Package ‘bnormnlr’

February 19, 2015

Type Package

Title Bayesian Estimation for Normal Heteroscedastic Nonlinear Regression Models

Version 1.0

Date 2014-12-08

Author Nicolas Molano-Gonzalez, Marta Corrales Bossio, Maria Fernanda Zarate, Edilberto Cepeda-Cuervo.

Maintainer Nicolas Molano-Gonzalez <nmolanog@unal.edu.co>


License GPL-2

Depends mvtnorm, numDeriv

Suggests car, coda, MASS

NeedsCompilation no

Repository CRAN

Date/Publication 2014-12-10 16:26:13

R topics documented:

bnormnlr-package ....................................................... 2
bnlr ................................................................. 3
chainsum ............................................................ 5
infocrit ............................................................... 7

Index 9
bnormnlr-package

Bayesian Estimation for Normal Heteroscedastic Nonlinear Regression Models

Description


Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>bnormnlr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.0</td>
</tr>
<tr>
<td>Date:</td>
<td>2014-12-09</td>
</tr>
<tr>
<td>License:</td>
<td>GPL-2</td>
</tr>
</tbody>
</table>

The package provides three functions: bnlr to perform Bayesian estimation for heteroscedastic normal nonlinear regression models; chainsum to summarize the MCMC chains obtained from bnlr and infocrit to extract information criteria measures from the model fit.

Author(s)

Nicolas Molano-Gonzalez, Marta Corrales Bossio, Maria Fernanda Zarate, Edilberto Cepeda-Cuervo. Maintainer: Nicolas Molano-Gonzalez <nmolanog@unal.edu.co>

References


Examples

```r
utils::data(muscle, package = "MASS")
###mean and variance functions
fmu<-function(param,cov){ param[1] + param[2]*exp(-cov/exp(param[3])))
fgsma<-function(param,cov){drop(exp(cov%*%param))}

###Note: use more MCMC chains (i.e NC=10000) for more accurate results.
mlb<bnlr(y=muscle$Length,f1=fmu,f2=fgsma,x=muscle$Conc,
z=cbind(1,muscle$Conc),bta0=c(20,-30,0),gma0=c(2,0),Nc=1200)
```
bnlr

Bayesian Estimation for Normal Heteroscedastic Nonlinear Regression Models.

Description


Usage

bnlr(y, f1, f2, f1g = NULL, f2g = NULL, x, z, bta0, gma0, b = rep(0, length(bta0)), B = diag(10^6, length(bta0)), g = rep(0, length(gma0)), G = diag(10^6, length(gma0)), Nc)

Arguments

y
A vector with the response variable.

f1
Non-linear function to specify the mean of the model. This function must have two arguments (param and cov) and must return a number in order to be accepted by the function. See details.

f2
Non-linear function to specify the variance of the model. This function must have two arguments (param and cov) and must return a positive number in order to be accepted by the function. See details.

f1g
A function which returns the gradient of the function f1 with respect to argument param. This function must have two arguments (param and cov) and must return a vector where each entry corresponds to the derivative of f1 with respect to param[i] evaluated at param.

f2g
A function which returns the gradient of the function f2 with respect to argument param. This function must have two arguments (param and cov) and must return a vector where each entry corresponds to the derivative of f2 with respect to param[i] evaluated at param.

x
Matrix of covariates associated with f1. This will be passed to f1 as argument cov. Thus f1(param0,x) should return a vector of length dim(x)[1] with the nonlinear function evaluated at each x[i,] for the parameter values param0.

z
Matrix of covariates associated with f2. This will be passed to f2 as argument cov. Thus f2(param0,z) should return a vector of length dim(z)[1] with the nonlinear function evaluated at each z[i,] for the parameter values param0.

bta0
Initial values for the parameters associated with f1. This vector will be passed to f1 together with x, so f1(bta0,x) should return a vector of length dim(x)[1] with the nonlinear function evaluated at each x[i,] for the parameter values bta0.
Initial values for the parameters associated with $f_2$. This vector will be passed to $f_2$ together with $z$, so $f_1(bta_0,x)$ should return a vector of length $\text{dim}(z)[1]$ with the nonlinear function evaluated at each $z[i]$, for the parameter values $gma0$.

