

The mlogit Package

August 27, 2008

Version 0.1-1

Date 2008-08-27

Title multinomial logit model

Author Yves Croissant <yves.croissant@let.ish-lyon.cnrs.fr>

Maintainer Yves Croissant <yves.croissant@let.ish-lyon.cnrs.fr>

Depends R (>= 1.8.0), maxLik, sandwich

Suggests Ecdat, lmtest, car

Description Estimation of the multinomial logit model with alternative and/or individual specific variables

License GPL (>=2)

URL <http://www.r-project.org>

R topics documented:

hmfctest	1
logitform	3
mlogit	5
mlogit.data	8

Index	11
--------------	-----------

`hmfctest`*Hausman-McFadden Test*

Description

Test the IIA hypothesis (independence of irrelevant alternatives) for a multinomial logit model.

Usage

```
hmfctest(x, ...)  
## S3 method for class 'mlogit':  
hmfctest(x, z, ...)  
## S3 method for class 'formula':  
hmfctest(x, alt.subset, ...)
```

Arguments

<code>x</code>	an object of class <code>mlogit</code> or a formula,
<code>z</code>	an object of class <code>mlogit</code> or a subset of alternatives for the <code>mlogit</code> method. This should be the same model as <code>x</code> estimated on a subset of alternatives,
<code>alt.subset</code>	a subset of alternatives,
<code>...</code>	further arguments passed to <code>mlogit</code> for the <code>formula</code> method.

Details

This is an implementation of the Hausman's consistency test for multinomial logit models. If the independence of irrelevant alternatives applies, the probability ratio of every two alternatives depends only on the characteristics of these alternatives. Consequently, the results obtained on the estimation with all the alternatives or only on a subset of them are consistent, but more efficient in the first case. On the contrary, only the results obtained from the estimation on a relevant subset are consistent. To compute this test, one needs a model estimated with all the alternatives and one model estimated on a subset of alternatives. This can be done by providing two objects of class `mlogit`, one object of class `mlogit` and a character vector indicating the subset of alternatives, or a formula and a subset of alternatives.

Value

an object of class `"htest"`.

Author(s)

Yves Croissant

References

Hausman, J.A. and D. McFadden (1984), A Specification Test for the Multinomial Logit Model, *Econometrica*, **52**, pp.1219–1240.

Examples

```
## from Greene's Econometric Analysis p. 731

data("ModeChoice", package="Ecdat")
ModeChoice <- mlogit.data(ModeChoice, choice="mode", shape="long",
                          alt.levels=c("air", "train", "bus", "car"))

## Create a variable of income only for the air mode

ModeChoice$avinc <- (ModeChoice$alt=='air')*ModeChoice$hinc

## Estimate the model on all alternatives, with car as the base level
## like in Greene's book.

x <- mlogit(mode~ttme+gc+avinc, ModeChoice, reflevel="car")

## Estimate the same model for ground modes only (the variable avinc
## must be dropped because it is 0 for every observation

g <- mlogit(mode~ttme+gc, ModeChoice, reflevel="car",
            alt.subset=c("car", "bus", "train"))

## Compute the test

hmfptest(x, g)
```

logitform

Model formula for logit models

Description

Two kinds of variables are used in logit models: alternative specific and individual specific variables. `logitform` provides a relevant class to deal with this specificity and suitable methods to extract the elements of the model.

Usage

```
logitform(object)
## S3 method for class 'formula':
logitform(object)
## S3 method for class 'list':
logitform(object)
is.logitform(object)
## S3 method for class 'logitform':
model.matrix(object, data, ...)
## S3 method for class 'logitform':
model.frame(formula, data, ...)
```

```
## S3 method for class 'logitform':
terms(x, ...)
## S3 method for class 'logitform':
update(object,new, ...)
```

Arguments

<code>object, x</code>	for the <code>logitform</code> function, a formula or a list, for the <code>update</code> and <code>model.matrix</code> methods, a <code>logitform</code> object,
<code>new</code>	the new formula for the <code>update</code> method,
<code>formula</code>	a <code>logitform</code> object,
<code>data</code>	a <code>data.frame</code> ,
<code>...</code>	further arguments.

