Package 'difNLR'

May 3, 2023

Type Package

Title DIF and DDF Detection by Non-Linear Regression Models

Version 1.4.2-1

Date 2023-05-02

Author Adela Hladka [aut, cre],
Patricia Martinkova [aut],
Karel Zvara [ctb]

Maintainer Adela Hladka <hladka@cs.cas.cz>

Depends R (>= 3.1)

Imports calculus, ggplot2 (>= 3.4.0), msm, nnet, plyr, stats, VGAM

Suggests ShinyItemAnalysis

Description Detection of differential item functioning (DIF) among dichotomously scored items and differential distractor functioning (DDF) among unscored items with non-linear regression procedures based on generalized logistic regression models (Hladka & Martinkova, 2020, <doi:10.32614/RJ-2020-014>).

License GPL-3

LazyData yes

RoxygenNote 7.2.3

BugReports https://github.com/adelahladka/difNLR/issues

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2023-05-03 10:50:02 UTC

R topics documented:

`difNLR-package` *DIF and DDF Detection by Non-Linear Regression Models.*

Description

The difNLR package contains method for detection of differential item functioning (DIF) based on non-linear regression. Both uniform and non-uniform DIF effects can be detected when considering one focal group. The method also allows to test the difference in guessing or inattention parameters between reference and focal group. DIF detection method is based either on likelihood-ratio test, F-test, or Wald's test of a submodel. Package also offers methods for detection of differential distractor functioning (DDF) based on multinomial log-linear regression model and newly methods for DIF detection among ordinal data via adjacent category logit and cumulative logit regression models.

Details

Package: difNLR
Type: Package
Version: 1.4.2-1
Date: 2023-05-03
Depends: R (>= 3.1)
Imports: calculus, ggplot2 (>= 3.4.0), msm, nnet, plyr, stats, VGAM
Suggests: ShinyItemAnalysis
License: GPL-3
BugReports: https://github.com/adelahladka/difNLR/issues
Encoding: UTF-8

Functions

- ddfMLR
- difNLR
- difORD
- estimNLR
- formulaNLR
- MLR
- NLR
- ORD
- startNLR

Datasets

- GMAT
- GMAT2
- MSATB

Note

This package was supported by grant funded by Czech Science foundation under number GJ15-15856Y.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Agresti, A. (2010). Analysis of ordinal categorical data. Second edition. John Wiley & Sons.

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Hladka, A. (2021). Statistical models for detection of differential item functioning. Dissertation thesis. Faculty of Mathematics and Physics, Charles University.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

Kingston, N., Leary, L., & Wightman, L. (1985). An exploratory study of the applicability of item response theory methods to the Graduate Management Admission Test. ETS Research Report Series, 1985(2): 1–64.

Martinkova, P., Drabinova, A., Liaw, Y. L., Sanders, E. A., McFarland, J. L., & Price, R. M. (2017). Checking equity: Why differential item functioning analysis should be a routine part of developing conceptual assessments. CBE–Life Sciences Education, 16(2), rm2, doi:10.1187/cbe.16100307.

Swaminathan, H. & Rogers, H. J. (1990). Detecting differential item functioning using logistic regression procedures. Journal of Educational Measurement, 27(4), 361–370, doi:10.1111/j.1745-3984.1990.tb00754.x

Vlckova, K. (2014). Test and item fairness. Master's thesis. Faculty of Mathematics and Physics, Charles University.

See Also

Useful links:

- Report bugs at https://github.com/adelahladka/difNLR/issues

---

`coef.ddfMLR` *Extract model coefficients from an object of* `"ddfMLR"` *class.*

Description

S3 method for extracting estimated model coefficients from an object of `"ddfMLR"` class.

Usage

```
## S3 method for class 'ddfMLR'
coef(object, SE = FALSE, simplify = FALSE, IRTpars = TRUE, CI = 0.95, ...)
```

Arguments

`object`	an object of `"ddfMLR"` class.
`SE`	logical: should the standard errors of estimated parameters be also returned? (default is `FALSE`).
`simplify`	logical: should the estimated parameters be simplified to a matrix? (default is `FALSE`).
`IRTpars`	logical: should the estimated parameters be returned in IRT parameterization? (default is `TRUE`).
`CI`	numeric: level of confidence interval for parameters, default is `0.95` for 95% confidence interval.
`...`	other generic parameters for `coef()` function.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

`ddfMLR` for DDF detection among nominal data.
`coef` for generic function extracting model coefficients.

Examples

```
## Not run:
# loading data
data(GMATtest, GMATkey)
Data <- GMATtest[, 1:20] # items
group <- GMATtest[, "group"] # group membership variable
key <- GMATkey # correct answers

# testing both DDF effects
(x <- ddfMLR(Data, group, focal.name = 1, key))

# estimated parameters
coef(x)
# includes standard errors
coef(x, SE = TRUE)
# includes standard errors and simplifies to matrix
coef(x, SE = TRUE, simplify = TRUE)
# intercept-slope parameterization
```

```
coef(x, IRTpars = FALSE)
# intercept-slope parameterization, simplifies to matrix, turn off confidence intervals
coef(x, IRTpars = FALSE, simplify = TRUE, CI = 0)

## End(Not run)
```

`coef.difNLR` *Extract model coefficients from an object of* `"difNLR"` *class.*

Description

S3 method for extracting model coefficients from an object of `"difNLR"` class.

Usage

```
## S3 method for class 'difNLR'
coef(object, SE = FALSE, simplify = FALSE, IRTpars = TRUE, CI = 0.95, ...)
```

Arguments

`object`	an object of `"difNLR"` class.
`SE`	logical: should the standard errors of estimated parameters be also returned? (default is `FALSE`).
`simplify`	logical: should the estimated parameters be simplified to a matrix? (default is `FALSE`).
`IRTpars`	logical: should the estimated parameters be returned in IRT parameterization? (default is `TRUE`).
`CI`	numeric: level of confidence interval for parameters, default is `0.95` for 95% confidence interval.
`...`	other generic parameters for `coef()` function.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

Karel Zvara
Faculty of Mathematics and Physics, Charles University

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

Swaminathan, H. & Rogers, H. J. (1990). Detecting differential item functioning using logistic regression procedures. Journal of Educational Measurement, 27(4), 361–370, doi:10.1111/j.1745-3984.1990.tb00754.x

See Also

`difNLR` for DIF detection among binary data using generalized logistic regression model.
`coef` for generic function extracting model coefficients.

Examples

```
## Not run:
# loading data
data(GMAT)
Data <- GMAT[, 1:20] # items
group <- GMAT[, "group"] # group membership variable

# testing both DIF effects using likelihood-ratio test and
# 3PL model with fixed guessing for groups
(x <- difNLR(Data, group, focal.name = 1, model = "3PLcg"))

# estimated parameters
coef(x)
# includes standard errors
coef(x, SE = TRUE)
# includes standard errors and simplifies to matrix
coef(x, SE = TRUE, simplify = TRUE)
# intercept-slope parameterization
coef(x, IRTpars = FALSE)
# intercept-slope parameterization, simplifies to matrix, turn off confidence intervals
coef(x, IRTpars = FALSE, simplify = TRUE, CI = 0)

## End(Not run)
```

`coef.difORD` *Extract model coefficients from an object of* `"difORD"` *class.*

Description

S3 method for extracting estimated model coefficients from an object of `"difORD"` class.

Usage

```
## S3 method for class 'difORD'
coef(object, SE = FALSE, simplify = FALSE, IRTpars = TRUE, CI = 0.95, ...)
```

Arguments

object	an object of "difORD" class.
SE	logical: should the standard errors of estimated parameters be also returned? (default is FALSE).
simplify	logical: should the estimated parameters be simplified to a matrix? (default is FALSE).
IRTpars	logical: should the estimated parameters be returned in IRT parameterization? (default is TRUE).
CI	numeric: level of confidence interval for parameters, default is 0.95 for 95% confidence interval.
...	other generic parameters for coef() function.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

difORD for DIF detection among ordinal data.
coef for generic function extracting model coefficients.

Examples

```
## Not run:
# loading data
data(Anxiety, package = "ShinyItemAnalysis")
Data <- Anxiety[, paste0("R", 1:29)] # items
group <- Anxiety[, "gender"] # group membership variable

# testing both DIF effects with adjacent category logit model
(x <- difORD(Data, group, focal.name = 1, model = "adjacent"))

# estimated parameters
coef(x)
# includes standard errors
```

```
coef(x, SE = TRUE)
# includes standard errors and simplifies to matrix
coef(x, SE = TRUE, simplify = TRUE)
# intercept-slope parameterization
coef(x, IRTpars = FALSE)
# intercept-slope parameterization, simplifies to matrix, turn off confidence intervals
coef(x, IRTpars = FALSE, simplify = TRUE, CI = 0)

## End(Not run)
```

---

ddfMLR *DDF detection for nominal data.*

Description

Performs DDF detection procedure for nominal data based on multinomial log-linear regression model and likelihood ratio test of a submodel.

Usage

```
ddfMLR(Data, group, focal.name, key, type = "both", match = "zscore", anchor = NULL,
       purify = FALSE, nrIter = 10, p.adjust.method = "none",
       alpha = 0.05, parametrization)
```

Arguments

Data	data.frame or matrix: dataset which rows represent unscored examinee answers (nominal) and columns correspond to the items. In addition, Data can hold the vector of group membership.
group	numeric or character: a dichotomous vector of the same length as nrow(Data) or a column identifier of Data.
focal.name	numeric or character: indicates the level of group which corresponds to focal group.
key	character: the answer key. Each element corresponds to the correct answer of one item.
type	character: type of DDF to be tested. Either "both" for uniform and non-uniform DDF (i.e., difference in parameters "a" and "b") (default), or "udif" for uniform DDF only (i.e., difference in difficulty parameter "b"), or "nudif" for non-uniform DDF only (i.e., difference in discrimination parameter "a"). Can be specified as a single value (for all items) or as an item-specific vector.
match	numeric or character: matching criterion to be used as an estimate of trait. Can be either "zscore" (default, standardized total score), "score" (total test score), or vector of the same length as number of observations in Data.

anchor — numeric or character: specification of DDF free items. Either NULL (default), or a vector of item names (column names of Data), or item identifiers (integers specifying the column number) determining which items are currently considered as anchor (DDF free) items. Argument is ignored if match is not "zscore" or "score".

purify — logical: should the item purification be applied? (default is FALSE).

nrIter — numeric: the maximal number of iterations in the item purification (default is 10).

p.adjust.method — character: method for multiple comparison correction. Possible values are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", and "none" (default). For more details see p.adjust.

alpha — numeric: significance level (default is 0.05).

parametrization — deprecated. Use coef.ddfMLR for different parameterizations.

Details

Performs DDF detection procedure for nominal data based on multinomial log-linear regression model and likelihood ratio test of submodel. Probability of selection the k-th category (distractor) is

$$P(y=k) = exp((a_k + a_k Dif * g) * (x - b_k - b_k Dif * g))) / (1 + \sum exp((a_l + a_l Dif * g) * (x - b_l - b_l Dif * g))),$$

where x is by default standardized total score (also called Z-score) and g is a group membership. Parameters a_k and b_k are discrimination and difficulty for the k-th category. Terms $a_k Dif$ and $b_k Dif$ then represent differences between two groups (reference and focal) in relevant parameters. Probability of correct answer (specified in argument key) is

$$P(y=k) = 1/(1 + \sum exp((a_l + a_l Dif * g) * (x - b_l - b_l Dif * g))).$$

Parameters are estimated via neural networks. For more details see multinom.

Missing values are allowed but discarded for item estimation. They must be coded as NA for both, Data and group arguments.

Value

The ddfMLR() function returns an object of class "ddfMLR". The output including values of the test statistics, p-values, and items marked as DDF is displayed by the print() method.

