Package ‘mclogit’

July 17, 2017

Type     Package
Title    Mixed Conditional Logit Models
Version  0.5.1
Date     2017-07-16
Author   Martin Elff
Maintainer Martin Elff <mclogit@elff.eu>
Description Specification and estimation of conditional logit models of binary responses and multinomial counts is provided, with or without alternative-specific random effects. The current implementation of the estimator for random effects variances uses a Laplace approximation (or PQL) approach and thus should be used only if groups sizes are large.
License  GPL-2
Depends  stats, Matrix
Imports  memisc, methods
LazyLoad Yes
BugReports http://github.com/melff/mclogit/issues
RoxygenNote 5.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2017-07-17 09:46:59 UTC

R topics documented:

electors ................................................. 2
getSummary-methods .................................. 3
mblogit ................................................. 4
mclogit ............................................... 6
mclogit.control ..................................... 7
mclogit.fit ......................................... 8
Transport .......................................... 9
Description

This is an artificial data set on electoral choice as influenced by class and party positions.

Usage

data(electors)

Format

A data frame containing the following variables:

- **class**: class position of voters
- **party**: party that runs for election
- **Freq**: frequency by which each party list is chosen by members of each class
- **time**: time variable, runs from zero to one
- **econ.left**: economic-policy "leftness" of each party
- **welfare**: emphasis of welfare expansion of each party
- **auth**: position on authoritarian issues

Examples

```r
data(electors)

summary(mclogit(
  cbind(Freq,interaction(time,class))~econ.left+welfare+auth,
  data=electors))

summary(mclogit(
  cbind(Freq,interaction(time,class))~econ.left/class+welfare/class+auth/class,
  data=electors))

summary(mclogit(
  cbind(Freq,interaction(time,class))~econ.left/class+welfare/class+auth/class,
  random=-1|party.time,
  data=within(electors,party.time<-interaction(party,time))))

summary(mclogit(
  cbind(Freq,interaction(time,class))~econ.left/(class*time)+welfare/class+auth/class,
  random=-1|party.time,
  data=within(electors,{
    party.time <- interaction(party,time)
    econ.left.sq <- (econ.left-mean(econ.left))^2
  }))
)
Description

`getSummary` methods for use by `mtable`.

Usage

```r
## S3 method for class 'mblogit'
getSummary(obj, alpha=.05, ...
## S3 method for class 'mclogit'
getSummary(obj, alpha=.05, 
  rearrange=NULL, ...
```

Arguments

- `obj` an object returned by `mblogit` or `mclogit`
- `alpha` level of the confidence intervals; their coverage should be 1-alpha/2
- `rearrange` an optional named list of character vectors. Each element of the list designates a column in the table of estimates, and each element of a character vector refers to a coefficient. Names of list elements become column heads and names of the character vector elements become coefficient labels.
- `...` further arguments; ignored.

Examples

```r
## Not run
summary(classd.model <- mclogit(cbind(Freq,choice.set)~
  (econdim.sq+nonmatdim1.sq+nonmatdim2.sq)+
  (econdim1+nonmatdim1+nonmatdim2)+
  (econdim1+nonmatdim1+nonmatdim2):classd,
  data=mvote.int.classd.random=1|mvote.int/eb,
  subset=classd!="Farmers"))
myGetSummary.classd <- function(x) getSummary.mclogit(x, 
  rearrange=list("Econ. Left/Right"=c(
    "Squared effect"="econdim1.sq",
    "Linear effect"="econdim1",
    "x Intermediate/Manual worker"="econdim1:classdIntermediate",
    "x Service class/Manual worker"="econdim1:classdService class",
    "x Self-employed/Manual worker"="econdim1:classdSelf-employed",
  ),
  "Lib./Auth."=c(
    "Squared effect"="nonmatdim1.sq",
```
The function `mblogit` fits multinomial logit models for categorical and multinomial count responses with fixed alternatives, where the logits are relative to a baseline category.
### mblogit

#### Usage

