Package ‘cbq’

October 11, 2020

Title  Conditional Binary Quantile Models
Version  0.2.0.2
Author  Xiao Lu
Maintainer  Xiao LU <xiao.lu.research@gmail.com>
License  MIT + file LICENSE
Encoding  UTF-8
LazyData  true
Biarch  true
Depends  R (>= 3.4.0)
Imports  methods, Formula, Rcpp (>= 0.12.0), rstan (>= 2.18.1), rstantools (>= 2.0.0)
LinkingTo  BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)
SystemRequirements  GNU make
RoxygenNote  7.1.0
NeedsCompilation  yes
Repository  CRAN
Date/Publication  2020-10-11 09:10:02 UTC

R topics documented:

  cbq-package .......................................................... 2
  cbq ................................................................. 2
  coef.cbq .......................................................... 5
  dald ............................................................... 6
  inverse ............................................................ 6
  is.dichotomous .................................................... 7
Description

Bayesian estimation of conditional binary quantile models.

References


Description

The main function for running the conditional binary quantile model. The function returns a cbq cbq object that can be further investigated using standard functions such as plot, print, coef, and predict.
cbq

Usage

cbq(
    formula,
    data,
    q = NULL,
    vi = FALSE,
    nsim = 1000,
    grad_samples = 1,
    elbo_samples = 100,
    tol_rel_obj = 0.01,
    output_samples = 2000,
    burnin = NULL,
    thin = 1,
    CIsize = 0.95,
    nchain = 1,
    seeds = 12345,
    inverse_distr = FALSE,
    offset = 1e-20,
    mc_core = TRUE
)

Arguments

formula An object of class "Formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
data A data frame containing the variables in the model.
q The quantile value.
vi Indicating whether variational inference should be used instead of MCMC sampling procedure.
nsim The number of iterations.
grad_samples Passed to vb (positive integer), the number of samples for Monte Carlo estimate of gradients, defaulting to 1.
elbo_samples Passed to vb (positive integer), the number of samples for Monte Carlo estimate of ELBO (objective function), defaulting to 100. (ELBO stands for "the evidence lower bound").
tol_rel_obj Passed to vb (positive double), the convergence tolerance on the relative norm of the objective, defaulting to 0.01.
output_samples Passed to vb (positive integer), number of posterior samples to draw and save, defaults to 1000.
burnin The number of burnin iterations.
thin Thinning parameter.
CIsize The size of confidence interval.
nchain The number of parallel chains.
seeds Random seeds to replicate the results.
inverse_distr  If FALSE, the ALD will not be reversed. The default is FALSE.
offset  Offset values to enhance sampling stability. The default value is 1e-20.
mcore  Indicating whether the estimation will be run in multiple parallel chains. The default is TRUE.

Details
The model can be passed either as a combination of a formula and a data frame data, as in lm(). Convergence diagnostics can be performed using either print(object,"mcmc") or plot(object,"mcmc").

Value
A cbq object, which can be further analyzed with its associated plot.cbq, coef.cbq and print.cbq functions.
An object of class cbq contains the following elements:

Call  The matched call.
formula  Symbolic representation of the model.
q  The quantile value.
nsim  The number of MCMC iterations.
burnin  The number of burnin periods.
thin  Thinning.
seeds  Random seeds.
Csize  The size of confidence interval.
data  Data used.
x  Covaraites used.
y  The dependent variable.
xnames  Names of the covariates.
stanfit  Outputs from stan.
sampledf  A matrix of posterior samples.
summaryout  A summary based on posterior samples.
npars  Number of covariates.
ulbs  Lower and upper confidence bounds.
means  Estimates at the mean.
vi  Indicating whether variational inference has been performed.
output_samples  Sample outputs.
fixed_var  Variables estimated using fixed effects.
random_var  Variables estimated using random effects.
xq  Variables indicating the choice sets.
Author(s)
Xiao Lu

References

Examples

```r
# Simulate the data
x <- rnorm(50)
y <- ifelse(x > 0, 1, 0)
dat <- as.data.frame(cbind(y, x))

# Estimate the CBQ model
model <- cbq(y ~ x, dat, 0.5, nchain = 1, mc_core = FALSE)

# Show the results
print(model)
coef(model)
plot(model)
```

---

**coef.cbq**

**Extract CBQ Coefficients**

**Description**

Create a table of coefficient results from a cbq object.

**Usage**

```r
## S3 method for class 'cbq'
coef(object, ...)
```

**Arguments**

- **object**
  - A cbq object.

- **...**
  - Further arguments passed to or from other methods.

**Value**

A table of coefficients with their corresponding lower and upper bounds.
**dald**

*Probability density function of asymmetric Laplace distributions*

**Description**

*dald* calculates probability densities of asymmetric Laplace distributions.

**Usage**

```r
dald(x, mu, p, sigma)
```

**Arguments**

- **x**: Random variable.
- **mu**: Position parameter.
- **p**: Quantile.
- **sigma**: Scale parameter.

