Package 'QuantifQuantile'

August 13, 2015

Type Package

Title Estimation of Conditional Quantiles using Optimal Quantization

Version 2.2

Date 2015-08-13

Author Isabelle Charlier and Davy Paindaveine and Jerome Saracco

Maintainer Isabelle Charlier <ischarli@ulb.ac.be>

Description Estimation of conditional quantiles using optimal quantization. Construction of an optimal grid of N quantizers, estimation of conditional quantiles and data driven selection of the size N of the grid. Graphical illustrations for the selection of N and of resulting estimated curves or surfaces when the dimension of the covariate is one or two.

License GPL (>= 2.0)

Depends rgl, parallel

NeedsCompilation no

Repository CRAN

Date/Publication 2015-08-13 20:37:00

R topics documented:

choice.grid 2
gironde 3
plot.QuantifQuantile 4
print.QuantifQuantile 5
QuantifQuantile 6
QuantifQuantile.d 8
QuantifQuantile.d2 11
summary.QuantifQuantile 13

Index **14**

choice.grid *Choice of the quantization grids*

Description

This function provides ng optimal quantization grids for X, with N fixed.

Usage

```
choice.grid(X, N, ng = 1, p = 2)
```

Arguments

X	vector or matrix that we want to quantize.
N	size of the quantization grids.
ng	number of quantization grids needed.
p	L_p norm optimal quantization.

Details

This function works for any dimension of X. If the covariate is univariate, X is a vector while X is a matrix with d rows when the covariate is d-dimensional.

Value

An array of dimension d*N*ng that corresponds to ng quantization grids.

References

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimator based on optimal quantization: from theory to practice*, Submitted.

Pages, G. (1998) *A space quantization method for numerical integration*, Journal of Computational and Applied Mathematics, 89(1), 1-38

See Also

QuantifQuantile, QuantifQuantile.d2 and QuantifQuantile.d

Examples

```
X <- runif(300,-2,2)
N <- 10
ng <- 20
choice.grid(X,N,ng)
```

gironde *A real dataset called gironde*

Description

gironde is a real dataset collected by the INSEE (French National Institute of Statistics and Economic Studies) and extracted by SIDDT from IRSTEA (French National Research Institute of Science and Technology for Environment and Agriculture) of Grenoble. It contains different variables collected in 542 towns/villages in Gironde, France.

Usage

```
data(gironde)
```

Format

The format is a list of 4 components: $employment: A data-frame of numerical variables containing, for 542 towns/villages in Gironde (France), the percentages of farmers ($farmers), of tradesmen and handicraftsmen ($tradesmen), of managers and executives ($managers), of workers and employees ($workers), of unemployed workers ($unemployed), of middle-range employees ($middleemp), of retired people ($retired), the employement rate ($employrate) and the average income ($income);

$housing: A data-frame of both numerical and categorical variables containing, for 542 towns/villages in Gironde (France), the population density ($population), the percentages of primary residences ($primaryres), of houses ($houses), of home owners living in their primary residence ($owners) and of council housing ($council);

$services: A data-frame of categorical variables containing, for 542 towns/villages in Gironde (France), the numbers of butchers ($butcher), of bakers ($baker), of post offices ($postoffice), of dentists ($dentist), of grocery stores ($grocery), of child care day nurseries ($nursery), of doctors ($doctor), of chemists ($chemist) and of restaurants ($restaurant);

$environment: A data-frame of numerical variables containing, for 542 towns/villages in Gironde (France) the percentages of buildings ($building), of water ($water), of vegetation ($vegetation) and of agricultural land ($agricul).

Source

INSEE (French National Institute of Statistics and Economic Studies) and SIDDT of IRSTEA Grenoble, France, http://siddt.irstea.fr/.

`plot.QuantifQuantile` *Plot of estimated conditional quantiles using optimal quantization.*

Description

This function plots the estimated conditional quantiles by default. It can also illustrate our data driven selection criterion for `N` by providing the plot of the bootstrap estimated values of integrated squared error ISE(N) versus `N`.

