

Package ‘mnlogit’

October 16, 2015

Type Package

Title Multinomial Logit Model

Version 1.2.4

Date 2015-10-16

Suggests VGAM, nnet

Imports mlogit, lmtest, Formula, stats

Author Asad Hasan, Wang Zhiyu, Alireza S. Mahani

Maintainer Asad Hasan <asad.hasan@sentrana.com>

Description Time and memory efficient estimation of multinomial logit models using maximum likelihood method. Numerical optimization performed by Newton-Raphson method using an optimized, parallel C++ library to achieve fast computation of Hessian matrices. Motivated by large scale multiclass classification problems in econometrics and machine learning.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-10-16 17:29:23

R topics documented:

Fish	2
Hypothesis Testing	3
mnlogit	4
Index	9

Fish

Choice of Fishing Mode

Description

A data.frame containing data on choices of recreational fishing mode. Data may depend on both the individual and the alternative.

number of observations : 1182

country : United States

Usage

```
data(Fish)
```

Format

A dataframe containing :

mode - recreation mode choice, one of : beach, pier, boat and charter

price - price for a mode for an individual

catch - fish catch rate for a mode for an individual

income - monthly income of an individual

chid - chooser ID: serial number of the individual

Source

Data taken from the R package **mlogit** by Yves Croissant, which lists the source as:

Herriges, J. A. and C. L. Kling (1999) “Nonlinear Income Effects in Random Utility Models”, *Review of Economics and Statistics*, **81**, 62-72.

References

Cameron, A.C. and P.K. Trivedi (2005) *Microeconometrics : methods and applications*, Cambridge, pp. 463–466, 486 and 491–495.

Description

Three hypothesis tests applicable to any MLE (Likelihood ratio test, Wald test, Rao score test) and the Hausman-McFadden test for IIA (independence of irrelevant alternatives) are provided.

Usage

```
lrtest(object, ...)
waldtest(object, ...)
scoretest(object, ...)
## S3 method for class 'mnlogit'
lrtest(object, ...)
## S3 method for class 'mnlogit'
scoretest(object, ...)
## S3 method for class 'mnlogit'
waldtest(object, ...)

hmftest(x, ...)
## S3 method for class 'formula'
hmftest(x, alt.subset, ...)
## S3 method for class 'mnlogit'
hmftest(x, z, ...)
```

Arguments

<code>object</code>	An fitted model which is an object of class <code>mnlogit</code> .
<code>...</code>	For <code>lrtest</code> and <code>waldtest</code> a fitted <code>mnlogit</code> object or a <code>formula</code> object maybe given. However for <code>scoretest</code> ONLY an fitted <code>mnlogit</code> is accepted. For <code>hmftest</code> either a subset of alternatives or an <code>mnlogit</code> object estimated using a subset of alternatives must be given.
<code>x</code>	A fitted object of class <code>mnlogit</code> or a <code>formula</code> object.
<code>z</code>	An object of class <code>mnlogit</code> or a subset of alternatives for the <code>hmftest.mnlogit</code> method. Must be the same model as <code>x</code> estimated on a subset of alternatives.
<code>alt.subset</code>	A subset of alternatives to do testing on.

Value

An object of class `htest`, with elements:

<code>statistic</code>	The value of the test statistic.
<code>parameter</code>	the value of the underlying test distribution's parameter. In this case, the number of degrees of freedom of chi-squared distribution.
<code>data.name</code>	The data frame used.

p.value	Probability for accepting the null hypothesis.
method	The name of the hypothesis test.
alternative	Alternative hypothesis.

Author(s)

Asad Hasan, Wang Zhiyu, Alireza S. Mahani

References

Code for the last two tests (Score and IIA test) is gratefully adapted from the CRAN package **mlogit**, while the first 2 tests are performed by functions in the CRAN package **lmtest**.

Croissant, Yves. *Estimation of multinomial logit models in R: The mlogit Packages*. <http://cran.r-project.org/package=mlogit>

Achim Zeileis, Torsten Hothorn (2002) *Diagnostic Checking in Regression Relationships*, R News 2(3), 7-10. <http://CRAN.R-project.org/doc/Rnews/>.

