Package 'polytomous'

July 2, 2014

Type Package

Title Polytomous logistic regression for fixed and mixed effects

Version 0.1.6

Date 2013-12-20

Author Antti Arppe

Maintainer Antti Arppe <antti.arppe@iki.fi>

Description Logistic regression modeling for polytomous settings (more than two categorical outcomes) with both fixed and mixed effect predictors, and univariate and bivariate analysis of categorical, unordered data.

Acknowledgements Ideas and input provided by R. Harald Baayen as well as Terrance M. Nearey in the development of the polytomous Poisson reformulation function allowing for mixed-effects logistic regression modeling of polytomous outcome settings are greatly appreciated.

Depends R(>= 2.10.1), stats, MASS, Hmisc, lme4

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2013-12-20 13:39:44

R topics documented:

`anova.polytomous` *Analysis of Model Fit for Polytomous Logistic Regression models*

Description

Calculate an analysis of individual variable contributions or model comparisons for one or more Polytomous Logistic Regression models.

Usage

```
## S3 method for class 'polytomous'
anova(object, ..., statistic = "deviance", test = "Chisq",
   outcome.specific = FALSE)
```

Arguments

`object, ...`	objects of class `"polytomous"`, typically the result of a call to `polytomous`, or a list of `objects` for the `polytomouslist` method.
`statistic`	a character string, determining the statistic for evaluating model fit, by default `"deviance"`, alternatively `"AIC"` or `"BIC"`.
`test`	a character string, determining the statistical method by which the significance of the comparison are done, by default the Chi-squared test (`"Chisq"`); currently no other methods are implemented. If set to `NULL`, no significance testing will be undertaken.
`outcome.specific`	a logical, which, if set `TRUE` in the case of a single `"polytomous"` `object` fit using `heuristic="one.vs.rest"`, will result in the presentation of the application of `anova.glm` on the outcome-specific reduction of `deviance` of the constituent binary models; by default set `FALSE` resulting in a conventional ANOVA table.

Details

Specifying a single object gives a table with sequential analysis of predictor impact with respect to the selected statistic of model fit. That is, the reductions in the residual statistic as each term of the formula is added in turn are given in as the rows of a table, plus the residual statistic values themselves.

If more than one object is specified, the table has a row for the residual degrees of freedom and selected statistic for each model. For all but the first model, the change in degrees of freedom and the statistic is also given. (This only makes statistical sense if the models are nested.) It is conventional to list the models from smallest to largest, but this is up to the user.

The table will optionally contain test statistics (and P values) comparing the reduction in deviance for the row to the residuals. Only a comparison of models or contributions of their components by the chi-squared test has been implemented, which is applicable only for the `"deviance"` statistic.

The comparison between two or more models by `anova.polytomous`, redirected to the `anova.polytomouslist` method, will only be valid if they are fitted to the same dataset and with the same heuristic. `anova.polytomouslist` will look for such discrepancies, resulting in an error when detected.

If `outcome.specific=TRUE`, the function will alternatively output the outcome-specific reductions of `deviance` for sequantially added predictors for each of the constituent binary models, using `anova.glm`. Thus, `outcome.specific=TRUE` is only applicable for a single `object` of the class `"polytomous"` fit with `heuristic="one.vs.rest"` and when `statistic="deviance"`.

Value

An object of class `"anova"` inheriting from class `"data.frame"`, when `outcome.specific=FALSE`.

When `outcome.specific=TRUE`, the function will produce a list with the following components:

`model` A list of outcome-specific results procuded by `anova.glm` on the constituent binary models.

`deviance` A table with outcome-specific reductions of `deviance` (columns) for each sequentially added predictor (rows).

`df` A table with outcome-specific reductions of degrees of freedom (columns) for each sequentially added predictor (rows).

`p.values` A table with outcome-specific evaluations (using the Chi-squared test) of the significance of reduction in `deviance` (columns) for each sequentially added predictor (rows).

Author(s)

Antti Arppe

References

Antti. A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

See Also

`polytomous`, `anova.glm`

Examples

```
data(think)
think.polytomous1 <- polytomous(Lexeme ~ Agent * Patient, data=think)
anova(think.polytomous1)

## Not run:
anova(think.polytomous1, statistic="AIC")
anova(think.polytomous1, statistic="BIC", test=NULL)

## End(Not run)

anova(think.polytomous1, outcome.specific=TRUE)

think.polytomous2 <- polytomous(Lexeme ~ Agent * Patient + Manner,
   data=think)
anova(think.polytomous1, think.polytomous2)

## Not run:
anova(think.polytomous1, think.polytomous2, statistic="AIC", test=NULL)

## End(Not run)
```

---

associations *Calculate measures of association for a two-way contingency table*

Description

`associations` takes a two-way contingency table of two categorical, unordered variables (with possibly multiple nominal values), and calculates a range of measures of association between the two variables.

Usage

```
associations(ctable, alpha=0.05, p.zero.correction = 1/sum(ctable)^2)
```

Arguments

`ctable`	a two-way contingency table cross-tabulating the co-occurrence counts of two categorical, unordered variables, with possible multiple nominal values.
`alpha`	the significance threshold (P-value) to be used in certain calculations; by default `alpha=0.05`.
`p.zero.correction`	a (very) small value to be substituted when `P=0` for the meaningful calculation of certain statistics based on logarithmic functions; by default specified according to the sum frequency of the contingecy table `ctable`.

Value

A list with the following components:

`alpha.X2` α: significance estimate of *Pearson* chi-squared (χ^2) test of independence (homogeneity); the probability of falsely rejecting the null hypothesis of independence when in actual fact it is true.

`alpha.G2` α: significance estimate of *Log-likelihood* ratio (G^2) test of independence (homogeneity).

`beta` β: the probability of falsely accepting, i.e. failing to reject, the null hypothesis of independence when in fact the alternative hypothesis of dependence is true, equal to $1 - power$.

`power` Probability of (correctly) rejecting the null hypothesis of independence when it is indeed false and the alternative hypothesis of dependence is true, equal to $1 - \beta$.

`effect.size` Cohen's *Effect Size* (Cohen 1988).

`likelihood.ratio` Log-likelihood ratio.

`cramers.v` Cramer's *V* (Cramer 1946).

`lambda.RC` Goodman-Kruskall $\lambda(R|C)$ indicating how much knowing the values of the independent *Column* variable increases the prediction accuracy of the values of the dependent *Row* variable, over always selecting the *Row* mode value (Goodman and Kruskall 1954).

`lambda.CR` Goodman-Kruskall $\lambda(C|R)$ indicating how much knowing the values of the independent *Row* variable increases the prediction accuarcy of the values of the dependent *Column* variable, over always selecting the *Column* mode value.

`tau.RC` Goodman-Kruskall $\tau(R|C)$ indicating how much knowing the values of the independent *Column* variable increases the prediction accuracy of the probabilities of values of the dependent *Row* variable, over a baseline of knowing only the overall probabilities of the classes of the dependent *Row* variable (Liebetrau 1983).

`tau.CR` Goodman-Kruskall $\tau(C|R)$ indicating how much knowing the values of the independent *Row* variable increases the prediction accuracy of the probabilities of values of the dependent *Column* variable, over a baseline of knowing only the overall probabilities of the classes of the dependent *Column* variable.

`uc.RC` Theil's Uncertainty Coefficient $UC(R|C)$, indicating how much knowing the values of the independent *Column* variable decreases uncertainty about the values of the dependent *Row* variable (Theil 1970).

`uc.CR` Theil's Uncertainty Coefficient $UC(C|R)$, indicating how much knowing the values of the independent *Row* variable decreases uncertainty about the values of the dependent *Column* variable.

`uc.sym` Theil's symmetric Uncertainty Coefficient UC, indicating the aggregate of how much knowing the values of either the *Row* or the *Column* variables decreases uncertainty about the values of each other.

`p.lambda.RC` Probability of observing $\lambda(R|C)$ by chance, when the distribution in the underlying sampling population is in fact homogeneous.

`p.lambda.CR` Probability of gaining $\lambda(C|R)$ by chance.

`p.tau.RC` Probability of gaining $\tau(R|C)$ by chance.

`p.tau.CR` Probability of gaining $\tau(C|R)$ by chance.

`p.uc.RC` Probability of gaining $UC(R|C)$ by chance.

`p.uc.CR` Probability of gaining $UC(C|R)$ by chance.

`var.lambda.RC` Variance of $\lambda(R|C)$.

`var.lambda.CR` Variance of $\lambda(C|R)$.

`var.tau.RC` Variance of $\tau(R|C)$.

`var.tau.CR` Variance of $\tau(C|R)$.

`var.uc.RC` Variance of $UC(R|C)$.

`var.uc.CR` Variance of $UC(C|R)$.

`ASE.lambda.RC` Asymptotic standard error of $\lambda(R|C)$.

`ASE.lambda.CR` Asymptotic standard error of $\lambda(C|R)$.

`ASE.tau.RC` Asymptotic standard error of $\tau(R|C)$.

`ASE.tau.CR` Asymptotic standard error of $\tau(C|R)$.

`ASE.uc.RC` Asymptotic standard error of $UC(R|C)$.

`ASE.uc.CR` Asymptotic standard error of $UC(C|R)$.

`noncentrality` Noncentrality parameter.

Acknowledgements

I appreciate having had access to a similar function script `measures.R` by Marc Schwartz, from whom I have also received valuable assistance in finding sources for the computation of the variances, and thus the other statistics based on them.

Author(s)

Antti Arppe

References

Agresti, A. (2002) Categorical Data Analysis (2nd edition). Hoboken: John Wiley & Sons, Hoboken.

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Cohen, J. (1988) Statistical power analysis for the behavioral sciences, (2nd edition). Hillsdale: Lawrence Erlbaum Associates.

Cramer, H. (1946) Mathematical Methods in Statistics. Princeton: Princeton University Press.

Goodman, L. A. and W. H. Kruskal (1954) Measures of Association for Cross- Classifications. Journal of the American Statistical Association, Vol. 49, No. 268 (December 1954), pp. 732–764.

Liebetrau, A. M. (1983) Measures of Association. Sage University Paper series on Quantitative Applications in the Social Sciences, 07-032. Beverly Hills and London: Sage Publications.

