Package ‘multinomRob’

February 20, 2015

Version 1.8-6.1
Date 2013/02/15
Title Robust Estimation of Overdispersed Multinomial Regression Models
Author Walter R. Mebane, Jr. <wrm1@cornell.edu>, Jasjeet Singh Sekhon <sekhon@berkeley.edu>
Maintainer Jasjeet Singh Sekhon <sekhon@berkeley.edu>
Description MNL and overdispersed multinomial regression using robust (LQD and tanh) estimation
Depends R (>= 1.7.1), rgenoud (>= 2.9), MASS (>= 7.1-8), mvtnorm (>= 0.6-3)
License GPL (>= 2)
URL http://sekhon.berkeley.edu/
NeedsCompilation yes
Repository CRAN
Date/Publication 2013-02-16 09:09:47

R topics documented:

Multinomial Multivariate-T Regression ........................................... 2
Multinomial Regression ............................................................. 3
Multinomial Regression Tanh Estimator ....................................... 6
Multinomial Regression Tanh Estimator Gauss-Newton Optimization ..... 8
rmultinomial .......................................................... 11
Robust Multinomial Regression ............................................... 12

Index 18
Multinomial Multivariate-T Regression

Description

multinomT fits the multinomial multivariate-t regression for grouped count data. This function is not meant to be called directly by the user. It is called by multinomRob, which constructs the various arguments.

Usage

multinomT(Yp, Xarray, xvec, jacstack, start = NA, nobsvec, fixed.df = NA)

Arguments

Yp  Matrix (observations by alternatives) of outcome proportions. Values must be between 0 and 1. Missing data (NA values) are not allowed.
Xarray  Array of regressors. dim(Xarray) = c(observations, parameters, alternatives).
xvec  Matrix (parameters by alternatives) that represents the model structure. It has a 1 for an estimated parameter, an integer greater than 1 for an estimated parameter constrained equal to another estimated parameter (all parameters constrained to be equal to one another have the same integer value in xvec) and a 0 otherwise.
jacstack  Array of regressors used to facilitate computing the gradient and the hessian matrix. dim(jacstack) = c(observations, unique parameters, alternatives).
start  A list of starting values of three kinds of parameters: start$beta, the values for the regression coefficients; start$Omega, the values for the variance-covariance matrix; start$df, the value for the multivariate-t degrees of freedom parameter.
nobsvec  Vector of the total number of counts for each observation.
fixed.df  The degrees of freedom to be used for the multivariate-t distribution. When this is specified, the DF will not be estimated.

Details

The function often provides good starting values for multinomRob’s LQD estimator, but the standard errors it reports are not correct, in part because they ignore heteroscedasticity.

Value

call  Names and values of all of the arguments which were passed to the function. See match.call for further details.
logL  Log likelihood.
deviance  Deviance.
Multinomial Regression

par
A list of three kinds of parameter estimates: par$beta, the estimates for the regression coefficients; par$omega, the estimates for the variance-covariance matrix; par$df, the estimate of the multivariate-t degrees of freedom parameter.

se
Vector of standard errors for the regression coefficients. WARNING: these are not correct in part because the model ignores heteroscedasticity.

optim
Returned by optim.

pred
A matrix of predicted probabilities with the same dimention as Yp.

Author(s)

Walter R. Mebane, Jr., University of Michigan, <wmebane@umich.edu>, http://www-personal.umich.edu/~wmebane

Jasjeet S. Sekhon, UC Berkeley, <sekhon@berkeley.edu>, http://sekhon.berkeley.edu/

References


For additional documentation please visit http://sekhon.berkeley.edu/robust/.

See Also

match.call.optim.

Multinomial Regression

Multinomial Regression Maximum Likelihood Estimator with Overdispersion

Description

multinomMLE estimates the coefficients of the multinomial regression model for grouped count data by maximum likelihood, then computes a moment estimator for overdispersion and reports standard errors for the coefficients that take overdispersion into account. This function is not meant to be called directly by the user. It is called by multinomRob, which constructs the various arguments.

Usage

multinomMLE(Y, ypos, Xarray, xvec, jacstack, itmax=100, xvar.labels, choice.labels, MLEonly=FALSE, print.level=0)
Multinomial Regression

Arguments

Y
Matrix (observations by alternatives) of outcome counts. Values must be non-
negative. Missing data (NA values) are not allowed.