A list of class "ddfMLR" with the following arguments:

Sval the values of likelihood ratio test statistics.

mlrPAR the estimates of final model.

mlrSE standard errors of the estimates of final model.

parM0 the estimates of null model.

parM1 the estimates of alternative model.

`llM0` log-likelihood of null model.

`llM1` log-likelihood of alternative model.

`AIC0` AIC of null model.

`AIC1` AIC of alternative model.

`BIC0` BIC of null model.

`BIC1` BIC of alternative model.

`DDFitems` either the column identifiers of the items which were detected as DDF, or `"No DDF item detected"` in case no item was detected as DDF.

`type` character: type of DDF that was tested.

`purification` `purify` value.

`nrPur` number of iterations in item purification process. Returned only if `purify` is `TRUE`.

`ddfPur` a binary matrix with one row per iteration of item purification and one column per item. `"1"` in i-th row and j-th column means that j-th item was identified as DDF in i-th iteration. Returned only if `purify` is `TRUE`.

`conv.puri` logical indicating whether item purification process converged before the maximal number `nrIter` of iterations. Returned only if `purify` is `TRUE`.

`p.adjust.method` character: method for multiple comparison correction which was applied.

`pval` the p-values by likelihood ratio test.

`adj.pval` the adjusted p-values by likelihood ratio test using `p.adjust.method`.

`df` the degress of freedom of likelihood ratio test.

`alpha` numeric: significance level.

`Data` the data matrix.

`group` the vector of group membership.

`group.names` levels of grouping variable.

`key` key of correct answers.

`match` matching criterion.

For an object of class `"ddfMLR"` several methods are available (e.g. `methods(class = "ddfMLR")`).

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Agresti, A. (2010). Analysis of ordinal categorical data. Second edition. John Wiley & Sons.

Hladka, A. (2021). Statistical models for detection of differential item functioning. Dissertation thesis. Faculty of Mathematics and Physics, Charles University.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

See Also

`plot.ddfMLR` for graphical representation of item characteristic curves.
`coef.ddfMLR` for extraction of item parameters with their standard errors.
`logLik.ddfMLR`, `AIC.ddfMLR`, `BIC.ddfMLR` for extraction of log-likelihood and information criteria.

`p.adjust` for multiple comparison corrections.
`multinom` for estimation function using neural networks.

Examples

```
## Not run:
# loading data
data(GMATtest, GMATkey)
Data <- GMATtest[, 1:20] # items
group <- GMATtest[, "group"] # group membership variable
key <- GMATkey # correct answers

# testing both DDF effects
(x <- ddfMLR(Data, group, focal.name = 1, key))

# graphical devices
plot(x, item = "Item1", group.names = c("Group 1", "Group 2"))
plot(x, item = x$DDFitems)
plot(x, item = 1)

# AIC, BIC, log-likelihood
AIC(x)
BIC(x)
logLik(x)

# AIC, BIC, log-likelihood for the first item
AIC(x, item = 1)
BIC(x, item = 1)
logLik(x, item = 1)

# estimated parameters
coef(x)
coef(x, SE = TRUE)
coef(x, SE = TRUE, simplify = TRUE)

# testing both DDF effects with Benjamini-Hochberg adjustment method
```

```
ddfMLR(Data, group, focal.name = 1, key, p.adjust.method = "BH")

# testing both DDF effects with item purification
ddfMLR(Data, group, focal.name = 1, key, purify = TRUE)

# testing uniform DDF effects
ddfMLR(Data, group, focal.name = 1, key, type = "udif")
# testing non-uniform DDF effects
ddfMLR(Data, group, focal.name = 1, key, type = "nudif")

# testing both DDF effects with total score as matching criterion
ddfMLR(Data, group, focal.name = 1, key, match = "score")

## End(Not run)
```

`difNLR` *DIF detection using non-linear regression method.*

Description

Performs DIF detection procedure for dichotomous data based on non-linear regression model (generalized logistic regression) and either likelihood-ratio test, F-test, or Wald's test of a submodel.

Usage

```
difNLR(
  Data, group, focal.name, model, constraints, type = "all", method = "nls",
  match = "zscore", anchor = NULL, purify = FALSE, nrIter = 10, test = "LR",
  alpha = 0.05, p.adjust.method = "none", start, initboot = TRUE, nrBo = 20,
  sandwich = FALSE
)
```

Arguments

`Data`	data.frame or matrix: dataset which rows represent scored examinee answers (`"1"` correct, `"0"` incorrect) and columns correspond to the items. In addition, `Data` can hold the vector of group membership.
`group`	numeric or character: a dichotomous vector of the same length as `nrow(Data)` or a column identifier of `Data`.
`focal.name`	numeric or character: indicates the level of `group` which corresponds to focal group.
`model`	character: generalized logistic regression model to be fitted. See **Details**.
`constraints`	character: which parameters should be the same for both groups. Possible values are any combinations of parameters `"a"`, `"b"`, `"c"`, and `"d"`. See **Details**.

`type` character: type of DIF to be tested. Possible values are `"all"` for detecting difference in any parameter (default), `"udif"` for uniform DIF only (i.e., difference in difficulty parameter `"b"`), `"nudif"` for non-uniform DIF only (i.e., difference in discrimination parameter `"a"`), `"both"` for uniform and non-uniform DIF (i.e., difference in parameters `"a"` and `"b"`), or combination of parameters `"a"`, `"b"`, `"c"`, and `"d"`. Can be specified as a single value (for all items) or as an item-specific vector.

`method` character: method used to estimate parameters. The options are `"nls"` for non-linear least squares (default), `"likelihood"` for maximum likelihood method with `"L-BFGS-B"` algorithm, or `"irls"` for maximum likelihood method with iteratively reweighted least squares (available only for `model = "2PL"`).

`match` numeric or character: matching criterion to be used as an estimate of trait. Can be either `"zscore"` (default, standardized total score), `"score"` (total test score), or vector of the same length as number of observations in `Data`.

`anchor` numeric or character: specification of DIF free items. Either `NULL` (default), or a vector of item names (column names of `Data`), or item identifiers (integers specifying the column number) determining which items are currently considered as anchor (DIF free) items. Argument is ignored if `match` is not `"zscore"` or `"score"`.

`purify` logical: should the item purification be applied? (default is `FALSE`).

`nrIter` numeric: the maximal number of iterations in the item purification (default is 10).

`test` character: test to be performed for DIF detection. Can be either `"LR"` for likelihood ratio test of a submodel (default), `"W"` for Wald test, or `"F"` for F-test of a submodel.

`alpha` numeric: significance level (default is 0.05).

`p.adjust.method` character: method for multiple comparison correction. Possible values are `"holm"`, `"hochberg"`, `"hommel"`, `"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, and `"none"` (default). For more details see `p.adjust`.

`start` numeric: initial values for estimation of parameters. If not specified, starting values are calculated with `startNLR` function. Otherwise, list with as many elements as a number of items. Each element is a named numeric vector of length 8 representing initial values for parameter estimation. Specifically, parameters `"a"`, `"b"`, `"c"`, and `"d"` are initial values for discrimination, difficulty, guessing, and inattention for reference group. Parameters `"aDif"`, `"bDif"`, `"cDif"`, and `"dDif"` are then differences in these parameters between reference and focal group.

`initboot` logical: in case of convergence issues, should be starting values re-calculated based on bootstrapped samples? (default is `TRUE`; newly calculated initial values are applied only to items/models with convergence issues).

`nrBo` numeric: the maximal number of iterations for calculation of starting values using bootstrapped samples (default is 20).

`sandwich` logical: should be sandwich estimator used for covariance matrix of parameters when using `method = "nls"`? Default is `FALSE`.

Details

DIF detection procedure based on non-linear regression is the extension of the logistic regression procedure (Swaminathan & Rogers, 1990) accounting for possible guessing and/or inattention when responding (Drabinova & Martinkova, 2017; Hladka & Martinkova, 2020).

The unconstrained form of 4PL generalized logistic regression model for probability of correct answer (i.e., $y = 1$) is

$$P(y = 1) = (c+cDif*g)+(d+dDif*g-c-cDif*g)/(1+exp(-(a+aDif*g)*(x-b-bDif*g))),$$

where x is by default standardized total score (also called Z-score) and g is a group membership. Parameters a, b, c, and d are discrimination, difficulty, guessing, and inattention. Terms $aDif$, $bDif$, $cDif$, and $dDif$ then represent differences between two groups (reference and focal) in relevant parameters.

This 4PL model can be further constrained by `model` and `constraints` arguments. The arguments `model` and `constraints` can be also combined. Both arguments can be specified as a single value (for all items) or as an item-specific vector (where each element correspond to one item).

The `model` argument offers several predefined models. The options are as follows: `Rasch` for 1PL model with discrimination parameter fixed on value 1 for both groups, `1PL` for 1PL model with discrimination parameter fixed for both groups, `2PL` for logistic regression model, `3PLcg` for 3PL model with fixed guessing for both groups, `3PLdg` for 3PL model with fixed inattention for both groups, `3PLc` (alternatively also `3PL`) for 3PL regression model with guessing parameter, `3PLd` for 3PL model with inattention parameter, `4PLcgdg` for 4PL model with fixed guessing and inattention parameter for both groups, `4PLcgd` (alternatively also `4PLd`) for 4PL model with fixed guessing for both groups, `4PLcdg` (alternatively also `4PLc`) for 4PL model with fixed inattention for both groups, or `4PL` for 4PL model.

The `model` can be specified in more detail with `constraints` argument which specifies what parameters should be fixed for both groups. For example, choice `"ad"` means that discrimination (parameter `"a"`) and inattention (parameter `"d"`) are fixed for both groups and other parameters (`"b"` and `"c"`) are not. The `NA` value for `constraints` means no constraints.

Missing values are allowed but discarded for item estimation. They must be coded as `NA` for both, `Data` and `group` arguments.

In case that model considers difference in guessing or inattention parameter, the different parameterization is used and parameters with standard errors are re-calculated by delta method. However, covariance matrices stick with alternative parameterization.

Value

The `difNLR()` function returns an object of class `"difNLR"`. The output including values of the test statistics, p-values, and items detected as function differently is displayed by the `print()` method.

Object of class `"difNLR"` is a list with the following components:

`Sval` the values of the test statistics.

`nlrPAR` the estimates of final model.

`nlrSE` the standard errors of estimates of final model.

`parM0` the estimates of null model.

`seM0` the standard errors of estimates of null model.

`covM0` the covariance matrices of estimates of null model.

`llM0` log-likelihood of null model.

`parM1` the estimates of alternative model.

`seM1` the standard errors of estimates of alternative model.

`covM1` the covariance matrices of estimates of alternative model.

`llM1` log-likelihood of alternative model.

`DIFitems` either the column identifiers of the items which were detected as DIF, or `"No DIF item detected"` in case no item was detected as function differently.

`model` fitted model.

`constraints` constraints for the `model`.

`type` character: type of DIF that was tested. If parameters were specified, the value is `"other"`.

`types` character: the parameters (specified by user, `type` has value `"other"`) which were tested for difference.

`p.adjust.method` character: method for multiple comparison correction which was applied.

`pval` the p-values of the test.

`adj.pval` the adjusted p-values of the test using `p.adjust.method`.

`df` the degress of freedom of the test.

`test` used test.

`purification` `purify` value.

`nrPur` number of iterations in item purification process. Returned only if `purify` is `TRUE`.

`difPur` a binary matrix with one row per iteration of item purification and one column per item. `"1"` in i-th row and j-th column means that j-th item was identified as DIF in i-th iteration. Returned only if `purify` is `TRUE`.

`conv.puri` logical: indicating whether item purification process converged before the maximal number `nrIter` of iterations. Returned only if `purify` is `TRUE`.

`method` used estimation method.

`conv.fail` numeric: number of convergence issues.

`conv.fail.which` the identifiers of the items which did not converge.

`alpha` numeric: significance level.

`Data` the data matrix.

`group` the vector of group membership.

`group.names` names of groups.

`match` matching criterion.