Theil, H. (1970) On the Estimation of Relationships Involving Qualitative Variables. The American Journal of Sociology, Vol. 76, No. 1 (July 1970), pp. 103–154.

See Also

See also chisq.posthoc, chisq.test.

Examples

```
data(think)
ctable <- table(think$Lexeme, think$Agent)
associations(ctable)

associations(table(think$Agent, think$Patient))
```

---

chisq.posthoc *Calculate cellwise posthoc analyses for a two-way contingency table*

Description

chisq.posthoc takes a contingency table crosstabulating two categorical, unordered variables, the overall independence/dependence of which has been evaluated with chisq.test, and calculates several variations of posthoc analyses concerning the impact of individual cells containing frequencies of value pairings of the two categorical variables, assessing the degree to which these individual cellwise observed values diverge (or not) from an overall hypothetical homogeneous distribution.

Usage

```
chisq.posthoc(ctable, alpha = 0.05, reorder = "none",
   std.pearson.residual.min = 2, correct=FALSE)
```

Arguments

ctable	a two-way contigency table crosstabulating two, possible multiple-valued categorical, unordered variables
alpha	a numerical value between 0 and 1 specifying the critical P-value threshold for significance; by default set to 0.05
reorder	a character string specifying whether the rows or columns of ctable, or both, should be reordered according to their descending marginal frequencies; possible values "none" (default), "both", "rows", or "cols"; if none of these are provided, ctable will be left as it is.
std.pearson.residual.min	the minimum absolute value for considering a cellwise standardized Pearson residual in the contingency table to deviate significantly from the expected value (representing a homogeneous distribution); by default equal to 2.
correct	a logical indicating whether to apply Yates' continuity correction when computing the test statistic; by default set to FALSE

Details

The cellwise posthoc analyses are variations based on the overall chi-squared test for homogeneity/heterogeneity of frequency distributions represented in contingency tables as implemented in `chisq.test`.

Though for smaller contingency tables the suggested absolute minimum cellwise value for a standardized Pearson residual to be considered to signal a potentially significant cellwise divergence (in relation to an overall homogeneous distribution for the entire table) is 2 or more, this absolute threshold value should probably be increased to 3 or more in the case of larger contingency tables (Agresti 2002).

With respect to minimum cell-wise expected counts, in contrast to `chisq.test` no warnings will be output.

Value

A list with the following components:

`ctable` the two-way contingency table, reordered as specified by the `reorder` argument.

`X2.df` the minimum value of the chi-squared test statistic for the degrees of freedom (`df=(nrow(ctable)-1)*(ncol(ctable`
of the contingency table so that the distribution of counts in the contingency table can be considered to overall diverge significantly (from the expected values representing a homogeneoues distribution), so that the probability of observing by chance such a distribution are at most `P=alpha`.

`X2.df1` the minimum value of the chi-squared test statistic for the minimum `df=1`.

`cells` a list of different assessments of the divergences of the cellwise values from expected values representing a homogeneous distribution, consisting of the following elements:

- `X2` the cellwise contributions to the chi-squared statistic, with the sign indicating whether the observed value is greater or less than the expected value.
- `X2.df.sign` the cellwise assessment of whether the chi-squared value of an individual cell by itself exceeds the overall minimum chi-squared value `X2.df` for a significantly non-homogeneous distribution of counts; having the values '+' or '-' when this is the case, with the sign indicating whether the cellwise observed value is greater or less than the expected value, or '0' otherwise (indicating no significant divergence).
- `X2.df1.sign` the cellwise assessment of whether the chi-squared value of an individual cell exceeds the minimum chi-squared value `X2.df` when `df=1`; having the values '+', '-', or '0'.
- `std.pearson.residuals` the cellwise standardized Pearson residuals.
- `std.pearson.residuals.sign` the cellwise assessment of whether the absolute value of a standardized Pearson residual is greater than `std.pearson.residual.min`, having the values '+', '-', or '0'.

Author(s)

Antti Arppe

References

Agresti, A. (2002) Categorical Data Analysis (2nd edition). Hoboken: John Wiley & Sons, Hoboken.

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Liebetrau, A. M. (1983) Measures of Association. Sage University Paper series on Quantitative Applications in the Social Sciences, 07-032. Beverly Hills and London: Sage Publications.

See Also

See also `chisq.test`, `associations`.

Examples

```
data(think)
ctable <- table(think$Lexeme, think$Agent)
chisq.posthoc(ctable)

chisq.posthoc(table(think$Agent, think$Patient))
```

---

crosstable.statistics *Calculate prediction accuracy statistics for a contingency table*

Description

`crosstable.statistics` takes a contingency table of observed vs. predicted values for a binary or polytomous response variable as input, and calculates a range of statistics about prediction accuracy.

Usage

```
crosstable.statistics(ctable)
```

Arguments

`ctable`	A contingency table cross-classifying observed and predicted values.

Value

A list with the following components:

`accuracy` Overall prediction accuracy

`recall.predicted` Recall of prediction for each outcome value

`precision.predicted` Precision of prediction for each outcome value

`lambda.prediction` $\lambda_{prediction}$: improvement in prediction accuracy over baseline of always predicting mode

`tau.classification` $\tau_{classification}$: improvement in classification accuracy over baseline of homogeneous distribution of predicted outcomes

`d.lambda.prediction` $d(\lambda_{prediction})$: used for calculating `p.lambda.prediction`

`d.tau.classification` $d(\tau_{classification})$: used for calculating `p.tau.classification`

`p.lambda.prediction` $P(\lambda_{prediction})$: probability of reaching $\lambda_{prediction}$ by chance

`p.tau.classification` $P(\tau_{classification})$: probability of reaching $\tau_{classification}$ by chance

Author(s)

Antti Arppe

References

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Arppe, A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

Menard, S. (1995). Applied Logistic Regression Analysis. Sage University Paper Series on Quantitative Applications in the Social Sciences 07-106. Thousand Oaks: Sage Publications.

See Also

See also `model.statistics`.

Examples

```
ctable=matrix(c(30, 10, 5, 60),2,2)
crosstable.statistics(ctable)
```

---

extract.exemplars *Extract a subset of exemplary exemplars from a dataset*

Description

A function that extracts a subset of exemplary exemplars from a dataset that has been used to fit a polytomous logistic regression model. Hierarchical agglomeratative clustering (HAC) is used to divide the dataset into distinct subsets in terms of their features/properties; from each such cluster a single exemplar with an outcome that has the highest expected probability for all the exemplars within the cluster is selected. Consequently, the number of exemplars that is extracted from the dataset equals the number of clusters into which the dataset is divided.

Usage

```
extract.exemplars(model.polytomous, model.hclust=NULL, n.clusters=10, p.bins=0,
   features=FALSE)
```

Arguments

`model.polytomous`
: an object of class `"polytomous"` that has been fitted with the function `polytomous`.

`model.hclust`
: an object of class `"hclust"` that has been produced by applying the function `hclust` to the dataset. If none is provided `NULL` (default), one will be automatically created using the elements `data` (which must consist of logical features/properties) and `formula` included in the polytomous model inputted as the argument `model.polytomous`; the lables in `formula` will specify the features that will be used in the clustering.

`n.clusters`
: a numeric argument specifying the number of clusters into which the dataset will be divided by applying the function `cutree` on the hierarchical agglomerative clustering model; by default `=10`.

`p.bins`
: a numeric argument specifying whether the exact probability estimates for the outcomes as provided by `model.polytomous` will be used in selecting the individual exemplars from the clusters (the default case determined by setting the value as `=0`), or the number of equal interval probability bins into which the exact probability estimates will be divided. In the latter case, the exemplar selected from the highest probability bin might not have received the absolutely highest probability estimate.

`features`
: a logical (by default `=FALSE`) indicating whether the number of features/properties evident in the individual exemplars in the dataset and belonging to the set of features specified in the `formula` will be used as a secondary ranking factor in addition to the outcome-specific probability estimates (or probability bins). When `features=TRUE` and `p.bins >= 2`, the exemplar with the highest number of features/properties will be selected from the highest probability bin per each cluster.

Details

The hierarchical agglomerative clustering, if done automatically within the function, will be undertaken with the function `hclust` using `method="ward"` on a distance matrix created using the `dist` function with `method="binary"`. Therefore, the `data` and `formula`in `model.polytomous` must consist of logical features/properties. The transformation of multinomial features to logical ones can be undertaken with `multinomial2logical`.

Value

`extract.exemplars` returns a list with the following components:

`indices` A numeric vector of indices of the exemplar rows in the dataset that have been extracted.

`outcomes` A factor containing the outcomes in each of the individual exemplars.

`max.probs` A numeric vector of the exact probability estimates for the outcome apparent in each exemplar.

`properties` A factor with the set of features/properties evident in each exemplar; the properties are separated by a semicolon and space.

Author(s)

Antti Arppe

References

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Divjak, D. and A. Arppe (2013). Extracting prototypes from exemplars. What can corpus data tell us about concept representation? Cognitive Linguistics, 24 (2): 221-274.

See Also

`extract.prototypes`, `polytomous`, `hclust`, `dist`

Examples

```
## Not run:
data(think)
think.logical <- multinomial2logical(data=think, outcome="Lexeme",
   variables=names(think)[2:24])
think.formula <- as.formula(paste("Lexeme",
   paste(grep("(Other)|(None)|(FiniteVerbChain)|(Overt)",
   names(think.logical)[-1],value=TRUE,invert=TRUE),collapse=" + "),sep=" ~ "))
think.polytomous <- polytomous(think.formula, data=think.logical)
extract.exemplars(think.polytomous,n.clusters=50)

## End(Not run)

## For more details, see vignette.
```

---

extract.prototypes *Extract the prototypical features for a set of outcomes*

Description

A function that extracts for each outcome included in a polytomous logistic regression model fitted with the function `polytomous` the set of features/properties that together can be interpreted to represent the prototypical characteristics of the outcome in question.

Usage

```
extract.prototypes(model.polytomous, p.critical=.05)
```

Arguments

model.polytomous	an object of class "polytomous" that has been fitted with the function polytomous.
p.critical	a numeric value specifying the critical p-level (by default =.05) for the coefficient (i.e. odds/logodds) estimated by polytomous for each feature/property included in the regression function to be considered statistically significant

Details

This function in effect automatically selects and groups together in a convenient manner per each outcome in the polytomous logistic regression model the set of significant explanatory features/properties.

