Package ‘SITQR’

October 11, 2021

Title SITQR

Version 0.0.1

Description Estimation and inference for single index thresholding in quantile regression by smoothing the indicator function in threshold.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Imports rgenoud (>= 5.7), quantreg (>= 5.18), parallel, GeneBook, MASS, pracma

Depends R (>= 3.2), stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

R topics documented:

prof_smooth 1
prof_unsmooth_taus 3
rqs_constant_test 5
rq_bias 7
rq_mix_boot 8
rq_wald 9

Index **11**

`prof_smooth` *Estimate the threshold parameter by smoothing*

Description

`prof_smooth` seeks to minimize the profiled objective function:

$$S(\boldsymbol{\psi}; h_n) = \frac{1}{n}\sum_{i=1}^{n} \rho_\tau\{y_i - \boldsymbol{z}_i^T \widetilde{\boldsymbol{\beta}}(\boldsymbol{\psi}) - \boldsymbol{z}_i'^T \widetilde{\boldsymbol{\delta}}(\boldsymbol{\psi}) G(\frac{x_{1i} + \boldsymbol{x}_{i2}^T \boldsymbol{\psi}}{h_n})\},$$

where $\widetilde{\boldsymbol{\beta}}(\boldsymbol{\psi})$ and $\widetilde{\boldsymbol{\beta}}(\boldsymbol{\psi})$ are estimation by quantile regression given $\boldsymbol{\psi}$. Here the smooth function $G(x) = \Phi(x) + x\phi(x)$ where $\Phi(x)$ and $\phi(x)$ are the cdf and pdf of standard normal respectively.

Usage

```
prof_smooth(
  z,
  tz,
  x2,
  x1,
  y,
  tau,
  bdw,
  starting,
  method,
  cl.setup,
  Domains.Matrix
)
```

Arguments

z	a $n \times d$ matrix for regression variables corresponding to $\boldsymbol{\beta}$.
tz	a $n \times d'$ matrix for regression variables corresponding to $\boldsymbol{\delta}$.
x2	a $n \times p$ matrix for threshold variables.
x1	a vector for the threshold variable whose coefficient is 1.
y	a vector for the response.
tau	a value between 0 and 1. This is the quantile of interest.
bdw	the bandwidth for smoothing.
starting	starting value for optimization.
method	if there is a good starting value, method "optim" is suggested. Otherwise, method "genetic" implemented by `rgenoud::genoud` is suggested. When parallel computing is used in method "genetic", all inputs should be specified outside of the function `prof_smooth`.
cl.setup	the number of nodes. >1 indicates choosing parallel computing option in `rgenoud::genoud`. Default is 1.
Domains.Matrix	it is a $p \times 2$ matrix and each row provides the optimization boundary for each threshold parameter. It is only used when `rgenoud::genoud` is used.

Value

This function returns the estimation for threshold parameters.

Author(s)

Yingying Zhang, <yyzhang@fem.ecnu.edu.cn>.

Examples

```
GenerateData <- function(n,sig,delta){
    z1 = rexp(n, rate = 1)
    z2 = runif(n, 0, 1)
    x1 = rnorm(n, 1, 1)
    x2 = rnorm(n, 0, 1)
    z = cbind(1, z1, z2)
```

```
    tz = cbind(z1, z2)
    x = cbind(1, x1, x2)
    rank = runif(n,0,1)
    y = 1 + z1 + z2 + delta*(z1 + z2)*(-1+sig*qnorm(rank)+x1+x2>0) + qnorm(rank)
    return(list(z=z,tz=tz,x=x,y=y))
}
# with optim method   ####
testData <- GenerateData(500,0,1)
# The true value for threshold variable is c( -1, 1, 1).
starting <- c(-1,1)
n <- 500
bdw <- 2*sd(testData$x[,-3]%*%c(-1,1)+testData$x[,3])*(log(n)/sqrt(n))
fit1 <- prof_smooth(testData$z, testData$tz,
                        testData$x[,-3],testData$x[,3],
                        testData$y,tau=0.5,
                        bdw,starting=starting,
                        method="optim")
fit1
# with genetic method   ####
z <- testData$z
tz <- testData$tz
x2 <- testData$x[,-3]
x1 <- testData$x[,3]
y <- testData$y
tau <- 0.5
cl.setup <- 2
Domains.Matrix <- cbind(rep(-5,2),rep(5,2))
starting <- c(-1,1)
bdw <- 2*sd(testData$x[,-3]%*%c(-1,1)+testData$x[,3])*(log(n)/sqrt(n))
fit2 <- prof_smooth(z, tz, x2, x1, y,tau,bdw,starting,
                    method="genetic",cl.setup,Domains.Matrix)
fit2
```

