Package ‘ZIQSI’

November 6, 2024

Type Package

Title ZIQSI package

Version 0.1.0

Author Zirui Wang and Tianying Wang

Maintainer Zirui Wang <wzr23@mails.tsinghua.edu.cn>

Description This R document provides a novel zero-inflated semiparametric single-index quantile regression algorithm to predict the personal quantile curve for zero-inflated response with given samples.

License GPL-2

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0),
MASS,
lme4 (>= 1.1-23),
quantreg,
stats

Suggests aod,
PearsonDS

RoxygenNote 7.3.2

R topics documented:

AQE 1
proposed.nonsmooth.spline 3

Index **6**

`AQE` *AQE Estimation*

Description

This function estimates the average quantile difference for two different values of a covariate while controlling for other covariates.

Usage

```
AQE(y, Xl, Xq, indexl, indexq, value1, value2, taus, delta = 0.499, m, u)
```

Arguments

y	A numeric vector of the response variable.
Xl	A matrix of covariates for logistic regression.
Xq	A matrix of covariates for quantile regression.
indexl	An integer index indicating which covariate is being modified for logistic regression.
indexq	An integer index indicating which covariate is being modified for quantile regression.
value1	The first value of the covariate to be analyzed.
value2	The second value of the covariate to be analyzed.
taus	A numeric vector of quantiles to estimate.
delta	A numeric value for the delta parameter (default is 0.499).
m	An integer parameter, the order of B-spline in single-index quantile regression.
u	q*1 vector, usually using a vector where all values are equal and the norm is 1, the initial parameter for minimizing the profile likelihood function for quantile regression.

Value

A numeric vector of quantile estimates. Simulation Example

This script demonstrates the simulation of data and the usage of the AQE function.

Examples

```
# Set the random seed for reproducibility
set.seed(100002)

# Sample size
n = 500

# Generate the true model for simulation
## Probability of Y > 0 given covariate x
p = function(x1, x2, x3, x4, x5, gam0 = -0.4, gam1 = -0.480,
              gam2 = -0.022, gam3 = 0.021, gam4 = 0.015, gam5 = -0.009) {
  lc = gam0 + gam1 * x1 + gam2 * x2 + gam3 * x3 + gam4 * x4 + gam5 * x5
  exp(lc) / (1 + exp(lc))
}

## Define beta functions
bet1 = function(x) {(0.3 * sqrt(x) - x) * 2}
bet2 = function(x) {x^2 * 2.2}
bet3 = function(x) {(x^2 - 0.5 * x + 0.6) * 2 / 3}
bet4 = function(x) {-sin(x * 2 * pi) * 0.1}
bet5 = function(x) {-(0.3 * x^2 - x) * 2}
bet0 = function(x) {-147.7 * x - 50 * x^2 - 20}
bet = function(x,tau){x*tau*20.4}
## G_tau function
```

```
func = function(x, tau) {
  return(bet(x %*% rbind(bet1(tau), bet2(tau), bet3(tau), bet4(tau), bet5(tau)) + bet0(tau), tau))
}

# Given samples
x1 = rbinom(n, 1, 0.5)  # Medicament use
x2 = rnorm(n, 28, 2)  # bmi
x3 = rnorm(n, 92.5, 13)  # waist
x4 = rnorm(n, 80, 12)  # diastolic_bp
x5 = rnorm(n, 124, 18.5)  # systolic_bp
x0 = rep(1, n)

# Covariates of each sample
X = cbind(x0, x1, x2, x3, x4, x5)
u = runif(n)
b = rbinom(n, 1, p(x1, x2, x3, x4, x5))
w = bet(bet1(u) * x1 + bet2(u) * x2 + bet3(u) * x3 + bet4(u) * x4 + bet5(u) * x5 + bet0(u), u)

# Simulated quantiles of each sample
y = b * w

P_ZIQSI = AQE(y,X[,-1],X,indexl = 2, indexq = 3, value1 = 23, value2 = 28,taus = 0.5, m = 3, u = rep(sqrt(1/6),6))
```

---

proposed.nonsmooth.spline   *proposed.nonsmooth.spline*

Description

This function estimates the quantiles for given sample(s) on a grid of quantile levels.

