Package: mixcat (via r-universe)

October 6, 2026

Title Mixed Effects Cumulative Link and Logistic Regression Models

Version 1.0-4

Date 2019-12-20

Author Georgios Papageorgiou [aut, cre], John Hinde [aut]

Maintainer Georgios Papageorgiou <gpapageo@gmail.com>

Depends R (>= 2.8.1), statmod

LazyLoad yes

Description Mixed effects cumulative and baseline logit link models for the analysis of ordinal or nominal responses, with non-parametric distribution for the random effects.

License GPL (>= 2)

NeedsCompilation yes

Repository https://cran.r-universe.dev

Date/Publication 2019-12-20 11:20:02 UTC

RemoteUrl https://github.com/cran/mixcat

RemoteRef HEAD

RemoteSha 20275d8b7fbe789f9006c93580d0910d7be83c84

Contents

mixcat-package 2
npmlt 3
schizo 7
summary.npmreg 7

Index **9**

mixcat-package *Mixed effects cumulative link and logistic regression models*

Description

Mixed effects models for the analysis of binary or multinomial (ordinal or nominal) data with non-parametric distribution for the random effects. The main function is `npmlt` and it fits cumulative and baseline logit models.

Details

Package:	mixcat
Type:	Package
Version:	1.0-4
Date:	2019-12-20
License:	GPL (>=2)
LazyLoad:	no

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

For details on the GNU General Public License see http://www.gnu.org/copyleft/gpl.html or write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

Acknowledgments

Papageorgiou's work was supported by the Science Foundation Ireland Research Frontiers grant 07/RFP/MATF448.

Author(s)

Georgios Papageorgiou and John Hinde (2011)

Maintainer: Georgios Papageorgiou <gpapageo@gmail.com>

References

Papageorgiou, G. and Hinde, J. (2012). Multivariate generalized linear mixed models with semi-nonparametric and smooth nonparametric random effects densities. Statistics and Computing 22, 79-92

npmlt *Mixed effects cumulative link and logistic regression models*

Description

Fits cumulative logit and baseline logit and link mixed effects regression models with non- parametric distribution for the random effects.

Usage

```
npmlt(formula, formula.npo=~1, random=~1, id, k=1, eps=0.0001,
      start.int=NULL, start.reg=NULL, start.mp=NULL,
      start.m=NULL, link="clogit",
      EB=FALSE, maxit=500, na.rm=TRUE, tol=0.0001)
```

Arguments

formula	a formula defining the response and the fixed, proportional odds, effects part of the model, e.g. y ~ x.
formula.npo	a formula defining non proportional odds variables of the model. A response is not needed as it has been provided in formula. Intercepts need not be provided as they are always non proportional. Variables in formula.npo must be a subset of the variables that appear in the right hand side of formula, e.g. ~ x.
random	a formula defining the random part of the model. For instance, random = ~1 defines a random intercept model, while random = ~1+x defines a model with random intercept and random slope for the variable x. If argument k=1, the resulting model is a fixed effects model (see below). Variables in random must be a subset of the variables that appear in the right hand side of formula.
id	a factor that defines the primary sampling units, e.g. groups, clusters, classes, or individuals in longitudinal studies. These sampling units have their own random coefficient, as defined in random. If argument id is missing it is taken to be id=seq(N), where N is the total number of observations, suitable for overdispersed independent multinomial data.
k	the number of mass points and masses for the non-parametric (discrete) random effects distribution. If k=1 the function fits a fixed effects models, regerdless of the random specification, as with k=1 the random effects distribution is degenerate at zero.
eps	positive convergence tolerance $epsilon$. Convergence is declared when the maximum of the absolute value of the score vector is less than $epsilon$.
start.int	a vector of length (number of categories minus one) with the starting values the fixed intercept(s).
start.reg	a vector with the starting values for the regression coefficients. One starting value for the proportional odds effects and (number of categories minus one) starting values for the non proportional effects, in the same order as they appear in formula.

`start.mp` starting values for the mass points of the random effects distribution in the form: (k starting values for the intercepts, k starting values for the first random slope,...).

`start.m` starting values for the masses of the random effects distribution: a vector of length k with non-negative elements that sum to 1.

`link` for a cumulative logit model set `link="clogit"` (default). For a baseline logit model, set `link="blogit"`. Baseline category is the last category.