`prof_unsmooth_taus` *Estimate the common threshold parameter for multiple quantiles without smoothing*

Description

`prof_unsmooth_taus` seeks to minimize the profiled objective function:

$$S(\boldsymbol{\psi}; h_n) = \frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n}\rho_{\tau_k}\{y_i - \boldsymbol{z}_i^T\widetilde{\boldsymbol{\beta}}(\tau_k,\boldsymbol{\psi}) - \boldsymbol{z}_i'^T\widetilde{\boldsymbol{\delta}}(\tau_k,\boldsymbol{\psi})I(x_{1i} + \boldsymbol{x}_{i2}^T\boldsymbol{\psi} > 0)\},$$

where $\widetilde{\boldsymbol{\beta}}(\tau_k,\boldsymbol{\psi})$ and $\widetilde{\boldsymbol{\beta}}(\tau_k,\boldsymbol{\psi})$ are estimation by quantile regression given τ_k and $\boldsymbol{\psi}$.

Usage

```
prof_unsmooth_taus(
  z,
  tz,
  x2,
  x1,
```

```
  y,
  taus,
  starting,
  method,
  cl.setup,
  Domains.Matrix
)
```

Arguments

z	a $n \times d$ matrix for regression variables corresponding to $\boldsymbol{\beta}$.
tz	a $n \times d'$ matrix for regression variables corresponding to $\boldsymbol{\delta}$.
x2	a $n \times p$ matrix for threshold variables.
x1	a vector for the threshold variable whose coefficient is 1.
y	a vector for the response.
taus	a value between 0 and 1. This is the quantile of interest.
starting	starting value for optimization.
method	if there is a good starting value, method "optim" is suggested. Otherwise, method "genetic" implemented by `rgenoud::genoud` is suggested. When parallel computing is used in method "genetic", all inputs should be specified outside of the function `prof_unsmooth_taus`.
cl.setup	the number of nodes. >1 indicates choosing parallel computing option in `rgenoud::genoud`. Default is 1.
Domains.Matrix	it is a $p \times 2$ matrix and each row provides the optimization boundary for each threshold parameter. It is only used when `rgenoud::genoud` is used.

Value

This function returns the unsmoothed estimation for the common threshold parameters by multiple quantiles.

Author(s)

Yingying Zhang, <yyzhang@fem.ecnu.edu.cn>.

Examples

```
GenerateData <- function(n,sig,delta){
    z1 = rexp(n, rate = 1)
    z2 = runif(n, 0, 1)
    x1 = rnorm(n, 1, 1)
    x2 = rnorm(n, 0, 1)
    z = cbind(1, z1, z2)
    tz = cbind(z1, z2)
    x = cbind(1, x1, x2)
    rank = runif(n,0,1)
    y = 1 + z1 + z2 + delta*(z1 + z2)*(-1+sig*qnorm(rank)+x1+x2>0) + qnorm(rank)
    return(list(z=z,tz=tz,x=x,y=y))
}
# with optim method   ####
testData <- GenerateData(500,0,1)
# The true value for threshold variable is c( -1, 1, 1).
```

```
starting <- c(-1,1)
n <- 500
taus <- c(0.5,0.7)
fit1 <- prof_unsmooth_taus(testData$z, testData$tz,
                        testData$x[,-3],testData$x[,3],
                        testData$y,taus,
                        starting=starting,
                        method="optim")
fit1
# with genetic method   ####
z <- testData$z
tz <- testData$tz
x2 <- testData$x[,-3]
x1 <- testData$x[,3]
y <- testData$y
cl.setup <- 2
Domains.Matrix <- cbind(rep(-5,2),rep(5,2))
starting <- c(-1,1)
fit2 <- prof_unsmooth_taus(z, tz, x2, x1, y, taus, starting,
                           method="genetic",cl.setup,Domains.Matrix)
fit2
```