Value

extract.prototypes returns a named list (by outcomes) consisting per each outcome of a single-column matrix with the significant feature/properties as rownames and the associated estimated odds as values, sorted in decreasing order for each outcome.

Author(s)

Antti Arppe

References

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Divjak, D. & A. Arppe (2013). Extracting prototypes from exemplars. What can corpus data tell us about concept representation? Cognitive Linguistics, 24 (2): 221-274

See Also

extract.exemplars, codepolytomous

Examples

```
data(think)
think.polytomous <- polytomous.one.vs.rest(Lexeme ~ Agent + Patient, data=think,
   heuristic="one.vs.rest")
extract.prototypes(think.polytomous)

## For more details, see vignette.
```

instance2narrowcount *Transformation of an uncounted instance-specific data table into a count data table in the* "narrow" *format*

Description

Transforms a data table with uncounted instance-by-instance information on the co-occurrences of individual outcomes with various predictor values into a count table in the "narrow" format, with frequency counts for the outcomes in conjunction with unique combinations of predictor variable values.

Usage

```
instance2narrowcount(data.table, variables, outcome = "OUTCOME",
   variables.default = NULL, outcome.ordered = NULL,
   numeric2discrete = function(x) cut2(x,levels.mean=TRUE,g=g.numeric),
   g.numeric = 2)
```

Arguments

data.table	a data table with instance-by-instance information on the occurrence of individual outcome variable variables in conjunction with specific values of predictor variables.
variables	a list of the names of predictor variables (columns in data.table) to be included in the creation of the count.table.
outcome	a name of the outcome variable (column in data.table); by default "OUTCOME".
variables.default	a list indicating for selected categorical predictors the value(s) that should be designated as the default/reference levels; by default NULL, in which case the original default/reference levels as specified for predictors in the object referred to by the data.table argument will be used.
outcome.ordered	a list specifying the order of the categories for the outcome/response variable; by default NULL, in which case the original order specified in the object referred to by the data argument will be used.
numeric2discrete	a function to transform a continuous numeric predictor into a discrete set of numeric values, by default cut2 from the Hmisc package with the preset parameters levels.mean=TRUE and g=g.numeric (by default =2). If set to NULL, each value of each numeric predictor will be treated as a discrete value of its own.
g.numeric	a parameter to be passed to the numeric2discrete function (parameter g for Hmisc::cut2(..., g=g.numeric, ...), or a user-defined function), determining the desired number of values for each numeric predictor; by default equal to 2.

Details

Transforms a data table with uncounted instance-by-instance information on the co-occurrences of individual outcomes with various predictor values into a count table in the `"narrow"` count format, with frequency counts for the outcomes in conjunction with unique combinations of predictor variable values.

Note that numeric variables will remain numeric despite the reduction of their distinct values using the `numeric2discrete` function.

Value

A count data table with the frequency counts for each unique combination of outcomes and predictor variable values. In addition to columns with values for each included predictor, the count data table has the following common columns:

`"Proportion"` the relative proportion of the specific outcome in conjunction with the specific combination of selected predictor variables (in relation to all the outcomes for the particular unique combination of predictor variables.

`"Count"` the frequency count of the specific outcome value in conjunction with the specific combination of selected predictor variables.

`outcome` the name of the response variable designated by the `outcome` argument; by default `"OUTCOME"`

`"Observation"` the index number for each unique combination of values of selected predictor variables.

Author(s)

Antti Arppe

References

Antti. A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

See Also

`polytomous.poisson.reformulation`, `wide2narrowcount`

Examples

```
data(think)
think.counts <- instance2narrowcount(think, c("Agent","Patient"), "Lexeme")
think.counts

think.poisson <- glm(Count ~ Observation + Lexeme + Lexeme:Agent + Lexeme:Patient,
   data=think.counts, family=poisson)
summary(think.poisson)

think.polytomous.poisson1 <- polytomous(Lexeme ~ Agent + Patient, data=think.counts,
   frequency="Count", heuristic="poisson.reformulation")
summary(think.polytomous.poisson1)
```

```
think.polytomous.poisson2 <- polytomous(Lexeme ~ Agent + Patient, data=think,
   heuristic="poisson.reformulation")
summary(think.polytomous.poisson2)

## Not run:

library(lme4)
think.counts2 <- instance2narrowcount(think, c("Agent","Patient","Section"), "Lexeme")
think.poisson.lmer <- lmer(Count ~ (1|Observation) + Lexeme + Lexeme:Agent +
   Lexeme:Patient + (1|Section), data=think.counts2, family=poisson)
summary(think.poisson.lmer)

think.polytomous.lmer <- polytomous(Lexeme ~ Agent + Patient + (1|Section), data=think)
summary(think.polytomous.lmer)


## End(Not run)
```

`model.statistics` *Calculate statistics for goodness of fit and prediction accuracy for a model*

Description

Calculate a range of goodness of fit measures for an model object fitted with some multivariate statistical method that yields probability estimates for outcomes.

Usage

```
model.statistics(observed, predicted, p.values, frequency = NA,
   outcomes = NULL, p.normalize = TRUE, cross.tabulation = TRUE,
   p.zero.correction=1/(nrow(p.values)*ncol(p.values))^2, ...)
```

Arguments

`observed`	observed values of the response variable
`predicted`	predicted values of the response variable; typically the outcome estimated to have the highest probability
`p.values`	matrix of probabilities for all values of the response variable (i.e outcomes)
`frequency`	A numeric vector (or the name of a column in the input data frame) with the frequencies of the instance. If absent (set to `NA`), each exemplar is assigned a frequency equal to 1.
`outcomes`	the outcome categories
`p.normalize`	if `TRUE`, probabilities are normalized so that `sum(P)` of all outcomes for each datapoint is equal to 1

`cross.tabulation`
: if `TRUE`, statistics on the crosstabulation of observed and predicted response values are calculated with `crosstable.statistics`

`p.zero.correction`
: a function to adjust slightly response/outcome-specific probability estimates which are exactly P=0; necessary for the proper calculation of pseudo-R-squared statistics; by default calculated on the basis of the dimensions of the matrix of probabilities `p.values` as `1/(nrow(p.values)*ncol(p.values))^2`

`...`
: further control arguments to be passed from and to other functions.

Value

A list with the following components:

`loglikelihood.null` Loglikelihood for null model

`loglikelihood.model` Loglikelihood for fitted model

`deviance.null` Null deviance

`deviance.model` Model deviance

`R2.likelihood` (McFadden's) R-squared

`R2.nagelkerke` Nagelkerke's R-squared

`crosstable` Crosstabulation of observed vs. predicted outcomes.

`crosstable.statistics(crosstable)` Various statistics calculated on the crosstabulation (`crosstable`) with `crosstable.statistics`, if `cross.tabulation=TRUE`

Author(s)

Antti Arppe

References

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Arppe, A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

Hosmer, D. W., Jr., and S. Lemeshow (2000) Applied Regression Analysis (2nd edition). New York: Wiley.

See Also

`crosstable.statistics`, `polytomous.one.vs.rest`, `polytomous.poisson.reformulation`, `polytomous`

Examples

```
## None for the time being
```

`multinomial2logical` *Transformation of a data frame with multinomial variable columns into a* `"logical"` *format*

Description

Transforms a data frame (or an object that can be coerced in such) consisting of multinomial variables as columns into a logical format, where each unique value of each multinomial variable is represented by a corresponding logical variable.

Usage

```
multinomial2logical(data, outcome=NULL, variables=NULL,
   variable.value.separator="")
```

Arguments

`data`	a data frame (or an object that can be coerced into such) with columns of multinomial variable.s
`outcome`	the name of the outcome variable (column in `data`) that will be retained as a multinomial variable column in the resultant data frame; by default `NULL`.
`variables`	a list of the names of predictor variables (columns in `data`) to be included in the creation of the resultant data frame.
`variable.value.separator`	a character (string) used in creating new column names for the resultant data frame by pasting together variable names and their respective variable values; by default `""`

Details

Transforms a data frame (or an object that can be coerced in such) consisting of multinomial variables as columns into a logical format, where each unique value of each multinomial variable is represented by a corresponding logical variable. The logical format allows for an easy specification of individual variable values as predictors in the formula of a polytomous logistic regression model.

Value

A data frame with logical variables representing each unique value of all (or selected) multinomial variable columns in `data`. When such a value of a multinomial variable is present on an observation row in the original `data`, the value of the corresponding logical variable is `TRUE`, whereas otherwise it is `FALSE`. Each logical variable is named as a combination of the corresponding original multinomial variable (column) name and variable value, separated by the `variable.value.separator` (by default ‘’). For instance, the logical variable corresponding to any occurrence of `"Group"` class of the variable `"Agent"` in the `think` dataset is named `"AgentGroup"`.

If an `outcome` column is specified, the resultant data frame will have as its first column the corresponding multinomial variable; this is necessary for using the resultant data frame as the input data for `polytomous`.

Author(s)

Antti Arppe

References

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Arppe, A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

See Also

polytomous, polytomous.one.vs.rest, polytomous.poisson.reformulation, instance2narrowcount, wide2narrowcount

Examples

```
data(think)
think.logical <- multinomial2logical(think, outcome="Lexeme",
   variables=c("Agent","Patient"))
think.polytomous1 <- polytomous(Lexeme ~ AgentIndividual + AgentGroup +
  PatientAbstraction + PatientActivity, data=think.logical)
summary(think.polytomous1)

think.polytomous2 <- polytomous(Lexeme ~ AgentIndividual + AgentGroup +
   PatientAbstraction + PatientActivity, data=think.logical,
   heuristic="poisson.reformulation",
   outcome.ordered=c("harkita","miettia","pohtia","ajatella"))
summary(think.polytomous2)
```

---

nominal *Univariate and bivariate statistics for categorical, unordered variables*

Description

nominal takes a data frame with categorical (i.e. nominal) variables and calculates a range of categorical statistics and posthoc analyses.