Ypos
Matrix indicating which elements of Y are counts to be analyzed (TRUE) and
which are values to be skipped (FALSE). This allows the set of outcome alter-
natives to vary over observations.

Xarray
Array of regressors. dim(Xarray) = c(observations, parameters, alternatives).

xvec
Matrix (parameters by alternatives) that represents the model structure. It has a 1
for an estimated parameter, an integer greater than 1 for an estimated parameter
constrained equal to another estimated parameter (all parameters constrained to
be equal to one another have the same integer value in xvec) and a 0 otherwise.

jacstack
Array of regressors used to facilitate computing the gradient and the hessian
matrix. dim(jacstack) = c(observations, unique parameters, alternatives).

itmax
The maximum number of iterations to be done in the Gauss-Newton optimiza-
tion.

xvar.labels
Vector of labels for observations.

choice.labels
Vector of labels for outcome alternatives.

MLEonly
If TRUE, then only the standard maximum-likelihood MNL model is estimated—
i.e., no overdispersion parameter is estimated.

print.level
Specify 0 for minimal printing (error messages only) or 3 to print details about
the MLE computations.

Details

Following the generalized linear models approach, the coefficient parameters in an overdispersed
multinomial regression model may be estimated using the likelihood for a standard multinomial re-
gression model. A moment estimator may be used for the dispersion parameter, given the coefficient
estimates, with little efficiency loss.

Value

multinomMLE returns a list containing the following objects. The returned objects are:

coefficients
The maximum likelihood coefficient estimates in matrix format. The value 0 is
used in the matrix to fill in for values that do not correspond to a regressor.

coeffvec
A vector containing the maximum likelihood coefficient estimates.

dispersion
Moment estimate of the dispersion: mean sum of squared orthogonalized resid-
uals (adjusted for degrees of freedom lost to estimated coefficients).

se
The MLE coefficient estimate standard errors derived from the asymptotic co-
variance estimated using the Hessian matrix (observed information).

se.opg
The MLE coefficient estimate standard errors derived from the asymptotic co-
variance estimated using the outer product of the gradient (expected informa-
tion) divided by the moment estimate of the dispersion. Not provided if MLEonly==TRUE.
Multinomial Regression

se.hes  The MLE coefficient estimate standard errors derived from the asymptotic covariance estimated using the Hessian matrix (observed information). Same as se; included for backward compatibility.

se.sw  The MLE coefficient estimate standard errors derived from the asymptotic covariance estimated using the estimated asymptotic sandwich covariance estimate. Not provided if mleonly=TRUE.

se.vec  se in vector form.

se.opg.vec  se.opg in vector form.

se.hes.vec  se.hes in vector form.

se.sw.vec  se.sw in vector form.

A  The outer product of the gradient (expected information) divided by the moment estimate of the dispersion.

B  The inverse of the hessian matrix (observed formation).

covmat  Sandwich estimate of the asymptotic covariance of the maximum likelihood coefficient estimates.

iters  Number of Gauss-Newton iterations.

error  Exit error code.

GNlist  List reporting final results of the Gauss-Newton optimization. Elements: coefficients, vector of coefficient parameters (same as coeffvec value in list returned by multinomMLE); tvec, matrix of coefficient parameters (same as coefficients value in list returned by multinomMLE); formation, inverse Hessian matrix; score, score (or gradient element) matrix; LLvals, list containing log-likelihood value; convflag, TRUE/FALSE convergence flag; iters, number of iterations done in final Gauss-Newton stage; posdef, TRUE if Hessian is positive definite.

sigma2  Moment estimate of the dispersion: mean sum of squared orthogonalized residuals (adjusted for degrees of freedom lost to estimated coefficients).

Y  The same Y matrix that was supplied as input, except modified by having done Y[!Ypos] <- 0.

Ypos  The same Ypos matrix that was supplied as input.

fitted.prob  The matrix of predicted probabilities for each category for each observation based on the coefficient estimates.

jacstack  The same jacstack that was supplied as an input argument.