For an object of class `"difNLR"` several methods are available (e.g., `methods(class = "difNLR")`).

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

Karel Zvara
Faculty of Mathematics and Physics, Charles University

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Hladka, A. (2021). Statistical models for detection of differential item functioning. Dissertation thesis. Faculty of Mathematics and Physics, Charles University.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

Swaminathan, H. & Rogers, H. J. (1990). Detecting differential item functioning using logistic regression procedures. Journal of Educational Measurement, 27(4), 361–370, doi:10.1111/j.1745-3984.1990.tb00754.x

See Also

`plot.difNLR` for graphical representation of item characteristic curves and DIF statistics.
`coef.difNLR` for extraction of item parameters with their standard errors.
`predict.difNLR` for prediction.
`fitted.difNLR` and `residuals.difNLR` for extraction of fitted values and residuals.
`logLik.difNLR`, `AIC.difNLR`, `BIC.difNLR` for extraction of log-likelihood and information criteria.

`p.adjust` for multiple comparison corrections.
`nls` for nonlinear least squares estimation.
`startNLR` for calculation of initial values of fitting algorithms in `difNLR()`.

Examples

```
# loading data
data(GMAT)
Data <- GMAT[, 1:20] # items
group <- GMAT[, "group"] # group membership variable
```

```
# testing both DIF effects using likelihood-ratio test and
# 3PL model with fixed guessing for groups
(x <- difNLR(Data, group, focal.name = 1, model = "3PLcg"))
## Not run:
# graphical devices
plot(x, item = x$DIFitems)
plot(x, item = "Item1")
plot(x, item = 1, group.names = c("Group 1", "Group 2"))
plot(x, plot.type = "stat")

# coefficients
coef(x)
coef(x, SE = TRUE)
coef(x, SE = TRUE, simplify = TRUE)

# fitted values
fitted(x)
fitted(x, item = 1)

# residuals
residuals(x)
residuals(x, item = 1)

# predicted values
predict(x)
predict(x, item = 1)

# predicted values for new subjects
predict(x, item = 1, match = 0, group = 0)
predict(x, item = 1, match = 0, group = 1)

# AIC, BIC, log-likelihood
AIC(x)
BIC(x)
logLik(x)

# AIC, BIC, log-likelihood for the first item
AIC(x, item = 1)
BIC(x, item = 1)
logLik(x, item = 1)

# testing both DIF effects using Wald test and
# 3PL model with fixed guessing for groups
difNLR(Data, group, focal.name = 1, model = "3PLcg", test = "W")

# testing both DIF effects using F test and
# 3PL model with fixed guessing for groups
difNLR(Data, group, focal.name = 1, model = "3PLcg", test = "F")

# testing both DIF effects using
# 3PL model with fixed guessing for groups and sandwich estimator
# of the covariance matrices
difNLR(Data, group, focal.name = 1, model = "3PLcg", sandwich = TRUE)
```

```
# testing both DIF effects using LR test,
# 3PL model with fixed guessing for groups
# and Benjamini-Hochberg correction
difNLR(Data, group, focal.name = 1, model = "3PLcg", p.adjust.method = "BH")

# testing both DIF effects using LR test,
# 3PL model with fixed guessing for groups
# and item purification
difNLR(Data, group, focal.name = 1, model = "3PLcg", purify = TRUE)

# testing both DIF effects using 3PL model with fixed guessing for groups
# and total score as matching criterion
difNLR(Data, group, focal.name = 1, model = "3PLcg", match = "score")

# testing uniform DIF effects using 4PL model with the same
# guessing and inattention
difNLR(Data, group, focal.name = 1, model = "4PLcgdg", type = "udif")

# testing non-uniform DIF effects using 2PL model
difNLR(Data, group, focal.name = 1, model = "2PL", type = "nudif")

# testing difference in parameter b using 4PL model with fixed
# a and c parameters
difNLR(Data, group, focal.name = 1, model = "4PL", constraints = "ac", type = "b")

# testing both DIF effects using LR test,
# 3PL model with fixed guessing for groups
# using maximum likelihood estimation method with L-BFGS-B algorithm
difNLR(Data, group, focal.name = 1, model = "3PLcg", method = "likelihood")

# testing both DIF effects using LR test and 2PL model
# using maximum likelihood estimation method with iteratively reweighted least squares algorithm
difNLR(Data, group, focal.name = 1, model = "2PL", method = "irls")

## End(Not run)
```

`difORD` *DIF detection among ordinal data.*

Description

Performs DIF detection procedure for ordinal data based either on adjacent category logit model or on cumulative logit model and likelihood ratio test of a submodel.

Usage

```
difORD(Data, group, focal.name, model = "adjacent", type = "both", match = "zscore",
       anchor = NULL, purify = FALSE, nrIter = 10, p.adjust.method = "none",
       alpha = 0.05, parametrization)
```

Arguments

`Data`	data.frame or matrix: dataset which rows represent ordinaly scored examinee answers and columns correspond to the items. In addition, `Data` can hold the vector of group membership.
`group`	numeric or character: a dichotomous vector of the same length as `nrow(Data)` or a column identifier of `Data`.
`focal.name`	numeric or character: indicates the level of `group` which corresponds to focal group.
`model`	character: logistic regression model for ordinal data (either `"adjacent"` (default) or `"cumulative"`). See **Details**.
`type`	character: type of DIF to be tested. Either `"both"` for uniform and non-uniform DIF (default), or `"udif"` for uniform DIF only, or `"nudif"` for non-uniform DIF only. Can be specified as a single value (for all items) or as an item-specific vector.
`match`	numeric or character: matching criterion to be used as an estimate of trait. Can be either `"zscore"` (default, standardized total score), `"score"` (total test score), or vector of the same length as number of observations in `Data`.
`anchor`	numeric or character: specification of DIF free items. Either `NULL` (default), or a vector of item names (column names of `Data`), or item identifiers (integers specifying the column number) determining which items are currently considered as anchor (DIF free) items. Argument is ignored if `match` is not `"zscore"` or `"score"`.
`purify`	logical: should the item purification be applied? (default is `FALSE`).
`nrIter`	numeric: the maximal number of iterations in the item purification (default is 10).
`p.adjust.method`	character: method for multiple comparison correction. Possible values are `"holm"`, `"hochberg"`, `"hommel"`, `"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, and `"none"` (default). For more details see `p.adjust`.
`alpha`	numeric: significance level (default is 0.05).
`parametrization`	deprecated. Use `coef.difORD` for different parameterizations.

Details

Calculates DIF likelihood ratio statistics based either on adjacent category logit model or on cumulative logit model for ordinal data.

Using adjacent category logit model, logarithm of ratio of probabilities of two adjacent categories is

$$log(P(y = k)/P(y = k - 1)) = b_0k + b_1 * x + b_2k * g + b_3 * x : g,$$

where x is by default standardized total score (also called Z-score) and g is a group membership.

Using cumulative logit model, probability of gaining at least k points is given by 2PL model, i.e.,

$$P(y >= k) = exp(b_0k + b_1 * x + b_2k * g + b_3 * x : g)/(1 + exp(b_0k + b_1 * x + b_2k * g + b_3 * x : g)).$$

The category probability (i.e., probability of gaining exactly k points) is then $P(y = k) = P(y >= k) - P(y >= k + 1)$.

Both models are estimated by iteratively reweighted least squares. For more details see `vglm`.

Missing values are allowed but discarded for item estimation. They must be coded as `NA` for both, `Data` and `group` parameters.

Value

The `difORD()` function returns an object of class `"difORD"`. The output including values of the test statistics, p-values, and items marked as DIF is displayed by the `print()` method.

A list of class `"difORD"` with the following arguments:

`Sval` the values of likelihood ratio test statistics.

`ordPAR` the estimates of the final model.

`ordSE` standard errors of the estimates of the final model.

`parM0` the estimates of null model.

`parM1` the estimates of alternative model.

`llM0` log-likelihood of null model.

`llM1` log-likelihood of alternative model.

`AICM0` AIC of null model.

`AICM1` AIC of alternative model.

`BICM0` BIC of null model.

`BICM1` BIC of alternative model.

`DIFitems` either the column identifiers of the items which were detected as DIF, or `"No DIF item detected"` in case no item was detected as DIF.

`model` model used for DIF detection.

`type` character: type of DIF that was tested.

`purification` `purify` value.

`nrPur` number of iterations in item purification process. Returned only if `purify` is `TRUE`.

`difPur` a binary matrix with one row per iteration of item purification and one column per item. `"1"` in i-th row and j-th column means that j-th item was identified as DIF in i-th iteration. Returned only if `purify` is `TRUE`.

`conv.puri` logical indicating whether item purification process converged before the maximal number `nrIter` of iterations. Returned only if `purify` is `TRUE`.

`p.adjust.method` character: method for multiple comparison correction which was applied.

`pval` the p-values by likelihood ratio test.

`adj.pval` the adjusted p-values by likelihood ratio test using `p.adjust.method`.

`df` the degress of freedom of likelihood ratio test.

`alpha` numeric: significance level.

`Data` the data matrix.

`group` the vector of group membership.

`group.names` levels of grouping variable.

`match` matching criterion.

For an object of class `"difORD"` several methods are available (e.g., `methods(class = "difORD")`).

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Agresti, A. (2010). Analysis of ordinal categorical data. Second edition. John Wiley & Sons.

Hladka, A. (2021). Statistical models for detection of differential item functioning. Dissertation thesis. Faculty of Mathematics and Physics, Charles University.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

See Also

`plot.difORD` for graphical representation of item characteristic curves.
`coef.difORD` for extraction of item parameters with their standard errors.
`predict.difORD` for calculation of predicted values.
`logLik.difORD`, `AIC.difORD`, `BIC.difORD` for extraction of log-likelihood and information criteria.

`p.adjust` for multiple comparison corrections.
`vglm` for estimation function using iteratively reweighted least squares.

Examples

```
# loading data
data(Anxiety, package = "ShinyItemAnalysis")
Data <- Anxiety[, paste0("R", 1:29)] # items
group <- Anxiety[, "gender"] # group membership variable

# testing both DIF effects with adjacent category logit model
(x <- difORD(Data, group, focal.name = 1, model = "adjacent"))
## Not run:
# graphical devices
plot(x, item = 6)
plot(x, item = "R6")
plot(x, item = "R6", group.names = c("Males", "Females"))

# estimated parameters
coef(x)
coef(x, SE = TRUE) # with SE
coef(x, SE = TRUE, simplify = TRUE) # with SE, simplified
```

```
# AIC, BIC, log-likelihood
AIC(x)
BIC(x)
logLik(x)

# AIC, BIC, log-likelihood for the first item
AIC(x, item = 1)
BIC(x, item = 1)
logLik(x, item = 1)

# testing both DIF effects with Benjamini-Hochberg adjustment method
difORD(Data, group, focal.name = 1, model = "adjacent", p.adjust.method = "BH")

# testing both DIF effects with item purification
difORD(Data, group, focal.name = 1, model = "adjacent", purify = TRUE)

# testing uniform DIF effects
difORD(Data, group, focal.name = 1, model = "adjacent", type = "udif")
# testing non-uniform DIF effects
difORD(Data, group, focal.name = 1, model = "adjacent", type = "nudif")

# testing both DIF effects with total score as matching criterion
difORD(Data, group, focal.name = 1, model = "adjacent", match = "score")

testing both DIF effects with cumulative logit model
(x <- difORD(Data, group, focal.name = 1, model = "cumulative"))
# graphical devices
plot(x, item = 7, plot.type = "cumulative")
plot(x, item = 7, plot.type = "category")

# estimated parameters
coef(x, simplify = TRUE)

## End(Not run)
```

`estimNLR` *Non-Linear Regression DIF models estimation.*

Description

Estimates parameters of non-linear regression models for DIF detection using either non-linear least squares or maximum likelihood method.