Usage

```
nominal(formula, data, sort.bivariate = NULL, std.pearson.residual.min = 2,
   correct = FALSE, report.interval = 100, factor2logical = FALSE)

## S3 method for class 'nominal'
```

```
print(x, max.print = 10,
   posthoc = "std.pearson.residuals.sign",
   assoc = ifelse("univariate" %in% class(x),
     list(c("N", "alpha.X2", "uc.12", "uc.21")),
     list(c("N1", "N2", "N12", "uc.12", "uc.21"))),
   sort.key = NULL, ...)

## S3 method for class 'nominal'
summary(object, posthoc = "std.pearson.residuals.sign",
   assoc = ifelse("univariate" %in% class(object),
     list(c("N", "alpha.X2", "uc.12", "uc.21")),
     list(c("N1", "N2", "N12", "uc.12", "uc.21"))),
   sort.key = NULL, ...)

## S3 method for class 'summary.nominal'
print(x, max.print = 10, ...)
```

Arguments

formula	a formula (see Details specifying either (1) univariate analysis focusing on the relationship of one dependent categorical outcome variable (with possible multiple classes) and one or more categorical explanatory independent variables or (2) bivariate analysis scrutinizing the interrelationships of two or more categorical independent variables.
data	data frame (or an object coercible by as.data.frame to a data frame) containing the variables specified in the formula as columns. In the case of univariate analysis, the dependent outcome variable and the corresponding column can either be a multiple-value factor or a set of logical/binary variables. For both univariate and bivariate analyses, if factor2logical=FALSE (default), all independent variables and the respective columns in the data must be logical/binary; if factor2logical=TRUE, the variables can be factors which will then be automatically transformed into corresponding new logical/binary variables using the multinomial2logical function.
x	an object of class "nominal", usually resulting from a call to nominal; or an object of class "summary.nominal", usually resulting from a call to summary.nominal.
object	an object of class "nominal", usually resulting from a call to nominal.
sort.bivariate	a character string, any one in the set c("uc","lambda","tau") specifying one three asymmetric measures of association (see associations), or NULL, specifying in the case of bivariate analysis whether the two contrasted categorical variables with be sorted so that the occurrence of the first categorical (logical/binary) variable in question (category1 in the output) explains more of the occurrence (or absence) of the second categorical variable (category2), in contrast to the other way around. If e.g. sort.bivariate="uc", the uc.21>uc.12, i.e. $UC(2\|1) > UC(1\|2)$. By default set to NULL so that no sorting will take place.
std.pearson.residual.min	the minimum absolute value for considering in univariate analysis a cellwise

standardized Pearson residual in the contingency table to deviate significantly from the expected value (representing a homogeneous distribution); by default equal to 2 (see chisq.posthoc).

correct — a logical control parameter indicating whether to apply Yates' continuity correction when computing the (X^2) statistic in the chi-squared test (cf. chisq.posthoc and chisq.test); by default set to FALSE.

report.interval — a numeric variable indicating the interval at which the progress of the calculation of the statistics will be reported, by default set to =100. This is useful when the number of explanatory independent variables is great in the case of univariate analysis, and especially when their pairwise combinations are high in bivariate analysis, which is often the case.

posthoc — a character string (or vector of strings) specifying which of the posthoc analyses of a chi-squared test (any one of c("X2", "X2.df.sign", "X2.df1.sign", "std.pearson.residual output by chisq.posthoc) will be output and included in sumry.table that is generated by print.nominal or summary.nominal. By default set to "std.pearson.residuals.sign"; if set to NULL, all posthoc analyses are excluded.

assoc — A character vector indicating which of the measures of association output by associations will be output and included in sumry.table generated by print.nominal or summary.nominal. By default set to c("N", "alpha.X2", "uc.12", "uc.21") in univariate analysis, and to c("N1", "N2", "N12", "uc.12", "uc.21") in bivariate analysis. If set to NULL, all measures of association are excluded.

sort.key — A character string specifying the measure of association, or some other statistic output by nominal, according to which the results will be sorted in decreasing order (any one of c("uc.12", "uc.21", "lambda.12", "lambda.21", "tau.12", "tau.21") for both univariate and bivariate analyses, and also c("N", "alpha.X2") for univariate analysis, and c("N1", "N2", "N12") for bivariate analysis). By default set to NULL, in which case no sorting is done.

factor2logical — a logical indicating whether (multiple-value) factors in data referred to in formula should be automatically transformed into a corresponding set of binary/logical variables to be used in the ensuing analysis; by default set to =FALSE, in which case reference to non-logical variables will result in an error.

max.print — the maximum number of rows of the parameter to be output when printing with print.summary.polytomous; by default set to 20; if set to NA all rows will be output.

... — further arguments passed to or from other methods.

Details

The specification of the formula determines whether a univariate or bivariate analysis is calculated. In univariate analysis, one of the variables, e.g. an outcome, is considered dependent of each of a number of other, independent variables, i.e. explanatory predictors. In bivariate analysis no such single dependent variable is assumed; in contrast, the interest is in the pairwise interrelationships of the selected variables. In such a case, each variable in these pairings is in both a dependent and independent role.

Typical usages for univariate analysis are:

```
nominal(formula=dependent ~ independent1 + independent2 + ..., data=...)
nominal(formula=dependent1 + dependent2 + ... ~ independent1 + ..., data=...)
nominal(formula=dependent1 + dependent2 + ... ~ ., data=...)
nominal(formula=dependent ~ ., data=...)
```

Typical usages for bivariate analysis are:

```
nominal(formula=. ~ independent1 + independent2 + ..., data=...)
nominal(formula=. ~ ., data=...)
```

In the univariate case, the formula has the form '`dependent ~ independent1 + independent2 + ...`'. Here, the dependent variable is assumed to be a multiple-value factor and the independent variables as binary/logical ones in `data`, Alternatively, the dependent variable can also be represented as multiple binary/logical variables, so that formula takes the form '`dependent1 + dependent2 + ... ~ independent1 + ...`'. In such a case, the values of the specified dependent variable values must cover the entire data. In the special case '`dependent ~ .`', the '.' is taken to represent all the other variables in `data` except the dependent one.

In the bivariate case, the formula has the form '`. ~ independent1 + independent2 + ...`'. Here, the specified independent variables are assumed to be logical/binary variables and are each contrasted against all the others pairwise to study their interrelationships. In the special case '`. ~ .`', all the variables in the data are contrasted against each other pairwise. N.B. in this latter case, too, all the variables are assumed to be logical/binary ones.

Value

For univariate analysis, a list with the following components:

`univariate` a named list for each independent variable value (see `independents` below), containing for each a list with two elements, (1) `posthoc` with the results of `chisq.posthoc` and (2) `assoc` with the results of `associations`, both for the crosstabulation of the independent variable in question and dependent variable values

`std.pearson.residuals.sign` a matrix of the cellwise assessments, for each independent variable (rows) per each dependent outcome variable value (columns), of whether the absolute value of a standardized Pearson residual is greater than `std.pearson.residual.min` (when the independent variable is present, i.e. `TRUE`), having the values '+', '-', or '0'.

`std.pearson.residuals` a matrix of cellwise standardized Pearson residuals, for each independent variable (rows) per each dependent outcome variable value (columns).

`X2.df.sign` a matrix of the cellwise assessments, for each independent variable (rows) per each dependent outcome variable value (columns), of whether the chi-squared contribution of an individual cell by itself exceeds the overall minimum chi-squared value, based on the degrees of freedom for the full dimensions of the crosstabulation (cf. `X2.df` in `chisq.posthoc`), for a significantly non-homogeneous distribution of counts; having the values '+' or '-' when this is the case, with the sign indicating whether the cellwise observed value is greater or less than the expected value, or '0' otherwise (indicating no significant divergence)

`X2.df1.sign` a matrix of the cellwise assessments, for each independent variable (rows) per each dependent outcome variable value (columns), of whether the chi-squared contribution of an individual cell by itself exceeds the minimum chi-squared value, based on degrees of freedom as `=1` (cf. `X2.df1` in `chisq.posthoc`), for a significantly non-homogeneous distribution of

counts; having the values '+' or '-' when this is the case, with the sign indicating whether the cellwise observed value is greater or less than the expected value, or '0' otherwise (indicating no significant divergence)

`X2` a matrix of the cellwise contributions, for each independent variable (rows) per each dependent outcome variable (columns), to the chi-squared statistic for each outcome variable value (columns) per each independent variable (rows), with the sign indicating whether the observed value is greater or less than the expected value.

`assocs` a named list for each independent variable with the various measures of association calculated with `associations` between the independent variable in question and the values of the dependent outcome variable.

In the notation of the association measures, the number codes in e.g. `uc.12` and `uc.21` refer to the direction (conditionality) of the asymmetric measures. For instance, `uc.12` corresponds to $UC(independent|dependent)$, i.e. the reduction of uncertainty concerning the `independent` variable when knowing the dependent `outcome` variable, whereas `uc.21` corresponds to $UC(dependent|independent)$, i.e. the reduction of uncertainty concerning the `dependent` outcome variable when knowing the `independent` predictor variable.

`dependents` a character string with the name of the dependent variable.

`dependent.values` a character string vector with the multiple values, i.e. classes, of the dependent variable.

`independents` a character string vector with the names of the independent variables.

For bivariate analysis, a list containing the following components:

`bivariate` a data frame with the various measures of association, calculated with `associations`, between the pairings of the independent variables (categories).

The two independent variables to be contrasted, `category1` and `category2`, are provided in columns 1-2, the frequency of the first category `N1`, the frequency of the second category `N2`, and their joint frequency `N12`, in columns 3-5, followed by the rest of the association measures.

In the notation of the association measures, the number codes in e.g. `uc.12` and `uc.21` refer to the direction (conditionality) of the asymmetric measures. For instance, `uc.12` corresponds to $UC(category1|category2)$, i.e. the reduction of uncertainty concerning `category1` when knowing `category2`, whereas `uc.21` corresponds to $UC(category2|category1)$, i.e. the reduction of uncertainty concerning `category2` when knowing `category1`.

`independents` a character string vector with the names of the independent variables.

For `summary.nominal`, the list of results is supplemented with one additional element:

`sumry.table` a data frame with various statistics selected out of the results of `nominal` according to the arguments `posthoc` and `assoc` provided to `summary.nominal` (see above).