Author(s)

Walter R. Mebane, Jr., University of Michigan, <wmebane@umich.edu>, http://www-personal.umich.edu/~wmebane

Jasjeet S. Sekhon, UC Berkeley, <sekhon@berkeley.edu>, http://sekhon.berkeley.edu/

References


For additional documentation please visit http://sekhon.berkeley.edu/robust/.
Examples

```r
# make some multinomial data
x1 <- rnorm(50);
x2 <- rnorm(50);
p1 <- exp(x1)/(1+exp(x1)+exp(x2));
p2 <- exp(x2)/(1+exp(x1)+exp(x2));
p3 <- 1 - (p1 + p2);
y <- matrix(0, 50, 3);
for (i in 1:50) {
  y[i,] <- rmultinomial(1000, c(p1[i], p2[i], p3[i]));
}

# perturb the first 5 observations
y[1:5,c(1,2,3)] <- y[1:5,c(3,1,2)];
y1 <- y[,1];
y2 <- y[,2];
y3 <- y[,3];

# put data into a dataframe
dtf <- data.frame(x1, x2, y1, y2, y3);

# Do MLE estimation. The following model is NOT identified if we
# try to estimate the overdispersed MNL.
dtf <- data.frame(y1=c(1,1), y2=c(2,1), y3=c(1,2), x=c(0,1))
summary(multinomRob(list(y1 ~ 0, y2 ~ x, y3 ~ x), data=dtf, MLEonly=TRUE))
```

Multinomial Regression Tanh Estimator

**Multinomial Regression Hyperbolic Tangent (Tanh) Estimator**

Description

`multinomTanh` fits the overdispersed multinomial regression model for grouped count data using the hyperbolic tangent (tanh) estimator. This function is not meant to be called directly by the user. It is called by `multinomRob`, which constructs the various arguments.

Usage

```r
multinomTanh(Y, Ypos, X, jacstack, xvec, tvec, pop, s2,
             xvar.labels, choice.labels, print.level = 0)
```

Arguments

- **Y** Matrix (observations by alternatives) of outcome counts. Values must be non-negative. Missing data (NA values) are not allowed.
- **Ypos** Matrix indicating which elements of `Y` are counts to be analyzed (TRUE) and which are values to be skipped (FALSE). This allows the set of outcome alternatives to vary over observations.
Multinomial Regression Tanh Estimator

X
Array of regressors. \( \text{dim}(X) = c(\text{observations}, \text{parameters}, \text{alternatives}) \).

jacstack
Array of regressors used to facilitate computing the gradient and the hessian matrix. \( \text{dim(jacstack)} = c(\text{observations}, \text{unique parameters}, \text{alternatives}) \).

xvec
Matrix (parameters by alternatives) that represents the model structure. It has a 1 for an estimated parameter, an integer greater than 1 for an estimated parameter constrained equal to another estimated parameter (all parameters constrained to be equal to one another have the same integer value in xvec) and a 0 otherwise.

tvec
Starting values for the regression coefficient parameters, as a matrix (parameters by alternatives). Parameters that are involved in equality constraints are repeated in tvec.

pop
Vector giving the total number of counts for each observation. In general, \( \text{pop} \leftarrow \text{apply}(Y \ast \text{ifelse}(Ypos, 1, 0), 1, \text{sum}) \).

s2
Overdispersion value. In multinomRob this is the square of the LQD scale estimate.

xvar.labels
Vector of labels for observations.

choice.labels
Vector of labels for outcome alternatives.

print.level
Specify 0 for minimal printing (error messages only) or 2 to print details about the tanh computations.

Details

The tanh estimator is a redescending M-estimator. Given an estimate of the scale of the overdispersion, the tanh estimator estimates the coefficient parameters of the linear predictors of the multinomial regression model.

Value

multinomTanh returns a list of 5 objects. The returned objects are:

mtanh
List of tanh estimation results from function mGNTanh.

weights
The matrix of tanh weights for the orthogonalized residuals. The matrix has the same dimensions as the outcome count matrix \( Y \). The first column of the matrix has names for the observations, and the remaining columns contain the weights. Each of the latter columns has a name derived from the choice.labels vector: column \( i+1 \) is named \( \text{paste}("\text{weights}:", \text{choice.labels}[i], \text{sep}="\") \).