Details

a `logitform` is a formula for which the right hand side may contain two parts: the first one contains the alternative specific variables and the second one the individual specific variables. The two parts are separated by a “|” sign. If a standard formula is written, it is assumed that there are no individual specific variables. Specific methods are provided to build correctly the model matrix and to update the formula. The `logitform` function is not intended to be use directly. While using the `mlogit` function, the first argument is automatically coerced to a `logitform` object.

Value

an object of class `logitform`.

Author(s)

Yves Croissant

Examples

```
v <-logitform(y~x1+x2|z1+z2)
update(v, .~.+x3-x1|. +z3-z1)
update(v, .~-x1-x2|. )
update(v, .~.|.-z1-z2)
v <-logitform(y~x1+x2)
update(v, .~-x1+x3)
update(v, .~.|z1+z2)

data("Fishing", package="Ecdat")
colnames(Fishing)[4:11] <- c("pr.beach", "pr.pier", "pr.boat", "pr.charter",
                           "ca.beach", "ca.pier", "ca.boat", "ca.charter")
Fish <- mlogit.data(Fishing, varying=c(4:11), shape="wide", choice="mode")
f <- logitform(mode~pr+ca|income)
X <- model.matrix(f, data=Fish)
X[1:6,]
```

mlogit	<i>Multinomial logit model</i>
--------	--------------------------------

Description

Estimation by maximum likelihood of the multinomial logit model, with alternative-specific and/or individual specific variables.

Usage

```
mlogit(formula, data, subset, weights, na.action,
        alt.subset = NULL, reflevel = NULL, ...)
## S3 method for class 'mlogit':
print(x, digits = max(3, getOption("digits") - 2),
      width = getOption("width"), ...)
## S3 method for class 'mlogit':
summary(object, ...)
## S3 method for class 'summary.mlogit':
print(x, digits = max(3, getOption("digits") - 2),
      width = getOption("width"), ...)
## S3 method for class 'mlogit':
print(x, digits = max(3, getOption("digits") - 2),
      width = getOption("width"), ...)
## S3 method for class 'mlogit':
logLik(object, ...)
## S3 method for class 'mlogit':
vcov(object, ...)
## S3 method for class 'mlogit':
residuals(object, outcome = TRUE, ...)
## S3 method for class 'mlogit':
fitted(object, outcome = TRUE, ...)
## S3 method for class 'mlogit':
df.residual(object, ...)
## S3 method for class 'mlogit':
terms(x, ...)
## S3 method for class 'mlogit':
estfun(x, ...)
## S3 method for class 'mlogit':
bread(x, ...)
## S3 method for class 'mlogit':
model.matrix(object, ...)
## S3 method for class 'mlogit':
update(object, new, ...)
```

Arguments

`x`, `object` an object of class `mlogit`

<code>formula</code>	a symbolic description of the model to be estimated,
<code>new</code>	an updated formula for the <code>update</code> method,
<code>data</code>	the data,
<code>subset</code>	an optional vector specifying a subset of observations,
<code>weights</code>	an optional vector of weights,
<code>na.action</code>	a function which indicates what should happen when the data contains 'NA's,
<code>alt.subset</code>	a vector of character strings containing the subset of alternative on which the model should be estimated,
<code>reflevel</code>	the base alternative (the one for which the coefficients of individual-specific variables are normalized to 0),
<code>digits</code>	the number of digits,
<code>width</code>	the width of the printing,
<code>outcome</code>	a boolean which indicates, for the <code>fitted</code> and the <code>residuals</code> methods whether a matrix (for each choice, one value for each alternative) or a vector (for each choice, only a value for the alternative chosen) should be returned,
<code>...</code>	further arguments.