`rqs_constant_test` *Testing the constancy of threshold parameters across multiple quantiles by mixed boostrap method*

Description

`rqs_constant_test` aims at testing the hypothesis that

$$H_0 : \boldsymbol{\psi}(\tau) = \boldsymbol{\psi}_0 \; for \; all \; \tau \in \mathcal{T} \; v.s. \; H_1 : \exists \tau' \in \mathcal{T} \; such \; that \; \boldsymbol{\psi}(\tau') \neq \boldsymbol{\psi}_0.$$

When $\boldsymbol{\psi}_0$ is unknown, the estimator obtained by `prof_unsmoothed_taus` can be used.

Usage

```
rqs_constant_test(
  z,
  tz,
  x2,
  x1,
  y,
  taus,
  bdws,
  phats,
  psi0,
  boot_times,
  probs = 0.95
)
```

Arguments

z	a $n \times d$ matrix for regression variables corresponding to $\boldsymbol{\beta}$.
tz	a $n \times d'$ matrix for regression variables corresponding to $\boldsymbol{\delta}$.
x2	a $n \times p$ matrix for threshold variables.
x1	a vector for the threshold variable whose coefficient is 1.
y	a vector for the response.
taus	a grid of quantiles in $\mathcal{T}$.
bdws	the bandwidth for each quantile. Under H_0, the same bandwidth can be used for each τ.
phats	a matrix with each column corresponding to threshold estimation for each τ in taus.
psi0	when $\boldsymbol{\psi}_0$ is unknown, the estimator obtained by `profiled_unsmoothed_taus` can be used.
boot_times	bootstrap times.
probs	confidence level.

Value

This function returns an object with 5 objects.

`KS` the Kolmogorov-Smirnov type statistic.

`C.value` the criticle value calculated by mixed bootstrap method.

`Accept` Accept=1 means to accept H_0. Accept=0 means to reject H_0.

`r` a list of length equal to taus, the bootstrap estimation for threshold parameters for each τ.

`Biastaus` a matrix with each column corresponding to the bias estimation for threshold parameters for each τ.

Author(s)

Yingying Zhang, <yyzhang@fem.ecnu.edu.cn>.

Examples

```
GenerateData <- function(n,sig,delta){
    z1 = rexp(n, rate = 1)
    z2 = runif(n, 0, 1)
    x1 = rnorm(n, 1, 1)
    x2 = rnorm(n, 0, 1)
    z = cbind(1, z1, z2)
    tz = cbind(z1, z2)
    x = cbind(1, x1, x2)
    rank = runif(n,0,1)
    y = 1 + z1 + z2 + delta*(z1 + z2)*(-1+sig*qnorm(rank)+x1+x2>0) + qnorm(rank)
    return(list(z=z,tz=tz,x=x,y=y))
}

testData <- GenerateData(500,0,1)
# The true value for threshold variable is c( -1, 1, 1).
starting <- c(-1,1)
phats <- matrix(0,2,2)
```

```
taus <- c(0.5,0.7)
n <- 500
bdw <- 2*sd(testData$x[,-3]%*%c(-1,1)+testData$x[,3])*(log(n)/sqrt(n))
for(i in 1:length(taus)){
   phats[,i] <- prof_smooth(testData$z, testData$tz,testData$x[,-3],
                        testData$x[,3],testData$y,tau=taus[i],
                        bdw,starting,method="optim")
}
bdws <- rep(bdw,2)
psi0 <- c(-1,1)
fit2 <- rqs_constant_test(testData$z,testData$tz,testData$x[,-3],
                     testData$x[,3],testData$y,taus,
                     bdws,phats,psi0,boot_times = 10,probs=0.95)
```

`rq_bias` *Estimate the bias of smoothed estimation for threshod parameters*

Description

`rq_bias` aims at estimating bias of smoothed estimation for threshold parameter and helps to implement bias correction.

