Package ‘BSquare’

February 19, 2015

Type Package
Title Bayesian Simultaneous Quantile Regression
Version 1.1
Date 2013-05-04
Author Luke Smith & Brian Reich
Maintainer Luke Smith <lukebrawleysmith@gmail.com>
Depends quadprog, quantreg, VGAM
Suggests survival
Description This package models the quantile process as a function of predictors.
License GPL (>= 2)
NeedsCompilation yes
Repository CRAN
Date/Publication 2013-05-24 12:04:13

R topics documented:

dqreg ......................................................... 2
qreg ......................................................... 2
qreg_spline ................................................. 4
qr_plot ...................................................... 7

Index 8
dqreg

Plot

Description

Plots posterior density for parametric basis functions.

Usage

dqreg(object, y, X)

Arguments

object An object of class qreg.
y A vector of response values to evaluate the density.
X A vector of predictors.

qreg

Quantile regression with parametric basis functions.)

Description

MCMC code for the quantile regression model of Reich and Smith, 2013.

Usage

qreg(X,Y=NULL,Y_low=NULL,Y_high=NULL,status=NULL,
    L=4,base="Gaussian",varying_effect=NULL,
    tau=seq(0.05,0.95,0.05),
    burn=10000, iters=50000)

Arguments

X Matrix of predictors with the first column consisting of all ones and all other values between -1 and 1.
Y A vector of responses.
Y_low, Y_high Vectors of endpoints for interval-censored values.
status Censoring status taking values
0 if uncensored
1 if left-censored on the interval (-Inf,Y)
2 if right-censored on the interval (Y,Inf)
3 if censored on the interval (Y_low,Y_high).
L The number of basis functions in quantile function
The centering distribution which can take values "Gaussian", "t", "logistic", "gamma", "weibull", or "ALAP."

If varying_effect = j, then only the covariates in the first j columns of X have different effects on different quantile levels.

Vector of quantile levels for output.

Number of MCMC samples to discard as burn-in.

Number of MCMC samples to generate after the burn-in.

See http://www4.stat.ncsu.edu/~reich/QR/ for more detailed descriptions and examples.

Posterior samples of the quantile function.

Log pseudo-maximum likelihood statistic for model comparisons.

The example is used to illustrate the method. In practice MCMC chains should be longer.


See Also
dqreg qr_plot

Examples

#Continuous data example
#Load the air quality data
data(airquality)
ozone <- airquality[,1]
solar <- airquality[,2]

#Remove missing observations
missing <- is.na(ozone+solar)
ozone <- ozone[!missing]
solar <- solar[!missing]
solar_std<-1.8*(solar - min(solar))/(max(solar)-min(solar)) - 0.9

#Fit the model and plot results
X<-cbind(1,solar_std)
#use longer chains in practice
fit<qreg(X,ozone,L=4,base="gamma", iters = 1000, burn = 1000)
qr_plot(fit,2, main = "Solar Effect")

#Right-censored data example
library(survival)
data(veteran)
trt<-ifelse(veteran[,1]==2,-1,1)
logtime<-log(veteran[,3])
event<-veteran[,4]
status<-ifelse(event==1,0,2)
X<-cbind(1,trt)
#use longer chains in practice
fit<qreg(X,Y=logtime,status=status, iters =1000, burn = 1000)
qr_plot(fit,index=2,main="Treatment effect")

qreg_spline

Quantile regression with splines.

Description

Quantile regression using spline basis functions for the quantile process.

Usage

qreg_spline(X, Y = NULL, Y_low = NULL, Y_high = NULL, status = NULL, knots_inter = c(0.1, 0.5, 0.9), Pareto = TRUE, varying_effect = NULL, tau = seq(0.05, 0.95, 0.05), burn = 10000, iters = 50000, q_low = 0.01, q_high = 0.99, sig_a = .1, sig_b = .1, mu_var = 10^2, cbf_var = 10^3, tail_mean = -1.2, tail_var = .4, cbf_eps = 0.5, theta_eps = 0.5, tuning_theta = 1, tuning_tail = rep(1, 2), cred_low = 0.025, cred_high = 0.975, seed = 1, verbose = TRUE)
qreg_spline

Arguments

\(X\) Matrix of predictors with the first column consisting of all ones and all other values between -1 and 1.

\(Y\) A vector of responses.

\(Y_{\text{low}}, Y_{\text{high}}\) Vectors of endpoints for interval-censored values.

\(\text{status}\) Censoring status taking values
0 if uncensored
1 if left-censored on the interval (-Inf, Y)
2 if right-censored on the interval (Y, Inf)
3 if censored on the interval (Y_{\text{low}}, Y_{\text{high}}).

\(\text{knots}_{\text{inter}}\) An ordered sequence of numbers in (0,1) specifying knot locations.

\(\text{Pareto}\) An indicator for whether Pareto or exponential tails are fit, corresponding to heavy and light tailed densities respectively. Distributions that decay slowly in the tail (e.g. t-distribution) are better.

\(\text{varying}_{\text{effect}}\) If varying_effect = j, then only the covariates in the first j columns of X have different effects on different quantile levels.