**Value**

probability density of \( x \).

---

**inverse**

*Inverse function*

**Description**

*inverse* generates inverse function of any given function.

**Usage**

```r
inverse(f, mu, p, sigma, lower = -10000, upper = 10000)
```

**Arguments**

- **f**: pald function
- **mu**: Position parameter.
- **p**: Quantile.
- **sigma**: Scale parameter.
- **lower**: Lower bound.
- **upper**: Upper bound.

**Value**

inversed pald
is.dichotomous

Check if a predictor is dichotomous, adopted from package circGLM

Description
Check if a predictor is dichotomous, adopted from package circGLM

Usage
is.dichotomous(x)

Arguments
x A character or numerical vector to be tested.

Value
A logical, TRUE if the x has dummy coding (0, 1), FALSE otherwise.

pald
Cumulative density function of asymmetric Laplace distributions

Description
pald calculates cumulative densities of asymmetric Laplace distributions.

Usage
pald(x, mu, p, sigma)

Arguments
x Random variable.
mu Position parameter.
p Quantile.
sigma Scale parameter.

Value
cumulative probability density of x.
plot.cbq  

Description

General plot function for cbq objects, which dispatches the chosen type of plotting to the corresponding function.

Usage

## S3 method for class 'cbq'
plot(x, type = "trace", ...)

Arguments

x A cbq object to be plotted.
type Character string giving the type of plotting. The options are "trace" for trace plots, "coef" for coefficient plots. The default is the traceplot.
... Additional arguments to be passed to subsequent plot functions.

Value

None.

plot_coef.cbq  

Description

Plot traceplots from a cbq object.

Usage

plot_coef.cbq(object, ...)

Arguments

object A cbq object.
... Additional parameters to be passed to the plot function.

Value

None.
plot_trace.cbq

Make traceplots for cbq

Description
Plot traceplots from a cbq object.

Usage
plot_trace.cbq(object, ...)

Arguments
object A cbq object.
... Additional parameters to be passed to the traceplot function.

Value
None.

predict.cbq
Predictions based on the fitted parameter values

Description
Create a vector of predictions from a cbq object.

Usage
## S3 method for class 'cbq'
predict(object, data, ci = 0.95, ...)

Arguments
object A cbq object.
data Data used for prediction.
ci Confidence interval. The default is 0.95.
... Further arguments passed to or from other methods.

Value
A vector of predictions.
print.cbq  

*Print cbq object*

**Description**

General print function for cbq objects, which dispatches the chosen type of printing to the corresponding function.

**Usage**

```r
## S3 method for class 'cbq'
print(x, type = "text", ...)
```

**Arguments**

- `x`: A cbq object to be printed.
- `type`: Character string giving the type of printing, such as "text", "mcmc", "coef".
- `...`: Additional arguments to be passed to print functions.

**Value**

None.

---

print_coef.cbq  

*Print cbq coefficients*

**Description**

Print cbq coefficients

**Usage**

```r
print_coef.cbq(object, digits = 3)
```

**Arguments**

- `object`: A cbq object.
- `digits`: Number of digits to display.

**Value**

None.
print_mcmc.cbq  

Print the mcmc results from a cbq object

Description

This prints a number of diagnostics about the results of a cbq objects

Usage

print_mcmc.cbq(object, ...)

Arguments

object  

A cbq object.

...  

Additional arguments to be passed to the print function.

Value

None.

print_text.cbq  

Print the main results from a cbq object.

Description

Print the main results from a cbq object.

Usage

print_text.cbq(object, digits = 3)

Arguments

object  

A cbq object.

digits  

Number of digits to display.
qald \hspace{1cm} \textit{Quantile function of asymmetric Laplace distributions}

**Description**

qald calculates quantiles values of asymmetric Laplace distributions.

**Usage**

\texttt{qald(y, mu, p, sigma)}

**Arguments**

- **y**: quantile value.
- **mu**: Position parameter.
- **p**: Quantile.
- **sigma**: Scale parameter.

**Value**

quantile value.

rald \hspace{1cm} \textit{Random number generator of asymmetric Laplace distributions}

**Description**

rald generates random numbers from asymmetric Laplace distributions.

**Usage**

\texttt{rald(n, mu, p, sigma)}

**Arguments**

- **n**: Number of random numbers to be generated.
- **mu**: Position parameter.
- **p**: Quantile.
- **sigma**: Scale parameter.

**Value**

random numbers.
Index

cbq, 2
cbq-package, 2
coef.cbq, 4, 5
dald, 6
inverse, 6
is.dichotomous, 7
pald, 7
plot.cbq, 4, 8
plot_coef.cbq, 8
plot_trace.cbq, 9
predict.cbq, 9
print.cbq, 4, 10
print_coef.cbq, 10
print_mcmc.cbq, 11
print_text.cbq, 11
qald, 12
rald, 12
vb, 3