Details

Let J being the number of alternatives. The formula may include alternative-specific and individual specific variables. For the latter, $J-1$ coefficients are estimated for each variable. Alternative and individual specific variables are separated by a `|`. For example, if x_1 and x_2 are alternative specific and z_1 and z_2 are individual specific, the formula `y~x1+x2|z1+z2` describe a model with one coefficient for x_1 and x_2 and $J-1$ coefficients for z_1 and z_2 . $J-1$ intercepts are also estimated. A model without intercepts is defined by the formula `y~x1+x2-1|z1+z2`. To obtain alternative specific coefficients for the alternative-specific variable x_2 , use `y~x1+x2+x2:alt|z1+z2` (replace `alt` by the relevant variable name if the alternative index is provided). Models with only alternative-specific or individual-specific variables are respectively estimated by `y~x1+x2` and `y~1|z1+z2`.

The model is estimated with the `maxLik` package and the Newton-Raphson method, using analytic gradient and hessian.

Value

An object of class "mlogit", a list with elements:

<code>coefficients</code>	the named vector of coefficients,
<code>logLik</code>	the value of the log-likelihood,
<code>hessian</code>	the hessian of the log-likelihood at convergence,
<code>gradient</code>	the gradient of the log-likelihood at convergence,
<code>call</code>	the matched call,
<code>est.stat</code>	some information about the estimation (time used, optimisation method),
<code>freq</code>	the frequency of choice,

residuals the residuals,
 fitted.values
 the fitted values,
 formula the formula (a logitform object),
 expanded.formula
 the formula (a formula object),
 model the model frame used,
 index the index of the choice and of the alternatives.

Author(s)

Yves Croissant

References

McFadden, D. (1973) Conditional Logit Analysis of Qualitative Choice Behavior, in P. Zarembka ed., *Frontiers in Econometrics*, New-York: Academic Press.

McFadden, D. (1974) "The Measurement of Urban Travel Demand", *Journal of Public Economics*, 3, pp. 303-328.

Train, K. (2004) *Discrete Choice Modelling, with Simulations*, Cambridge University Press.

See Also

`mlogit.data` to shape the data. `multinom` from package `nnet` performs the estimation of the multinomial logit model with individual specific variables

Examples

```
## Cameron and Trivedi's Microeconometrics p.493
## There are two alternative specific variables : pr (price) and ca (catch)
## and four fishing mode : beach, pier, boat, charter

data("Fishing",package="Ecdat")
colnames(Fishing)[4:11] <- c("pr.beach","pr.pier","pr.boat","pr.charter",
                           "ca.beach","ca.pier","ca.boat","ca.charter")
Fish <- mlogit.data(Fishing,varying=c(4:11),shape="wide",choice="mode")

## a pure "conditional" model without intercepts

summary(mlogit(mode~pr+ca-1,data=Fish))

## a pure "multinomial model

summary(mlogit(mode~1|income,data=Fish))

## which can also be estimated using multinom (package nnet)

library(nnet)
```

```
summary(multinom(mode~income,data=Fishing))

## a "mixed" model

m <- mlogit(mode~pr+ca|income,data=Fish)
summary(m)

## robust inference with the sandwich package

library(lmtest)
coefTest(m,vcov=sandwich)
```

mlogit.data	<i>data.frame for logit model</i>
-------------	-----------------------------------

Description

shape a `data.frame` in a suitable form for the use of the `mlogit` function.

Usage

```
mlogit.data(x, choice, shape = c("wide", "long"), varying = NULL,
            sep=".", alt.var = NULL, id.var = NULL,
            alt.levels = NULL, opposite = NULL, ...)
```