```r
mblogit(formula, data = parent.frame(), random = NULL, subset,
weights = NULL, na.action = getOption("na.action"), model = TRUE,
x = FALSE, y = TRUE, contrasts = NULL, control = mclogit.control(...),
...)
```

#### Arguments

- **formula**: the model formula. The response must be a factor or a matrix of counts.
- **data**: an optional data frame, list or environment (or object coercible by `as.data.frame` to a data frame) containing the variables in the model. If not found in `data`, the variables are taken from `environment(formula)`, typically the environment from which `glm` is called.
- **random**: an optional formula that specifies the random-effects structure or `NULL`.
- **subset**: an optional vector specifying a subset of observations to be used in the fitting process.
- **weights**: an optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector.
- **na.action**: a function which indicates what should happen when the data contain NAs. The default is set by the `na.action` setting of `options`, and is `na.fail` if that is unset. The 'factory-fresh' default is `na.omit`. Another possible value is `NULL`, no action. Value `na.exclude` can be useful.
- **model**: a logical value indicating whether `model frame` should be included as a component of the returned value.
- **x, y**: logical values indicating whether the response vector and model matrix used in the fitting process should be returned as components of the returned value.
- **contrasts**: an optional list. See the `contrasts.arg` of `model.matrix.default`.
- **control**: a list of parameters for the fitting process. See `mclogit.control`.
- **...**: arguments to be passed to `mclogit.control`.

#### Details

The function `mblogit` internally rearranges the data into a 'long' format and uses `mclogit.fit` to compute estimates. Nevertheless, the 'user data' is unaffected.

#### Value

`mblogit` returns an object of class "mblogit", which has almost the same structure as an object of class "glm". The difference are the components `coefficients`, `residuals`, `fitted.values`, `linear.predictors`, and `y`, which are matrices with number of columns equal to the number of response categories minus one.
Conditional Logit Models and Mixed Conditional Logit Models

Description

`mclogit` fits conditional logit models and mixed conditional logit models to count data and individual choice data, where the choice set may vary across choice occasions.

Conditional logit models without random effects are fitted by Fisher-scoring/IWLS. The implementation of mixed conditional logit currently is limited to PQL and random intercepts.

Usage

```r
mclogit(formula, data=parent.frame(), random=NULL,
        subset, weights = NULL, offset=NULL, na.action = getOption("na.action"),
        model = TRUE, x = FALSE, y = TRUE, contrasts=NULL,
        start=NULL,
        control=mclogit.control(...), ...)
```

Arguments

- **formula**: a model formula: a symbolic description of the model to be fitted. The left-hand side contains a two-column matrix. The first column contains the choice counts or choice indicators (alternative is chosen=1, is not chosen=0). The second column contains unique numbers for each choice set.
- **data**: an optional data frame, list or environment (or object coercible by `as.data.frame`) containing the variables in the model. If not found in data, the variables are taken from `environment(formula)`, typically the environment from which `glm` is called.
- **random**: an optional formula that specifies the random-effects structure or NULL.
- **weights**: an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector.
- **offset**: an optional model offset. Currently only supported for models without random effects.
- **subset**: an optional vector specifying a subset of observations to be used in the fitting process.
- **na.action**: a function which indicates what should happen when the data contain NAs. The default is set by the `na.action` setting of `options`, and is `na.fail` if that is unset. The ‘factory-fresh’ default is `na.omit`. Another possible value is `NULL`, no action. Value `na.exclude` can be useful.
**mclogit.control**

- **start**
  - an optional numerical vector of starting values for the conditional logit parameters.
- **model**
  - a logical value indicating whether model frame should be included as a component of the returned value.
- **x, y**
  - logical values indicating whether the response vector and model matrix used in the fitting process should be returned as components of the returned value.
- **contrasts**
  - an optional list. See the contrasts.arg of model.matrix.default.
- **control**
  - a list of parameters for the fitting process. See `mclogit.control`
- ...
  - arguments to be passed to `mclogit.control`

**Value**

`mclogit` returns an object of class "mclogit", which has almost the same structure as an object of class "glm".

**Examples**