Usage

```
rq_bias(z, tz, x2, x1, y, tau, bdw, phat, Lam)
```

Arguments

`z`	a $n \times d$ matrix for regression variables corresponding to $\boldsymbol{\beta}$.
`tz`	a $n \times d'$ matrix for regression variables corresponding to $\boldsymbol{\delta}$.
`x2`	a $n \times p$ matrix for threshold variables.
`x1`	a vector for the threshold variable whose coefficient is 1.
`y`	a vector for the response.
`tau`	a value between 0 and 1. This is the quantile of interest.
`bdw`	the bandwidth for smoothing.
`phat`	the estimated threshold parameters by `profiled_smooth`.
`Lam`	the estimated covariance matrix by bootstrap method. We suggest to use the covariance estimation by mixed bootstrap method rather than the wald method.

Value

This function returns the bias estimation for smoothed estimation of threshold parameters.

Author(s)

Yingying Zhang, <yyzhang@fem.ecnu.edu.cn>.

Examples

```
GenerateData <- function(n,sig,delta){
    z1 = rexp(n, rate = 1)
    z2 = runif(n, 0, 1)
    x1 = rnorm(n, 1, 1)
    x2 = rnorm(n, 0, 1)
    z = cbind(1, z1, z2)
    tz = cbind(z1, z2)
    x = cbind(1, x1, x2)
    rank = runif(n,0,1)
    y = 1 + z1 + z2 + delta*(z1 + z2)*(-1+sig*qnorm(rank)+x1+x2>0) + qnorm(rank)
    return(list(z=z,tz=tz,x=x,y=y))
}
testData <- GenerateData(500,0,1)
# The true value for threshold variable is c( -1, 1, 1).
starting <- c(-1,1)
n <- 500
bdw <- 2*sd(testData$x[,-3]%*%c(-1,1)+testData$x[,3])*(log(n)/sqrt(n))
fit1 <- prof_smooth(testData$z, testData$tz,testData$x[,-3],
                        testData$x[,3],testData$y,tau=0.5,
                        bdw,starting,method="optim")
phats = matrix(fit1,2,1)
fit2 <- rq_mix_boot(testData$z,testData$tz,testData$x[,-3],
                    testData$x[,3],testData$y,taus=0.5,
                    bdws = bdw,phats,boot_times = 1000)
Lam  = fit2$Lam.boot[[1]]
phat = fit1
fit3 <- rq_bias(testData$z,testData$tz,testData$x[,-3],
                    testData$x[,3],testData$y,tau=0.5,
                    bdw,phat,Lam)
```

`rq_mix_boot` *Estimate the asymptotic covariance matrix for threshold parameters by mixed bootstrap*

Description

`rq_mix_boot` aims at estimating the asymptotic covariance matrix by mixed bootstrap method. It updates the regression coefficients by calculating a mean and then update the threshold parameters given the updated regression coefficients.

Usage

```
rq_mix_boot(z, tz, x2, x1, y, taus, bdws, phats, boot_times = 1000)
```

Arguments

`z`	a $n \times d$ matrix for regression variables corresponding to $\boldsymbol{\beta}$.
`tz`	a $n \times d'$ matrix for regression variables corresponding to $\boldsymbol{\delta}$.
`x2`	a $n \times p$ matrix for threshold variables.
`x1`	a vector for the threshold variable whose coefficient is 1.
`y`	a vector for the response.

taus	a vector with values between 0 and 1. These are the quantiles of interest.
bdws	a vector with bandwidth for each τ in taus.
phats	a matrix with each column corresponding to threshold estimation for each τ in taus.
boot_times	bootstrap times.

Value

This function returns an object with 3 objects.