If \( \text{sum}(\text{Ypos}[i,] == \text{FALSE}) > 0 \), then values of NA appear in weights[i,], with \( \text{sum(is.na(weights[i,]))} = \text{sum(!Ypos[i,])} \). The NA values will be the last values in the affected row of the weights matrix, regardless of which outcome alternatives were unavailable for the observation.

hdiag
The matrix of weights used to fully studentize the orthogonalized residuals. The matrix has the same dimensions as the outcome count matrix \( Y \). The first column of the matrix has names for the observations, and the remaining columns contain the weights. Each of the latter columns has a name derived from the choice.labels vector: column \( i+1 \) is named \( \text{paste}(\text{"Hdiag:"}, \text{choice.labels}[i], \text{sep}="\") \).

If \( \text{sum}(\text{Ypos}[i,] == \text{FALSE}) > 0 \), then values of 0 appear in hdiag[i,], with \( \text{sum(is.na(hdiag[i,]))} = \text{sum(!Ypos[i,])} \). The 0 values created for this reason will be the last values in the affected row.
List of predicted outcome counts, studentized residuals and standardized residuals.

tvec

The tanh coefficient estimates in matrix format. The matrix has one column for each outcome alternative. The label for each row of the matrix gives the names of the regressors to which the coefficient values in the row apply. The regressor names in each label are separated by a forward slash (/), and NA is used to denote that no regressor is associated with the corresponding value in the matrix. The value 0 is used in the matrix to fill in for values that do not correspond to a regressor.

Author(s)

Walter R. Mebane, Jr., University of Michigan, <wmebane@umich.edu>, http://www-personal.umich.edu/~wmebane

Jasjeet S. Sekhon, UC Berkeley, <sekhon@berkeley.edu>, http://sekhon.berkeley.edu/

References


For additional documentation please visit http://sekhon.berkeley.edu/robust/.

Description

mGntanh uses Gauss-Newton optimization to compute the hyperbolic tangent (tanh) estimator for the overdispersed multinomial regression model for grouped count data. This function is not meant to be called directly by the user. It is called by multinomRob, which constructs the various arguments.

Usage

mGntanh(bstart, sigma2, resstart, Y, Ypos, Xarray, xvec, tvec, jacstack, itmax = 100, print.level = 0)
Arguments

- **bstart**: Vector of starting values for the coefficient parameters.
- **sigma2**: Value of the dispersion parameter (variance). The estimator does not update this value.
- **resstart**: Array of initial orthogonalized (but not standardized) residuals.
- **Y**: Matrix (observations by alternatives) of outcome counts. Values must be non-negative. Missing data (NA values) are not allowed.
- **Ypos**: Matrix indicating which elements of Y are counts to be analyzed (TRUE) and which are values to be skipped (FALSE). This allows the set of outcome alternatives to vary over observations.
- **xarray**: Array of regressors. dim(Xarray) = c(observations, parameters, alternatives).
- **xvec**: Matrix (parameters by alternatives) that represents the model structure. It has a 1 for an estimated parameter, an integer greater than 1 for an estimated parameter constrained equal to another estimated parameter (all parameters constrained to be equal to one another have the same integer value in xvec) and a 0 otherwise.
- **tvec**: Starting values for the regression coefficient parameters, as a matrix (parameters by alternatives). Parameters that are involved in equality constraints are repeated in tvec.
- **jacstack**: Array of regressors used to facilitate computing the gradient and the Hessian matrix. dim(jacstack) = c(observations, unique parameters, alternatives).
- **itmax**: Maximum number of Gauss-Newton stages. Each stage does at most 100 Gauss-Newton steps.
- **print.level**: Specify 0 for minimal printing (error messages only) or 2 to print details about the tanh computations.

Details

The tanh estimator is a redescending M-estimator. Given an estimate of the scale of the overdispersion, the tanh estimator estimates the coefficient parameters of the linear predictors of the multinomial regression model.

Value

mGNtanh returns a list of 16 objects. The returned objects are:

- **coefficients**: The tanh coefficient estimates in matrix format. The matrix has one column for each outcome alternative. The label for each row of the matrix gives the names of the regressors to which the coefficient values in the row apply. The regressor names in each label are separated by a forward slash (/), and NA is used to denote that no regressor is associated with the corresponding value in the matrix. The value 0 is used in the matrix to fill in for values that do not correspond to a regressor.
- **coeffvec**: A vector containing the tanh coefficient estimates.
- **dispersion**: Value of the dispersion parameter (variance). This is the value specified in the argument sigma2 in the call to the function.
w

Vector of weights based on the tanh estimator’s psi function for each observation.

psi

Vector of values of the tanh estimator’s psi function for each observation.