For univariate analysis, each row consists of the statistics for each indepedent variable (in relation to the values of the dependent outcome variable values which are given as columns by default). For bivariate analysis, each row consists of the statistics for a (unique) pairing of two independent variables.

Author(s)

Antti Arppe

References

Agresti, A. (2002) Categorical Data Analysis (2nd edition). Hoboken: John Wiley & Sons, Hoboken.

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Cohen, J. (1988) Statistical power analysis for the behavioral sciences, (2nd edition). Hillsdale: Lawrence Erlbaum Associates.

Cramer, H. (1946) Mathematical Methods in Statistics. Princeton: Princeton University Press.

Goodman, L. A. and W. H. Kruskal (1954) Measures of Association for Cross- Classifications. Journal of the American Statistical Association, Vol. 49, No. 268 (December 1954), pp. 732–764.

Liebetrau, A. M. (1983) Measures of Association. Sage University Paper series on Quantitative Applications in the Social Sciences, 07-032. Beverly Hills and London: Sage Publications.

Theil, H. (1970) On the Estimation of Relationships Involving Qualitative Variables. The American Journal of Sociology, Vol. 76, No. 1 (July 1970), pp. 103–154.

See Also

See also `chisq.posthoc`, `associations`.

Examples

```
data(think)
think.logical <- multinomial2logical(think, c("Agent","Patient"),
   outcome="Lexeme")

think.univariate <- nominal(Lexeme ~ ., data=think.logical)
summary(think.univariate)

think.bivariate <- nominal(. ~ ., data=think.logical[-1])
summary(think.bivariate)
```

`plot.polytomous` *Plot function for selected results of* `polytomous`.

Description

This function presents visually the estimated logodds or odds, or expected probabilities for a model fitted with `polytomous` and its auxiliary functions `polytomous.one.vs.rest` or `polytomous.poisson.reformulation`

Usage

```
## S3 method for class 'polytomous'
plot(x, values="probabilities", ...)

## S3 method for class 'polytomous.parameters'
plot(x, values="logodds",
type="density", predictors=NULL, outcomes=NULL, panes="single",
lty=NULL, col=NULL, mfrow=NULL, main=NULL,
legend.position="topright", ...)

## S3 method for class 'polytomous.probabilities'
plot(x, type="density",
select="all", panes="single", lty=NULL, col=NULL, pch=NULL,
mfrow=NULL, main=NULL, legend.position="topright", ...)
```

Arguments

x	A object of the class "polytomous" produced by polytomous or its auxiliary functions polytomous.one.vs.rest or polytomous.poisson.reformulation, consisting of a list including estimated logodds or odds for predictors and estimated probabilities of outcomes for outcome-predictor combinations.
values	A character string specifiying whether expected "probabilities" (default) or estimated "logodds" or "odds" should be plotted.
type	A character string spefifying the type of plot to be drawn; "density" is available for both value types as default, while a histogram ("hist") is available only for plot.polytomous.parameters and sorted values ("sort") only for plot.polytomous.probabilities.
panes	A character string specifying whether a "single" pane (default) integrating all component plots, or "multiple" panes for each individual component plot are to be plotted. If "multiple" panes are selected, the number or rows and columns is specified automatically. Alternatively, one can invoke the plotting of multiple panes by explicitly specifying the appropriate number of rows and columns with the parameter mfrow (N.B. this overrides panes="single").
predictors	A regular expression specifying which predictors and their values should be included in the plot(s); by default NULL so that all predictors incorporated in the "polytomous" model will be included.
outcomes	A list of outcomes to be included in the plot; by default NULL so that all outcomes will be considered.
select	For the method plot.polytomous.probabilities, a character string specifying which instance-wise probability estimates should be plotted; by default "all", other values are "max" for instance-wise maximum probabilities, "min" for instance-wise minimum probabilities, "maxmin", "minmax" for both maximum and minimum instance-wise probabilities. Alternatively, a numeric vector c(1,2,...) specifying selected ranks of the instance-wise probability estimates can be provided, with 1 corresponding to the instance-wise maximum probability estimates.

`lty, col, pch, mfrow, main, legend.position`
: Specifications of various graphical parameters (see par) to be used in the plots; if any of these is set to =NULL default settings will be used (for legend.position, the default value is topright). Note that lty is relevant only to plot.polytomous.parameters(..., ty and plot.polytomous.probabilities(..., type="density", ...), and pch only to plot.polytomous.probabilities(..., type="sort", ...).

`...`
: Arguments to be passed to methods, such as graphical parameters (see par).

Value

A plot of the selected type is produced on the graphics device.

Author(s)

Antti Arppe

References

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Arppe, A. (2009) Linguistic choices vs. probabilities – how much and what can linguistic theory explain? In: Featherston, S. and S. Winkler, (eds.) The Fruits of Empirical Linguistics. Volume 1: Process. Berlin: de Gruyter, pp. 1–24.

Antti. A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

See Also

polytomous, polytomous.one.vs.rest, polytomous.poisson.reformulation

Examples

```
data(think)
think.polytomous <- polytomous(Lexeme ~ Agent + Patient + Section,
   data=think)

plot(think.polytomous, values="logodds")
plot(think.polytomous, values="logodds", type="hist", panes="multiple")
plot(think.polytomous, values="logodds", type="density", panes="multiple")
plot(think.polytomous, values="logodds", type="density", panes="multiple",
   predictors="Section*")
plot(think.polytomous, values="logodds", type="density", panes="multiple",
   predictors="Patient*")
plot(think.polytomous, values="logodds", type="hist", panes="multiple", col=1:4)
plot(think.polytomous, values="logodds", type="density", panes="single",
   outcomes=c("ajatella","miettia","pohtia","harkita"))

plot(think.polytomous, values="probabilities")
plot(think.polytomous, values="probabilities", panes="multiple")
```

```
plot(think.polytomous, values="probabilities", select="max")
plot(think.polytomous, values="probabilities", select=c(1:3))
plot(think.polytomous, values="probabilities", panes="multiple", select=c(1:3))
plot(think.polytomous, values="probabilities", type="sort", legend.position="topleft")
plot(think.polytomous, values="probabilities", type="sort", pch=".",
   legend.position="topleft")
plot(think.polytomous, values="probabilities", type="sort", pch=".", panes="multiple")
```

`polytomous` *Fitting polytomous logistic regression models for fixed or mixed effects predictors.*

Description

`polytomous` is a top-level function that is used to select one of the heuristics for fitting a polytomous logistic regression model for fixed or mixed effects predictors; the arguments supplied to `polytomous` are passed on to the appropriate function implementing the indicated heuristic.

Usage

```
polytomous(formula, data, heuristic = "one.vs.rest", ...)

## S3 method for class 'polytomous'
print(x, max.print = 10, ...)
```

Arguments

`formula`	an object of class `"formula"` (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model formula specification are given under `Details`.
`data`	a data frame (or an object coercible by `as.data.frame` to a data frame) containing the variables specified in the model. The data may be represented either in the uncounted `"instance"` or the `"wide"` count format; see `Details`.
`heuristic`	the heuristic to be used in fitting the model; by default `"one.vs.rest"`; currently the only alternative is `"poisson.reformulation"`. Fitting mixed-effects models will require using the `"poisson.reformulation"` heuristic.
`x`	An object of the class `"polytomous"` fitted with `polytomous` to be printed with `print.polytomous`.
`max.print`	The maximum number of rows of the `parameter` to be output when printing with `print.summary.polytomous`; by default equal to 10; if set to `NA`, all rows will be output.
`...`	further arguments passed to or from other methods.

Details

A typical predictor has the form `response ~ terms` where `response` is the name of the factorial response vector indicating the possible polytomous outcomes (a column in the `data` argument) and `terms` is a series of terms which specifies a linear predictor for `response`. In this case, the object referred to by the `data` argument is expected to be in the `"narrow"` format, with one row for each observation of an outcome and predictors. Then, the predictors may either be multinomial factors or binary variables (with the values `TRUE, FALSE`).

Alternatively, `response` can consist of the names of the individual outcome values separated by `"|"`, e.g. `response1|response2|response3 ~ ...`. In such a case, the object referred to by `data` argument is expected to be in the `"wide"` format so that is contains a column for each of the indicated outcomes giving their frequency of occurrence for a combination of predictor values indicated on the same row in the `data` argument.

A terms specification of the form `first + second` indicates all the terms in `first` together with all the terms in `second` with any duplicates removed.

A specification of the form `first:second` indicates the the set of terms obtained by taking the interactions of all terms in `first` with all terms in `second`. The specification `first*second` indicates the _cross_ of `first` and `second`. This is the same as `first + second + first:second`.

A specification of the form `fixed|random` indicates a fixed effect predictor for which the impact of random effect (grouping factor) is to be evaluated, a formula with such terms specifying a mixed-effects model. Mixed-effects models with random effect terms can only be specified in the formula when using the `"poisson.reformulation"` heuristic. If `formula` contains random terms, `heuristic` will be automatically switched to `"poisson.reformulation"`.

The actual fitting of the polytomous logistic regression model is undertaken by the function determined by the `heuristic` argument, i.e. `polytomous.one.vs.rest` or `polytomous.poisson.reformulation`. Models with only fixed-effect predictors will be fit using `glm`; models with also random-effect predictors will be fit using `lmer`.

Value

`polytomous` returns an object of class `"polytomous"`, a list containing at least following components:

`model` the underlying model(s) fitted using `glm` or `glmer`.

`data` the originally supplied `data` argument object.

`frequency` the originally supplied `frequency` argument.

`logodds` a matrix of the outcome-by-predictor logodds estimated for the model.

`odds` a matrix of the outcome-by-predictor odds (`exp(logodds)`) estimated for the model.

`p.values` a matrix of the estimates of the significances of the outcome-by-predictor logodds/odds estimated for the model.

`fitted` a matrix of the fitted outcome-specific probability estimates corresponding to the original data table in uncounted `"instance"` or `"narrow"` count or `"wide"` count format.

`statistics` a range of descriptive statistics describing the goodness of fit and classification performance of the model; see `model.statistics` and `crosstable.statistics`.

`formula` the formula specification used to fit the model.

`outcomes` the outcome categories.

`heuristic` the heuristic used to fit the model.