Usage

```
estimNLR(y, match, group, formula, method, lower, upper, start)

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

```
## S3 method for class 'estimNLR'
vcov(object, sandwich = FALSE, ...)
```

Arguments

y	numeric: binary vector of responses.
match	numeric: vector of matching criterion.
group	numeric: binary vector of group membership. "0" for reference group, "1" for focal group.
formula	formula: specification of the model. Can be obtained by formulaNLR() function.
method	character: method used to estimate parameters. The options are "nls" for non-linear least squares (default), "likelihood" for maximum likelihood method, and "irls" for maximum likelihood estimation with iteratively reweighted least squares. See **Details**.
lower	numeric: lower bounds for parameters of model specified in formula.
upper	numeric: upper bounds for parameters of model specified in formula.
start	numeric: initial parameters. Can be obtained by startNLR() function.
x	an object of "estimNLR" class.
...	other generic parameters for S3 methods.
object	an object of "estimNLR" class.
sandwich	logical: should be sandwich estimator used for covariance matrix of parameters when using method = "nls"? Default is FALSE.

Details

Function offers either non-linear least squares estimation via nls function, maximum likelihood method with "L-BFGS-B" method via optim function, or maximum likelihood method with iteratively reweighted least squares via glm function.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Hladka, A. (2021). Statistical models for detection of differential item functioning. Dissertation thesis. Faculty of Mathematics and Physics, Charles University.

Examples

```
# loading data
data(GMAT)
y <- GMAT[, 1] # item 1
match <- scale(rowSums(GMAT[, 1:20])) # standardized total score
group <- GMAT[, "group"] # group membership variable

# formula for 3PL model with the same guessing
M <- formulaNLR(model = "3PLcg", type = "both")

# starting values for 3PL model with the same guessing for item 1
start <- startNLR(GMAT[, 1:20], group, model = "3PLcg", parameterization = "classic")
start <- start[[1]][M$M0$parameters]

# non-linear least squares
fitNLSM0 <- estimNLR(
  y = y, match = match, group = group,
  formula = M$M0$formula, method = "nls",
  lower = M$M0$lower, upper = M$M0$upper, start = start
)
fitNLSM0

coef(fitNLSM0)
logLik(fitNLSM0)
vcov(fitNLSM0)
vcov(fitNLSM0, sandwich = TRUE)
fitted(fitNLSM0)
residuals(fitNLSM0)

# maximum likelihood method
fitLKLM0 <- estimNLR(
  y = y, match = match, group = group,
  formula = M$M0$formula, method = "likelihood",
  lower = M$M0$lower, upper = M$M0$upper, start = start
)
fitLKLM0

coef(fitLKLM0)
logLik(fitLKLM0)
vcov(fitLKLM0)
fitted(fitLKLM0)
residuals(fitLKLM0)

# iteratively reweighted least squares for 2PL model
M <- formulaNLR(model = "2PL", parameterization = "logistic")
fitIRLSM1 <- estimNLR(
  y = y, match = match, group = group,
  formula = M$M1$formula, method = "irls"
)
fitIRLSM1

coef(fitIRLSM1)
```

```
logLik(fitIRLSM1)
vcov(fitIRLSM1)
fitted(fitIRLSM1)
residuals(fitIRLSM1)
```

`fitted.difNLR` *Fitted values and residuals for an object of* `"difNLR"` *class.*

Description

S3 methods for extracting fitted values and residuals for an object of `"difNLR"` class.

Usage

```
## S3 method for class 'difNLR'
fitted(object, item = "all", ...)

## S3 method for class 'difNLR'
residuals(object, item = "all", ...)
```

Arguments

`object`	an object of `"difNLR"` class.
`item`	numeric or character: either character `"all"` to apply for all converged items (default), or a vector of item names (column names of `Data`), or item identifiers (integers specifying the column number).
`...`	other generic parameters for S3 methods.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

Karel Zvara
Faculty of Mathematics and Physics, Charles University

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

Swaminathan, H. & Rogers, H. J. (1990). Detecting differential item functioning using logistic regression procedures. Journal of Educational Measurement, 27(4), 361–370, doi:10.1111/j.1745-3984.1990.tb00754.x

See Also

`difNLR` for DIF detection among binary data using generalized logistic regression model.
`fitted` for generic function extracting fitted values.
`residuals` for generic function extracting residuals.

Examples

```
## Not run:
# loading data
data(GMAT)
Data <- GMAT[, 1:20] # items
group <- GMAT[, "group"] # group membership variable

# testing both DIF effects using likelihood-ratio test and
# 3PL model with fixed guessing for groups
(x <- difNLR(Data, group, focal.name = 1, model = "3PLcg"))

# fitted values
fitted(x)
fitted(x, item = 1)

# residuals
residuals(x)
residuals(x, item = 1)

## End(Not run)
```

`formulaNLR` *Formula for non-linear regression DIF model.*

Description

Function returns the formula of the non-linear regression DIF model based on model specification and DIF type to be tested.

Usage

```
formulaNLR(model, constraints = NULL, type = "all", parameterization = "classic", outcome)
```

Arguments

`model`	character: generalized logistic regression model for which starting values should be estimated. See **Details**.
`constraints`	character: which parameters should be the same for both groups. Possible values are any combinations of parameters `"a"`, `"b"`, `"c"`, and `"d"`. Default value is `NULL`. See **Details**.
`type`	character: type of DIF to be tested. Possible values are `"all"` for detecting difference in any parameter (default), `"udif"` for uniform DIF only (i.e., difference in difficulty parameter `"b"`), `"nudif"` for non-uniform DIF only (i.e., difference in discrimination parameter `"a"`), `"both"` for uniform and non-uniform DIF (i.e., difference in parameters `"a"` and `"b"`), or combination of parameters `"a"`, `"b"`, `"c"`, and `"d"`. Can be specified as a single value (for all items) or as an item-specific vector.
`parameterization`	character: parameterization of regression coefficients. Possible options are `"classic"` (IRT parameterization), `"alternative"` (default) and `"logistic"` (logistic regression). See **Details**.
`outcome`	character: name of outcome to be printed in formula. If not specified `"y"` is used.

Details

The unconstrained form of 4PL generalized logistic regression model for probability of correct answer (i.e., $y = 1$) is

$$P(y = 1) = (c+cDif*g)+(d+dDif*g-c-cDif*g)/(1+exp(-(a+aDif*g)*(x-b-bDif*g))),$$

where x is by default standardized total score (also called Z-score) and g is a group membership. Parameters a, b, c, and d are discrimination, difficulty, guessing, and inattention. Terms $aDif$, $bDif$, $cDif$, and $dDif$ then represent differences between two groups (reference and focal) in relevant parameters.

The `model` argument offers several predefined models. The options are as follows: `Rasch` for 1PL model with discrimination parameter fixed on value 1 for both groups, `1PL` for 1PL model with discrimination parameter fixed for both groups, `2PL` for logistic regression model, `3PLcg` for 3PL model with fixed guessing for both groups, `3PLdg` for 3PL model with fixed inattention for both groups, `3PLc` (alternatively also `3PL`) for 3PL regression model with guessing parameter, `3PLd` for 3PL model with inattention parameter, `4PLcgdg` for 4PL model with fixed guessing and inattention parameter for both groups, `4PLcgd` (alternatively also `4PLd`) for 4PL model with fixed guessing for both groups, `4PLcdg` (alternatively also `4PLc`) for 4PL model with fixed inattention for both groups, or `4PL` for 4PL model.

Three possible parameterization can be specified in `"parameterization"` argument: `"classic"` returns IRT parameters of reference group and differences in these parameters between reference and focal group. `"alternative"` returns IRT parameters of reference group, the differences in parameters `"a"` and `"b"` between two groups and parameters `"c"` and `"d"` for focal group. `"logistic"` returns parameters in logistic regression parameterization and it is available only for 2PL model.

Value

A list of two models. Both includes formula, parameters to be estimated and their lower and upper constraints.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

difNLR

Examples

```
# 3PL model with the same guessing for both groups
# to test both types of DIF
formulaNLR(model = "3PLcg", type = "both")

# 4PL model with the same guessing and inattention
# to test uniform DIF
formulaNLR(model = "4PLcgdg", type = "udif")

# 2PL model to test non-uniform DIF
formulaNLR(model = "2PL", type = "nudif")

# 4PL model to test all possible DIF
# with alternative parameterization
formulaNLR(model = "4PL", type = "all", parameterization = "alternative")

# 4PL model with fixed a and c parameter
# to test difference in b with alternative parameterization
formulaNLR(model = "4PL", constraints = "ac", type = "b", parameterization = "alternative")

# 2PL model with logistic parameterization
formulaNLR(model = "2PL", parameterization = "logistic")
```

genNLR *Generates data set based on generalized logistic regression DIF and DDF models.*

Description

Generates dichotomous, nominal, and ordinal data based on generalized logistic regression models for DIF and DDF detection.

Usage

```
genNLR(N = 1000, ratio = 1, itemtype = "dich", a, b, c, d, mu = 0, sigma = 1)
```

Arguments

N	numeric: number of rows representing respondents. (default is 1000).
ratio	numeric: ratio of respondents number in reference and focal group.
itemtype	character: type of items to be generated. Options are "dich" (default) for dichotomous item based on non-linear regression model for DIF detection (see difNLR for details), "nominal" for nominal items based on multinomial model for DDF detection (see ddfMLR for detail), and "ordinal" for ordinal data based on adjacent category logit model (for details see difORD).
a	numeric: matrix representing discriminations with m rows (where m is number of items). Needs to be provided. See **Details**.
b	numeric: numeric: matrix representing difficulties with m rows (where m is number of items). Needs to be provided. See **Details**.
c	numeric: matrix representing guessings (lower asymptotes) with m rows (where m is number of items). Default is NULL. See **Details**.
d	numeric: matrix representing inattentions (upper asymptotes) with m rows (where m is number of items). Default is NULL. See **Details**.
mu	numeric: a mean vector of the underlying distribution. The first value corresponds to reference group, the second to focal group. Default is 0 value for both groups.
sigma	numeric: a standard deviation vector of the underlying distribution. The first value corresponds to reference group, the second to focal group. Default is 1 value for both groups.

Details

The a, b, c and d are numeric matrices with m rows (where m is number of items) representing parameters of regression models for DIF and DDF detection.

For option itemtype = "dich", matrices should have two columns. The first column represents parameters of the reference group and the second of the focal group. In case that only one column is provided, parameters are set to be the same for both groups.

For options `itemtype = "nominal"` and `itemtype = "ordinal"`, matrices `c` and `d` are ignored. Matrices `a` and `b` contain parameters for distractors. For example, when item with 4 different choices is supposed to be generated, user provide matrices with 6 columns. First 3 columns correspond to distractors parameters for reference group and last three columns for focal group. The number of choices can differ for items. Matrices `a` and `b` need to consist of as many columns as is the maximum number of distractors. Items with less choices can containt NAs.

Value

A `data.frame` containing `N` rows representing respondents and `m + 1` columns representing `m` items. The last column is group membership variable with coding `"0"` for reference group and `"1"` for focal group.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

See Also

`difNLR`, `difORD`, `ddfMLR`

Examples

```
# seed
set.seed(123)
# generating parameters for dichotomous data with DIF, 5 items
a <- matrix(runif(10, 0.8, 2), ncol = 2)
b <- matrix(runif(10, -2, 2), ncol = 2)
c <- matrix(runif(10, 0, 0.25), ncol = 2)
d <- matrix(runif(10, 0.8, 1), ncol = 2)
# generating dichotomous data set with 300 observations (150 each group)
genNLR(N = 300, a = a, b = b, c = c, d = d)
# generating dichotomous data set with 300 observations (150 each group)
# and different mean and standard deviation for underlying distribution
genNLR(N = 300, a = a, b = b, c = c, d = d, mu = c(1, 0), sigma = c(1, 2))
# generating dichotomous data set with 300 observations (250 reference group, 50 focal)
```

```
genNLR(N = 300, ratio = 5, a = a, b = b, c = c, d = d)

