRUE,prob = c(0.3,0.3,0.4)),
            sample(1:3,20,replace = TRUE,prob = c(0.2,0.4,0.4)),
            sample(1:4,20,replace = TRUE,prob = c(.1,.3,.4,.2)))


### apply pick on d1 ###
# binarize at the mean of
# each row and column
d1_rowmean <- pick(d1)
d1_colmean <- pick(d1,byItem = TRUE)

# binarize at the cutoff=2
d1_cut <- pick(d1,cutoff = 2,byItem = TRUE)

# binarize at different cutoffs (per row)
# for example at the median of each row
med_cuts <- apply(d1,1,median)
d1_cuts <- pick(d1,cutoff = med_cuts)

# binarize at different cutoffs (per column)
# for example at the median of each column
med_cuts_col <- apply(d1,2,median)
d1_cuts_col <- pick(d1,cutoff = med_cuts_col,byItem = TRUE)


# binarize at the k=2 higher values
# per row and column
d1_krow <- pick(d1,k = 2)
d1_kcol <- pick(d1,k = 2,byItem = TRUE)

## End(Not run)
```

Plato7 *Plato's Seven Works*

Description

This dataset contains statistical information about Plato's seven works. The underlying problem to this dataset is the fact that the chronological order of Plato's works is unknown. Scholars only know that Republic was his first work, and Laws his last work. For each work, Cox and Brandwood (1959) extracted the last five syllables of each sentence. Each syllable is classified as long or short which gives 32 types. Consequently, we obtain a percentage distribution across the 32 scenarios for each of the seven works. The dataset has been borrowed from the package **smacof** (De Leeuw and Mair, 2009).

Usage

```
data(Plato7)
```

Format

Data frame containing syllable percentages of Plato's 7 works.

References

Cox, D. R. & Brandwood, L. (1959). On a discriminatory problem connected with the work of Plato. Journal of the Royal Statistical Society (Series B), 21, 195-200.

De Leeuw, J.& Mair, P. (2009). Multidimensional Scaling Using Majorization: SMACOF in R. Journal of Statistical Software, 31(3), 1-30. URL http://www.jstatsoft.org/v31/i03/.

Examples

```
## Not run:
data(Plato7)
str(Plato7)

## End(Not run)
```

plot.mdf `plot` *function for* `"mdf"` *class objects.*

Description

Generic function for plotting S3 class `"mdf"` objects. This function, is plotting the rows of the conditional adjacency matrix (CAM) which are nonparametric estimates of the item response functions. The plot is produced using the `ggplot` function from the package **ggplot2**.

Usage

```
## S3 method for class 'mdf'
plot(x, select=NULL, plot.type="IRF", ...)
```

Arguments

x	Object of class mdf
select	: in this argument the user can provide a subset of items he would like them to be explicitly plotted. If select=NULL the estimated IRF for every item in the scale is plotted. When plot.type="persons" this argument is ignored.
plot.type	: Determines the type of plot that is returned. By default, plot.type="IRF", which returns the estimated IRFs for the items in the MUDFOLD scale. The user can set plot.type="scale" in order to get plotted the unidimensional MUDFOLD scale. Setting plot.type="persons" will return the distribution of the person parameters on the latent scale.
...	Other arguments passed on to ggplot plotting method.

Details

The plot method is used to obtain a graphical representation of the estimated rank order of the items, the item response functions, and the distribution of the person parameters. As estimates of the IRFs are considered the rows of the CAM. For interpolating the missing diagonal elements of the CAM, we make use of the na.approx function from the package **zoo**.

Author(s)

Spyros E. Balafas (auth.), Wim P. Krijnen (auth.), Wendy J. Post (contr.), Ernst C. Wit (auth.)

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

References

W.H. Van Schuur.(1984). *Structure in Political Beliefs: A New Model for Stochastic Unfolding with Application to European Party Activists*. CT Press.

W.J. Post. (1992). *Nonparametric Unfolding Models: A Latent Structure Approach*. M & T series. DSWO Press.

W.J. Post and T.AB. Snijders. (1993).Nonparametric unfolding models for dichotomous data. Methodika.

A. Zeileis and G. Grothendieck. (2005). zoo: S3 Infrastructure for Regular and Irregular Time Series. Journal of Statistical Software, 14(6), 1-27. doi:10.18637/jss.v014.i06

H. Wickham. (2009). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.

H. Wickham. (2007). Reshaping Data with the reshape Package. Journal of Statistical Software, 21(12), 1-20. URL http://www.jstatsoft.org/v21/i12/.

Examples