Examples

```
library(mnlogit)
data(Fish, package = "mnlogit")

# Unconstrained model
fm <- formula(mode ~ price | income | catch)
fit <- mnlogit(fm, Fish)
# Constrained model - intercep dropped
fm.c <- formula(mode ~ price | income - 1 | catch)
fit.c <- mnlogit(fm.c, Fish)

## MLE hypothesis tests
lrtest(fit, fit.c)
waldtest(fit, fit.c)
scoretest(fit, fit.c)

## IIA test
alt.subset <- c("beach", "boat", "charter")
hmfctest(fit, alt.subset)
```

mnlogit

Fast estimation of multinomial logit models

Description

Time and memory efficient estimation of multinomial logit models using maximum likelihood method. Targeted at large scale multiclass classification problems in econometrics and machine learning. Numerical optimization is performed by the Newton-Raphson method using an optimized, parallel C++ library to achieve fast computation of Hessian matrices. The user interface closely related to the CRAN package **mlogit**.

Usage

```

mnlogit(formula, data, choiceVar=NULL, maxiter = 50, ftol = 1e-6,
        gtol = 1e-6, weights = NULL, ncores = 1, na.rm = TRUE,
        print.level=0, linDepTol = 1e-6, start=NULL, alt.subset=NULL, ...)
## S3 method for class 'mnlogit'
fitted(object, outcome=TRUE, ...)
## S3 method for class 'mnlogit'
residuals(object, outcome=TRUE, ...)
## S3 method for class 'mnlogit'
df.residual(object, ...)
## S3 method for class 'mnlogit'
terms(x, ...)
## S3 method for class 'mnlogit'
update(object, new, ...)
## S3 method for class 'mnlogit'
print(x, digits = max(3, getOption("digits") - 2),
      width = getOption("width"),
      what = c("obj", "eststat", "modsize"), ...)
## S3 method for class 'mnlogit'
vcov(object, ...)
## S3 method for class 'mnlogit'
logLik(object, ...)
## S3 method for class 'mnlogit'
summary(object, ...)
## S3 method for class 'mnlogit'
print.summary(x, digits = max(3, getOption("digits") - 2),
             width = getOption("width"), ... )
## S3 method for class 'mnlogit'
index(object, ...)
## S3 method for class 'mnlogit'
predict(object, newdata = NULL, probability = TRUE,
        returnData=FALSE, choiceVar=NULL, ...)
## S3 method for class 'mnlogit'
coef(object, order=FALSE, as.list = FALSE, ...)

```

Arguments

formula	formula object or string specifying the model to be estimated (see Note).
data, newdata	A data.frame object with data organized in the 'long' format (see Note). This can also be a <code>mlogit.data</code> class object. <code>newdata</code> is used in the <code>predict</code> method.
choiceVar	A string naming the column in 'data' which has the list of choices. <i>Note:</i> This argument is not used if <code>data</code> or <code>newdata</code> is a <code>mlogit.data</code> object.
maxiter	An integer indicating maximum number of Newton's iterations. If <code>maxiter</code> ≤ 0 , then only Hessian, gradient and the loglikelihood are calculated at initial point.
ftol	A real number indicating tolerance on the difference of two subsequent loglikelihood values.
gtol	A real number indicating tolerance on norm of the gradient.

<code>weights</code>	Optional vector of (positive) frequency weights, one for each observation.
<code>ncores</code>	An integer indicating number of processors allowed for Hessian calculations.
<code>na.rm</code>	a logical variable which indicates whether rows of the data frame containing NAs will be removed.
<code>print.level</code>	An integer which controls the amount of information to be printed during execution.
<code>linDepTol</code>	Tolerance for detecting linear dependence between columns in input data. Dependent columns are removed from the estimation.
<code>start</code>	Named vector of coefficients to use as initial guess. Use naming convention as given by <code>names(coefit())</code> , where <code>fit</code> is a <code>mnlogit</code> class object.
<code>alt.subset</code>	Subset of alternatives to perform estimation on.
<code>...</code>	Currently unused.
<code>object, x</code>	An object of class <code>mnlogit</code> .
<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>new</code>	An formula for the update method. It must obey all rules specified for the <code>formula</code> argument.
<code>digits</code>	Number of digits to print.
<code>width</code>	The width of printing.
<code>what</code>	Specifies what to print. Default option is <code>'obj'</code> is the print function for <code>mnlogit</code> objects. Option <code>'eststat'</code> prints estimation stats and option <code>'mdsize'</code> prints model size information.
<code>probability</code>	If TRUE predict output the probability matrix, otherwise the choice with the highest probability for each observation is returned.
<code>returnData</code>	If TRUE a data attribute is added to the returned object.
<code>order</code>	If TRUE coefficients are ordered by variable name.
<code>as.list</code>	Returns estimated model coefficients grouped by variable type.