# generating parameters for nominal data with DDF, 5 items,
# each item 3 choices
a <- matrix(runif(20, 0.8, 2), ncol = 4)
b <- matrix(runif(20, -2, 2), ncol = 4)
# generating nominal data set with 300 observations (150 each group)
genNLR(N = 300, itemtype = "nominal", a = a, b = b)
# generating nominal data set with 300 observations (250 reference group, 50 focal)
genNLR(N = 300, itemtype = "nominal", ratio = 5, a = a, b = b)

# generating parameters for nominal data with DDF, 5 items,
# items 1 and 2 have 2 choices, items 3, 4 and 5 have 3 choices
a <- matrix(runif(20, 0.8, 2), ncol = 4)
a[1:2, c(2, 4)] <- NA
b <- matrix(runif(20, -2, 2), ncol = 4)
b[1:2, c(2, 4)] <- NA
# generating nominal data set with 300 observations (150 each group)
genNLR(N = 300, itemtype = "nominal", a = a, b = b)
# generating nominal data set with 300 observations (250 reference group, 50 focal)
genNLR(N = 300, itemtype = "nominal", ratio = 5, a = a, b = b)
```

GMAT *Dichotomous dataset based on GMAT with the same total score distribution for groups.*

Description

The GMAT is a generated dataset based on parameters from Graduate Management Admission Test (GMAT, Kingston et al., 1985). First two items were considered to function differently in uniform and non-uniform way respectively. The dataset represents responses of 2,000 subjects to multiple-choice test of 20 items. A correct answer is coded as 1 and incorrect answer as 0. The column group represents group membership, where 0 indicates reference group and 1 indicates focal group. Groups are the same size (i.e. 1,000 per group). The distributions of total scores (sum of correct answers) are the same for both reference and focal group (Martinkova et al., 2017). The column criterion represents generated continuous variable which is intended to be predicted by test.

Usage

```
data(GMAT)
```

Format

A GMAT data frame consists of 2,000 observations on the following 22 variables:

Item1-Item20 dichotomously scored items of the test

group group membership vector, "0" reference group, "1" focal group

criterion continuous critetion intended to be predicted by test

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Kingston, N., Leary, L., & Wightman, L. (1985). An exploratory study of the applicability of item response theory methods to the Graduate Management Admission Test. ETS Research Report Series, 1985(2): 1–64.

Martinkova, P., Drabinova, A., Liaw, Y. L., Sanders, E. A., McFarland, J. L., & Price, R. M. (2017). Checking equity: Why differential item functioning analysis should be a routine part of developing conceptual assessments. CBE–Life Sciences Education, 16(2), rm2, doi:10.1187/cbe.16100307.

See Also

GMATtest, GMATkey

GMAT2 *Dichotomous dataset based on GMAT.*

Description

The GMAT2 is a generated dataset based on parameters from Graduate Management Admission Test (GMAT, Kingston et al., 1985). First two items were considered to function differently in uniform and non-uniform way respectively. The dataset represents responses of 1,000 subjects to multiple-choice test of 20 items. A correct answer is coded as 1 and incorrect answer as 0. The column group represents group membership, where 0 indicates reference group and 1 indicates focal group. Groups are the same size (i.e. 500 per group).

Usage

```
data(GMAT2)
```

Format

A GMAT2 data frame consists of 1,000 observations on the following 21 variables:

Item1-Item20 dichotomously scored items of the test

group group membership vector, "0" reference group, "1" focal group

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Kingston, N., Leary, L., & Wightman, L. (1985). An exploratory study of the applicability of item response theory methods to the Graduate Management Admission Test. ETS Research Report Series, 1985(2): 1–64.

Martinkova, P., Drabinova, A., Liaw, Y. L., Sanders, E. A., McFarland, J. L., & Price, R. M. (2017). Checking equity: Why differential item functioning analysis should be a routine part of developing conceptual assessments. CBE–Life Sciences Education, 16(2), rm2, doi:10.1187/cbe.16100307.

See Also

GMAT2test, GMAT2key

GMAT2key *Key of correct answers for* GMAT2test *dataset.*

Description

The GMAT2key is a vector of factors representing correct answers of generated GMAT2test data set based on Graduate Management Admission Test (GMAT) data set (Kingston et al., 1985).

Usage

```
data(GMAT2key)
```

Format

A nominal vector with 20 values representing correct answers to items of GMAT2test dataset. For more details see GMAT2test.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Kingston, N., Leary, L., & Wightman, L. (1985). An exploratory study of the applicability of item response theory methods to the Graduate Management Admission Test. ETS Research Report Series, 1985(2): 1–64.

Martinkova, P., Drabinova, A., Liaw, Y. L., Sanders, E. A., McFarland, J. L., & Price, R. M. (2017). Checking equity: Why differential item functioning analysis should be a routine part of developing conceptual assessments. CBE–Life Sciences Education, 16(2), rm2, doi:10.1187/cbe.16100307.

See Also

GMAT2, GMAT2test

GMAT2test *Dataset based on GMAT.*

Description

The GMAT2test is a generated dataset based on parameters from Graduate Management Admission Test (GMAT, Kingston et al., 1985). First two items were considered to function differently in uniform and non-uniform way respectively. The data set represents responses of 1,000 subjects to multiple-choice test of 20 items. Aditionally, 4 possible answers on all items were generated, coded A, B, C, and D. The column group represents group membership, where 0 indicates reference group and 1 indicates focal group. Groups are the same size (i.e. 500 per group).

Usage

```
data(GMAT2test)
```

Format

A GMAT2test data frame consists of 1,000 observations on the following 21 variables:

Item1-Item20 nominal items of the test coded A, B, C, and D

group group membership vector, "0" reference group, "1" focal group

Correct answers are presented in GMAT2key data set.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Kingston, N., Leary, L., & Wightman, L. (1985). An exploratory study of the applicability of item response theory methods to the Graduate Management Admission Test. ETS Research Report Series, 1985(2): 1–64.

Martinkova, P., Drabinova, A., Liaw, Y. L., Sanders, E. A., McFarland, J. L., & Price, R. M. (2017). Checking equity: Why differential item functioning analysis should be a routine part of developing conceptual assessments. CBE–Life Sciences Education, 16(2), rm2, doi:10.1187/cbe.16100307.

See Also

GMAT2, GMAT2key

GMATkey *Key of correct answers for* GMATtest *dataset.*

Description

The GMATkey is a vector of factors representing correct answers of generated GMATtest data set based on Graduate Management Admission Test (GMAT, Kingston et al., 1985).

Usage

```
data(GMATkey)
```

Format

A nominal vector with 20 values representing correct answers to items of GMATtest dataset. For more details see GMATtest.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Kingston, N., Leary, L., & Wightman, L. (1985). An exploratory study of the applicability of item response theory methods to the Graduate Management Admission Test. ETS Research Report Series, 1985(2): 1–64.

Martinkova, P., Drabinova, A., Liaw, Y. L., Sanders, E. A., McFarland, J. L., & Price, R. M. (2017). Checking equity: Why differential item functioning analysis should be a routine part of developing conceptual assessments. CBE–Life Sciences Education, 16(2), rm2, doi:10.1187/cbe.16100307.

See Also

GMAT, GMATtest

GMATtest *Dataset based on GMAT with the same total score distribution for groups.*

Description

The GMATtest is a generated dataset based on parameters from Graduate Management Admission Test (GMAT, Kingston et al., 1985). First two items were considered to function differently in uniform and non-uniform way respectively. The dataset represents responses of 2,000 subjects to multiple-choice test of 20 items. Aditionally, 4 possible answers on all items were generated, coded A, B, C, and D. The column group represents group membership, where 0 indicates reference group and 1 indicates focal group. Groups are the same size (i.e. 1,000 per group). The distributions of total scores (sum of correct answers) are the same for both reference and focal group (Martinkova et al., 2017). The column criterion represents generated continuous variable which is intended to be predicted by test.

Usage

```
data(GMATtest)
```

Format

A GMATtest data frame consists of 2,000 observations on the following 22 variables:

Item1-Item20 nominal items of the test coded A, B, C, and D

group group membership vector, "0" reference group, "1" focal group

criterion continuous critetion intended to be predicted by test

Correct answers are presented in GMATkey data set.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Kingston, N., Leary, L., & Wightman, L. (1985). An exploratory study of the applicability of item response theory methods to the Graduate Management Admission Test. ETS Research Report Series, 1985(2): 1–64.

Martinkova, P., Drabinova, A., Liaw, Y. L., Sanders, E. A., McFarland, J. L., & Price, R. M. (2017). Checking equity: Why differential item functioning analysis should be a routine part of developing conceptual assessments. CBE–Life Sciences Education, 16(2), rm2, doi:10.1187/cbe.16100307.

See Also

GMAT, GMATkey

---

logLik.ddfMLR *Log-likelihood and information criteria for an object of "ddfMLR" class.*

Description

S3 methods for extracting log-likelihood, Akaike's information criterion (AIC) and Schwarz's Bayesian criterion (BIC) for an object of "ddfMLR" class.

Usage

```
## S3 method for class 'ddfMLR'
logLik(object, item = "all", ...)

## S3 method for class 'ddfMLR'
AIC(object, item = "all", ...)

## S3 method for class 'ddfMLR'
BIC(object, item = "all", ...)
```

Arguments

`object`	an object of `"ddfMLR"` class.
`item`	numeric or character: either character `"all"` to apply for all converged items (default), or a vector of item names (column names of `Data`), or item identifiers (integers specifying the column number).
`...`	other generic parameters for S3 methods.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

`ddfMLR` for DDF detection among nominal data.
`logLik` for generic function extracting log-likelihood.
`AIC` for generic function calculating AIC and BIC.

Examples

```
## Not run:
# loading data
data(GMATtest, GMATkey)
Data <- GMATtest[, 1:20] # items
group <- GMATtest[, "group"] # group membership variable
key <- GMATkey # correct answers

# testing both DDF effects
(x <- ddfMLR(Data, group, focal.name = 1, key))

# AIC, BIC, log-likelihood
```

```
AIC(x)
BIC(x)
logLik(x)

# AIC, BIC, log-likelihood for the first item
AIC(x, item = 1)
BIC(x, item = 1)
logLik(x, item = 1)

## End(Not run)
```

`logLik.difNLR` *Log-likelihood and information criteria for an object of* `"difNLR"` *class.*

Description

S3 methods for extracting log-likelihood, Akaike's information criterion (AIC) and Schwarz's Bayesian criterion (BIC) for an object of `"difNLR"` class.

Usage

```
## S3 method for class 'difNLR'
logLik(object, item = "all", ...)

## S3 method for class 'difNLR'
AIC(object, item = "all", ...)

## S3 method for class 'difNLR'
BIC(object, item = "all", ...)
```

Arguments

`object`	an object of `"difNLR"` class.
`item`	numeric or character: either character `"all"` to apply for all converged items (default), or a vector of item names (column names of `Data`), or item identifiers (integers specifying the column number).
`...`	other generic parameters for S3 methods.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences

<martinkova@cs.cas.cz>

Karel Zvara
Faculty of Mathematics and Physics, Charles University

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

Swaminathan, H. & Rogers, H. J. (1990). Detecting differential item functioning using logistic regression procedures. Journal of Educational Measurement, 27(4), 361–370, doi:10.1111/j.1745-3984.1990.tb00754.x

See Also

difNLR for DIF detection among binary data using generalized logistic regression model.
logLik for generic function extracting log-likelihood.
AIC for generic function calculating AIC and BIC.

Examples

```
## Not run:
# loading data
data(GMAT)
Data <- GMAT[, 1:20] # items
group <- GMAT[, "group"] # group membership variable

# testing both DIF effects using likelihood-ratio test and
# 3PL model with fixed guessing for groups
(x <- difNLR(Data, group, focal.name = 1, model = "3PLcg"))

# AIC, BIC, log-likelihood
AIC(x)
BIC(x)
logLik(x)

# AIC, BIC, log-likelihood for the first item
AIC(x, item = 1)
BIC(x, item = 1)
logLik(x, item = 1)

## End(Not run)
```

`logLik.difORD` *Log-likelihood and information criteria for an object of* `"difORD"` *class.*

Description

S3 methods for extracting log-likelihood, Akaike's information criterion (AIC) and Schwarz's Bayesian criterion (BIC) for an object of `"difORD"` class.

Usage

```
## S3 method for class 'difORD'
logLik(object, item = "all", ...)