```
## Not run:
data(ANDRICH)
fit <- mudfold(ANDRICH)
plot(fit, plot.type= "scale")
plot(fit, plot.type= "IRF")
plot(fit, plot.type= "persons")
plot(fit, select="DONTBELIEV", plot.type= "IRF")

## End(Not run)
```

`print.mdf` `print` *method for* `"mdf"` *class objects resulted from the* `mudfold` *function.*

Description

S3 generic function for printing `"mdf"` class objects.

Usage

```
## S3 method for class 'mdf'
print(x, ...)
```

Arguments

`x`	Object of class `"mdf"`
`...`	further arguments passed on to the `print` method.

Author(s)

Spyros E. Balafas (auth.), Wim P. Krijnen (auth.), Wendy J. Post (contr.), Ernst C. Wit (auth.)

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

References

W.H. Van Schuur.(1984). *Structure in Political Beliefs: A New Model for Stochastic Unfolding with Application to European Party Activists*. CT Press.

W.J. Post. (1992). *Nonparametric Unfolding Models: A Latent Structure Approach*. M & T series. DSWO Press.

W.J. Post. and T.AB. Snijders (1993). *Nonparametric unfolding models for dichotomous data*. Methodika.

Examples

```
## Not run:
data(ANDRICH)
fit <- mudfold(ANDRICH)
fit
print(fit)

## End(Not run)
```

---

summary.mdf summary *method for S3 class* "mdf" *objects.*

Description

Generic function that is used in order to summarize information from "mdf" class objects.

Usage

```
## S3 method for class 'mdf'
summary(object, boot=FALSE, diagnostics=FALSE, type="perc", ...)
```

Arguments

object	: Object of class "mdf" resulted from the function mudfold or as.mudfold.
boot	: This argument applies when the nboot argument in the mudfold function is not NULL. If boot=FALSE (default) then no bootstrap information is returned by the summary. When boot=TRUE, confidence intervals, standard errors, biases, calculated from the bootstrap iterations for each parameter are given with the output. If the bootstrap estimate of the scale does not agree with the the scale of the item selection algorithm, then a summary of the bootstrap estimate of the scale is also given in the output.
diagnostics	: Logical argument that controls if scale diagnostics such as the conditional adjacency matrix and the star matrix will be included in the output. If diagnostics=FALSE (default) no diagnostics are returned. If diagnostics=TRUE,the conditional adjacency matrix and the star matrix are also included in the output.
type	: A string that determines the type of confidence intervals that will be calculated. This argument is passed to the boot.ci function from the R package **boot**. Available options are c("norm","basic", "perc", "bca"). See ?boot.ci for more information.
...	Other arguments passed on to the function boot.ci from the R package **boot**.

Details

A summary of the MUDFOLD scale that has been calculated with the mudfold function.

Value

The output of the `summary.mdf()` is a list with two main components. The first component of the list is a `data.frame` with scale statistics and the second component is a list with item statistics. If `diagnostics=TRUE` another component with diagnostic matrices is also included in the output. When the bootstrap scale estimate does not agree with the obtained MUDFOLD estimate a summary of the bootstrap scale will be given in the output.

Author(s)

Spyros E. Balafas (auth.), Wim P. Krijnen (auth.), Wendy J. Post (contr.), Ernst C. Wit (auth.)

Maintainer: Spyros E. Balafas (<s.balafas@rug.nl>)

References

W.H. Van Schuur.(1984). *Structure in Political Beliefs: A New Model for Stochastic Unfolding with Application to European Party Activists*. CT Press.

W.J. Post. (1992). *Nonparametric Unfolding Models: A Latent Structure Approach*. M & T series. DSWO Press.

W.J. Post. and T.AB. Snijders (1993). *Nonparametric unfolding models for dichotomous data*. Methodika.

Examples

```
## Not run:
data(ANDRICH)
fit <- mudfold(ANDRICH, nboot=100)
summary(fit, boot=TRUE, diagnostics=TRUE)
summary(fit, boot=TRUE, diagnostics=FALSE)
summary(fit, boot=FALSE, diagnostics=TRUE)
summary(fit, boot=FALSE, diagnostics=FALSE)


## End(Not run)
```

Index