Value

An object of class `mnlogit`, with elements:

<code>coefficients</code>	the named vector of coefficients.
<code>logLik</code>	the value of the log-likelihood function at exit.
<code>gradient</code>	the gradient of the log-likelihood function at exit.
<code>hessian</code>	the Hessian of the log-likelihood function at exit.
<code>est.stat</code>	Newton Raphson stats.
<code>fitted.values</code>	Estimated probabilities of the alternative selected in each observation.
<code>probabilities</code>	the probability matrix: (i, j) entry denotes the probability of the j th alternative being chosen in the i th observation.

residuals	The residual. Has attribute outcome which is the probability of not choosing the selected alternative.
df	The number of estimated coefficients in the model.
AIC	The AIC value of the fitted model.
choices	The vector of alternatives's names.
model.size	Information about number of parameters in model.
ordered.coeff	Vector of coefficients ordered by variable name.
model	The data.frame used in model estimation.
freq	The relative frequency of each choice in input data.
formula	The formula specifying the model.
call	The mnlogit function call that user made,

Note

1. The data must be in the 'long' format. This means that for each observation there must be as many rows as there are alternatives (which should be grouped together).
2. The formula should be specified in the format: responseVar ~ choice specific variables with generic coefficients | individual specific variables | choice specific variables with choice specific coefficients. These are the 3 available variable types.
3. Any type of variables may be omitted. To omit use "1" as a placeholder.
4. An alternative specific intercept is included by default in the estimation. To omit it, use a '-1' or '0' anywhere in the formula.

Author(s)

Asad Hasan, Wang Zhiyu, Alireza S. Mahani

References

Croissant, Yves. *Estimation of multinomial logit models in R: The mlogit Packages*. <http://cran.r-project.org/package=mlogit>

Train, K. (2004) *Discrete Choice Methods with Simulation*, Cambridge University Press.

Examples

```
library(mnlogit)
data(Fish, package = "mnlogit")
fm <- formula(mode ~ price | income | catch)

fit <- mnlogit(fm, Fish, ncores = 2)

## Not run:
fit <- mnlogit(fm, Fish, choiceVar="alt", ncores = 2) # same effect as previous
summary(fit)
print(fit)
```

```
predict(fit)
print(fit, what = "eststat")
print(fit, what = "modsize")

# Formula examples (see also Note)
fm <- formula(mode ~ 1 | income) # Only type-2 with intercept
fm <- formula(mode ~ price - 1) # Only type-1, no intercept
fm <- formula(mode ~ 1 | 1 | catch) # Only type-3, including intercept

## End(Not run)
```


Index

*Topic **datasets**

Fish, [2](#)

*Topic **mnlogit, logistic, classification,
multinomial, mlogit, parallel**

mnlogit, [4](#)

coef.mnlogit (mnlogit), [4](#)

df.residual.mnlogit (mnlogit), [4](#)

Fish, [2](#)

Fishing (Fish), [2](#)

fitted.mnlogit (mnlogit), [4](#)

hmftest (Hypothesis Testing), [3](#)

Hypothesis Testing, [3](#)

index (mnlogit), [4](#)

logLik.mnlogit (mnlogit), [4](#)

lrtest (Hypothesis Testing), [3](#)

mnlogit, [4](#)

predict.mnlogit (mnlogit), [4](#)

print.est.stats (mnlogit), [4](#)

print.mnlogit (mnlogit), [4](#)

print.model.size (mnlogit), [4](#)

print.summary.mnlogit (mnlogit), [4](#)

residuals.mnlogit (mnlogit), [4](#)

scoretest (Hypothesis Testing), [3](#)

summary.mnlogit (mnlogit), [4](#)

terms.mnlogit (mnlogit), [4](#)

update.mnlogit (mnlogit), [4](#)

vcov.mnlogit (mnlogit), [4](#)

waldtest (Hypothesis Testing), [3](#)