`EB` if `EB=TRUE` the empirical Bayes estimates of the random effects are calculated and stored in the component `eBayes`. Further, fitted values of the linear predictor (stored in the component `fitted`) and fitted probabilities (stored in object `prob`) are obtained at the empirical Bayes estimates of the random effects. Otherwise, if `EB=FALSE` (default), empirical Bayes estimates are not calculated and fitted values of the linear predictors and probabilities are calculated at the zero value of the random effects.

`maxit` integer giving the maximal number of iterations of the fitting algorithm until convergence. By default this number is set to 500.

`na.rm` a logical value indicating whether NA values should be stripped before the computation proceeds.

`tol` positive tolerance level used for calculating generalised inverses (g-inverses). Consider matrix $A = PDP^T$, where $D = Diag\{eigen_i\}$ is diagonal with entries the eigen values of A. Its g-inverse is calculated as $A^- = PD^-P^T$, where D^- is diagonal with entries $1/eigen_i$ if $eigen_i > tol$, and 0 otherwise.

Details

Maximizing a likelihood over an unspecified random effects distribution results in a discrete mass point estimate of this distribution (Laird, 1978; Lindsay, 1983). Thus, the terms 'non-parametric' (NP) and 'discrete' random effects distribution are used here interchangeably. Function `npmlt` allows the user to choose the number k of mass points/masses of the discrete distribution, a choice that should be based on the log-likelihood. Note that the mean of the NP distribution is constrained to be zero and thus for `k=1` the fitted model is equivalent to a fixed effects model. For `k>1` and a random slope in the model, the mass points are bivariate with a component that corresponds to the intercept and another that corresponds to the slope.

General treatments of non-parametric modeling can be found in Aitkin, M. (1999) and Aitkin et al. (2009). For more details on multinomial data see Hartzel et al (2001).

The response variable `y` can be binary or multinomial. A binary response should take values 1 and 2, and the function `npmlt` will model the probability of 1. For an ordinal response, taking values $1, \ldots, q$, a cumulative logit model can be fit. Ignoring the random effects, such a model, with formula `y~x`, takes the form

$$log\frac{P(Y \leq r)}{1 - P(Y \leq r)} = \beta_r + \gamma x,$$

where $\beta_r, r = 1, \ldots, q-1$, are the cut-points and γ is the slope. Further, if argument formula.npo is specified as `~x`, the model becomes

$$log\frac{P(Y \leq r)}{1 - P(Y \leq r)} = \beta_r + \gamma_r x,$$

Similarly, for a nominal response with q categories, a baseline logit model can be fit. The fixed effects part of the model, y~x, takes the form,

$$log\frac{P(Y=r)}{P(Y=q)} = \beta_r + \gamma x,$$

where $r = 1, \ldots, q-1$. Again, formula.npo can be specified as ~x, in which case slope γ will be replaced by category specific slopes, γ_r.

The user is provided with the option of specifying starting values for some or all the model parameters. This option allows for starting the algorithm at different starting points, in order to ensure that it has convered to the point of maximum likelihood. Further, if the fitting algorithm fails, the user can start by fitting a less complex model and use the estimates of this model as starting values for the more complex one.

With reference to the tol argument, the fitting algorithm calculates g-inverses of two matrices: 1. the information matrix of the model, and 2. the covariance matrix of multinomial proportions. The covariance matrix of a multinomial proportion p of length q is calculated as $Diag\{p*\} - p * p*^T$, where $p*$ is of length $q-1$. A g-inverse for this matrix is needed because elements of $p*$ can become zero or one.

Value

The function npmlt returns an object of class 'npmreg', a list containing at least the following components:

call	the matched call.
formula	the formula supplied.
formula.npo	the formula for the non proportional odds supplied.
random	the random effects formula supplied.
coefficients	a named vector of regression coefficients.
mass.points	a vector or a table that contains the mass point estimates.
masses	the masses (probabilities) corresponding to the mass points.
vcvremat	the estimated variance-covariance matrix of the random effects.
var.cor.mat	the estimated variance-covariance matrix of the random effects, with the upper triangular covariances replaced by the corresponding correlations.
m2LogL	minus twice the maximized log-likelihood of the chosen model.
SE.coefficients	a named vector of standard errors of the estimated regression coefficients.
SE.mass.points	a vector or a table that contains the the standard errors of the estimated mass points.
SE.masses	the standard errors of the estimated masses.
VRESE	the standard errors of the estimates of the variances of random effects.
CVmat	the inverse of the observed information matrix of the model.
eBayes	if EB=TRUE it contains the empirical Bayes estimates of the random effects. Otherwise it contains vector(s) of zeros.