## S3 method for class 'difORD'
AIC(object, item = "all", ...)

## S3 method for class 'difORD'
BIC(object, item = "all", ...)
```

Arguments

`object`	an object of `"difORD"` class.
`item`	numeric or character: either character `"all"` to apply for all converged items (default), or a vector of item names (column names of `Data`), or item identifiers (integers specifying the column number).
`...`	other generic parameters for S3 methods.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

`difORD` for DIF detection among ordinal data.
`logLik` for generic function extracting log-likelihood.
`AIC` for generic function calculating AIC and BIC.

Examples

```
## Not run:
# loading data
data(Anxiety, package = "ShinyItemAnalysis")
Data <- Anxiety[, paste0("R", 1:29)] # items
group <- Anxiety[, "gender"] # group membership variable

# testing both DIF effects with adjacent category logit model
(x <- difORD(Data, group, focal.name = 1, model = "adjacent"))

# AIC, BIC, log-likelihood
AIC(x)
BIC(x)
logLik(x)

# AIC, BIC, log-likelihood for the first item
AIC(x, item = 1)
BIC(x, item = 1)
logLik(x, item = 1)

## End(Not run)
```

MLR *DDF likelihood ratio statistics based on multinomial log-linear regression model.*

Description

Calculates DDF likelihood ratio statistics for nominal data based on multinomial log-linear model.

Usage

```
MLR(Data, group, key, type = "both", match = "zscore", anchor = 1:ncol(Data),
    p.adjust.method = "none", alpha = 0.05, parametrization)
```

Arguments

Data	data.frame or matrix: dataset which rows represent unscored examinee answers (nominal) and columns correspond to the items.
group	numeric: binary vector of group membership. "0" for reference group, "1" for focal group.
key	character: the answer key. Each element corresponds to the correct answer of one item.
type	character: type of DDF to be tested. Either "both" for uniform and non-uniform DDF (i.e., difference in parameters "a" and "b") (default), or "udif" for uniform DDF only (i.e., difference in difficulty parameter "b"), or "nudif" for non-uniform DDF only (i.e., difference in discrimination parameter "a"). Can be specified as a single value (for all items) or as an item-specific vector.

`match` numeric or character: matching criterion to be used as an estimate of trait. Can be either `"zscore"` (default, standardized total score), `"score"` (total test score), or vector of the same length as number of observations in `Data`.

`anchor` character or numeric: specification of DIF free items. A vector of item identifiers (integers specifying the column number) specifying which items are currently considered as anchor (DIF free) items. Argument is ignored if `match` is not `"zscore"` or `"score"`.

`p.adjust.method` character: method for multiple comparison correction. Possible values are `"holm"`, `"hochberg"`, `"hommel"`, `"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, and `"none"` (default). For more details see `p.adjust`.

`alpha` numeric: significance level (default is 0.05).

`parametrization` deprecated. Use `coef.ddfMLR` for different parameterizations.

Details

$$P(y = k) = exp(b_0k+b_1k*x+b_2k*g+b_3k*x*g)/(1+\sum exp(b_0l+b_1l*x+b_2l*g+b_3l*x*g)),$$

where x is by default standardized total score (also called Z-score) and g is a group membership. Probability of correct answer (specified in argument `key`) is

$$P(y = k) = 1/(1 + \sum exp(b_0l + b_1l * x + b_2l * g + b_3l * x * g)).$$

Parameters are estimated via neural networks. For more details see `multinom`.

Value

A list with the following arguments:

`Sval` the values of likelihood ratio test statistics.

`pval` the p-values by likelihood ratio test.

`adj.pval` the adjusted p-values by likelihood ratio test using `p.adjust.method`.

`df` the degress of freedom of likelihood ratio test.

`par.m0` the estimates of null model.

`par.m1` the estimates of alternative model.

`se.m0` standard errors of parameters in null model.

`se.m1` standard errors of parameters in alternative model.

`cov.m0` list of covariance matrices of item parameters for null model.

`cov.m1` list of covariance matrices of item parameters for alternative model.

`ll.m0` log-likelihood of m0 model.

`ll.m1` log-likelihood of m1 model.

`AIC.m0` AIC of m0 model.

`AIC.m1` AIC of m1 model.

`BIC.m0` BIC of m0 model.

`BIC.m1` BIC of m1 model.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Agresti, A. (2010). Analysis of ordinal categorical data. Second edition. John Wiley & Sons.

Hladka, A. (2021). Statistical models for detection of differential item functioning. Dissertation thesis. Faculty of Mathematics and Physics, Charles University.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

See Also

p.adjust multinom

Examples

```
## Not run:
# loading data
data(GMATtest, GMATkey)
Data <- GMATtest[, 1:20] # items
group <- GMATtest[, "group"] # group membership variable
key <- GMATkey # correct answers

# testing both DDF effects
MLR(Data, group, key, type = "both")

# testing uniform DDF effects
MLR(Data, group, key, type = "udif")

# testing non-uniform DDF effects
MLR(Data, group, key, type = "nudif")

## End(Not run)
```

MSATB *Dichotomous dataset of Medical School Admission Test in Biology.*

Description

The MSATB dataset consists of the responses of 1,407 subjects (484 males, 923 females) to admission test to medical school in the Czech republic. It contains 20 selected items from original test while first item was previously detected as differently functioning (Vlckova, 2014). A correct answer is coded as 1 and incorrect answer as 0. The column gender represents gender of students, where 0 indicates males (reference group) and 1 indicates females (focal group).

Usage

```
data(MSATB)
```

Format

A MSATB data frame consists of 1,407 observations on the following 21 variables:

Item dichotomously scored items of the test

gender gender of respondents, "0" males, "1" females

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Vlckova, K. (2014). Test and item fairness. Master's thesis. Faculty of Mathematics and Physics, Charles University.

See Also

MSATBtest, MSATBkey

MSATBkey *Key of correct answers for* MSATBtest *dataset.*

Description

The MSATBkey is a vector of factors representing correct answers of MSATBtest dataset.

Usage

```
data(MSATBkey)
```

Format

A nominal vector with 20 values representing correct answers to items of MSATBtest dataset. For more details see MSATBtest.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Vlckova, K. (2014). Test and item fairness. Master's thesis. Faculty of Mathematics and Physics, Charles University.

See Also

MSATB, MSATBtest

MSATBtest *Dataset of School Admission Test in Biology.*

Description

The MSATBtest dataset consists of the responses of 1,407 subjects (484 males, 923 females) to multiple-choice admission test to medical school in the Czech republic. It contains 20 selected items from original test while first item was previously detected detected as differently functioning (Vlckova, 2014). Possible answers were A, B, C, and D, while any combination of these can be correct. The column gender represents gender of students, where 0 indicates males (reference group) and 1 indicates females (focal group).

Usage

```
data(MSATBtest)
```

Format

A MSATBtest data frame consists of 1,407 observations on the following 21 variables:

Item nominal items of the test

gender gender of respondents, "0" males, "1" females

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Vlckova, K. (2014). Test and item fairness. Master's thesis. Faculty of Mathematics and Physics, Charles University.

See Also

MSATB, MSATBkey

NLR *DIF statistics based on non-linear regression model.*

Description

Calculates either DIF likelihood ratio statistics or F statistics for dichotomous data based on non-linear regression model (generalized logistic regression model).

Usage

```
NLR(Data, group, model, constraints = NULL, type = "all", method = "nls",
   match = "zscore", anchor = 1:ncol(Data), start, p.adjust.method = "none", test = "LR",
    alpha = 0.05, initboot = TRUE, nrBo = 20, sandwich = FALSE)
```

Arguments

Data	data.frame or matrix: dataset which rows represent scored examinee answers ("1" correct, "0" incorrect) and columns correspond to the items.
group	numeric: binary vector of group membership. "0" for reference group, "1" for focal group.
model	character: generalized logistic regression model to be fitted. See **Details**.
constraints	character: which parameters should be the same for both groups. Possible values are any combinations of parameters "a", "b", "c", and "d". Default value is NULL. See **Details**.
type	character: type of DIF to be tested. Possible values are "all" for detecting difference in any parameter (default), "udif" for uniform DIF only (i.e., difference in difficulty parameter "b"), "nudif" for non-uniform DIF only (i.e., difference in discrimination parameter "a"), "both" for uniform and non-uniform DIF (i.e., difference in parameters "a" and "b"), or combination of parameters "a", "b", "c", and "d". Can be specified as a single value (for all items) or as an item-specific vector.
method	character: method used to estimate parameters. The options are "nls" for non-linear least squares (default), "likelihood" for maximum likelihood method with "L-BFGS-B" algorithm, or "irls" for maximum likelihood method with iteratively reweighted least squares (available only for model = "2PL").
match	character or numeric: matching criterion to be used as estimate of trait. Can be either "zscore" (default, standardized total score), "score" (total test score), or numeric vector of the same length as number of observations in Data.
anchor	character or numeric: specification of DIF free items. A vector of item identifiers (integers specifying the column number) specifying which items are currently considered as anchor (DIF free) items. Argument is ignored if match is not "zscore" or "score".

`start` numeric: initial values for estimation of parameters. If not specified, starting values are calculated with `startNLR` function. Otherwise, list with as many elements as number of items. Each element is a named numeric vector of length 8 representing initial values for parameter estimation. Specifically, parameters `"a"`, `"b"`, `"c"`, and `"d"` are initial values for discrimination, difficulty, guessing, and inattention for reference group. Parameters `"aDif"`, `"bDif"`, `"cDif"`, and `"dDif"` are then differences in these parameters between reference and focal group.

`p.adjust.method` character: method for multiple comparison correction. Possible values are `"holm"`, `"hochberg"`, `"hommel"`, `"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, and `"none"` (default). For more details see `p.adjust`.

`test` character: test to be performed for DIF detection. Can be either `"LR"` for likelihood ratio test of a submodel (default), `"W"` for Wald test, or `"F"` for F-test of a submodel.

`alpha` numeric: significance level (default is 0.05).

`initboot` logical: in case of convergence issues, should be starting values re-calculated based on bootstraped samples? (default is `TRUE`; newly calculated initial values are applied only to items/models with convergence issues).

`nrBo` numeric: the maximal number of iterations for calculation of starting values using bootstraped samples (default is 20).

`sandwich` logical: should be sandwich estimator used for covariance matrix of parameters when using `method = "nls"`? Default is `FALSE`.

Details

Calculation of the test statistics using DIF detection procedure based on non-linear regression (extension of logistic regression procedure; Swaminathan and Rogers, 1990; Drabinova and Martinkova, 2017).

The unconstrained form of 4PL generalized logistic regression model for probability of correct answer (i.e., $y = 1$) is

$$P(y = 1) = (c+cDif*g)+(d+dDif*g-c-cDif*g)/(1+exp(-(a+aDif*g)*(x-b-bDif*g))),$$

where x is by default standardized total score (also called Z-score) and g is a group membership. Parameters a, b, c, and d are discrimination, difficulty, guessing, and inattention. Terms $aDif$, $bDif$, $cDif$, and $dDif$ then represent differences between two groups (reference and focal) in relevant parameters.

This 4PL model can be further constrained by `model` and `constraints` arguments. The arguments `model` and `constraints` can be also combined. Both arguments can be specified as a single value (for all items) or as an item-specific vector (where each element correspond to one item).

The `model` argument offers several predefined models. The options are as follows: `Rasch` for 1PL model with discrimination parameter fixed on value 1 for both groups, `1PL` for 1PL model with discrimination parameter fixed for both groups, `2PL` for logistic regression model, `3PLcg` for 3PL model with fixed guessing for both groups, `3PLdg` for 3PL model with fixed inattention for both groups, `3PLc` (alternatively also `3PL`) for 3PL regression model with guessing parameter, `3PLd` for 3PL model with inattention parameter, `4PLcgdg` for 4PL model with fixed guessing and inattention

parameter for both groups, 4PLcgd (alternatively also 4PLd) for 4PL model with fixed guessing for both groups, 4PLcdg (alternatively also 4PLc) for 4PL model with fixed inattention for both groups, or 4PL for 4PL model.

The model can be specified in more detail with constraints argument which specifies what parameters should be fixed for both groups. For example, choice "ad" means that discrimination (parameter "a") and inattention (parameter "d") are fixed for both groups and other parameters ("b" and "c") are not. The NA value for constraints means no constraints.

In case that the model considers a difference in guessing or inattention parameter, different parameterization is used and parameters with standard errors are re-calculated by delta method.

Value

A list with the following arguments:

Sval the values of test statistics.

pval the p-values by test.

adjusted.pval adjusted p-values by p.adjust.method.

df the degrees of freedom of test.

test used test.

par.m0 the matrix of estimated item parameters for null model.

se.m0 the matrix of standard errors of item parameters for null model.

cov.m0 list of covariance matrices of item parameters for null model.

par.m1 the matrix of estimated item parameters for alternative model.

se.m1 the matrix of standard errors of item parameters for alternative model.

cov.m1 list of covariance matrices of item parameters for alternative model.

conv.fail numeric: number of convergence issues.

conv.fail.which the indicators of the items which did not converge.

ll.m0 log-likelihood of null model.

ll.m1 log-likelihood of alternative model.

startBo0 the binary matrix. Columns represents iterations of initial values re-calculations, rows represents items. The value of 0 means no convergence issue in null model, 1 means convergence issue in null model.

startBo1 the binary matrix. Columns represents iterations of initial values re-calculations, rows represents items. The value of 0 means no convergence issue in alternative model, 1 means convergence issue in alternative model.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

Karel Zvara
Faculty of Mathematics and Physics, Charles University

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Hladka, A. (2021). Statistical models for detection of differential item functioning. Dissertation thesis. Faculty of Mathematics and Physics, Charles University.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

Swaminathan, H. & Rogers, H. J. (1990). Detecting differential item functioning using logistic regression procedures. Journal of Educational Measurement, 27(4), 361–370, doi:10.1111/j.1745-3984.1990.tb00754.x

See Also

p.adjust

Examples