A

The outer product of the gradient (expected information) divided by the moment estimate of the dispersion.

B

The inverse of the Hessian matrix (observed formation).

covmat

Sandwich estimate of the asymptotic covariance of the tanh coefficient estimates.

iters

Number of Gauss-Newton iterations.

error

Error code: 0, no errors; 2, sum(w) < nobs*(ncats-1)/2 (weights are too small); 32, Hessian not positive definite in the final Newton step.

GNlist

List reporting final results of the Gauss-Newton optimization. Elements: coefficients, vector of coefficient parameters (same as coeffvec value in list returned by mGNtanh); tvec, matrix of coefficient parameters (same as coefficients value in list returned by mGNtanh); formation, inverse Hessian matrix; score, score (or gradient element) matrix; llvals, list containing weighted (llvals$LL) and unweighted (llvals$LU) log-likelihood values; convflag, TRUE/FALSE convergence flag; iters, number of iterations done in final Gauss-Newton stage; posdef, TRUE if Hessian is positive definite.

tanhsigma2

The tanh overdispersion parameter estimate, which is a weighted moment estimate of the dispersion: weighted mean sum of squared orthogonalized residuals (adjusted for effective sample size after weighting and degrees of freedom lost to estimated coefficients).

Y

The same Y matrix that was supplied as input, except modified by having done Y[!Ypos] <- 0.

Ypos

The same Ypos matrix that was supplied as input.

probmat

The matrix of predicted probabilities for each category for each observation based on the coefficient estimates.

jacstack

The same jacstack that was supplied as an input argument.

xarray

The same xarray that was supplied as an input argument.

Author(s)

Walter R. Mebane, Jr., University of Michigan, <wmebane@umich.edu>, http://www-personal.umich.edu/~wmebane

Jasjeet S. Sekhon, UC Berkeley, <sekhon@berkeley.edu>, http://sekhon.berkeley.edu/

References


For additional documentation please visit http://sekhon.berkeley.edu/robust/.
Description
Generates a random count vector for one observation of a multinomial distribution for \( n \) trials with probability vector \( pr \).

Usage
\[
\text{rmultinomial}(n = 5, \ pr = c(0.5, 0.5), \ \text{long} = \text{FALSE})
\]

Arguments
- \( n \) Number of trials.
- \( pr \) Probability vector.
- \( \text{long} \) TRUE to choose one generator, FALSE to choose another one.

Details
Generates a random count vector for one observation of a multinomial distribution for \( n \) trials with probability vector \( pr \).

Value
\( x \) Vector of counts.

Note
This function is only used in the examples and not in the \texttt{multinomRob} code.

Author(s)
Walter R. Mebane, Jr., University of Michigan, \texttt{wmebane@umich.edu}, \url{http://www-personal.umich.edu/~wmebane}

Jasjeet S. Sekhon, UC Berkeley, \texttt{sekhon@berkeley.edu}, \url{http://sekhon.berkeley.edu/}

Examples
\[
\text{rmultinomial}(10, \ c(.3, .3, .4));
\]
Robust Multinomial Regression

Multinomial Robust Estimation

Description

multinomRob fits the overdispersed multinomial regression model for grouped count data using the hyperbolic tangent (tanh) and least quartile difference (LQD) robust estimators.

Usage

multinomRob(model, data, starting.values=NULL, equality=NULL, genoud.parms=NULL, print.level=0, iter = FALSE, maxiter = 10, multinom.t=1, multinom.t.df=NA, MLEonly=FALSE)

Arguments

model

The regression model specification. This is a list of formulas, with one formula for each category of outcomes for which counts have been measured for each observation. For example, in the following,

model=list(y1 ~ x1, y2 ~ x2, y3 ~ 0)

the outcome variables containing counts are y1, y2 and y3, and the linear predictor for y1 is a coefficient times x1 plus a constant, the linear predictor for y2 is a coefficient times x2 plus a constant, and the linear predictor for y3 is zero. Each formula has the format countvar ~ rhs, where countvar is the name of a vector, in the dataframe referenced by the data argument, that gives the counts for all observations for one category. RHS denotes the righthand side of a formula using the usual syntax for formulas, where each variable in the formula is the name of a vector in the dataframe referenced by the data argument. For example, a RHS specification of var1 + var2*var3 would specify that the regressors are to be var1, var2, var3, the terms generated by the interaction var2:var3, and the constant.