```r
data(transport)

summary(mclogit(
  cbind(resp,suburb)~distance+cost, 
  data=transport 
))

data(electors)

summary(mclogit(
  cbind(Freq,interaction(time,class))~econ.left/class+welfare/class+auth/class, 
  random=-1|party.time, 
  data=within(electors,party.time<-interaction(party,time)))
```

---

**mclogit.control**

*Control Parameters for the Fitting Process*

**Description**

`mclogit.control` returns a list of default parameters that control the fitting process of `mclogit`.

**Usage**

```r
mclogit.control(epsilon = 1e-08, 
                maxit = 25, trace=TRUE)
```

**Arguments**

- **epsilon**
  - positive convergence tolerance $\epsilon$; the iterations converge when $|\text{dev} - \text{dev}_{old}| / (|\text{dev}| + 0.1) < \epsilon$.
- **maxit**
  - integer giving the maximal number of IWLS or PQL iterations.
- **trace**
  - logical indicating if output should be produced for each iteration.
Value

A list.

\[\text{mclogit.fit} \quad \text{Internal functions used for model fit.}\]

Description

These functions are exported and documented for use by other packages. They are not intended for end users.

Usage

\[
\text{mclogit.fit}(y, s, w, X, \text{start} = \text{NULL}, \text{offset} = \text{NULL}, \text{control} = \text{mclogit.control}())
\]

\[
\text{mmcmlogit.fitPQL}(y, s, w, X, Z, G, \text{groups}, \text{start}, \text{offset} = \text{NULL}, \text{control} = \text{mclogit.control}())
\]

Arguments

- \(y\) a response vector. Should be binary.
- \(s\) a vector identifying individuals or covariate strata
- \(w\) a vector with observation weights.
- \(X\) a model matrix; required.
- \(Z\) the random effects design matrix.
- \(G\) a list of design matrices for the (co-)variance parameters.
- \(\text{groups}\) a list of grouping factors.
- \(\text{start}\) an optional numerical vector of starting values for the coefficients.
- \(\text{offset}\) an optional model offset. Currently only supported for models without random effects.
- \(\text{control}\) a list of parameters for the fitting process. See \text{mclogit.control}.

Value

A list with components describing the fitted model.
Description

This is an artificial data set on choice of means of transport based on cost and walking distance.

Usage

data(Transport)

Format

A data frame containing the following variables:

- **transport** means of transportation that can be chosen.
- **suburb** identifying number for each suburb
- **distance** walking distance to bus or train station
- **cost** cost of each means of transportation
- **working** size of working population of each suburb
- **prop.true** true choice probabilities
- **resp** choice frequencies of means of transportation
Index

*Topic datasets
  electors, 2
  Transport, 9
*Topic models
  mclogit, 6
*Topic regression
  mclogit, 6

AIC.mclogit (mclogit), 6
anova.mclogit (mclogit), 6
as.data.frame, 5, 6

BIC.mclogit (mclogit), 6
deviance.mclogit (mclogit), 6

electors, 2
fitted.mblogit (mblogit), 4
fitted.mclogit (mclogit), 6

getSummary, 3
getSummary-methods, 3
getSummary.mblogit
  (getSummary-methods), 3
getSummary.mclogit
  (getSummary-methods), 3
glm, 5, 7

logLik.mclogit (mclogit), 6

mblogit, 3, 4
mclogit, 3, 6
mclogit.control, 5, 7, 7, 8
mclogit.fit, 5, 8
mmclogit.fitPQL (mclogit.fit), 8
mtable, 3

na.exclude, 5, 6
na.fail, 5, 6
na.omit, 5, 6

options, 5, 6
predict.mblogit (mblogit), 4
predict.mclogit (mclogit), 6
print.mblogit (mblogit), 4
print.mclogit (mclogit), 6
print.mmclogit (mblogit), 4
print.summary.mblogit (mblogit), 4
print.summary.mclogit (mclogit), 6
print.summary.mmclogit (mblogit), 4

residuals.mclogit (mclogit), 6

summary.mblogit (mblogit), 4
summary.mclogit (mclogit), 6
summary.mmclogit (mblogit), 4

Transport, 9

vcov.mclogit (mclogit), 6