Mean of the normal prior distribution of the mean parameters. Should have same length as $bta0$.

Covariance matrix of the normal prior distribution of the mean parameters.

Mean of the normal prior distribution of the variance parameters. Should have same length as $gma0$.

Covariance matrix of the normal prior distribution of the variance parameters.

Number MCMC simulations of the posterior distributions of the regression parameters given the data.

The matrices $x$ and $z$ should have the same number of rows as observations are in vector $y$. The functions $f_1$ and $f_2$ should be constructed in such a way that $f_1(bta0,x)$ and $f_2(gma0,z)$ returns a vector of the same length of $y$. $f_1g$ and $f_2g$ should be constructed in such a way that $f_1g(bta0,x)$ and $f_2g(gma0,z)$ returns a matrix where each row corresponds to the gradient (with respect to param) evaluated at $bta0$ ($gma0$) given the covariate values of $x[i]$, $(z[i])$.

A list with the following objects:

- **chains**: A matrix where MCMC simulations of the posterior distributions of the regression parameters given the data are stored. Rows correspond to MCMC simulation and columns correspond to the regression parameters. Parameters associated with $f_1$ are denoted by $bta_i$, parameters associated with $f_2$ are denoted by $gma_i$. In this matrix also is stored the Deviance of each iteration.

- **accept.bta**: An integer presenting the number of samples accepted by the Metropolis-Hastings algorithm for the mean parameters.

- **accept.gma**: An integer presenting the number of samples accepted by the Metropolis-Hastings algorithm for the variance parameters.

- **y**: Response variable used in the fit.

- **x**: Covariates associated with the mean used in the fit.

- **z**: Covariates associated with the variance used in the fit.

- **f1**: Function used to model the mean.

- **f2**: Function used to model the variance.

**Author(s)**

Nicolas Molano-Gonzalez, Marta Corrales Bossio, Maria Fernanda Zarate, Edilberto Cepeda-Cuervo.
References


Examples

```r
### Example of heteroscedastic model, using gradient
library(car)
library(coda)
utils::data(muscle, package = "MASS")

# Mean and variance functions
fsigma <- function(param, cov) { drop(exp(exp(cov%*%param)))}

# Simulate heteroscedastic data
muscle$Length <- fmu(c(28.9632978, -34.2274097, -0.4972977), muscle$Conc) +
   rnorm(60, 0, sqrt(exp(log(2) + .8 * muscle$Conc)))

g < - function(param, cov) {
  cbind(1, exp(-cov/exp(param[3])), param[2] * exp(-cov/exp(param[3])) * cov/exp(param[3]))
}

# Gradients
fmu.g <- function(param, cov) { cbind(1, exp(-cov/exp(param[3])), param[2] * exp(-cov/exp(param[3])) * cov/exp(param[3]))}
fsigma.g <- function(param, cov) { cbind(drop(exp(exp(cov%*%param))), drop(exp(exp(cov%*%param)) * cov[, 2]))}

# Without gradient
mlb <- bnlr(y=muscle$Length, f1=fmu, f2=fsigma, x=muscle$Conc, z=cbind(1, muscle$Conc),
            bta0=c(20, -30, 0), gma0=c(.5, .5), NC=500)

# With gradient
m2b <- bnlr(y=muscle$Length, f1=fmu, f2=fsigma, x=muscle$Conc, z=cbind(1, muscle$Conc),
            bta0=c(20, -30, 0), gma0=c(.5, .5), NC=500)

chainsum(mlb$chains, burn=1:50)
chainsum(m2b$chains, burn=1:50)
infocrit(mlb, 1:50)
infocrit(m2b, 1:50)

# Note: use more MCMC chains (i.e. NC=10000) for more accurate results.
```

**chainsum**  
*Function to summarize chain results of MCMC simulation obtained from bulr*
Description

This function reports mean and desired quantiles of the samples obtained via Gibbs sampler of the posterior distribution of the parameters.