The set of outcome alternatives may be specified to vary over observations, by putting in a negative value for alternatives that do not exist for particular observations. If the value of an outcome variable is negative for an observation, then that outcome is considered not available for that observation. The predicted counts for that observation are defined only for the available observations and are based on the linear predictors for the available observations. The same set of coefficient parameter values are used for all observations. Any observation for which fewer than two outcomes are available is omitted.

Observations with missing data (NA) in any outcome variable or regressor are omitted (listwise deletion).

In a model that has the same regressors for every category, except for one category for which there are no regressors in order to identify the model (the reference category), the RHS specification must be given for all the categories except...
the reference category. The formula for the reference category must include a
RHS specification that explicitly omits the constant, e.g., countvar ~ -1 or
countvar ~ 0. The number of coefficient parameters to be estimated equals the
number of terms generated by all the formulas, subject to equality constraints
that may be specified using the equality argument.

data
The dataframe that contains all the variables referenced in the model argument,
which are the data to be analyzed.

starting.values
Starting values for the regression coefficient parameters, as a vector. The pa-
rameter ordering matches the ordering of the formulas in the model argument:
parameters for the terms in the first formula appear first, then come parameters
for the terms in the second formula, etc. In practice it will usually be better to
start by letting multinomRob find starting values by using the multinom.t op-
tion, then using the results from one run as starting values for a subsequent run
done with, perhaps, a larger population of operators for rgenoud.

equality
List of equality constraints. This is a list of lists of formulas. Each formula has
the same format as in the model specification, and must include only a subset of
the outcomes and regressors used in the model specification formulas. All the
coefficients specified by the formulas in each list will be constrained to have the
same value during estimation. For example, in the following,
multinomRob(model=list(y1 ~ x1, y2 ~ x2, y3 ~ 0), data=dtf, equality=list(list(y1
list(y1 ~ x1, y2 ~ x2, y3 ~ 0))
and the coefficients of x1 and x2 are constrained equal by
equality=list(list(y1 ~ x1 + 0, y2 ~ x2 + 0))
In the equality formulas it is necessary to say + 0 so the intercepts are not
involved in the constraints. If a parameter occurs in two different lists in the
equality= argument, then all the parameters in the two lists are constrained to
be equal to one another. In the output this is described as consolidating the lists.

genoud.parms
List of named arguments used to control the rgenoud optimizer, which is used
to compute the LQD estimator.

print.level
Specify 0 for minimal printing, 1 to print more detailed information about LQD
and other intermediate computations, 2 to print details about the tanh computa-
tions, or 3 to print details about starting values computations.

iter
TRUE means to iterate between LQD and tanh estimation steps until either the
algorithm converges, the number of iterations specified by the maxiter argu-
ment is reached, or if an LQD step occurs that produces a larger value than the
previous step did for the overdispersion scale parameter. This option is often
improves the fit of the model.

maxiter
The maximum number of iterations to be done between LQD and tanh estima-
tion steps.

multinom.t
1 means use the multinomial multivariate-t model to compute starting values for
the coefficient parameters. But if the MNL results are better (as judged by the
LQD fit), MNL values will be used instead. 0 means use nonrobust maximum
likelihood estimates for a multinomial regression model. 2 forces the use of the
multivariate-t model for starting values even if the MNL estimates provide better starting values for the LQD. Note that with multinom.t=1 or multinom.t=2, multivariate-t starting values will not be used if the model cannot generate valid standard errors. To force the use of multivariate-t estimates even in this circumstance, see the multinom.t.df argument.

