Package ‘mclogit’

February 20, 2015

Type Package
Title Mixed Conditional Logit
Version 0.3-1
Date 2014-10-13
Author Martin Elff
Maintainer Martin Elff <elff@gmx.com>
Description This package provides a function to estimate parameters for
the conditional logit model (also with multinomial counts), and for the
mixed conditional logit model, or conditional logit with random effects
(random intercepts only, no random slopes yet).
The current implementation of random effects is limited to
the PQL technique, which requires large cluster sizes.
License GPL-2
Depends stats, Matrix
Enhances memisc
LazyLoad Yes
NeedsCompilation no
Repository CRAN
Date/Publication 2014-10-14 05:27:20

R topics documented:

electors .............................................................. 2
getSummary.mclogit ........................................... 3
mclogit ............................................................. 4
mclogit.control ................................................ 6
Transport ...................................................... 7

Index 8
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
  )))
```
getSummary.mclogit

Description

A `getSummary` method for use by `mtable`

Usage

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

Arguments

- `obj`: an object returned by `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(classsd.model <- mclogit(cbind(Freq, choice.set)~
  (econdim1.sq+nonmatdim1.sq+nonmatdim2.sq)+
  (econdim1+nonmatdim1+nonmatdim2)*
  (econdim1+nonmatdim1+nonmatdim2):classsd,  
  data=mvoteint.classsd, random=~1|mvoteint/eb,  
  subset=classsd!="Farmers"))
mygetsummary.classsd <- function(x) getSummary.mclogit(x, rearrange=list(  
  "Econ. Left/Right"=c(  
    "Squared effect"="econdim1.sq",  
    "Linear effect"="econdim1",  
    " x Intermediate/Manual worker"="econdim1:classsdIntermediate",  
    " x Service class/Manual worker"="econdim1:classsdService class",  
    " x Self-employed/Manual worker"="econdim1:classsdSelf-employed"  
  ),  
  "Lib./Auth."=c(  
    "Squared effect"="nonmatdim1.sq",  
    "Linear effect"="nonmatdim1",  
    " x Intermediate/Manual worker"="nonmatdim1:classsdIntermediate",  
    " x Service class/Manual worker"="nonmatdim1:classsdService class",  
    " x Self-employed/Manual worker"="nonmatdim1:classsdSelf-employed"  
  ))
```
mclogit

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.
mclogit

Usage

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

Arguments

formula a model formula: a symbolic description of the model to be fitted. The left-hand 
          side contains is expected to be 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. 
          If individual-level data is used, choice sets correspond to the individuals, if ag- 
          gregated data with choice counts are used, choice sets may e.g. correspond to 
          covariate classes within clusters. 

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.

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.

start.theta an optional numerical vector of starting values for the variance parameters.

model a logical value indicating whether model frame should be included as a compo- 
       nent 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
mclogit.control

Details

mclogit tries first to fit the model using the IRLS algorithm of \texttt{glm.fit}, which has the advantage that starting values are not needed in most cases. If convergence cannot achieved, it tries to minimize the deviance using \texttt{optim} with method "BFGS".

Value

mclogit returns an object of class "mlogit", which has almost the same structure as an object of class "\texttt{glm}". The difference are the components \texttt{coefficients}, \texttt{residuals}, \texttt{fitted.values}, \texttt{linear.predictors}, and \texttt{y}, which are matrices with number of columns equal to the number of response categories minus one.

Examples

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

\textit{Control Parameters for the Fitting Process}

Description

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

Usage

\texttt{mclogit.control(epsilon = 1e-08,}
\texttt{ maxit = 25, trace=TRUE)}

Arguments

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

Value

A list.

Choice of Means of Transport

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, 7
*Topic models
  mclogit, 4
*Topic regression
  mclogit, 4

as.data.frame, 5

electors, 2

getSummary, 3
getSummary.mclogit, 3
glm, 6
glm.fit, 6

mclogit, 3, 4
mclogit.control, 5, 6
mtable, 3

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

optim, 6
options, 5

Transport, 7