Package ‘GAQuantReg’

May 18, 2015

Type Package

Title Gloablly adaptive quantile regression

Version 1.0

Date 2015-05-11

Author Qi Zheng, Limin Peng and Xuming He

Maintainer Qi Zheng <zheng.qi85@gmail.com>

Description Globally adaptive quantile regression for model selection in ultra-high dimensional heterogeneous data

Suggests quantreg, MASS, flare

License GPL (>= 2)

Dependes quantreg, MASS, R(>=3.1.2)

R topics documented:

`GAQuantReg-package` *Globally adaptive quantile regression for model selection in high dimensional data*

Description

The packages provides

(1) Globally adaptive quantile regression at at multiple specified quantile levels: vecrqalasso();

(2) Globally adaptive quantile regression at a specified compact set: grqalasso().

(3) Tuning parameter selector for globally Lasso-type quantile estimator: lambdaCV().

(4) GIC type tuning parameter selector for globally adaptive Lasso type quantile estimator: lambdaGIC().

Details

Package:	GAQuantReg
Type:	Package
Version:	1.0
Date:	2015-05-12
License:	GPL (>= 2)

Author(s)

Qi Zheng, Limin Peng and Xuming He

Maintainer: Qi Zheng <zheng.qi85@gmail.com>

References

Zheng et al (2015, Annals of Statistics) Globally adaptive quantile regression with ultra-high dimensional data.

`grqalasso` *GIC type tuning parameter selector for globally adaptive quantile regression*

Description

This is a functon to select a tuning parameter for globally adaptive quantile quantile based on the GIC criterion proposed by Zheng, Peng and He (2015).

Usage

```
grqalasso(y,x,tau=c(0.1,0.9), method="BC", weights="unif",
               grid=seq(0, nrow(x)/20, length.out=nrow(x)/4), tol=1e-10)
```

Arguments

`y`	The response vector.
`x`	The covariates matrix.
`tau`	The quantile interval of interest. The default is [0.1, 0.9].
`method`	The method to obtain the tuning parameter for the high dimensional Lasso-type estimator. The default is "BC", which adopts the tuning parameter selector proposed by Belloni and Chernozhukov (2011). The other option is "CV5", which selects the tuning parameter by 5-fold Cross Validation.
`weights`	The type of adaptive weights. There are three different types: "unif", "absmean", and "pointwise". The default is "unif", which calculate the adaptive weights according to the infinity norm of quantile coefficient, and adaptive weights are the same at all given quantile levels. The "absmean" penalties measure the L1 norm of quantile coefficients. The third type pointwise" are the ordinary adaptive lasso penalties, and vary across taus.
`grid`	The GIC grid. The default is of n/4 equally spaced points from 0 to n/20.
`tol`	Tolerance, the default is 1e-10.

Details

The function first choose the quantile levels needed in the given region, and then it will call the function verrqalasso().

Value

The function returns a list containing

`taus`	all quantile levels considered in the given region
`estbeta`	the corresponding solutions at the given quantiles
`lambda.cv`	tuning parameter for the Lasso-type estimator
`lambda.gic`	tuning parameter for the globally adaptive quantile regression
`method`	the method used to select lambda.cv: either "BC" or "CV5"
`weights`	the adaptive weights used: "unif", "absmean" or "pointwise"

Note

If the option "CV5" is specified, the procedure may be time-consuming. (We are developing a faster CV version)

Author(s)

Qi Zheng (zheng.qi85@gmail.com)

References

Zheng et al (2015, Annals of Statistics) Globally adaptive quantile regression with ultra-high dimensional data.

Examples

```
library(MASS)
library(quantreg)

p=400;                                              ######## model dim
n=200;                                              ######## sample size

######## covariance matrix
Sigma=matrix(0,p,p);
pho=0.5;
J=seq(1,p,1);
for (i in 1:p){Sigma[i,]=pho^(abs(i-J))};

######## covariate matrix
Z=mvrnorm(n=n , rep(0,p), Sigma,  empirical = FALSE); ######## Generate covariates
X=cbind(rep(1,n),Z);                                  ######## Add intercept
########################################################

######## regression coefficients
beta=rep(0,p+1);
beta[2]=2;  beta[3]=1.5;   beta[6]=3; beta[9]=1;  beta[13]=0.9;  beta[17]=1;

######## errors and response variable
epsilon=rnorm(n,0,sqrt(2));                           quantile
```

```
Y=X%*%beta+epsilon;

object=grqalasso(Y,Z,tau=c(0.2,0.8),method="BC", weights="absmean")
```

`lambdaCV` *Tuning parameter selector for the globally Lasso type quantile regression based on 5-fold cross validation*

Description

This is a functon to select a tuning parameter for the Lasso type estimator based on the 5-fold cross-validation. The Lasso type estimator will be used to construct the adaptive penalities for the globally adaptive quantile quantile proposed by Zheng, Peng and He (2015).

