Package 'bayesQRsurvey'

September 26, 2025

Title Bayesian quantile regression models for complex survey data analysis

Version 0.1.0

Description Bayesian quantile regression models for complex survey data analysis, both single and multiple-output are supported. To speed up the calculations, all algorithms are implemented in C++ using 'Rcpp', 'RcppArmadillo', and 'RcppEigen', and called from R.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

LinkingTo Rcpp,
RcppArmadillo,
RcppEigen

Imports Rcpp,
stats,
graphics,
methods,
plotly,
geometry,
pracma,
ggplot2,
rlang,
posterior

Suggests MASS,
knitr,
rmarkdown,
grDevices,
testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

SystemRequirements BLAS, LAPACK

URL https://github.com/torodriguezt/bayesQRsurvey

BugReports https://github.com/torodriguezt/bayesQRsurvey/issues

Contents

---

bqr.svy	*Bayesian quantile regression for complex survey data*

Description

`bqr.svy` implements Bayesian methods for estimating quantile regression models for complex survey data analysis regarding single (univariate) outputs. To improve computational efficiency, the Markov Chain Monte Carlo (MCMC) algorithms are implemented in C++.

Usage

```
bqr.svy(
  formula,
  weights = NULL,
  data = NULL,
  quantile = 0.5,
  method = c("ald", "score", "approximate"),
  prior = NULL,
  niter = 50000,
  burnin = 10000,
  thin = 1,
  verbose = TRUE
)
```

Arguments

formula	a symbolic description of the model to be fit.
weights	an optional numerical vector containing the survey weights. If NULL, equal weights are used.
data	an optional data frame containing the variables in the model.
quantile	numerical scalar or vector containing quantile(s) of interest (default=0.5).
method	one of `"ald"`, `"score"` and `"approximate"` (default=`"ald"`).
prior	a `bqr_prior` object of class "prior". If omitted, a vague prior is assumed (see [prior](#)).
niter	number of MCMC draws.
burnin	number of initial MCMC draws to be discarded.
thin	thinning parameter, i.e., keep every keepth draw (default=1).
verbose	logical flag indicating whether to print progress messages (default=TRUE).

Details

The function bqr.svy can estimate three types of models, depending on method specification.

- "ald" – asymmetric Laplace working likelihood
- "score" – score based working likelihood function
- "approximate" – pseudolikelihood function based on a Gaussian approximation

Value

An object of class "bqr.svy", containing:

beta	Posterior mean estimates of regression coefficients.
draws	Posterior draws from the MCMC sampler.
accept_rate	Average acceptance rate (if available).
quantile	The quantile(s) fitted.
prior	Prior specification used.
formula, terms, model	
	Model specification details.
runtime	Elapsed runtime in seconds.

References

Nascimento, M. L. & Gonçalves, K. C. M. (2024). Bayesian Quantile Regression Models for Complex Survey Data Under Informative Sampling. *Journal of Survey Statistics and Methodology*, 12(4), 1105–1130.

Examples

```
# Generate population data
set.seed(123)
N    <- 10000
x1_p <- runif(N, -1, 1)
x2_p <- runif(N, -1, 1)
y_p  <- 2 + 1.5 * x1_p - 0.8 * x2_p + rnorm(N)

# Generate sample data
n <- 500
z_aux <- rnorm(N, mean = 1 + y_p, sd=.5)
p_aux <- 1 / (1 + exp(2.5 - 0.5 * z_aux))
s_ind <- sample(1:N, n, replace = FALSE, prob = p_aux)
y_s   <- y_p[s_ind]
x1_s  <- x1_p[s_ind]
x2_s  <- x2_p[s_ind]
w     <- 1 / p_aux[s_ind]
data  <- data.frame(y = y_s, x1 = x1_s, x2 = x2_s, w = w)

# Basic usage with default method ('ald') and priors (vague)
fit1 <- bqr.svy(y ~ x1 + x2, weights = w, data = data)

# Specify informative priors
prior<- prior(
 beta_x_mean = c(2, 1.5, -0.8),
 beta_x_cov = diag(c(0.25, 0.25, 0.25))),
```

```
  sigma_shape = 1,
 sigma_rate = 1
)
fit2 <- bqr.svy(y ~ x1 + x2, weights = w, data = data, prior = prior)

# Specify different methods
fit_score  <- bqr.svy(y ~ x1 + x2, weights = w, data = data, method = "score")
fit_approx <- bqr.svy(y ~ x1 + x2, weights = w, data = data, method = "approximate")
```

---

mo.bqr.svy   *Multiple-Output Bayesian quantile regression for complex survey data*

Description

mo.bqr.svy implements a Bayesian approach to multiple-output quantile regression for complex survey data analysis. The method builds a quantile region based on a directional approach. To improve computational efficiency, an Expectation-Maximization (EM) algorithm is implemented instead of the usual Markov Chain Monte Carlo (MCMC).

