Package ‘cbq’

January 9, 2020

Title Conditional Binary Quantile Models
Version 0.1.0.0
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)
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 6.1.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2020-01-09 11:20:02 UTC

R topics documented:
cbq-package .................................................. 2
cbq .......................................................... 2
coeff.cbq ..................................................... 5
dald .......................................................... 5
inverse ....................................................... 6
is.dichotomous ............................................. 6
pald .......................................................... 7
cbq-package

cbq: An R Package for Estimating Conditional Binary Quantile Models

Description

Bayesian estimation of conditional binary quantile models.

References


cbq

Fitting conditional binary quantile models

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.

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)
cbq

Arguments

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

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.
cbq

burnin  The number of burnin periods.
thin  Thinning.
seeds  Random seeds.
CIsize  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

# 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)

# Show the results
print(model)
coef(model)
plot(model)
**coef.cbq**

<table>
<thead>
<tr>
<th><strong>Extract CBQ Coefficients</strong></th>
</tr>
</thead>
</table>

**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**

<table>
<thead>
<tr>
<th><strong>Probability density function of asymmetric Laplace distributions</strong></th>
</tr>
</thead>
</table>

**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**

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**

```r
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**

*Plot cbq object*

**Description**

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

**Usage**

```r
## 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  
*Make coefficient plots for cbq*

**Description**

Plot traceplots from a cbq object.

**Usage**

```r
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**

```r
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**
```r
## 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**

```r
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**

*Quantile function of asymmetric Laplace distributions*

**Description**

qald calculates quantiles values of asymmetric Laplace distributions.

**Usage**

```
qald(y, mu, p, sigma)
```

**Arguments**

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

**Value**

quantile value.
rald

*Random number generator of asymmetric Laplace distributions*

**Description**

*rald* generates random numbers from asymmetric Laplace distributions.

**Usage**

```
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
coeff.cbq, 3, 5
dald, 5

inverse, 6
is.dichotomous, 6

pald, 7
plot.cbq, 3, 7
plot_coef.cbq, 8
plot_trace.cbq, 8
predict.cbq, 9
print.cbq, 3, 9
print_coef.cbq, 10
print_mcmc.cbq, 10
print_text.cbq, 11

qald, 11
rald, 12

vb, 3