Usage

```
lambdaCV(y, x, taus, K=5, len = 2 * sqrt(nrow(x)), tol=1e-10)
```

Arguments

`y`	The response vector.
`x`	The covariates matrix.
`taus`	The tau grid between 0 and 1.
`K`	The K-fold cross-validation, and the default is 5.
`len`	The number of grid points for Cross-Validation. The default is 2 * sqrt(nrow(x)).
`tol`	Tolerance, the default is 1e-10.

Details

1. Two-layer Cross-Validation search are included: (Step 1): the initial search is conducted on the grid { 1, 2, 4, . . . } until the estimated coeffs are small enough. (Step 2): the next search will conducted around two grid points with the smallest CV value from (Step 1). It can be seen as a refined CV search. The new grid is of legnth len.

Value

The function returns the selected tuning parameter.

Note

This procedure is time-consuming for ultra-high dimensional data, especially in step 1. (We are developing a faster CV version)

Author(s)

Qi Zheng (zheng.qi85@gmail.com)

Examples

```
library(MASS)
library(quantreg)

p=400;                                              ######## model dim
n=200;                                              ######## sample size

######## covariance matrix
Sigma=matrix(0,p,p);
pho=0.5;
J=seq(1,p,1);
for (i in 1:p){Sigma[i,]=pho^(abs(i-J))};

######## covariate matrix
Z=mvrnorm(n=n , rep(0,p), Sigma,  empirical = FALSE); ######## Generate covariates
Z=pnorm(Z);
for(i in 1:p){
        Z[,i]=Z[,i]/sqrt(sum(Z[,i]^2)/n);
}
X=cbind(rep(1,n),Z);

######## regression coefficients
alpha=rep(0,p+1);
alpha[2]=2;
alpha[3]=1.5;
gamma=rep(0,p+1);
gamma[9]=2;

######## errors and response variable
epsilon=rnorm(n);
Y=X%*%alpha+(X%*%gamma)*epsilon;

######## taus grid
tauL=0.3;
tauU=0.7;
J=seq(tauL,tauU,0.01);

lambda=lambdaCV(Y,X,J);
```

`lambdaGIC` *GIC type tuning parameter selector for globally adaptive quantile regression*

Description

This is a functon to select a tuning parameter for globally adaptive quantile quantile based on the GIC criterion proposed by Zheng, Peng and He (2015).

Usage

```
lambdaGIC(y, x, tau, w, grid=seq(0, nrow(x)/20, length.out=nrow(x)/4), tol=1e-10)
```

Arguments

`y`	The response vector.
`x`	The covariates matrix.
`tau`	The tau grid between 0 and 1.
`w`	The adaptive penalties.
`grid`	The GIC grid. The default is of n/4 equally spaced points from 0 to n/20.
`tol`	Tolerance, the default is 1e-10.

Details

The GIC is calculated at the given tau grid. If tau is a single quantile, it reduces to the GIC proposed by Fan and Tang (2013); otherwise, If tau is a vector of quantile, then the globally GIC (Zheng et al, 2015) will be implmented at the specified quantile levels.

Value

The function returns the selected tuning parameter for globally adaptive quantile regression.

Note

This procedure requires standardized response and covariates.

Author(s)

Qi Zheng (zheng.qi85@gmail.com)

References

Fan, Y, and Tang, C. (2013, Journal of Royal Statistical Society, B.) Tuning parameter slection in high dimensional penalized likelihood.

Zheng et al (2015, Annals of Statistics) Globally adaptive quantile regression with ultra-high dimensional data.