```
## Not run:
# loading data
data(GMAT)
Data <- GMAT[, 1:20] # items
group <- GMAT[, "group"] # group membership variable

# testing both DIF effects using LR test (default)
# and model with fixed guessing for both groups
NLR(Data, group, model = "3PLcg")

# using F test
NLR(Data, group, model = "3PLcg", test = "F")

# testing both DIF effects with Benjamini-Hochberg correction
NLR(Data, group, model = "3PLcg", p.adjust.method = "BH")

# 4PL model with the same guessing and inattention
# to test uniform DIF
NLR(Data, group, model = "4PLcgdg", type = "udif")

# 2PL model to test non-uniform DIF
NLR(Data, group, model = "2PL", type = "nudif")
```

```
# 4PL model with fixed a and c parameter
# to test difference in b
NLR(Data, group, model = "4PL", constraints = "ac", type = "b")

# using maximum likelihood estimation method with L-BFGS-B algorithm
NLR(Data, group, model = "3PLcg", method = "likelihood")

# using maximum likelihood estimation method with iteratively reweighted least squares algorithm
NLR(Data, group, model = "2PL", method = "irls")

## End(Not run)
```

ORD *DIF likelihood ratio statistics for ordinal data.*

Description

Calculates DIF likelihood ratio statistics for ordinal data based either on adjacent category logit regression model or on cumulative logit regression model.

Usage

```
ORD(Data, group, model = "adjacent", type = "both", match = "zscore",
    anchor = 1:ncol(Data), p.adjust.method = "none",
    alpha = 0.05, parametrization)
```

Arguments

Data	data.frame or matrix: dataset which rows represent ordinaly scored examinee answers and columns correspond to the items.
group	numeric: binary vector of group membership. "0" for reference group, "1" for focal group.
model	character: logistic regression model for ordinal data (either "adjacent" (default) or "cumulative"). See **Details**.
type	character: type of DIF to be tested. Either "both" for uniform and non-uniform DIF (i.e., difference in parameters "a" and "b") (default), or "udif" for uniform DIF only (i.e., difference in difficulty parameter "b"), or "nudif" for non-uniform DIF only (i.e., difference in discrimination parameter "a"). Can be specified as a single value (for all items) or as an item-specific vector.
match	numeric or character: matching criterion to be used as an estimate of trait. Can be either "zscore" (default, standardized total score), "score" (total test score), or vector of the same length as number of observations in Data.
anchor	character or numeric: specification of DIF free items. A vector of item identifiers (integers specifying the column number) specifying which items are currently considered as anchor (DIF free) items. Argument is ignored if match is not "zscore" or "score".

`p.adjust.method`
: character: method for multiple comparison correction. Possible values are `"holm"`, `"hochberg"`, `"hommel"`, `"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, and `"none"` (default). For more details see `p.adjust`.

`alpha`
: numeric: significance level (default is 0.05).

`parametrization`
: deprecated. Use `coef.difORD` for different parameterizations.

Details

Calculates DIF likelihood ratio statistics based either on adjacent category logit model or on cumulative logit model for ordinal data.

Using adjacent category logit model, logarithm of ratio of probabilities of two adjacent categories is

$$log(P(y = k)/P(y = k - 1)) = b_0k + b_1 * x + b_2k * g + b_3 * x : g,$$

where x is by default standardized total score (also called Z-score) and g is a group membership.

Using cumulative logit model, probability of gaining at least k points is given by 2PL model, i.e.,

$$P(y >= k) = exp(b_0k + b_1 * x + b_2k * g + b_3 * x : g)/(1 + exp(b_0k + b_1 * x + b_2k * g + b_3 * x : g)).$$

The category probability (i.e., probability of gaining exactly k points) is then $P(y = k) = P(y >= k) - P(y >= k + 1)$.

Both models are estimated by iteratively reweighted least squares. For more details see `vglm`.

Value

A list with the following arguments:

`Sval` the values of likelihood ratio test statistics.

`pval` the p-values by likelihood ratio test.

`adj.pval` the adjusted p-values by likelihood ratio test using `p.adjust.method`.

`df` the degress of freedom of likelihood ratio test.

`par.m0` the estimates of null model.

`par.m1` the estimates of alternative model.

`se.m0` standard errors of parameters in null model.

`se.m1` standard errors of parameters in alternative model.

`cov.m0` list of covariance matrices of item parameters for null model.

`cov.m1` list of covariance matrices of item parameters for alternative model.

`ll.m0` log-likelihood of null model.

`ll.m1` log-likelihood of alternative model.

`AIC.m0` AIC of null model.

`AIC.m1` AIC of alternative model.

`BIC.m0` BIC of null model.

`BIC.m1` BIC of alternative model.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Agresti, A. (2010). Analysis of ordinal categorical data. Second edition. John Wiley & Sons.

Hladka, A. (2021). Statistical models for detection of differential item functioning. Dissertation thesis. Faculty of Mathematics and Physics, Charles University.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

See Also

p.adjust vglm

Examples

```
## Not run:
# loading data
data(Anxiety, package = "ShinyItemAnalysis")
Data <- Anxiety[, paste0("R", 1:29)] # items
group <- Anxiety[, "gender"] # group membership variable

# testing both DIF effects
ORD(Data, group, type = "both")

# testing uniform DIF effects
ORD(Data, group, type = "udif")

# testing non-uniform DIF effects
ORD(Data, group, type = "nudif")

# testing DIF using cumulative logit model
ORD(Data, group, model = "cumulative")

## End(Not run)
```

`plot.ddfMLR` *ICC plots for an object of* `"ddfMLR"` *class.*

Description

Plot method for an object of `"ddfMLR"` class using **ggplot2**.

The characteristic curves for an item specified in `item` argument are plotted. Plotted curves represent the best model.

Usage

```
## S3 method for class 'ddfMLR'
plot(x, item = "all", group.names, ...)
```

Arguments

`x`	an object of `"ddfMLR"` class.
`item`	numeric or character: either character `"all"` to apply for all items (default), or a vector of item names (column names of `Data`), or item identifiers (integers specifying the column number).
`group.names`	character: names of reference and focal group.
`...`	other generic parameters for `plot()` function.

Value

Returns list of objects of class `"ggplot"`.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

`ddfMLR` for DDF detection.
`ggplot` for general function to plot a `"ggplot"` object.

Examples

```
## Not run:
# loading data
data(GMATtest, GMATkey)
Data <- GMATtest[, 1:20] # items
group <- GMATtest[, "group"] # group membership variable
key <- GMATkey # correct answers

# testing both DDF effects
(x <- ddfMLR(Data, group, focal.name = 1, key))

# graphical devices
plot(x, item = "Item1", group.names = c("Group 1", "Group 2"))
plot(x, item = x$DDFitems)
plot(x, item = c(3, 1, 5))

## End(Not run)
```

`plot.difNLR` *ICC and test statistics plots for an object of* `"difNLR"` *class.*

Description

Plot method for an object of `"difNLR"` class using **ggplot2**.

Two types of plots are available. The first one is obtained by setting `plot.type = "cc"` (default). The characteristic curves for an item specified in `item` argument are plotted. Plotted curves represent the best model.

The second plot is obtained by setting `plot.type = "stat"`. The test statistics (either LR-test, or F-test, depends on argument `test`) are displayed on the Y axis, for each converged item. The detection threshold is displayed by a horizontal line and items detected as DIF are printed with the red color. Only parameters `size` and `title` are used.

Usage

```
## S3 method for class 'difNLR'
plot(
  x,
  plot.type = "cc",
  item = "all",
  group.names,
  draw.empirical = TRUE,
  draw.CI = FALSE,
  ...
)
```

Arguments

x	an object of "difNLR" class.
plot.type	character: type of plot to be plotted (either "cc" for characteristic curve (default), or "stat" for test statistics).
item	numeric or character: either character "all" to apply for all converged items (default), or a vector of item names (column names of Data), or item identifiers (integers specifying the column number).
group.names	character: names of reference and focal group.
draw.empirical	logical: should empirical probabilities be plotted as points? Default value is TRUE.
draw.CI	logical: should confidence intervals for predicted values be plotted? Default value is FALSE.
...	other generic parameters for plot() function.

Value

For an option plot.type = "stat", returns object of class "ggplot". In case of plot.type = "cc", returns list of objects of class "ggplot".

Outputs can be edited and modified as standard "ggplot" object including colours, titles, shapes or linetypes.

Note that option draw.CI = TRUE returns confidence intervals for predicted values as calculated by predict.difNLR. Confidence intervals may overlap even in case that item functions differently.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

Karel Zvara
Faculty of Mathematics and Physics, Charles University

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

Swaminathan, H. & Rogers, H. J. (1990). Detecting differential item functioning using logistic regression procedures. Journal of Educational Measurement, 27(4), 361–370, doi:10.1111/j.1745-3984.1990.tb00754.x

See Also

`difNLR` for DIF detection among binary data using generalized logistic regression model. `predict.difNLR` for prediction. `ggplot` for general function to plot a `"ggplot"` object.

Examples

```
## Not run:
# loading data
data(GMAT)
Data <- GMAT[, 1:20] # items
group <- GMAT[, "group"] # group membership variable

# testing both DIF effects using likelihood-ratio test and
# 3PL model with fixed guessing for groups
(x <- difNLR(Data, group, focal.name = 1, model = "3PLcg"))

