Package ‘multinbmod’

February 20, 2015

Version 1.0
Date 2012-01-14
Title Regression analysis of overdispersed correlated count data
Author Ivonne Solis-Trapala
Maintainer Ivonne Solis-Trapala
        <ivonne.solis-trapala@mrc-hnr.cam.ac.uk>
Description This is a likelihood approach for the regression analysis of overdispersed correlated count data with cluster varying covariates. The approach fits a multivariate negative binomial model by maximum likelihood and provides robust estimates of the regression coefficients.
License GPL-2
NeedsCompilation no
Repository CRAN
Date/Publication 2014-01-16 15:41:52

R topics documented:

  multinbmod-package ............................................ 1
  multinb.fit ..................................................... 2
  multinbmod ..................................................... 4
  summary.multinbmod ........................................... 5

Index 7

   multinbmod-package  Regression analysis of overdispersed correlated count data

Description

This is a robust likelihood approach for the regression analysis of overdispersed correlated counts data with cluster varying covariates. The approach fits a multivariate negative binomial model by maximum likelihood and provides robust estimates of the regression coefficients.

Details
Use function multinbmod to fit a multivariate negative binomial model by maximum likelihood. Robust estimates of regression parameters are provided.

Author(s)

Ivonne Solis-Trapala

Maintainer: Ivonne Solis-Trapala <i.solis-trapala@mrc-hnr.cam.ac.uk>

References


Examples

```r
id <- factor(rep(1:20L, rep(5, 20))
y <- rbinom(100L, mu = rexp(100L, 1)+rep(rexp(20L), rep(5, 20)), size=2.5)
x <- rbinom(100L, 1, .5)
dat <- data.frame(y = y, x = x, id = id)
multinbmod(y~x, data=dat, id=id)
summary(multinbmod(y~x, data=dat, id=id))
```

Description

This function is called by "multinbmod", but it can also be called directly

Usage

`multinb.fit(y, x, offset=1L, id, start.par, control=list())`
Arguments

y  Response vector.
x  Design matrix of covariates.
offset  Optional vector of offset values.
id  Variable indicating which subjects are correlated.
start.par  Vector of starting values for the parameters in the linear predictor (defaults to zero) and the overdispersion parameter (default to 0.5).
control  A list of parameters that control the convergence criteria. See "nlminb" for details.

Value

The return values is a list with components:

estimated regression coefficients

se from model  Estimated standard errors of regression coefficients.
robust se  Robust estimate of standard errors of regression coefficients.
t-values  Robust t-values.
covariance of beta estimates from model  Estimated covariance of estimated regression parameters.
robust covariance of beta estimates  Robust estimate of covariance of estimated regression coefficients
estimated phi  ML estimate of overdispersion parameter.
se(phi)  Its standard error.
-2 x log-likelihood

converged?  Logical.
iterations  Number of iterations required for convergence.

Author(s)

Ivonne Solis-Trapala

References


See Also

multinbmod
Examples

```r
id <- factor(rep(1:20L, rep(5, 20)))
y <- rnbinom(100L, mu = rexp(100L)+rep(rexp(20L,,3),rep(5,20))Lsize=2.5)
x<--rbinom(100L,1,.5)
dat <- data.frame(y = y, x = x, id = id)
multinb.fit(y,cbind(1,x),id=id)
```

multinbmod  
Regression analysis of overdispersed correlated count data

Description

This function fits a multivariate negative binomial model by Maximum Likelihood and calculates robust standard errors of the regression coefficients.

Usage

```r
multinbmod(formula, data, id, offset, start.coef = NULL, start.phi = NULL,control=list())
```

Arguments

- `formula`: A symbolic description of the model to be fit.
- `data`: An optional 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 "multinbfit" is called.
- `id`: A vector which identifies correlated subjects. The length of "id" should be the same as the number of observations. Data are assumed to be sorted so that observations on a cluster are contiguous rows for all entities in the formula.
- `offset`: Optional vector of offset values.
- `start.coef`: Vector of starting values for the parameters in the linear predictor. Defaults are set to zero.
- `start.phi`: Overdispersion parameter. This value must be positive. Default is set to 0.5.
- `control`: A list of parameters that control the convergence criteria. See "nlminb" for details.

Details

The marginal distribution of the \( j \)-th observation from a cluster \( i \) is assumed to be Negative Binomial with mean \( \mu_{ij} \) and variance \( \mu_{ij} + \phi \cdot \mu_{ij}^2 \). The covariance of two observations is \( \phi \) times the product of their means. The function provides robust estimates of the regression parameters.
**Value**

The return values is a list, an object of class "multinbfit". The components are:

- `converged` Logical.
- `coefficients` Estimated regression coefficients.
- `model.coef.se` Their standard errors.
- `robust.coef.se` Robust estimates of standard errors.
- `robust.t.values` Robust t-values.
- `mle.phi` Estimated overdispersion parameter.
- `phi.se` Its standard error.
- `minus2.loglik` \(-2 \times \text{log-likelihood}\).
- `call` The function call.

**Author(s)**

Ivonne Solis-Trapala

**References**


**Examples**

```r
id <- factor(rep(1:20, rep(5, 20)))
y <- rbinom(100, mu = rexp(100,1)+rep(rexp(20,.3),rep(20,10)),size=2.5)
x <- rbinom(100,1,.5)
dat <- data.frame(y = y, x = x, id = id)
multinbmod(y~x,data=dat,id=id)
summary(multinbmod(y~x,data=dat,id=id,control=list(iter.max=100)))
```

**Description**

It displays the output of multinbmod

**Usage**

```r
## S3 method for class 'multinbmod'
summary(object,...)
```
Arguments

object The multinbmod object.

... Any other arguments

Value

A short summary of the object is printed

Author(s)

Ivonne Solis-Trapala

See Also

multinbmod

Examples

id <- factor(rep(1:20, rep(5, 20)))
y <- rnbinom(100, mu = rexp(100,1)+rep(rexp(20,.3),rep(5,20)),size=2.5)
x<-rbinom(100,1,.5)
dat <- data.frame(y = y, x = x, id = id)
summary(myfit<-multinbmod(y~x,data=dat,id=id,control=list(iter.max=100)))
Index

*Topic **Clustered**
  multinbmod, 4

*Topic **count data**
  multinbmod, 4

*Topic **multivariate negative binomial model, overdispersed correlated count data**
  multinbmod-package, 1

*Topic **multivariate**
  multinbmod, 4

*Topic **negative binomial**
  multinbmod, 4

  multinb.fit, 2
  multinbmod, 4
  multinbmod-package, 1

  summary (summary.multinbmod), 5
  summary.multinbmod, 5