fitted	the fitted values of the linear predictors computed at the empirical Bayes estimates of the random effects, if EB=TRUE. Otherwise, if EB=FALSE (default) these fitted values are computed at the zero value of the random effects.
prob	the estimated probabilities of observing a response at one of the categories. These probabilities are computed at the empirical Bayes estimates of the random effects, if EB=TRUE. If EB=FALSE (default) these estimated probabilities are computed at the zero value of the random effects.
nrp	number of random slopes specified.
iter	the number of iterations of the fitting algorithm.
maxit	the maximal allowed number of iterations of the fitting algorithm until convergence.
flagcvm	last iteration at which eigenvalue(s) of covariance matrix of multinomial variable were less than tol argument.
flaginfo	last iteration at which eigenvalue(s) of model information matrix were less than tol argument.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

References

Aitkin, M. (1999). A general maximum likelihood analysis of variance components in generalized linear models. Biometrics 55, 117-128.

Aitkin, M., Francis, B., Hinde, J., and Darnell, R. (2009). Statistical Modelling in R. Oxford Statistical Science Series, Oxford, UK.

Hedeker, D. and Gibbons, R. (2006). Longitudinal Data Analysis. Wiley, Palo Alto, CA.

Hartzel, J., Agresti, A., and Caffo, B. (2001). Multinomial logit random effects models. Statistical Modelling, 1(2), 81-102.

Laird, N. (1978). Nonparametric maximum likelihood estimation of a mixing distribution. Journal of the American Statistical Association, 73, 805-811.

Lindsay, B. G. (1983). The geometry of mixture likelihoods, Part II: The exponential family. The Annals of Statistics, 11, 783-792.

See Also

summary.npmreg

Examples

```
data(schizo)
attach(schizo)

npmlt(y~trt*sqrt(wk),formula.npo=~trt,random=~1+trt,id=id,k=2,EB=FALSE)
```

schizo *National Institute of Mental Health shizophrenia study*

Description

Schizophrenia data from a randomized controlled trial with patients assigned to either drug or placebo group. "Severity of Illness" was measured, at weeks 0,1,...6, on a four category ordered scale: 1. normal or borderline mentally ill, 2. mildly or moderately ill, 3. markedly ill, and 4. severely or among the most extremely ill. Most of the observations where made on weeks 0,1,3, and 6.

Usage

```
data(schizo)
```

Format

A data frame with 1603 observations on 437 subjects. Four numerical vectors contain information on

`id` patient ID.

`y` ordinal response on a 4 category scale.

`trt` treatment indicator: 1 for drug, 0 for placebo.

`wk` week.

Source

http://tigger.uic.edu/~hedeker/ml.html

References

Hedeker, D. and Gibbons, R. (2006). Longitudinal Data Analysis. Wiley, Palo Alto, CA.

summary.npmreg *Summarizing mixed multinomial regression model fits*

Description

`summary` and `print` methods for objects of type `npmreg`.

Usage

```
## S3 method for class 'npmreg'
summary(object,digits = max(3, getOption("digits") - 3),...)
## S3 method for class 'npmreg'
print(x,digits = max(3, getOption("digits") - 3),...)
```

Arguments

object	an object of class npmreg.
x	an object of class npmreg.
digits	the minimum number of significant digits to be printed in values.
...	further arguments, which will mostly be ignored.

Details

The function npmlt returns an object of class "npmreg". The function summary (i.e., summary.npmreg) can be used to obtain or print a summary of the results, and the function print (i.e., print.npmreg) to print the results.

Value

Summary or print.

Author(s)

Georgios Papageorgiou <gpapageo@gmail.com>

See Also

npmlt

Examples

```
data(schizo)
attach(schizo)
fit1<-npmlt(y~trt*sqrt(wk),formula.npo=~trt,random=~1,id=id,k=2)
print(fit1)
summary(fit1)
```

Index

∗ **datasets**
schizo, 7
∗ **models**
mixcat-package, 2
npmlt, 3
summary.npmreg, 7
∗ **regression**
mixcat-package, 2
npmlt, 3
summary.npmreg, 7

mixcat *(*mixcat-package*)*, 2
mixcat-package, 2

npmlt, *2*, 3, *8*

print.npmreg *(*summary.npmreg*)*, 7

schizo, 7
summary.npmreg, *6*, 7