\(\tau\) Vector of quantile levels for output.

\(\text{burn}\) Number of MCMC samples to discard as burn-in.

\(\text{iters}\) Number of MCMC samples to generate after the burn-in.

\(q_{\text{low}}\) The quantile level below which the lower parametric tail is fit.

\(q_{\text{high}}\) The quantile level above which the upper parametric tail is fit.

\(\text{sig}_a\) The shape hyperparameter for the prior precision of the basis functions.

\(\text{sig}_b\) The scale hyperparameter for the prior precision of the basis functions.

\(\mu_{\text{var}}\) The prior variance of the prior mean of the basis functions.

\(\text{cbf}_{\text{var}}\) The prior variance of the constant basis functions.

\(\text{tail}_{\text{mean}}\) The prior mean for the log of the shape parameters for Pareto tails. Only used if Pareto = TRUE.

\(\text{tail}_{\text{var}}\) The prior variance for the log of the shape parameters for Pareto tails. Only used if Pareto = TRUE.

\(\text{cbf}_{\text{eps}}\) A parameter in [.1,.5] indicating the degree of stochastic thinning for the location basis functions. Lower is stronger.

\(\text{theta}_{\text{eps}}\) A parameter in [.1,.5] indicating the degree of stochastic thinning for the other basis functions. Lower is stronger.

\(\text{tuning}_{\text{theta}}\) The initial candidate variance for the parameters updated by random walk Metropolis Hastings.

\(\text{tuning}_{\text{tail}}\) The initial candidate variance for the tail parameters.

\(\text{cred}_{\text{low}}\) The lower limit of the posterior credible intervals.

\(\text{cred}_{\text{high}}\) The upper limit of the posterior credible intervals.

\(\text{seed}\) MCMC seed.

\(\text{verbose}\) An indicator for outputting real-time MCMC updates.
Details

See http://www4.stat.ncsu.edu/~reich/QR/ for more detailed descriptions and examples.

Value

- **q**: A (iters x N_tau x P) array of posterior quantile effects
- **q_lower**: An (N_tau x P) array of lower limits of credible sets for posterior quantile effects
- **q_upper**: An (N_tau x P) array of upper limit of credible sets for posterior quantile effects
- **q_mean**: An (N_tau x P) array of posterior means for posterior quantile effects
- **theta**: An (iters x M x P) array of posterior parameters
- **tuning_parms**: An (M x P) array of candidate precisions
- **acc_theta**: An (M x P) array of acceptance ratios of theta for the keepers
- **post_mu**: An (N_tau x M x P) array of posterior mean hyperparameters
- **post_sigma2**: An (N_tau x M x P) array of posterior precision hyperparameters
- **post_rho**: An (N_tau x M x P) array of posterior correlation hyperparameters
- **post_xi_low**: An (iters x 1) array of posterior lower tail shape parameters
- **post_xi_high**: An (iters x 1) array of posterior upper tail shape parameters
- **tau**: A vector specifying the quantile levels of interest (includes the thresholds)
- **MCMC_time**: The MCMC run time.
- **LPML**: The log pseudo marginal likelihood.
- **iters**: The number of MCMC iterations kept.
- **burn**: The number of MCMC iterations burned.

Note

The example is used to illustrate the method. In practice MCMC chains should be longer. Code for modeling multiple quantile functions correlated across space or time or for running on graphics processing units is available from the authors on request.

References


See Also

qr_plot
Examples

data(airquality)
ozone=airquality[,1]
solar=airquality[,2]

#Remove missing observations
missing=is.na(ozone) | is.na(solar)
ozone=ozone[!missing]
solar=solar[!missing]

#Create design matrix. First column must be all ones, others must be between -1 and 1
solar_std = 1.8 * (solar - min(solar))/(max(solar)-min(solar)) - 0.9
X = cbind(1,solar_std)

tau=seq(0.05,0.95,0.05)
use longer chains in practice
fit=qreg_spline(X,Y=ozone,its=1000, burn = 1000, knots_inter = c(.5))

qr_plot(fit,index=2, main = "Solar Effect")

---

qr_plot  
Plot

Description

Plot of posterior distribution of effects

Usage

qr_plot(object, index = 1, 
xlab = "Quantile level", ylab = "Covariate effect", 
main = "", col = gray(0.75), lwd = 1, add = FALSE)

Arguments

<table>
<thead>
<tr>
<th>object</th>
<th>An object of class qreg or qreg_spline</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>Integer describing which predictor to plot</td>
</tr>
<tr>
<td>xlab</td>
<td>X axis label</td>
</tr>
<tr>
<td>ylab</td>
<td>Y axis label</td>
</tr>
<tr>
<td>main</td>
<td>plot title</td>
</tr>
<tr>
<td>col</td>
<td>plot color</td>
</tr>
<tr>
<td>lwd</td>
<td>line width</td>
</tr>
<tr>
<td>add</td>
<td>Superimposed plot</td>
</tr>
</tbody>
</table>
Index

dqreg, 2, 3
qr_plot, 3, 6, 7
qreg, 2
qreg_spline, 4