

Package ‘mclgit’

January 2, 2012

Type Package

Title Mixed Conditional Logit

Version 0.2-3

Date 2011-12-07

Author Martin Elff

Maintainer Martin Elff <melff@essex.ac.uk>

Description This packages provides a function to estimate parameters for the mixed conditional logit model, or conditional logit with random effects. The current implementation is limited to random intercepts and to the PQL technique, mainly appropriate for large clusters.

License GPL-2

Depends stats, MASS, memisc, Matrix

LazyLoad Yes

Repository CRAN

Date/Publication 2011-12-09 09:21:18

R topics documented:

electors	2
mclgit	3
mclgit.control	4
Transport	5

Index	6
--------------	----------

 electors

Class, Party Position, and Electoral Choice

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

```
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
  })))
```

mclgit

*Conditional Logit Models and Mixed Conditional Logit Models***Description**

mclgit 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 slopes.

Usage

```
mclgit(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=mclgit.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 aggregated data with choice counts are used, choice sets may e.g. correspond to covariate classes within clusters. The right-hand of the formula contains choice predictors. It should be noted that constants are deleted from the formula as are predictors that do not vary within choice sets.
data	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>glm</code> is called.
random	an optional formula that specifies the random-effects structure or <code>NULL</code> .
weights	an optional vector of weights to be used in the fitting process. Should be <code>NULL</code> 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 <code>na.action</code> setting of <code>options</code> , and is <code>na.fail</code> if that is unset. The ‘factory-fresh’ default is <code>na.omit</code> . Another possible value is <code>NULL</code> , no action. Value <code>na.exclude</code> can be useful.

start.theta	an optional numerical vector of starting values for the variance parameters.
model	a logical value indicating whether <i>model frame</i> 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 <code>contrasts.arg</code> of <code>model.matrix.default</code> .
control	a list of parameters for the fitting process. See mclgit.control
...	arguments to be passed to <code>mclgit.control</code>

Details

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

Value

`mlogit` returns an object of class "mlogit", 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.

Examples

```
data(Transport)

summary(mclgit(
  cbind(resp,suburb)~distance+cost,
  data=Transport
))

data(electors)

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

mclgit.control

Control Parameters for the Fitting Process

Description

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

Usage

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

Arguments

epsilon positive convergence tolerance ϵ ; the iterations converge when $|dev - dev_{old}| / (|dev| + 0.1) < \epsilon$.

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.

Transport	<i>Choice of Means of Transport</i>
-----------	-------------------------------------

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, [5](#)

*Topic **models**

mclgit, [3](#)

*Topic **regression**

mclgit, [3](#)

as.data.frame, [3](#)

electors, [2](#)

glm, [4](#)

glm.fit, [4](#)

mclgit, [3](#)

mclgit.control, [4](#), [4](#)

na.exclude, [3](#)

na.fail, [3](#)

na.omit, [3](#)

optim, [4](#)

options, [3](#)

Transport, [5](#)