`data.format` the format type of the `data` argument object; having the value of either `"instance"`, `"narrow"`, or `"wide"`.

The functions implementing the various heuristics may provide additional heuristic-specific results, see `polytomous.one.vs.rest` and `polytomous.poisson.reformulation`.

Acknowledgments

Ideas and input provided by R. Harald Baayen as well as Terrance M. Nearey in the development of the `polytomous.poisson.reformulation` function allowing for mixed-effects logistic regression modeling of polytomous outcome settings are greatly appreciated.

Note

In addition to polytomous logistic regression modeling and auxiliary functions, the `polytomous` package also contains three functions for the univariate analysis if data with categorical (nominal), unordered variables, namely `associations`, `chisq.posthoc` and `nominal`.

Author(s)

Antti Arppe

References

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Arppe, A. (2009) Linguistic choices vs. probabilities – how much and what can linguistic theory explain? In: Featherston, S. & S. Winkler (eds.) The Fruits of Empirical Linguistics. Volume 1: Process. Berlin: de Gruyter, pp. 1–24.

Antti. A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

See Also

`polytomous.one.vs.rest`, `polytomous.poisson.reformulation`, `summary.polytomous`, `anova.polytomous`, `predict.polytomous`, `plot.polytomous`, `nominal`,`chisq.posthoc`, `associations`

Examples

```
data(think)
think.polytomous1 <- polytomous(Lexeme ~ Agent + Patient, data=think)
summary(think.polytomous1)

think.polytomous2 <- polytomous(Lexeme ~ Agent + Patient + Register, data=think)
summary(think.polytomous2)
```

```
think.polytomous.lmer1 <-  polytomous(Lexeme ~ Agent + Patient + (1|Register),
   data=think, heuristic="poisson.reformulation")
summary(think.polytomous.lmer1)

## Not run:
think.polytomous3 <- polytomous(Lexeme ~ Agent + Patient + Section,
   data=think)
summary(think.polytomous2)

think.polytomous.lmer2 <-  polytomous(Lexeme ~ Agent + Patient + (1|Section),
   data=think, heuristic="poisson.reformulation")
summary(think.polytomous.lmer2)

## End(Not run)
```

`polytomous.one.vs.rest` *Fitting polytomous logistic regression models with the one-vs-rest heuristic*

Description

A function fitting a polytomous logistic regression model based on the one-vs-rest heuristic. With the one-vs-rest heuristic, each individual outcome is contrasted with all the other outcomes lumped together.

Usage

```
polytomous.one.vs.rest(formula, data, frequency = NA, p.normalize = TRUE, ...)
```

Arguments

`formula`	an object of class `"formula"` (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under `Details` for the `polytomous` function.
`data`	a data frame (or object coercible by `as.data.frame` to a data frame) containing the variables specified in the model. The data may be represented either in the uncounted `"instance"` or the `"wide"` count format; see `Details` for the function `polytomous`.
`frequency`	A numeric vector (or the name of a column in the input data frame) with the frequencies of the instance. If absent (set to `NA`), each exemplar is assigned a frequency equal to 1.
`p.normalize`	a logical indicating whether outcome-specific probability estimates ought to be normalized so that sum(P(outcome|predictors))=1.
`...`	further control arguments passed to or from other methods, see `glm` or `model.statistics`.

Details

polytomous.one.vs.rest fits a polytomous logistic regression model with fixed-effects predictors using the one.vs.rest heuristic, where the occurrences of each individual outcome are contrasted with all the other outcomes (i.e. the "rest") lumped together. Consequently, a polytomous one-vs-rest model consits of a set of binary logistic regression models fitted with glm, with one such model for each possible outcome. Every such binary logistic regression is fitted independently of each other.

The outcome-predictor odds (which are simply exp(logodds)) indicate how much the occurrence of some predictor increases or decreases the odds-ratio of the outcome in question to occur, instead of any other outcome, all other predictors considered equal.

With multiple predictors, due to the independent fitting of the outcome-specific binary logistic regression models, the sums of probability estimates for particular predictor combinations provided by these binary models are not always exactly equal to one (though mostly quite close). Thus, it is recommended that the probability estimates are normalized by setting p.normalize as TRUE (which is the default setting).

Value

polytomous.one.vs.rest returns an object of class c("polytomous","one.vs.rest"), a list containing the following components:

model a list containing the underlying outcome-specific binary models fitted using glm.

data the originally supplied data argument object.

frequency the originally supplied frequency argument.

logodds a matrix of the outcome-by-predictor logodds estimated for the model.

odds a matrix of the outcome-by-predictor odds (exp(logodds)) estimated for the model.

p.values a matrix of the estimates of the significances of the outcome-by-predictor logodds/odds estimated for the model.

fitted a matrix of the fitted outcome-specific probability estimates corresponding to the original data table in uncounted "instance" or "wide" count format.

statistics a range of descriptive statistics describing the goodness of fit and classification performance of the model; see model.statistics and crosstable.statistics.

formula the formula specification used to fit the model.

outcomes the outcome categories.

heuristic the heuristic used to fit the model.

data.format the format type of the data argument object; having the value of either "instance", "narrow", or "wide".

Author(s)

Antti Arppe

References

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Arppe, A. (2009) Linguistic choices vs. probabilities – how much and what can linguistic theory explain? In: Featherston, S. and S. Winkler, (eds.) The Fruits of Empirical Linguistics. Volume 1: Process. Berlin: de Gruyter, pp. 1–24.

Rifkin, R. and A. Klautau (2004) In Defense of One-Vs-All Classification. Journal of Machine Learning Research, pp. 101–141.

See Also

polytomous, polytomous.one.vs.rest, anova.polytomous, predict.polytomous, model.statistics, glm

Examples

```
data(think)
think.polytomous <- polytomous.one.vs.rest(Lexeme ~ Agent + Patient, data=think,
   heuristic="one.vs.rest")
summary(think.polytomous)

think.polytomous$statistics
think.polytomous$logodds
think.polytomous$odds
think.polytomous$p.values

think.Agent_Patient.counts <- instance2narrowcount(think, c("Agent","Patient"),
   "Lexeme")
think.Agent_Patient.wide <- cbind(matrix(think.Agent_Patient.counts$Count,,4,
   byrow=TRUE, dimnames=list(NULL,c("ajatella","harkita","miettia","pohtia"))),
   unique(think.Agent_Patient.counts[c("Agent","Patient")]))
think.Agent_Patient.wide

think.Agent_Patient.counts2 <- wide2narrowcount(think.Agent_Patient.wide,
   variables=c("Agent","Patient"),
   outcomes=c("ajatella","harkita","miettia","pohtia"), outcome="Lexeme")
think.Agent_Patient.counts2
identical(think.Agent_Patient.counts,think.Agent_Patient.counts2)

think.polytomous2 <- polytomous(ajatella|harkita|miettia|pohtia ~ Agent + Patient,
   data=think.Agent_Patient.wide)
summary(think.polytomous2)

identical(think.polytomous$odds, think.polytomous2$odds)
identical(round(think.polytomous$odds,5),round(think.polytomous2$odds,5))
```

`polytomous.poisson.reformulation`
A function fitting a polytomous logistic regression model based on the Poisson-reformulation heuristic.

Description

A function fitting a polytomous logistic regression model based on the Poisson-reformulation heuristic. With the Poisson-reformulation heuristic, the polytomous setting is reformulated using one of the functions `instance2narrowcount` or `wide2narrowcount` as counts of outcome-predictor combinations, for which a logistic regression model can be fit using the `glm` or `lmer` functions with setting `family=poisson`. See `Details` for the further specifics of the Poisson reformulation.

Usage

```
polytomous.poisson.reformulation(formula, data, frequency = NA,
   variables.ordered = NULL, include.Observation = TRUE, ...)
```

Arguments

`formula`	an object of class `"formula"` (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under `Details` for the `polytomous` function.
`data`	a data frame (or object coercible by `as.data.frame` to a data frame) containing the variables specified in the model. The data may be represented either in the `"narrow"` or the `"wide"` format; see `Details` for the function `polytomous`.
`frequency`	a numeric vector (or the name of a column in the input data frame) with the frequencies of the instances. If absent (set to `NA`), each instance is assigned a frequency equal to 1.
`variables.ordered`	a list specifying the order of the predictor variables in the count data table resulting from the Poisson reformulation; by default `NULL`, in which case the order will be alphabetical; passed on to `instance2narrowcount` or `wide2narrowcount`.
`include.Observation`	A logical whether a factor index specifying unique predictor value combinations should be included in the count data table resulting from the Poisson reformulation; by default `TRUE`.
`...`	Control variables to be passed on to other functions, see `instance2narrowcount`, `wide2narrowcount` or `model.statistics`.

Details

With the Poisson reformulation heuristic, the polytomous setting is reformulated using one of the functions `instance2narrowcount` or `wide2narrowcount` as counts of outcome-predictor combinations, for which a logistic regression model can be fit using the `glm` or `lmer` functions with setting `family=poisson`.

With the Poisson reformulation, the original regression `formula` estimating probabilities of polytomous outcomes based on combinations of predictors is transformed into a formula estimating counts of polytomous outcomes in combination with (effectively discrete) predictor values. Consequently, the original outcome term is turned into a predictor and the original predictor terms are re-expressed as interaction terms between outcomes and predictors. For example, the original formula:

```
outcome ~ predictor1 + predictor2
```

is transformed into the following Poisson formula (provided as `formula.poisson` in the results):

```
Count ~ outcome + outcome:predictor1 + outcome:predictor2 +   Observation
```

The auxiliary predictor variable `Observation` is an index designating unique combinations of predictor values specified in the original formula.

When fitting a mixed model with random variables that have large numbers of values with frequencies close or equal to 1, it may be advisable in order to get proper estimation of their impact to set `include.Observation=FALSE`, as unique predictor value combinations indicated by each `Observation` value may effectively be convergent with the values of such random variables.

Value

`polytomous.poisson.reformulation` returns an object of class `c("polytomous","one.vs.rest", model.type)` (where `model.type` is either `"fixed"` or `"mixed"`), a list containing the following components:

`model` a list containing the underlying outcome-specific binary models fitted using `glm` in the case of a `"fixed"` effects model or `lmer` in case of a mixed effects model, with `family=poisson` in either case.