If the starting.values argument is not NULL, the starting values given in that argument are used and the multinom.t argument is ignored. Multinomial multivariate-t starting values are not available when the number of outcome alternatives varies over the observations.

multinom.t.df NA means that the degrees of freedom (DF) for the multivariate-t model (when used) should be estimated. If multinom.t.df is a number, that number will be used for the degrees of freedom and the DF will not be estimated. Only a positive number should be used. Setting multinom.t.df to a number also implies that, if multinom.t=1 or multinom.t=2, the multivariate-t starting values will be used (depending on the comparison with the MNL estimates if multinom.t=1 is set) even if the standard errors are not defined.

MLEonly If TRUE, then only the standard maximum-likelihood MNL model is estimated. No robust estimation model and no overdispersion parameter is estimated.

Details

The tanh estimator is a redescending M-estimator, and the LQD estimator is a generalized S-estimator. The LQD is used to estimate the scale of the overdispersion. Given that scale estimate, the tanh estimator is used to estimate the coefficient parameters of the linear predictors of the multinomial regression model.

If starting values are not supplied, they are computed using a multinomial multivariate-t model. The program also computes and reports nonrobust maximum likelihood estimates for the multinomial regression model, reporting sandwich estimates for the standard errors that are adjusted for a nonrobust estimate of the error dispersion.

Value

multinomRob returns a list of 15 objects. The returned objects are:

coefficients The tanh coefficient estimates in matrix format. The matrix has one column for each formula specified in the model argument. The name of each column is the name used for the count variable in the corresponding formula. The label for each row of the matrix gives the names of the regressors to which the coefficient values in the row apply. The regressor names in each label are separated by a forward slash (/), and NA is used to denote that no regressor is associated with the corresponding value in the matrix. The value 0 is used in the matrix to fill in for values that do not correspond to a model formula regressor.

se The tanh coefficient estimate standard errors in matrix format. The format and labelling used for the matrix is the same as is used for the coefficients. The standard errors are derived from the estimated asymptotic sandwich covariance estimate.
LQDsigma2 The LQD dispersion (variance) parameter estimate. This is the LQD estimate of the scale value, squared.

TANHsigma2 The tanh dispersion parameter estimate.

weights The matrix of tanh weights for the orthogonalized residuals. The matrix has one row for each observation in the data and as many columns as there are formulas specified in the `model` argument. The first column of the matrix has names for the observations, and the remaining columns contain the weights. Each of the latter columns has a name derived from the name of one of the count variables named in the `model` argument. If count1 is the name of the count variable used in the first formula, then the second column in the matrix is named `weights:count1`, etc.

If an observation has negative values specified for some outcome variables, indicating that those outcome alternatives are not available for that observation, then values of `NA` appear in the weights matrix for that observation, as many `NA` values as there are unavailable alternatives. The `NA` values will be the last values in the affected row of the weights matrix, regardless of which outcome alternatives were unavailable for the observation.

hdiag Weights used to fully studentize the orthogonalized residuals. The matrix has one row for each observation in the data and as many columns as there are formulas specified in the `model` argument. The first column of the matrix has names for the observations, and the remaining columns contain the weights. Each of the latter columns has a name derived from the name of one of the count variables named in the `model` argument. If count1 is the name of the count variable used in the first formula, then the second column in the matrix is named `hdiag:count1`, etc.

If an observation has negative values specified for some outcome variables, indicating that those outcome alternatives are not available for that observation, then values of 0 appear in the weights matrix for that observation, as many 0 values as there are unavailable alternatives. Values of 0 that are created for this reason will be the last values in the affected row of the weights matrix, regardless of which outcome alternatives were unavailable for the observation.

prob The matrix of predicted probabilities for each category for each observation based on the tanh coefficient estimates.

residuals.rotate Matrix of studentized residuals which have been made comparable by rotating each choice category to the first position. These residuals, unlike the student and standard residuals below, are no longer orthogonalized because of the rotation. These are the residuals displayed in Table 6 of the reference article.

residuals.student Matrix of fully studentized orthogonalized residuals.

residuals.standard Matrix of orthogonalized residuals, standardized by dividing by the overdispersion scale.

mnl List of nonrobust maximum likelihood estimation results from function `multinomMLE`.

multinomT List of multinomial multivariate-t estimation results from function `multinomT`.
Robust Multinomial Regression

**genoud**
List of LQD estimation results obtained by rgenoud optimization, from function genoudRob.

**mtanh**
List of tanh estimation results from function mGNTanh.

**error**
Exit error code, usually from function mGNTanh.

**iter**
Number of LQD-tanh iterations.

**Author(s)**
Walter R. Mebane, Jr., University of Michigan, <wmebane@umich.edu>, http://www-personal.umich.edu/~wmebane

Jasjeet S. Sekhon, UC Berkeley, <sekhon@berkeley.edu>, http://sekhon.berkeley.edu/

**References**

For additional documentation please visit http://sekhon.berkeley.edu/robust/.