Examples

```
library(quantreg)
library(flare)
data(eyedata)

######## data information
p=dim(x)[2];
n=length(y);

######## standardize data
meany=mean(y);
Y=y-meany;
normy=sqrt(mean(Y^2));
Y=Y/normy;

meanx=apply(x,2,mean);
X=x-outer(rep(1,dim(x)[1]),meanx);
normx=sqrt(apply(X^2,2,mean));
X=sweep(X,2,1/normx,"*");
```

```
X=cbind(rep(1,n),X);

######## quantile interval of interest
tauL=0.2;
tauU=0.8;
taus=seq(tauL,tauU,0.005);
tol=1e-10;

######## Lasso type estimator (initial estimator)
lambda=12;
lambda1=c(0,rep(lambda,p))
fitb=coef(rq(Y~0+X,method="lasso",tau=taus,lambda=lambda1));
Estimate=t(fitb*(abs(fitb)>=tol));

######## unif weights
Pilot=1/abs(Estimate);
Pilot[which(Pilot==Inf)]=1e+50;
unifwt=apply(Pilot,2,min);

lambda.gic=lambdaGIC(Y,X,tau=taus,w=unifwt);
```

`vecrqalasso` *Globally adaptive quantile regression at a given quantile vector.*

Description

This is a functon to conduct globally adaptive quantile regression at a given quantile vector based on the proposed method in Zheng, Peng and He (2015).

Usage

```
vecrqalasso(y,x,taus, method="BC", weights="unif",
                grid=seq(0,nrow(x)/20, length.out=nrow(X)/4), tol=1e-6)
```

Arguments

`y`	The response vector.
`x`	The covariates matrix.
`taus`	The tau grid between 0 and 1.
`method`	The method to obtain the tuning parameter for the high dimensional Lasso-type estimator. The default is "BC", which adopts the tuning parameter selector proposed by Belloni and Chernozhukov (2011). The other option is "CV5", which selects the tuning parameter by 5-fold Cross Validation.
`weights`	The type of adaptive weights. There are three different types: "unif", "absmean", and "pointwise". The default is "unif", which calculate the adaptive weights according to the infinity norm of quantile coefficient, and adaptive weights are the same at all given quantile levels. The "absmean" penalties measure the weighted L1 norm of quantile coefficients. The third type pointwise" are the ordinary adaptive lasso penalties, and vary across taus.
`grid`	The grid for GIC. The default is of n/4 equally spaced points from 0 to n/20.
`tol`	Tolerance, the default is 1e-10.

Details

1. The data will be standardized.

2. On the first stage, it implements a global Lasso-type quantile regression to obtain the initial estimator, from which the adaptive Lasso penalties are calculated. The tuning parameter can be

3. On the second stage, the adaptive Lasso quantile regression will be applied at the quantile vector with a uniform tuning parameter selected by a global GIC type selector.

Value

The function returns a list containing

`taus`	all quantile levels provided
`estbeta`	the corresponding solutions at the given quantiles
`lambda.cv`	tuning parameter for the Lasso-type estimator
`lambda.gic`	tuning parameter for the globally adaptive quantile regression
`method`	the method used to select lambda.cv: either "BC" or "CV5"
`weights`	the adaptive weights used: "unif", "absmean" or "pointwise"

Note

If the option "CV5" is specified, the procedure may be time-consuming. (We are developing a faster CV version).

Author(s)

Qi Zheng (zheng.qi85@gmail.com)

References

Zheng et al (2015, Annals of Statistics) Globally adaptive quantile regression with ultra-high dimensional data.

Examples

```
library(MASS)
library(quantreg)

p=400;                                            ######## model dim
n=200;                                            ######## sample size

######## covariance matrix
Sigma=matrix(0,p,p);
pho=0.5;
J=seq(1,p,1);
for (i in 1:p){Sigma[i,]=pho^(abs(i-J))};

######## covariate matrix
Z=mvrnorm(n=n , rep(0,p), Sigma,  empirical = FALSE); ######## Generate covariates
Z=pnorm(Z);
for(i in 1:p){
        Z[,i]=Z[,i]/sqrt(sum(Z[,i]^2)/n);
}
X=cbind(rep(1,n),Z);
```

```
######## regression coefficients
alpha=rep(0,p+1);
alpha[2]=2;
alpha[3]=1.5;
gamma=rep(0,p+1);
gamma[9]=2;

######## errors and response variable
epsilon=rnorm(n);
Y=X%*%alpha+(X%*%gamma)*epsilon;

######## taus grid
tauL=0.1;
tauU=0.9;
J=seq(tauL,tauU,0.01);

object=vecrqalasso(Y,Z,J);
```

Index