# item characteristic curves
plot(x)
plot(x, item = x$DIFitems)
plot(x, item = 1)
plot(x, item = "Item1")

# item characteristic curves without empirical probabilities
plot(x, item = 1, draw.empirical = FALSE)

# item characteristic curves without empirical probabilities but with CI
plot(x, item = 1, draw.empirical = FALSE, draw.CI = TRUE)

# graphical devices - test statistics
plot(x, plot.type = "stat")

## End(Not run)
```

---

plot.difORD *ICC plots for an object of "difORD" class.*

Description

Plot method for an object of `"difORD"` class using **ggplot2**.

The characteristic curves (category probabilities) for an item specified in `item` argument are plotted. Plotted curves represent the best model. For cumulative logit model, also cumulative probabilities may be plotted.

Usage

```
## S3 method for class 'difORD'
plot(x, item = "all", plot.type, group.names, ...)
```

Arguments

x	an object of "difORD" class.
item	numeric or character: either character "all" to apply for all converged items (default), or a vector of item names (column names of Data), or item identifiers (integers specifying the column number).
plot.type	character: which plot should be displayed for cumulative logit regression model. Either "category" (default) for category probabilities or "cumulative" for cumulative probabilities.
group.names	character: names of reference and focal group.
...	other generic parameters for plot() function.

Value

Returns list of objects of class "ggplot".

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

See Also

difORD for DIF detection among ordinal data.
ggplot for general function to plot a "ggplot" object.

Examples

```
## Not run:
# loading data
data(Anxiety, package = "ShinyItemAnalysis")
Data <- Anxiety[, paste0("R", 1:29)] # items
group <- Anxiety[, "gender"] # group membership variable

# testing both DIF effects with adjacent category logit model
(x <- difORD(Data, group, focal.name = 1, model = "adjacent"))
```

```
# graphical devices
plot(x, item = 6)
plot(x, item = "R6", group.names = c("Males", "Females"))

# testing both DIF effects with cumulative logit model
(x <- difORD(Data, group, focal.name = 1, model = "cumulative"))
plot(x, item = 7, plot.type = "cumulative")
plot(x, item = 7, plot.type = "category")

## End(Not run)
```

`predict.ddfMLR` *Predicted values for an object of* `"ddfMLR"` *class.*

Description

S3 method for predictions from the model used in the object of `"ddfMLR"` class.

Usage

```
## S3 method for class 'ddfMLR'
predict(object, item = "all", match, group, ...)
```

Arguments

`object`	an object of `"ddfMLR"` class.
`item`	numeric or character: either character `"all"` to apply for all converged items (default), or a vector of item names (column names of `Data`), or item identifiers (integers specifying the column number).
`match`	numeric: matching criterion for new observations.
`group`	numeric: group membership for new observations.
`...`	other generic parameters for `predict()` function.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

See Also

ddfMLR for DDF detection among nominal data using multinomial log-linear regression model.
predict for generic function for prediction.

Examples

```
## Not run:
# loading data
data(GMATtest, GMATkey)
Data <- GMATtest[, 1:20] # items
group <- GMATtest[, "group"] # group membership variable
key <- GMATkey # correct answers

# testing both DDF effects
(x <- ddfMLR(Data, group, focal.name = 1, key))

# fitted values
predict(x, item = 1)

# predicted values
predict(x, item = 1, match = 0, group = c(0, 1))
predict(x, item = x$DDFitems, match = 0, group = c(0, 1))

## End(Not run)
```

---

predict.difNLR *Predicted values for an object of "difNLR" class.*

Description

S3 method for predictions from the model used in the object of "difNLR" class.

Usage

```
## S3 method for class 'difNLR'
predict(
  object,
  item = "all",
  match,
  group,
  interval = "none",
  level = 0.95,
  ...
)
```

Arguments

`object`	an object of `"difNLR"` class.
`item`	numeric or character: either character `"all"` to apply for all converged items (default), or a vector of item names (column names of `Data`), or item identifiers (integers specifying the column number).
`match`	numeric: matching criterion for new observations.
`group`	numeric: group membership for new observations.
`interval`	character: type of interval calculation, either `"none"` (default) or `"confidence"` for confidence interval.
`level`	numeric: confidence level.
`...`	other generic parameters for `predict()` function.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

Karel Zvara
Faculty of Mathematics and Physics, Charles University

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

Swaminathan, H. & Rogers, H. J. (1990). Detecting differential item functioning using logistic regression procedures. Journal of Educational Measurement, 27(4), 361–370, doi:10.1111/j.1745-3984.1990.tb00754.x

See Also

`difNLR` for DIF detection among binary data using generalized logistic regression model.
`predict` for generic function for prediction.

Examples

```
## Not run:
# loading data
data(GMAT)
Data <- GMAT[, 1:20] # items
group <- GMAT[, "group"] # group membership variable

# testing both DIF effects using likelihood-ratio test and
# 3PL model with fixed guessing for groups
(x <- difNLR(Data, group, focal.name = 1, model = "3PLcg"))

# predicted values
summary(predict(x))
predict(x, item = 1)
predict(x, item = "Item1")

# predicted values for new observations - average score
predict(x, item = 1, match = 0, group = 0) # reference group
predict(x, item = 1, match = 0, group = 1) # focal group

# predicted values for new observations - various z-scores and groups
new.match <- rep(c(-1, 0, 1), 2)
new.group <- rep(c(0, 1), each = 3)
predict(x, item = 1, match = new.match, group = new.group)

# predicted values for new observations with confidence intervals
predict(x, item = 1, match = new.match, group = new.group, interval = "confidence")
predict(x, item = c(2, 4), match = new.match, group = new.group, interval = "confidence")

## End(Not run)
```

---

predict.difORD *Predicted values for an object of "difORD" class.*

Description

S3 method for predictions from the model used in the object of "difORD" class.

Usage

```
## S3 method for class 'difORD'
predict(object, item = "all", match, group, type = "category", ...)
```

Arguments

object	an object of "difORD" class.
item	numeric or character: either character "all" to apply for all converged items (default), or a vector of item names (column names of Data), or item identifiers (integers specifying the column number).

match	numeric: matching criterion for new observations.
group	numeric: group membership for new observations.
type	character: type of probability to be computed. Either "category" for category probabilities or "cumulative" for cumulative probabilities. Cumulative probabilities are available only for cumulative logit model.
...	other generic parameters for predict() function.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Hladka, A. & Martinkova, P. (2020). difNLR: Generalized logistic regression models for DIF and DDF detection. The R Journal, 12(1), 300–323, doi:10.32614/RJ2020014.

See Also

difORD for DIF detection among ordinal data using either cumulative logit or adjacent category logit model.
predict for generic function for prediction.

Examples

```
## Not run:
# loading data
data(Anxiety, package = "ShinyItemAnalysis")
Data <- Anxiety[, paste0("R", 1:29)] # items
group <- Anxiety[, "gender"] # group membership variable

# testing both DIF effects with cumulative logit model
(x <- difORD(Data, group, focal.name = 1, model = "cumulative"))

# fitted values
predict(x, item = "R6")

# predicted values
predict(x, item = "R6", match = 0, group = c(0, 1))
predict(x, item = "R6", match = 0, group = c(0, 1), type = "cumulative")
predict(x, item = c("R6", "R7"), match = 0, group = c(0, 1))

# testing both DIF effects with adjacent category logit model
(x <- difORD(Data, group, focal.name = 1, model = "adjacent"))
```

```
# fitted values
predict(x, item = "R6")

# predicted values
predict(x, item = "R6", match = 0, group = c(0, 1))
predict(x, item = c("R6", "R7"), match = 0, group = c(0, 1))

## End(Not run)
```

`startNLR` *Calculates starting values for non-linear regression DIF models.*

Description

Calculates starting values for `difNLR()` function based on linear approximation.

Usage

```
startNLR(Data, group, model, match = "zscore", parameterization = "alternative",
         simplify = FALSE)
```

Arguments

`Data`	Data data.frame or matrix: dataset which rows represent scored examinee answers (`"1"` correct, `"0"` incorrect) and columns correspond to the items.
`group`	numeric: binary vector of group membership. `"0"` for reference group, `"1"` for focal group.
`model`	character: generalized logistic regression model for which starting values should be estimated. See **Details**.
`match`	character or numeric: matching criterion to be used as estimate of trait. Can be either `"zscore"` (default, standardized total score), `"score"` (total test score), or numeric vector of the same length as number of observations in `Data`.
`parameterization`	character: parameterization of regression coefficients. Possible options are `"classic"` (IRT parameterization), `"alternative"` (default) and `"logistic"` (logistic regression). See **Details**.
`simplify`	logical: should initial values be simplified into the matrix? This is only applicable when parameterization is the same for all items.

Details

The unconstrained form of 4PL generalized logistic regression model for probability of correct answer (i.e., $y = 1$) is

$$P(y = 1) = (c+cDif*g)+(d+dDif*g-c-cDif*g)/(1+exp(-(a+aDif*g)*(x-b-bDif*g))),$$

where x is by default standardized total score (also called Z-score) and g is a group membership. Parameters a, b, c, and d are discrimination, difficulty, guessing, and inattention. Terms $aDif$, $bDif$, $cDif$, and $dDif$ then represent differences between two groups (reference and focal) in relevant parameters.

The `model` argument offers several predefined models. The options are as follows: `Rasch` for 1PL model with discrimination parameter fixed on value 1 for both groups, `1PL` for 1PL model with discrimination parameter fixed for both groups, `2PL` for logistic regression model, `3PLcg` for 3PL model with fixed guessing for both groups, `3PLdg` for 3PL model with fixed inattention for both groups, `3PLc` (alternatively also `3PL`) for 3PL regression model with guessing parameter, `3PLd` for 3PL model with inattention parameter, `4PLcgdg` for 4PL model with fixed guessing and inattention parameter for both groups, `4PLcgd` (alternatively also `4PLd`) for 4PL model with fixed guessing for both groups, `4PLcdg` (alternatively also `4PLc`) for 4PL model with fixed inattention for both groups, or `4PL` for 4PL model.

Three possible parameterization can be specified in `"parameterization"` argument: `"classic"` returns IRT parameters of reference group and differences in these parameters between reference and focal group. `"alternative"` returns IRT parameters of reference group, the differences in parameters `"a"` and `"b"` between two groups and parameters `"c"` and `"d"` for focal group. `"logistic"` returns parameters in logistic regression parameterization.

Value

A list containing elements representing items. Each element is a named numeric vector of length 8 with initial values for generalized logistic regression model.

Author(s)

Adela Hladka (nee Drabinova)
Institute of Computer Science of the Czech Academy of Sciences
Faculty of Mathematics and Physics, Charles University
<hladka@cs.cas.cz>

Patricia Martinkova
Institute of Computer Science of the Czech Academy of Sciences
<martinkova@cs.cas.cz>

References

Drabinova, A. & Martinkova, P. (2017). Detection of differential item functioning with nonlinear regression: A non-IRT approach accounting for guessing. Journal of Educational Measurement, 54(4), 498–517, doi:10.1111/jedm.12158.

Hladka, A. (2021). Statistical models for detection of differential item functioning. Dissertation thesis. Faculty of Mathematics and Physics, Charles University.

See Also

`difNLR`

Examples

```
# loading data
data(GMAT)
Data <- GMAT[, 1:20] # items
group <- GMAT[, "group"] # group membership variable

# starting values for 3PL model
startNLR(Data, group, model = "3PL")

# starting values for 3PL model
# simplified into single table
startNLR(Data, group, model = "3PL", simplify = TRUE)

# starting values for 3PL model
# with score as matching criterion
startNLR(Data, group, model = "3PL", match = "score")

# starting values for model specified for each item
startNLR(Data, group,
  model = c(
    rep("1PL", 5), rep("2PL", 5),
    rep("3PL", 5), rep("4PL", 5)
  )
)
```

Index