`data` the originally supplied `data` argument object.

`data.poisson` the Poisson reformulated count data table used to fit the model

`frequency` the originally supplied `frequency` argument.

`fitted.poisson` a two-column matrix of the fitted counts and associated probability estimates corresponding to the input Poisson count data table

`fitted` a matrix of the fitted outcome-specific probability estimates corresponding to the original data table in `"instance"`, `"narrow"` or `"wide"` format.

`coefficients` the original coefficients of the model

`logodds` a matrix of the outcome-by-predictor logodds estimated for the model.

`odds` a matrix of the outcome-by-predictor odds (`exp(logodds)`) estimated for the model.

`p.values` a matrix of the estimates of the significances of the outcome-by-predictor logodds/odds estimated for the model.

`statistics` a range of descriptive statistics describing the goodness of fit and classification performance of the model; see `model.statistics` and `crosstable.statistics`. For mixed models with random effects, `statistics` will also include their standard deviations (`sd.ranef`) and variances (`var.ranef`).

`formula` the formula specification used to fit the model.

`formula.poisson` the reformulated formula with counts as the response, used to fit the model.

`outcomes` the outcome categories.

`heuristic` the heuristic used to fit the model, being `"poisson.reformulation"`.

`data.format` the format type of the `data` argument object; having the value of either `"instance"`, `"narrow"`, or `"wide"`.

Author(s)

Antti Arppe, with ideas from R. Harald Baayen and Terrance M. Nearey

References

Arppe, A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

Faraway, J. J. (2006). Extending the Linear Model with R. Boca Raton: Chapman & Hall/CRC, 101-102.

Venables, W. N. and B. D. Ripley (2002). Modern Applied Statistics with S (4th edition). New York: Springer, 199-202.

See Also

instance2narrowcount, wide2narrowcount, model.statistics, polytomous, glm, lmer

Examples

```
data(think)
think.polytomous <- polytomous.poisson.reformulation(Lexeme ~ Agent + Patient,
   data=think)
think.polytomous$statistics
think.polytomous$odds
think.polytomous$p.values

## Not run:
library(nnet)
think.multinom <- multinom(Lexeme ~ Agent + Patient, data=think)
exp(coef(think.multinom))

## End(Not run)

think.counts <- instance2narrowcount(think, c("Agent","Patient"), "Lexeme")
think.poisson <- glm(Count ~ Observation + Lexeme + Lexeme:Agent +
   Lexeme:Patient, data=think.counts, family=poisson)
summary(think.poisson)

## Not run:
think.lmer <- polytomous.poisson.reformulation(Lexeme ~ Agent + Patient +
   (1|Section), data=think)
summary(think.lmer)
ranef(think.lmer)

## End(Not run)
```

predict.polytomous *Predict method for polytomous objects*

Description

Obtains predictions on the basis of a fitted "polytomous" object on data already incorporated in the object or on new data with the same predictors as the originally fitted model object.

Usage

```
## S3 method for class 'polytomous'
predict(object, newdata=NULL, type="response",
   p.normalize = TRUE, ...)
```

Arguments

object	objects of class "polytomous", typically the result of a call to polytomous.
newdata	optionally, a data frame in which to look for variables with which to predict. If omitted (i.e. set to NULL), the fitted linear predictors of the object are used.
type	the type of prediction requested. For the default type="link", the predictions are cumulative log-odds (estimated probabilities on logit scale), while type="response" yields the distributions of predicted probabilities over the outcome responses. The option type="terms" returns a matrix giving the fitted values of each term in the model formula on the linear predictor scale, whereas the option type="choice" produces the predicted individual discrete choices, given the selected predictors. The prediction type values "link" and "terms" are currently only implemented for the "one.vs.rest" heuristic, and are based on the function predict.glm.
p.normalize	a logical indicating whether outcome-specific probability estimates ought to be normalized so that sum(P(outcome|predictors))=1; only applicable and possibly desirable for "polytomous" objects fit with heuristic="one.vs.rest", since the constituent binary models are fit independently of each other.
...	further arguments passed to and from other functions.

Details

If newdata is omitted the predictions are based on the data used for the fit.

Value

a vector or matrix of predictions.

Author(s)

Antti Arppe

References

Antti. A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

See Also

polytomous, polytomous.one.vs.rest, polytomous.poisson.reformulation, predict.glm

Examples

```
data(think)
think.polytomous <- polytomous(Lexeme ~ Agent + Patient, data=think)
head(predict(think.polytomous, type="response"))
predict(think.polytomous, newdata=think[1:20,], type="choice")
```

---

ranef.polytomous *Extract the modes of the random effects for polytomous objects*

Description

A function to extract the conditional modes of the random effects from a fitted mixed-effects"polytomous" model object. For linear mixed models the conditional modes of the random effects are also the conditional means.

Usage

```
  ## S3 method for class 'polytomous'
ranef(object, ...)
```

Arguments

object	an object of a class of fitted models with random effects, typically an "polytomous" object of the type "mixed".
...	additional control arguments to be passed on to the underlying ranef function; see ranef (postVar, codedrop, whichel).

Details

If grouping factor i has k levels and j random effects per level the *i'th* component of the list returned by ranef is a data frame with k rows and j columns. The *k'th* face of this array is a positive definite symmetric j by j matrix. If there is only one grouping factor in the model the variance-covariance matrix for the entire random effects vector, conditional on the estimates of the model parameters and on the data will be block diagonal and this j by j matrix is the *k'th* diagonal block.

Value

A list of data frames, one for each grouping factor for the random effects. The number of rows in the data frame is the number of levels of the grouping factor. The number of columns is the dimension of the random effect associated with each level of the factor.

Author(s)

Antti Arppe

See Also

polytomous.poisson.reformulation, ranef

Examples

```
## Not run:
data(think)
think.lmer <- polytomous(Lexeme ~ Agent + Patient + (1|Section), data=think,
   heuristic="poisson.reformulation")
summary(think.lmer)
ranef(think.lmer)

think.lmer2 <- polytomous(Lexeme ~ Agent + Patient + (1|Section) + (1|Author),
   data=think, heuristic="poisson.reformulation")
summary(think.lmer)
ranef(think.lmer)

## End(Not run)
```

---

shanghainese *Shanghainese topic markers.*

Description

500 occurrences of the five most frequent topic markers 'ne, a, mo, zi, ma' in Shanghainese.

Usage

```
data(shanghainese)
```

Format

A data frame with 500 observations on the following 6 variables.

`TOPIC_MARKER` A factor specifying one of the five topic markers

`TOPIC_LENGTH` A numeric vector specifying the character length of the topic

TOPIC_POS A factor specifying the part-of-speech of the topic

FUNCTION A factor specifying function of the topic

COMMENT_TYPE A factor specifying type of the comment

GENRE A factor specifying the genre in which the topic marker had been used

Details

The five most frequent topic markers 'ne, a, mo, zi, ma' in Shanghainese were extracted SOURCES (REFERENCES). The shanghainese dataset contains a selection of 5 contextual features judged as most informative.

For extensive details of the data and its linguistic and statistical analysis, see Han, Arppe & Newman (forthc.).

References

Han, W., A. Arppe & J. Newman (forthc.) Topic marking in a Shanghainese corpus: From observation to prediction. Corpus Linguistics and Linguistic Theory. DOI: 10.1515/cllt-2013-0014.

Examples

```
## For examples see vignette
```

---

summary.polytomous *A summary of a Polytomous Logistic Regression model*

Description

A summarization method for an object of the class "polytomous".

Usage

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

## S3 method for class 'summary.polytomous'
print(x, digits = max(3, getOption("digits") - 3),
   parameter="odds", max.parameter=ifelse(parameter=="odds",10000,100),
   p.critical=.05, max.print=10, cycles=0, max.denominator=0, ...)
```

Arguments

object	An object of class "polytomous", resulting from a call to polytomous.
x	An object of class "summary.polytomous", usually resulting from a call to summary.polytomous.
digits	The number of significant digits to use when printing.
parameter	The set of parameters to output in printing; by default "odds", alternatively "logodds".
max.parameter	A value specifying an upper limit beyond which exceptionally large (and potentially unreliable and mostly nonsignificant) parameter values will be output as being beyond the scale, as "Inf", "1/Inf", "-Inf"; see Details for the default limit values and specific outputs for the different parameter types.
p.critical	The critical P-value for considering a parameter value significant; by default set to P=.05 as is common in the humanities.
max.print	The maximum number of rows of the parameter to be output when printing with print.summary.polytomous; by default set to 10; if set to NA all rows will be output.
cycles	A value to be passed on to the fractions function in the MASS package for representing odds as fractions; however, fractional representation will not result with the default value =0.
max.denominator	A value to be passed on to the fractions function in the MASS package for representing odds as fractions; however, fractional representation will not result with the default value =0.
...	further arguments passed to or from other methods.

Details

Calculates descriptive statistics of a fitted Polytomous Logistic Regression model and prints a nice summary of the key results.

Parameters for which the respective P-value is greater than the critical threshold value (set with p.critical by default as P=.05), i.e. which would be considered as not significant, are output within parentheses, e.g. "(1.1)".

For parameter="odds", the default maximum output limit value is max.parameter=10000. With this default value, "Inf" will be output if odds > 10000 and "1/Inf" if odds < 1/10000. For parameter="logodds", the default maximum output limit value is max.parameter=100. With this default value, "Inf" will be output if logodds > 100 and "-Inf" if logodds < -100.

Value

summary.polytomous returns an object of the class "summary.polytomous", a list with the following components:

formula The formula specified for the "polytomous" object.

odds, logodds, p.values The estimated odds, logodds, and their P-values

statistics A range of descriptive statistics calculated with model.statistics.

Author(s)

Antti Arppe

References

Antti. A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

See Also

polytomous, model.statistics

Examples

```
data(think)
think.polytomous <- polytomous(Lexeme ~ Agent + Patient, data=think)
print(summary(think.polytomous), digits=2, parameter="odds")
print(summary(think.polytomous), digits=4, parameter="logodds")

