

Package ‘conquer’

August 27, 2020

Type Package

Title Convolution-Type Smoothed Quantile Regression

Version 1.0.2

Date 2020-08-25

Description Fast and accurate convolution-type smoothed quantile regression. Implemented using Barzilai-Borwein gradient descent with a Huber regression warm start. Construct confidence intervals for regression coefficients using multiplier bootstrap.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

URL <https://github.com/XiaoouPan/conquer>

SystemRequirements C++11

Imports Rcpp (>= 1.0.3), Matrix, matrixStats, stats

LinkingTo Rcpp, RcppArmadillo (>= 0.9.850.1.0)

RoxygenNote 7.1.1

NeedsCompilation yes

Author Xuming He [aut],
Xiaoou Pan [aut, cre],
Kean Ming Tan [aut],
Wen-Xin Zhou [aut]

Maintainer Xiaoou Pan <xip024@ucsd.edu>

Repository CRAN

Date/Publication 2020-08-27 05:10:24 UTC

R topics documented:

conquer-package	2
conquer	2

Index	5
--------------	----------

conquer-package

Conquer: Convolution-Type Smoothed Quantile Regression

Description

Fast and accurate convolution-type smoothed quantile regression. Implemented using Barzilai-Borwein gradient descent with a Huber regression warm start. Construct confidence intervals for regression coefficients using multiplier bootstrap.

Author(s)

Xuming He <xmhe@umich.edu>, Xiaoou Pan <xip024@ucsd.edu>, Kean Ming Tan <keanming@umich.edu>, and Wen-Xin Zhou <wez243@ucsd.edu>

References

- Barzilai, J. and Borwein, J. M. (1988). Two-point step size gradient methods. *IMA J. Numer. Anal.* 8 141–148.
- Fernandes, M., Guerre, E. and Horta, E. (2019). Smoothing quantile regressions. *J. Bus. Econ. Statist.*, in press.
- He, X., Pan, X., Tan, K. M., and Zhou, W.-X. (2020). Smoothed quantile regression for large-scale inference. Preprint.
- Horowitz, J. L. (1998). Bootstrap methods for median regression models. *Econometrica* 66 1327–1351.
- Koenker, R. (2005). *Quantile Regression*. Cambridge University Press, Cambridge.
- Koenker, R. Package "quantreg".
- Koenker, R. and Bassett, G. (1978). Regression quantiles. *Econometrica* 46 33-50.
- Pan, X. and Zhou, W.-X. (2020). Multiplier bootstrap for quantile regression: Non-asymptotic theory under random design. *Information and Inference: A Journal of the IMA*, to appear.

conquer

Convolution-Type Smoothed Quantile Regression

Description

Fit a smoothed quantile regression via convolution-type smoothing method. The solution is computed using gradient descent with Barzilai-Borwein step size. Constructs $(1-\alpha)$ confidence intervals with multiplier bootstrap.

Usage

```
conquer(
  X,
  Y,
  tau = 0.5,
  kernel = c("Gaussian", "uniform", "parabolic", "triangular"),
  h = 0,
  checkSing = FALSE,
  tol = 1e-04,
  iteMax = 5000,
  ci = FALSE,
  alpha = 0.05,
  B = 1000
)
```

Arguments

X	A n by p design matrix. Each row is a vector of observation with p covariates. Number of observations n must be greater than number of covariates p .
Y	An n -dimensional response vector.
tau	(optional) The desired quantile level. Default is 0.5. Value must be between 0 and 1.
kernel	(optional) A character string specifying the choice of kernel function. Default is "Gaussian". Other choices are "Gaussian", "uniform", "parabolic" or "triangular".
h	(optional) The bandwidth parameter for kernel smoothing. Default is $\max(((\log(n)+p)/n)^{0.4}, 0.05)$. The default will be used if the input value is less than 0.05.
checkSing	(optional) A logical flag. Default is FALSE. If checkSing = TRUE, then it will check if the design matrix is singular before running conquer.
tol	(optional) Tolerance level of the gradient descent algorithm. The gradient descent algorithm terminates when the maximal entry of the gradient is less than tol. Default is 1e-04.
iteMax	(optional) Maximum number of iterations. Default is 5000.
ci	(optional) A logical flag. Default is FALSE. If ci = TRUE, then three types of confidence intervals (percentile, pivotal and normal) will be constructed via multiplier bootstrap.
alpha	(optional) The nominal level for $(1-\alpha)$ -confidence intervals. Default is 0.05. The input value must be in $(0, 1)$.
B	(optional) The size of bootstrap samples. Default is 1000.

Value

An object containing the following items will be returned:

`coeff` A $(p + 1)$ -vector of estimated quantile regression coefficients, including the intercept.

ite The number of iterations of the gradient descent algorithm for convergence.
 residual The residuals of the quantile regression fit.
 bandwidth The value of smoothing bandwidth.
 tau The desired quantile level.
 kernel The choice of kernel function.
 n The sample size.
 p The dimension of the covariates.
 perCI The percentile confidence intervals for regression coefficients. Not available if ci = FALSE.
 pivCI The pivotal confidence intervals for regression coefficients. Not available if ci = FALSE
 normCI The normal-based confidence intervals for regression coefficients. Not available if ci = FALSE

Author(s)

Xuming He <xmhe@umich.edu>, Xiaou Pan <xip024@ucsd.edu>, Kean Ming Tan <keanming@umich.edu>, and Wen-Xin Zhou <wez243@ucsd.edu>

References

Barzilai, J. and Borwein, J. M. (1988). Two-point step size gradient methods. *IMA J. Numer. Anal.* 8 141–148.
 Fernandes, M., Guerre, E. and Horta, E. (2019). Smoothing quantile regressions. *J. Bus. Econ. Statist.*, in press.
 He, X., Pan, X., Tan, K. M., and Zhou, W.-X. (2020). Smoothed quantile regression for large-scale inference. Preprint.
 Koenker, R. and Bassett, G. (1978). Regression quantiles. *Econometrica* 46 33-50.

Examples

```
n = 500; p = 10
beta = rep(1, p)
X = matrix(rnorm(n * p), n, p)
Y = 1 + X %*% beta + rt(n, 2)

## Smoothed quantile regression with Gaussian kernel
fit.Gauss = conquer(X, Y, tau = 0.5, kernel = "Gaussian")
beta.hat.Gauss = fit.Gauss$coeff

## Smoothe quantile regression with uniform kernel
fit.unif = conquer(X, Y, tau = 0.5, kernel = "uniform")
beta.hat.unif = fit.unif$coeff

## Construct three types of confidence intervals via multiplier bootstrap
fit = conquer(X, Y, tau = 0.5, kernel = "Gaussian", ci = TRUE)
ci.per = fit$perCI
ci.piv = fit$pivCI
ci.norm = fit$normCI
```

Index

conquer, [2](#)

conquer-package, [2](#)