Arguments

<code>x</code>	a <code>data.frame</code> ,
<code>choice</code>	the variable indicating the choice made,
<code>shape</code>	the shape of the <code>data.frame</code> : whether <code>long</code> if each row is an alternative or <code>wide</code> if each row is an observation,
<code>varying</code>	the indexes of the variables that are alternative specific,
<code>sep</code>	the separator of the variable name and the alternative name (only relevant for a <code>wide data.frame</code>),
<code>alt.var</code>	the name of the variable that contains the alternative index (for a <code>long data.frame</code> only) or the name under which the alternative index will be stored (the default name is <code>alt</code>),
<code>id.var</code>	the name of the variable that contains the choice index or the name under which the choice index will be stored,
<code>alt.levels</code>	the name of the alternatives: if null, for a <code>wide data.frame</code> , they are guessed from the variable names and the choice variable (both should be the same), for a <code>long data.frame</code> , they are guessed from the <code>alt.var</code> argument,
<code>opposite</code>	returns the opposite of the specified variables,
<code>...</code>	further arguments passed to <code>reshape</code> .

Value

A `data.frame` in long format, *i.e.* one line for each alternative. The first two columns contain the choice index and the alternative index. The choice variable is a boolean which indicates the choice made. This function use `reshape` if the `data.frame` is in wide format.

Author(s)

Yves Croissant

See Also

`reshape`

Examples

```
# ModeChoice is a long data.frame

data("ModeChoice", package="Ecdat")
ModeChoice <- mlogit.data(ModeChoice, choice="mode", shape="long",
  alt.levels=c("air", "train", "bus", "car"))

# Same but with an alt variable called altm

data("ModeChoice", package="Ecdat")
ModeChoice$altm <- rep(c("air", "train", "bus", "car"), 210)
ModeChoice <- mlogit.data(ModeChoice, choice="mode", shape="vert",
  alt.var="altm")

# Same but with an chid variable called ind

data("ModeChoice", package="Ecdat")
ModeChoice$ind <- rep(1:210, each=4)
ModeChoice <- mlogit.data(ModeChoice, choice="mode", shape="long",
  id.var="ind", alt.levels=c("air", "train", "bus", "car"))

# Same but with two own provided variables

data("ModeChoice", package="Ecdat")
ModeChoice$ind <- rep(1:210, each=4)
ModeChoice$altm <- rep(c("air", "train", "bus", "car"), 210)
ModeChoice <- mlogit.data(ModeChoice, choice="mode", shape="long",
  id.var="ind", alt.var="altm")

# Train is a wide data.frame with columns 'choiceid' is the choice
# index, the alternatives are named "ch1" and "ch2", the opposite
# of the variables is returned

data("Train", package="Ecdat")
Train <- mlogit.data(Train, choice="choice", shape="wide",
  varying=4:11, alt.levels=c("ch1", "ch2"),
  opposite=c("price", "time", "change", "comfort"), sep="")
```

```
# Car is a wide data.frame, indexes names are provided

data("Car", package="Ecdat")
Carl <-
mlogit.data(Car, varying=5:70, shape="wide", sep="", choice="choice",
            id.var="Mr", alt.var="Car", alt.levels=1:6)
```

Index

- *Topic **attribute**
 - mlogit.data, 8
- *Topic **htest**
 - hmfptest, 1
- *Topic **models**
 - logitform, 3
- *Topic **regression**
 - mlogit, 4

bread.mlogit (*mlogit*), 4

df.residual.mlogit (*mlogit*), 4

estfun.mlogit (*mlogit*), 4

fitted.mlogit (*mlogit*), 4

hmfptest, 1

is.logitform (*logitform*), 3

logitform, 3

logLik.mlogit (*mlogit*), 4

mlogit, 4

mlogit.data, 8

model.frame.logitform
 (*logitform*), 3

model.matrix.logitform
 (*logitform*), 3

model.matrix.mlogit (*mlogit*), 4

print.mlogit (*mlogit*), 4

print.summary.mlogit (*mlogit*), 4

residuals.mlogit (*mlogit*), 4

summary.mlogit (*mlogit*), 4

terms.logitform (*logitform*), 3

terms.mlogit (*mlogit*), 4

update.logitform (*logitform*), 3

update.mlogit (*mlogit*), 4

vcov.mlogit (*mlogit*), 4