Usage

```
mo.bqr.svy(
  formula,
  weights = NULL,
  data = NULL,
  quantile = 0.5,
  prior = NULL,
  U = NULL,
  gamma_U = NULL,
  n_dir = NULL,
  epsilon = 1e-06,
  max_iter = 1000,
  verbose = FALSE
)
```

Arguments

formula	a symbolic description of the model to be fit.
weights	an optional numerical vector containing the survey weights. If NULL, equal weights are used.
data	an optional data frame containing the variables in the model.
quantile	numerical scalar or vector containing quantile(s) of interest (default=0.5).
prior	a bqr_prior object of class "prior". If omitted, a vague prior is assumed (see prior).
U	an optional $d \times K$-matrix of directions, where d indicates the response variable dimension and K indicates indicates the number of directions.
gamma_U	an optional list with length equal to K for which each element corresponds to $d \times (d-1)$-matrix of ortoghonal basis for each row of U.

n_dir	numerical scalar corresponding to the number of directions (if U and gamma_U are not supplied).
epsilon	numerical scalar indicating the convergence tolerance for the EM algorithm (default = 1e-6).
max_iter	numerical scalar indicating maximum number of EM iterations (default = 1000).
verbose	logical flag indicating whether to print progress messages (default=FALSE).

Value

An object of class "mo.bqr.svy" containing:

call	The matched call
formula	The model formula
terms	The terms object
quantile	Vector of fitted quantiles
prior	List of priors used for each quantile
fit	List of fitted results for each quantile, each containing one sub-list per direction
coefficients	Coefficients from the first quantile
n_dir	Number of directions
U	Matrix of projection directions ($d \times K$)
Gamma_list	List of orthogonal complement bases, one per direction
n_obs	Number of observations
n_vars	Number of covariates
response_dim	Dimension of the response d

Examples

```
library(MASS)

# Generate population data
set.seed(123)
N    <- 10000
data <- mvrnorm(N,rep(0,3),matrix(c(4,0,2,0,1,1.5,2,1.5,9),3,3))
x_p  <- as.matrix(data[,1])
y_p  <- data[,2:3]+cbind(rep(0,N),x_p)

# Generate sample data
n <- 500
z_aux <- rnorm(N, mean = 1 + y_p, sd=.5)
p_aux <- 1 / (1 + exp(2.5 - 0.5 * z_aux))
s_ind <- sample(1:N, n, replace = FALSE, prob = p_aux)
y_s <- y_p[s_ind,]
x_s <- x_p[s_ind,]
w    <- 1 / p_aux[s_ind]
data_s<- data.frame(y1 = y_s[,1], y2 = y_s[,2],  x1 = x_s, w = w)

# Basic usage with default priors when U and gamma_U are given
fit1 <- mo.bqr.svy(cbind(y1, y2) ~ x1, weights = w, data = data_s, quantile = c(0.1, 0.2),
 U = matrix(c(0,1,1/sqrt(2),1/sqrt(2)),2),
 gamma_U = list(c(1,0),c(1/sqrt(2),-1/sqrt(2))))
```

```
# Basic usage with default priors when n_dir is given
fit2 <- mo.bqr.svy(cbind(y1, y2) ~ x1, weights = w, data = data_s, quantile = c(0.1, 0.2), n_dir=2)
```

---

plot.bqr.svy   *Plot Method for Bayesian Weighted Quantile Regression (Survey)*

Description

Plot method for objects of class `bqr.svy` produced by `bqr.svy()`. It can display fitted quantile
curves, coefficient–quantile profiles, MCMC trace plots, and posterior densities.