**Examples**

```r
# make some multinomial data
x1 <- rnorm(50);
x2 <- rnorm(50);
p1 <- exp(x1)/(1+exp(x1)+exp(x2));
p2 <- exp(x2)/(1+exp(x1)+exp(x2));
p3 <- 1 - (p1 + p2);
y <- matrix(0, 50, 3);
for (i in 1:50) {
y[i,] <- rmultinomial(1000, c(p1[i], p2[i], p3[i]));
}

# perturb the first 5 observations
y[1:5, c(1,2,3)] <- y[1:5, c(3,1,2)];
y1 <- y[,1];
y2 <- y[,2];
y3 <- y[,3];

# put data into a dataframe
dtf <- data.frame(x1, x2, y1, y2, y3);

## Set parameters for Genoud
## Not run:
## For production, use these kinds of parameters
zz.genoud.parms <- list(pop.size = 1000,
                       wait.generations = 10,
                       max.generations = 100,
                       scale.domains = 5,
                       print.level = 0)
```
'## end

## For testing, we are setting the parameters to run quickly. Don't use these for production
zz.genoud parms <- list( 
  pop.size = 10,
  wait.generations = 1,
  max.generations = 1,
  scale.domains = 5,
  print.level = 0
)

# estimate a model, with "y3" being the reference category
# true coefficient values are: (Intercept) = 0, x = 1
# impose an equality constraint
# equality constraint: coefficients of x1 and x2 are equal
mulrobe <- multinomRob(list(y1 ~ x1, y2 ~ x2, y3 ~ 0),
  dtf,
  equality = list(y1 ~ x1 + 0, y2 ~ x2 + 0)),
  genoud parms = zz.genoud parms,
  print.level = 3, iter=FALSE);
summary(mulrobe, weights=TRUE);

# Do only MLE estimation. The following model is NOT identified if we
# try to estimate the overdispersed MNL.
dtf <- data.frame(y1=c(1,1), y2=c(2,1), y3=c(1,2), x=c(0,1))
summary(multinomRob(list(y1 ~ 0, y2 ~ x, y3 ~ x), data=dtf, MLEonly=TRUE))
Index

*Topic distribution
  rmultinomial, 11

*Topic models
  Multinomial Multivariate-T Regression, 2
  Multinomial Regression, 3
  Multinomial Regression Tanh Estimator, 6
  Multinomial Regression Tanh Estimator Gauss-Newton Optimization, 8
  Robust Multinomial Regression, 12

*Topic multivariate
  rmultinomial, 11

*Topic optimize
  Multinomial Regression Tanh Estimator Gauss-Newton Optimization, 8

*Topic regression
  Multinomial Multivariate-T Regression, 2
  Multinomial Regression, 3
  Multinomial Regression Tanh Estimator, 6
  Multinomial Regression Tanh Estimator Gauss-Newton Optimization, 8
  Robust Multinomial Regression, 12

*Topic robust
  Multinomial Regression Tanh Estimator, 6
  Multinomial Regression Tanh Estimator Gauss-Newton Optimization, 8
  Robust Multinomial Regression, 12

match.call, 3
mGNTanh (Multinomial Regression Tanh Estimator Gauss-Newton Optimization), 8

Multinomial Multivariate-T Regression, 2
Multinomial Regression, 3
Multinomial Regression Tanh Estimator, 6
Multinomial Regression Tanh Estimator Gauss-Newton Optimization, 8
multinomMLE, 15
multinomMLE (Multinomial Regression), 3
multinomRob (Robust Multinomial Regression), 12
multinomT, 15
multinomT (Multinomial Multivariate-T Regression), 2
multinomTanh (Multinomial Regression Tanh Estimator), 6
optim, 3
rmultinomial, 11
Robust Multinomial Regression, 12