Usage

chainsum(chains, q = c(0.025, 0.5, 0.975), burn = NULL)

Arguments

- **chains**: A matrix where mcmc simulations of the posterior distributions of the regression parameters given the data are stored. Rows correspond to mcmc simulation and columns correspond to the regression parameters. Parameters associated with f1 are denoted by bta$i (i=1,2,...), parameters associated with f2 are denoted by gma$i (i=1,2,...). In this matrix also is stored the Deviance of each iteration.

- **q**: Vector of desired quantiles.

- **burn**: A vector indicating which samples must be discarded from the mcmc simulation.

Details

This function can accept any kind of matrix but is highly recommended to pass only the matrix produced by bnlr, in order to avoid misses.

Value

A matrix with summary statistics of the chains.

Author(s)

Nicolas Molano-Gonzalez, Marta Corrales Bossio, Maria Fernanda Zarate, Edilberto Cepeda-Cuervo.

References


Examples

```r
utils::data(muscle, package = "MASS")
plot(muscle$Conc, muscle$Length)

###mean and variance functions
fmuc<-function(param,cov){ param[1] + param[2]*exp(-cov/exp(param[3]))}
fsgma<-function(param,cov){drop(exp(cov%*%param))}

###Note: use more MCMC chains (i.e NC=10000) for more accurate results.
mlb<-bnlr(y=muscle$Length,f1=fmu,f2=fsgma,x=muscle$Conc,
```
Infocrit

Infocrit calculates the Expected Number of Parameters, DIC, AIC, and BIC for a bnlr fit.

**Description**

Function to calculate Expected Number of Parameters, DIC, AIC, and BIC for bnlr output.

**Usage**

infocrit(model, burn)

**Arguments**

- `model`: A list derived from bnlr function
- `burn`: A vector indicating which samples must be discarded from the mcmc simulation

**Value**

A vector with:

- `pd`: Expected Number of Parameters
- `dic`: Deviance Information Criterion
- `aic`: Akaike Information Criterion
- `bic`: Bayesian Information Criterion

**Author(s)**

Nicolas Molano-Gonzalez, Marta Corrales Bossio, Maria Fernanda Zarate, Edilberto Cepeda-Cuervo.

**References**


Examples

```
# Simulation of heteroscedastic model
utils::data(muscle, package = "MASS")

# mean and variance functions
fsgma <- function(param, cov) {drop(exp(cov * %*% param))}

# simulate heteroscedastic data
muscleLength <- fmu(c(28.9632978, -34.2274097, -0.4972977), muscle$Conc) +
    rnorm(60, 0, sqrt(exp(log(2) + 0.8 * muscle$Conc)))

# Note: use more MCMC chains (i.e. NC=10000) for more accurate results.
m2b <- bnlr(y=muscle$Length, f1=fmu, f2=fsgma, x=muscle$Conc,
    z=matrix(rep(1, length(muscle$Length)), ncol=1), bta0=c(20, -30, -1), gma0=2, NC=650)
m1b <- bnlr(y=muscle$Length, f1=fmu, f2=fsgma, x=muscle$Conc, z=cbind(1, muscle$Conc),
    bta0=c(20, -30, 0), gma0=c(.5, .5), NC=650)

chainsum(m1b$chains, burn=1:65)
chainsum(m2b$chains, burn=1:65)
infocrit(m1b, 1:65)
infocrit(m2b, 1:65)
```
Index

*Topic package
  bnormnlr-package, 2

bnlr, 3
bnormnlr (bnormnlr-package), 2
bnormnlr-package, 2

chainsum, 5

infocrit, 7