Usage

```
## S3 method for class 'bqr.svy'
plot(
  x,
  y = NULL,
  type = c("fit", "quantile", "trace", "density"),
  predictor = NULL,
  tau = NULL,
  which = NULL,
  add_points = TRUE,
  combine = TRUE,
  show_ci = FALSE,
  ci_probs = c(0.1, 0.9),
  at = NULL,
  grid_length = 200,
  points_alpha = 0.4,
  point_size = 1.5,
  line_size = 1.2,
  main = NULL,
  use_ggplot = TRUE,
  theme_style = c("minimal", "classic", "bw", "light"),
  color_palette = c("viridis", "plasma", "set2", "dark2"),
  add_h0 = TRUE,
  add_ols = FALSE,
  ols_fit = NULL,
  ols_weights = NULL,
  ...
)

## S3 method for class 'bwqr_fit'
plot(x, ...)

## S3 method for class 'bwqr_fit_multi'
plot(x, ...)
```

Arguments

x   Object of class `bqr.svy`.

y	Ignored (S3 signature).
type	One of `"fit"`, `"quantile"`, `"trace"`, `"density"`.
predictor	(fit) Name of a numeric predictor; if `NULL`, the first numeric predictor (excluding the response) is used.
tau	Quantile(s) to plot; must appear in `x$quantile`. If `NULL`, all available are used.
which	(quantile/trace/density) Coefficient name or index to display.
add_points	(fit) Logical; overlay observed data points.
combine	(fit) Logical; if multiple `tau`: `TRUE` overlays curves in one panel; `FALSE` uses one panel per quantile.
show_ci	(fit) Logical; draw credible bands.
ci_probs	(fit) Length-2 numeric vector with lower/upper probabilities for credible bands.
at	(fit) Named list of fixed values for non-`predictor` covariates (see Details).
grid_length	(fit) Integer; number of points in the predictor grid.
points_alpha	(fit) Point transparency in `[0,1]`.
point_size	(fit) Point size.
line_size	(fit/quantile) Line width for fitted/summary lines.
main	Optional main title.
use_ggplot	Logical; if `TRUE`, return a ggplot object.
theme_style	(ggplot) One of `"minimal"`, `"classic"`, `"bw"`, `"light"`.
color_palette	(ggplot) One of `"viridis"`, `"plasma"`, `"set2"`, `"dark2"`.
add_h0	(quantile) Logical; add a horizontal reference at $y = 0$.
add_ols	(quantile) Logical; add the OLS estimate (dotted line) for the selected coefficient.
ols_fit	(quantile) Optional precomputed `lm` object; if `NULL`, an `lm()` is fitted internally using `x$model` and `x$terms`.
ols_weights	(quantile) Optional numeric vector of weights when fitting OLS internally (length must match `nrow(x$model)`).
...	Accepted for compatibility; ignored by internal plotting code.

Details

Supported plot types:

- `type = "fit"`: Fitted quantile curves versus a single numeric predictor. Optionally overlay observed points and credible bands. Other covariates can be held fixed via `at`.

- `type = "quantile"`: A single coefficient as a function of the quantile τ. Optionally add a reference line at 0 and the corresponding OLS estimate.

- `type = "trace"`: MCMC trace for one selected coefficient at a chosen τ.

- `type = "density"`: Posterior density for one selected coefficient at a chosen τ.

Notes:

- `tau` must be included in `x$quantile`. If `NULL`, all available quantiles in the object are used.

- For `type = "fit"`, `predictor` must be a numeric column in the original model. If `NULL`, the first numeric predictor (different from the response) is chosen automatically.

- For type = "fit", at is a named list (list(var = value, ...)) used to fix other covariates while plotting versus predictor. Provide valid levels for factors.
- When use_ggplot = TRUE, a ggplot object is returned and the appearance is controlled by theme_style and color_palette. Otherwise, base graphics are used and the function returns invisible(NULL).

Value

invisible(NULL) for base R graphics, or a ggplot object if use_ggplot = TRUE.

Examples

```
data(mtcars)
fit <- bqr.svy(mpg ~ wt + hp + cyl, data = mtcars,
               quantile = c(0.25, 0.5, 0.75), method = "ald",
               niter = 20000, burnin = 10000, thin = 5)

plot(fit, type = "fit", predictor = "wt", show_ci = TRUE)
plot(fit, type = "quantile", which = "wt", add_h0 = TRUE, add_ols = TRUE)
plot(fit, type = "trace", which = "wt", tau = 0.5)
plot(fit, type = "density", which = "wt", tau = 0.5)
```

---

print.bayesQRsurvey   *Print methods for bayesQRsurvey model objects*

Description

print.bayesQRsurvey is an S3 method that prints the content of an S3 object of class bqr.svy or mo.bqr.svy to the console.

Usage

```
## S3 method for class 'bqr.svy'
print(x, digits = 3, ...)

## S3 method for class 'mo.bqr.svy'
print(x, digits = 3, ...)
```

---

prior   *Create prior for Bayesian quantile regression models for complex survey data*

Description

prior creates prior distributions for both single (bqr.svy) and multiple-output (mo.bqr.svy) Bayesian quantile regression models for complex survey data.

