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>

Description Implementation of Bayesian estimation in normal heteroscedastic nonlinear regression Models following Cepeda-Cuervo, (2001).

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 *Bayesian Estimation for Normal Heteroscedastic Nonlinear Regression Models*

Description

Implementation of Bayesian estimation in normal heteroscedastic nonlinear regression Models following Cepeda-Cuervo, (2001).

Details

Package:	bnormnlr
Type:	Package
Version:	1.0
Date:	2014-12-09
License:	GPL-2

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

Cepeda-Cuervo, E. (2001). Modelagem da variabilidade em modelos lineares generalizados. Unpublished Ph.D. tesis. Instituto de Matematicas. Universidade Federal do Rio do Janeiro.

Cepeda-Cuervo, E. and Gamerman, D. (2001). Bayesian modeling of variance heterogeneity in normal regression models. Brazilian Journal of Probability and Statistics 14.1: 207-221.

Cepeda-Cuervo, E. and Achcar, J.A. (2010). Heteroscedastic nonlinear regression models. Communications in Statistics-Simulation and Computation 39.2 : 405-419.

Examples

```
utils::data(muscle, package = "MASS")
###mean and variance functions
fmu<-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.
m1b<-bnlr(y=muscle$Length,f1=fmu,f2=fsgma,x=muscle$Conc,
z=cbind(1,muscle$Conc),bta0=c(20,-30,0),gma0=c(2,0),Nc=1200)
```

```
chainsum(m1b$chains,burn=1:200)
infocrit(m1b,1:8000)
```

---

bnlr *Bayesian Estimation for Normal Heteroscedastic Nonlinear Regression Models.*

Description

Implementation of Bayesian estimation in Heteroscedastic Nonlinear Regression Models following Cepeda-Cuervo, (2001).

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.

gma0	Initial values for the parameters associated with f2. This vector will be passed to f2 together with z, so f1(bta0,x) should return a vector of length dim(z)[1] with the nonlinear function evaluated at each z[i,] for the parameter values gma0.
b	Mean of the normal prior distribution of the mean parameters. Should have same length as bta0.
B	Covariance matrix of the normal prior distribution of the mean parameters.
g	Mean of the normal prior distribution of the variance parameters. Should have same length as gma0.
G	Covariance matrix of the normal prior distribution of the variance parameters.
Nc	Number mcmc simulations of the posterior distributions of the regression parameters given the data.

Details

The matrices x and z should have the same number of rows as observations are in vector y. The functions f1 and f2 should be constructed in such a way that f1(bta0,x) and f2(gma0,z) returns a vector of the same length of y. f1g and f2g should be constructed in such a way that f1g(bta0,x) and f2g(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,]).

Value

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 f1 are denoted by btai, parameters associated with f2 are denoted by gmai. 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	Fucntion used to model the mean.
f2	Fucntion used to model the variance.

Author(s)

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

References

Cepeda-Cuervo, E. (2001). Modelagem da variabilidade em modelos lineares generalizados. Unpublished Ph.D. tesis. Instituto de Matematicas. Universidade Federal do Rio do Janeiro.

Cepeda-Cuervo, E. and Gamerman, D. (2001). Bayesian modeling of variance heterogeneity in normal regression models. Brazilian Journal of Probability and Statistics 14.1: 207-221.

Cepeda-Cuervo, E. and Achcar, J.A. (2010). Heteroscedastic nonlinear regression models. Communications in Statistics-Simulation and Computation 39.2 : 405-419.

Examples

```
#######################################################
###Simulation of heteroscedastic model, using gradient
#######################################################
library(car)
library(coda)
utils::data(muscle, package = "MASS")
###mean and variance functions
fmu<-function(param,cov){ param[1] + param[2]*exp(-cov/exp(param[3]))}
fsgma<-function(param,cov){drop(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)))

####gradients
fmug<-function(param,cov){
cbind(1,exp(-cov/exp(param[3])),param[2]*exp(-cov/exp(param[3]))*cov/exp(param[3]))}
fsgmag<-function(param,cov){ cbind(drop(exp(cov%*%param)),drop(exp(cov%*%param))*cov[,2])}

###without gradient
m1b<-bnlr(y=muscle$Length,f1=fmu,f2=fsgma,x=muscle$Conc,z=cbind(1,muscle$Conc)
,bta0=c(20,-30,0),gma0=c(.5,.5),Nc=500)
###with gradient
m2b<-bnlr(y=muscle$Length,f1=fmu,f2=fsgma,x=muscle$Conc,z=cbind(1,muscle$Conc),
bta0=c(20,-30,0),gma0=c(.5,.5),Nc=500)

chainsum(m1b$chains,burn=1:50)
chainsum(m2b$chains,burn=1:50)
infocrit(m1b,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 bnlr*

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 btai (i=1,2,...), parameters associated with f2 are denoted by gmai (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 missuses.

Value

A matrix with summary statistics of the chains.

Author(s)

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

References

Carlin, B. P. & Louis, T. A. (2009), Bayesian Methods for Data Analysis, 3rd edn, CRC Press, New York.

Gamerman, D. & Lopes, H. F. (2006), Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, 2nd edn, CRC Press, New York.

Examples

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

###mean and variance functions
fmu<-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.
m1b<-bnlr(y=muscle$Length,f1=fmu,f2=fsgma,x=muscle$Conc,
```

```
z=cbind(1,muscle$Conc),bta0=c(20,-30,0),gma0=c(.5,.5),Nc=1200)
chainsum(m1b$chains,burn=1:200)
```

`infocrit` *Expected Number of Parameters, DIC, AIC and BIC for 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

Carlin, B. P. & Louis, T. A. (2009), Bayesian Methods for Data Analysis, 3rd edn, CRC Press, New York.

Gamerman, D. & Lopes, H. F. (2006), Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, 2nd edn, CRC Press, New York.

Examples

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

##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