Usage

```
proposed.nonsmooth.spline(
  y,
  Xl,
  Xq,
  xl,
  xq,
  taus = seq(0, 0.99, 0.01),
  delta = 0.499,
  m = 3,
  u
)
```

Arguments

y	n*1 vector, the observed outcome
Xl	n*p matrix, the observed covariates in logistic regression
Xq	n*q matrix, the observed covariates in quantile regression
xl	m*p matrix, the new covariates in logistic regression, with which conditional quantile function are estimated

xq	m*q matrix, the new covariates in quantile regression, with which conditional quantile function are estimated
taus	k*1 vector, the grid of target tau's of y
delta	constant, better to keep the default 0.499
m	numeric variable, the order of B-spline function
u	q*1 vector, usually using a vector where all values are equal and the norm is 1, the initial parameter for minimizing the profile likelihood function for quantile regression.

Value

quantiles of a m*k matrix, each row is the estimated quantiles for each new case

Examples

```
# Set the random seed for reproducibility
set.seed(10001)

# Sample size
n = 500

# Generate the true model for simulation
## Probability of Y > 0 given covariate x
p = function(x1, x2, x3, x4, x5, gam0 = -0.4, gam1 = -0.480,
              gam2 = -0.022, gam3 = 0.021, gam4 = 0.015, gam5 = -0.009) {
  lc = gam0 + gam1 * x1 + gam2 * x2 + gam3 * x3 + gam4 * x4 + gam5 * x5
  exp(lc) / (1 + exp(lc))
}

## Define beta functions
bet1 = function(x) {(0.3 * sqrt(x) - x) * 2}
bet2 = function(x) {x^2 * 2.2}
bet3 = function(x) {(x^2 - 0.5 * x + 0.6) * 2 / 3}
bet4 = function(x) {-sin(x * 2 * pi) * 0.1}
bet5 = function(x) {-(0.3 * x^2 - x) * 2}
bet0 = function(x) {-147.7 * x - 50 * x^2 - 20}
bet = function(x,u){x^4*u*10^(-5)/6+x^2*u*0.2/3}
## G_tau function
func = function(x, tau) {
 return(bet(x %*% rbind(bet1(tau), bet2(tau), bet3(tau), bet4(tau), bet5(tau)) + bet0(tau), tau))
}

# Given covariates for quantile estimation
X1 = c(0, 1)  # sex
X2 = qnorm(0.5, 28, 2)  # bmi
X3 = qnorm(0.5, 92.5, 13)  # waist
X4 = qnorm(0.5, 80, 12)  # diastolic_bp
X5 = qnorm(0.5, 124, 18.5)  # systolic_bp
X0 = cbind(c(rep(X1[1], 1), rep(X1[2], 1)), rep(X2, 2), rep(X3, 2), rep(X4, 2), rep(X5, 2))

# Given samples
x1 = rbinom(n, 1, 0.5)  # Medicament use
x2 = rnorm(n, 28, 2)  # bmi
x3 = rnorm(n, 92.5, 13)  # waist
x4 = rnorm(n, 80, 12)  # diastolic_bp
```

```
x5 = rnorm(n, 124, 18.5)  # systolic_bp
x0 = rep(1, n)

# Covariates of each sample
X = cbind(x0, x1, x2, x3, x4, x5)
u = runif(n)
b = rbinom(n, 1, p(x1, x2, x3, x4, x5))
w = bet(bet1(u) * x1 + bet2(u) * x2 + bet3(u) * x3 + bet4(u) * x4 + bet5(u) * x5 + bet0(u), u)

# Simulated quantiles of each sample
y = b * w

# Estimated quantile curves for the given covariate x
A = proposed.nonsmooth.spline(y = y, Xl = cbind(x1, x2, x3, x4, x5),
                                  Xq = cbind(x0, x1, x2, x3, x4, x5),
                                  xl = X0,
                                  xq = cbind(1, X0),
                                  taus = seq(0, 0.99, by = 0.01),
                                  delta = 0.499,
                                  m = 4,
                                  u = rep(1/sqrt(6), 6))
# plot the estimated quantile curve for the first sample
# plot(seq(0,0.99,0.01),A[1,])
# plot the estimated quantile curve for the second sample
# plot(seq(0,0.99,0.01),A[2,])
```

Index

AQE, 1

proposed.nonsmooth.spline, 3