Usage

```
prior(
  beta_x_mean = NULL,
  beta_x_cov = NULL,
  sigma_shape = 0.001,
  sigma_rate = 0.001,
  beta_y_mean = NULL,
  beta_y_cov = NULL
)
```

Arguments

beta_x_mean	vector of prior means for the regression coefficients. (default = NULL).
beta_x_cov	prior covariance matrix for the regression coefficients. (default = NULL).
sigma_shape	shape parameter for inverse Gamma prior for σ^2. (default = 0.001).
sigma_rate	rate parameter for inverse Gamma prior for σ^2. (default = 0.001).
beta_y_mean	prior means for the coefficients related to the variables that emerge from the product between the orthogonal basis and the outputs (default = NULL).
beta_y_cov	prior covariance matrix for the coefficients related to the variables that emerge from the product between the orthogonal basis and the outputs. (default = NULL).

Details

The function `prior` builds prior distributions for the three methods implemented in the function `bqr.svy` and for the multiple-output quantile regression implemented in the function `mo.bqr.svy`. Every nonspecified prior parameter will get the default value.

- `method = "ald"` in function `bqr.svy` allow the specification of hyperparameters `beta_x_mean`, `beta_x_cov`, `sigma_shape`, and `sigma_rate`.

- `method = "score"` in function `bqr.svy` allow the specification of hyperparameters `beta_x_mean` and `beta_x_cov`.

- `method = "approximate"` in function `bqr.svy` allow the specification of hyperparameters `beta_x_mean` and `beta_x_cov`.

- In function `mo.bqr.svy`, the specification of hyperparameters `beta_x_mean`, `beta_x_cov`, `sigma_shape`, `sigma_rate`, `beta_y_mean`, and `beta_y_cov` are allowed.

Value

An object of class `"prior"`.

See Also

bqr.svy, mo.bqr.svy, summary

Examples

```
#Simulate data
set.seed(123)
n  <- 200
x1 <- rnorm(n, 0, 1)
```

```
x2 <- runif(n, -1, 1)
w  <- runif(n, 0.5, 2)   # survey weights

y1 <- 2 + 1.5*x1 - 0.8*x2 + rnorm(n, 0, 1)
y2 <- 1 + 0.5*x1 - 0.2*x2 + rnorm(n, 0, 1)

data <- data.frame(y1 = y1, y2 = y2, x1 = x1, x2 = x2, w = w)


# Define a general informative prior
prior_general <- prior(
  beta_x_mean = c(2, 1.5, -0.8),
  beta_x_cov  = diag(c(0.25, 0.25, 0.25)),
  sigma_shape = 3,
  sigma_rate  = 2,
  beta_y_mean = 1,
  beta_y_cov  = 0.25
)

#Estimate the model parameters with informative prior


fit_ald <- bqr.svy(y1 ~ x1 + x2, weights = w, data = data,
                   prior = prior_general, method = "ald")

fit_scr <- bqr.svy(y1 ~ x1 + x2, weights = w, data = data,
                   prior = prior_general, method = "score")

fit_apx <- bqr.svy(y1 ~ x1 + x2, weights = w, data = data,
                   prior = prior_general, method = "approximate")

# Multiple-output method
fit_mo <- mo.bqr.svy(cbind(y1, y2) ~ x1 + x2, weights = w,
                     data = data, prior = prior_general, n_dir = 10)
```

---

summary.bayesQRsurvey   *Summary methods for bayesQRsurvey*

Description

summary.bayesQRsurvey is an S3 method that summarizes the output of the bqr.svy or mo.bqr.svy function. For the bqr.svy the posterior mean, posterior credible interval and convergence diagnostics are calculated. For the mo.bqr.svy the iterations for convergence, the MAP and the direction are calculated.

Usage

```
## S3 method for class 'bqr.svy'
summary(object, probs = c(0.025, 0.975), digits = 2, ...)

## S3 method for class 'mo.bqr.svy'
summary(object, digits = 4, ...)
```

Arguments

object	An object of class `mo.bqr.svy`.
probs	Two-element numeric vector with credible interval probabilities. Default `c(0.025, 0.975)`.
digits	Integer; number of decimals used by printing helpers. Default 4.
...	Unused.

Value

An object of class `summary.bqr.svy` with one block per τ.

An object of class `summary.mo.bqr.svy`.

Index