`index.boot` a list of length equal to taus, the bootstrap estimates for $\psi(\tau)$ for each τ.

`time.boot` a list of length equal to taus, the bootstrap calculation times for each τ.

`Lam.boot` a list of length equal to taus, the covariance matrix estimation for each τ.

Author(s)

Yingying Zhang, <yyzhang@fem.ecnu.edu.cn>.

Examples

```
GenerateData <- function(n,sig,delta){
    z1 = rexp(n, rate = 1)
    z2 = runif(n, 0, 1)
    x1 = rnorm(n, 1, 1)
    x2 = rnorm(n, 0, 1)
    z = cbind(1, z1, z2)
    tz = cbind(z1, z2)
    x = cbind(1, x1, x2)
    rank = runif(n,0,1)
    y = 1 + z1 + z2 + delta*(z1 + z2)*(-1+sig*qnorm(rank)+x1+x2>0) + qnorm(rank)
    return(list(z=z,tz=tz,x=x,y=y))
}
testData <- GenerateData(500,0,1)
starting <- c(-1,1)
# The true value for threshold variable is c( -1, 1, 1).
n <- 500
bdw <- 2*sd(testData$x[,-3]%*%c(-1,1)+testData$x[,3])*(log(n)/sqrt(n))
fit1 <- prof_smooth(testData$z, testData$tz,testData$x[,-3],
                        testData$x[,3],testData$y,tau=0.5,
                        bdw,starting,method="optim")
phats = matrix(fit1,2,1)
fit2 <- rq_mix_boot(testData$z,testData$tz,testData$x[,-3],
                    testData$x[,3],testData$y,taus=0.5,
                    bdws = bdw,phats,boot_times = 10)
```

`rq_wald` *Estimate the asymptotic covariance matrix for threshold parameters by Wald method*

Description

`rq_wald` aims at estimating the asymptotic covariance matrix denoted as $\Lambda(\tau)$ by Wald method.

Usage

```
rq_wald(z, tz, x2, x1, y, tau, bdw, phat)
```

Arguments

z	a $n \times d$ matrix for regression variables corresponding to $\boldsymbol{\beta}$.
tz	a $n \times d'$ matrix for regression variables corresponding to $\boldsymbol{\delta}$.
x2	a $n \times p$ matrix for threshold variables.
x1	a vector for the threshold variable whose coefficient is 1.
y	a vector for the response.
tau	a value between 0 and 1. This is the quantile of interest.
bdw	the bandwidth for smoothing.
phat	the estimated threshold parameters by `profiled_smooth`.

Value

This function returns the asymptotic covariance estimation for threshold parameters.

Author(s)

Yingying Zhang, <yyzhang@fem.ecnu.edu.cn>.

Examples

```
GenerateData <- function(n,sig,delta){
    z1 = rexp(n, rate = 1)
    z2 = runif(n, 0, 1)
    x1 = rnorm(n, 1, 1)
    x2 = rnorm(n, 0, 1)
    z = cbind(1, z1, z2)
    tz = cbind(z1, z2)
    x = cbind(1, x1, x2)
    rank = runif(n,0,1)
    y = 1 + z1 + z2 + delta*(z1 + z2)*(-1+sig*qnorm(rank)+x1+x2>0) + qnorm(rank)
    return(list(z=z,tz=tz,x=x,y=y))
}

testData <- GenerateData(500,0,1)
# The true value for threshold variable is c( -1, 1, 1).
starting <- c(-1,1)
fit1 <- prof_smooth(testData$z, testData$tz,testData$x[,-3],
                        testData$x[,3],testData$y,tau=0.5,
                        bdw=0.18,starting,method="optim")
fit2 <- rq_wald(testData$z,testData$tz,testData$x[,-3],
                testData$x[,3],testData$y,tau=0.5,
                bdw=0.18,phat = fit1)
```

Index

prof_smooth, 1
prof_unsmooth_taus, 3

rq_bias, 7
rq_mix_boot, 8
rq_wald, 9
rqs_constant_test, 5