## For more examples see examples(polytomous).
```

---

think *Finnish 'think' verbs.*

Description

3404 occurrences of four synonymous Finnish 'think' verbs ('ajatella': 1492; 'miettia': 812; 'pohtia': 713; 'harkita': 387) in newspaper and Internet newsgroup discussion texts

Usage

```
data(think)
```

Format

A data frame with 3404 observations on the following 27 variables.

`Lexeme` A factor specifying one of the four 'think' verb synonyms

`Polarity` A factor specifying whether the 'think' verb has negative polarity or not (=Other)

`Voice` A factor specifying whether the 'think' verb is in the passive voice or not (=Other)

`Mood` A factor specifying whether the 'think' verb is in the indicative or conditional mood or not (=Other)

`Person` A factor specifying whether the 'think' verb is in the first, second, third person or not (=None)

`Number` A factor specifying whether the 'think' verb is in the plural number or not (=Other)

`Covert` A factor specifying whether the agent/subject of the 'think' verb is explicitly expressed as a syntactic argument (=`Overt`), or only as a morphological feature of the 'think' verb (=`Covert`)

`ClauseEquivalent` A factor specifying whether the 'think' verb is used as a non-finite clause equivalent (=`ClauseEquivalent`) or as a finite verb for, (=`FiniteVerbChain`)

`Agent` A factor specifying the occurrence of Agent/Subject of the 'think' verb as either a Human Individual, Human Group, or as absent (=`None`)

`Patient` A factor specifying the occurrence of the Patient/Object argument among the semantic or structural subclasses as either an Human Individual/Group, Abstraction, Activity, Communication, an 'etta' ('that') clause (=`etta_CLAUSE`), DirectQuote, IndirectQuestion, Infinitive, Participle, or as absent (=`None`)

`Manner` A factor specifying the occurrrence of the Manner argument as any of its subclasses Generic, Negative (sufficiency), Positive (sufficiency), Frame, Agreement (Agree or Disagree), Joint, or as absent (=`None`)

`Time` A factor specifying the occurrence of Time argument (as a moment) as either of its subclasses definite (Definite), indefinite (Indefinite), or as absent (=`None`)

`Modality1` A factor specifying the main semantic subclasses of the entire Verb chain as either indicating Possibility, Necessity, or their absense (=`None`)

`Modality2` A factor specifying minor semantic subclasses of the entire Verb chain as indicating either a Temporal element (begin, end, continuation, etc.), External (cause), Volition, Accidental nature of the thinking process, or their absense (=`None`)

`Source` A factor specifying the occurrence of a Source argument or its absense (=`None`)

`Goal` A factor specifying the occurrence of a Goal argument or its absence (=`None`)

`Quantity` A factor specifying the occurrence of a Quantity argument, or its absence (=`None`)

`Location` A factor specifying the occurrence of a Location argument, or its absence (=`None`)

`Duration` A factor specifying the occurrence of a Duration argument, or its absence (=`None`)

`Frequency` A factor specifying the occurrence of a Frequency arument, or its absence (=`None`)

`MetaComment` A factor specifying the occurrence of a MetaComment, or its absence (=`None`)

`ReasonPurpose` A factor specifying the occurrence of a Reason or Purpose argument, or their absence (=`None`)

`Condition` A factor specifying the occurrence of a Condition argument, or its absence (=`None`)

`CoordinatedVerb` A factor specifying the occurrence of a Coordinated Verb (in relation to the 'think' verb), or its absence (=`None`)

`Register` A factor specifying whether the 'think' verb occurs in the newspaper subcorpus (=`hs95`) or the Internet newsgroup discussion corpus (=`sfnet`)

`Section` A factor specifying the subsection in which the 'think' verb occurs in either of the two subcorpora

`Author` A factor specifying the author of the text in which the 'think' verb occurs, if that author is identifiable - authors in the Internet newgroup discussion subcorpus are anonymized; unidentifiable/unknown author designated as (=`None`)

Details

The four most frequent synonyms meaning 'think, reflect, ponder, consider', i.e. 'ajatella, miettia, pohtia, harkita', were extracted from two months of newspaper text from the 1990s (Helsingin Sanomat 1995) and six months of Internet newsgroup discussion from the early 2000s (SFNET 2002-2003), namely regarding (personal) relationships (sfnet.keskustelu.ihmissuhteet) and politics (sfnet.keskustelu.politiikka). The newspaper corpus consisted of 3,304,512 words of body text (i.e. excluding headers and captions as well as punctuation tokens), and included 1,750 examples of the studied 'think' verbs. The Internet corpus comprised 1,174,693 words of body text, yielding 1,654 instances of the selected 'think' verbs. In terms of distinct identifiable authors, the newspaper sub-corpus was the product of just over 500 journalists and other contributors, while the Internet sub-corpus involved well over 1000 discussants. The `think` dataset contains a selection of 26 contextual features judged as most informative.

For extensive details of the data and its linguistic and statistical analysis, see Arppe (2008). For the full selection of contextual features, see the `amph` (2008) microcorpus.

Source

amph (2008) A micro-corpus of 3404 occurrences of the four most common Finnish THINK lexemes, 'ajatella, miettia, pohtia, and harkita', in Finnish newspaper and Internet newsgroup discussion texts, containing extracts and linguistic analysis of the relevant context in the original corpus data, scripts for processing this data, R functions for its statistical analysis, as well as a comprehensive set of ensuing results as R data tables. Compiled and analyzed by Antti Arppe. Available on-line at URL: http://www.csc.fi/english/research/software/amph/

Helsingin Sanomat (1995) ~22 million words of Finnish newspaper articles published in Helsingin Sanomat during January–December 1995. Compiled by the Research Institute for the Languages of Finland [KOTUS] and CSC – IT Center for Science, Finland. Available on-line at URL: http://www.csc.fi/kielipankki/

SFNET (2002-2003) ~100 million words of Finnish internet newsgroup discussion posted during October 2002–April 2003. Compiled by Tuuli Tuominen and Panu Kalliokoski, Computing Centre, University of Helsinki, and Antti Arppe, Department of General Linguistics, University of Helsinki, and CSC - IT Center for Science, Finland. Available on-line at URL: http://www.csc.fi/kielipankki/

References

Arppe, A. (2008) Univariate, bivariate and multivariate methods in corpus-based lexicography – a study of synonymy. Publications of the Department of General Linguistics, University of Helsinki, No. 44. URN: http://urn.fi/URN:ISBN:978-952-10-5175-3.

Arppe, A. (2009) Linguistic choices vs. probabilities – how much and what can linguistic theory explain? In: Featherston, S. and S. Winkler (eds.) The Fruits of Empirical Linguistics. Volume 1: Process. Berlin: de Gruyter, pp. 1–24.

Examples

```
## For examples see examples(polytomous)
```

`wide2narrowcount` *Transformation of a count data table from* `"wide"` *to* `"narrow"` *count format*

Description

Transforms a count data table in the `"wide"` into the `"narrow"` format, so that a polytomous logistic regression model can be fit with `heuristic="poisson.reformulation"` using `glm` or `lmer` with `family=poisson` for count data.

Usage

```
wide2narrowcount(data.table, variables, outcomes, outcome = "OUTCOME",
   variables.default = NULL, outcome.ordered = NULL)
```

Arguments

`data.table`	A data table in the `"wide"` format which contains a column for each of the indicated outcomes giving their frequency of occurrence for a combination of predictor values indicated on the same row in the `data.table`.
`variables`	The predictor variables (columns in `data.table`) to be included in the `count.table`.
`outcomes`	The outcome variables (columns in `data.table`) with the frequencies of the outcomes for the associated predictor variable value combinations.
`outcome`	A character string designating a name for the outcome variable; by default `"OUTCOME"`.
`variables.default`	a list indicating for selected categorical predictors the value(s) that should be designated as the default/reference levels; by default `NULL`, in which case the original default/reference levels as specified for predictors in the object referred to by the `data.table` argument will be used.
`outcome.ordered`	a list specifying the order of the categories for the outcome/response variable; by default `NULL`, in which case the original order specified with `outcomes` argument will be used.

Details

Transforms a count data table in the `"wide"` format into the `"narrow"` format, so that a polytomous logistic regression model can be fit with `heuristic="poisson.reformulation"` using codeglm or `lmer` with `family=poisson`.

Value

A count data table with the frequency counts for each unique combination of outcomes and predictor variable values. In addition to columns with values for each included predictor, the count data table has the following common columns:

`"Proportion"` the relative proportion of the specific outcome in conjunction with the specific combination of selected predictor variables (in relation to the sum frequency of all the outcomes for the particular unique combination of predictor variables).

`"Count"` the frequency count of the specific outcome value in conjunction with the specific combination of selected predictor variables.

`outcome` the name of the response variable designated by the `outcome` argument; by default `"OUTCOME"`

`"Observation"` the index number for each unique combination of values of selected predictor variables.

Author(s)

Antti Arppe

References

Antti. A. (in prep.) Solutions for fixed and mixed effects modeling of polytomous outcome settings.

See Also

`polytomous.poisson.reformulation`, `instance2narrowcount`

Examples

```
data(think)
think.Agent_Patient.counts <- instance2narrowcount(think, c("Agent","Patient"),
   "Lexeme")
think.Agent_Patient.wide <- cbind(matrix(think.Agent_Patient.counts$Count,,4,
   byrow=TRUE, dimnames=list(NULL,c("ajatella","harkita","miettia","pohtia"))),
   unique(think.Agent_Patient.counts[c("Agent","Patient")]))
think.Agent_Patient.wide

think.Agent_Patient.counts2 <- wide2narrowcount(think.Agent_Patient.wide,
   variables=c("Agent","Patient"),
   outcomes=c("ajatella","harkita","miettia","pohtia"), outcome="Lexeme")
think.Agent_Patient.counts2
identical(think.Agent_Patient.counts,think.Agent_Patient.counts2)

think.polytomous1 <- polytomous(ajatella|harkita|miettia|pohtia ~ Agent + Patient,
   data=think.Agent_Patient.wide, heuristic="one.vs.rest")
summary(think.polytomous1)

think.polytomous2 <- polytomous(ajatella|harkita|miettia|pohtia ~ Agent + Patient,
   data=think.Agent_Patient.wide, heuristic="poisson.reformulation",
   outcome="Lexeme")
summary(think.polytomous2)
```

Index