Usage

```
## S3 method for class 'QuantifQuantile'
plot(x, col.plot = c(1:(length(x$alpha) + 1)),
  ise = FALSE, ...)
```

Arguments

`x`	An object of class `QuantifQuantile`, which is the result of `QuantifQuantile` or `QuantifQuantile.d2`.
`col.plot`	Vector of size `length(x$alpha)+1`. The first entry corresponds to the color of the data points while the other colors are for the conditional quantiles curves, points or surfaces.
`ise`	Whether it plots the ISE curves in addition to the estimated quantile curves (if `ise=TRUE`, two different plots).
`...`	Arguments to be passed to `par`.

Details

If `X` is univariate, the graph is two-dimensional and if `X` is bivariate, it provides a 3D-graph using the `rgl` package. When only one value for `x` is considered, estimated conditional quantiles are plotted as points. When `x` is a grid of values, they are plotted as curves if `d=1` and surfaces if `d=2`.

When `ise=TRUE`, the first plot allows to adapt the choice of the grid for `N`, called `testN`. For example, if the curve is decreasing with `N`, it indicates that the values in `testN` are too small and the optimal `N` is larger.

References

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimation through optimal quantization*, Journal of Statistical Planning and Inference, 2015 (156), 14-30.

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimator based on optimal quantization: from theory to practice*, Submitted.

See Also

`QuantifQuantile`, `QuantifQuantile.d2` and `QuantifQuantile.d`

Examples

```
#for a univariate X
set.seed(644972)
n <- 300
X <- runif(300,-2,2)
Y <- X^2+rnorm(n)
res <- QuantifQuantile(X,Y,testN=seq(10,25,by=5))
plot(res,ise=TRUE)

## Not run:
set.seed(92536)
n <- 300
X <- runif(300,-2,2)
Y <- X^2+rnorm(n)
res <- QuantifQuantile(X,Y,testN=seq(10,25,by=5),x=1)
plot(res,ise=TRUE)


#for a bivariate X
#(a few seconds to execute)
set.seed(253664)
d <- 2
n <- 1000
X<-matrix(runif(d*n,-2,2),nr=d)
Y<-apply(X^2,2,sum)+rnorm(n)
res <- QuantifQuantile.d2(X,Y,testN=seq(80,130,by=10),B=20,tildeB=15)
plot(res,ise=TRUE)

set.seed(193854)
d <- 2
n <- 1000
X<-matrix(runif(d*n,-2,2),nr=d)
Y<-apply(X^2,2,sum)+rnorm(n)
res <- QuantifQuantile.d2(X,Y,testN=seq(110,140,by=10),x=as.matrix(c(1,0)),
B=30,tildeB=20)
plot(res,ise=TRUE)

## End(Not run)
```

---

print.QuantifQuantile *Print of QuantifQuantile results*

Description

This function displays a small description of QuantifQuantile results.

Usage

```
## S3 method for class 'QuantifQuantile'
print(x, ...)
```

Arguments

x	An object of class QuantifQuantile, which is the result of the QuantifQuantile, QuantifQuantile.d2 or QuantifQuantile.d functions.
...	Not used.

Author(s)

Isabelle Charlier, Davy Paindaveine, Jerome Saracco

References

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimation through optimal quantization*, Journal of Statistical Planning and Inference, 2015 (156), 14-30.

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimator based on optimal quantization: from theory to practice*, Submitted.

See Also

QuantifQuantile, QuantifQuantile.d2 and QuantifQuantile.d

plot.QuantifQuantile, summary.QuantifQuantile

Examples

```
set.seed(644972)
n <- 300
X <- runif(300,-2,2)
Y <- X^2+rnorm(n)
res <- QuantifQuantile(X,Y,testN=seq(10,25,by=5))
print(res)
```

QuantifQuantile *QuantifQuantile for X univariate*

Description

Estimation of conditional quantiles using optimal quantization when X is univariate.

Usage

```
QuantifQuantile(X, Y, alpha = c(0.05, 0.25, 0.5, 0.75, 0.95),
  x = seq(min(X), max(X), length = 100), testN = c(35, 40, 45, 50, 55),
  p = 2, B = 50, tildeB = 20, same_N = TRUE, ncores = 1)
```

Arguments

X	vector of covariates.
Y	vector of response variables.
alpha	vector of order of the quantiles.
x	vector of values for x in q_alpha(x).
testN	grid of values of N that will be tested.
p	L_p norm optimal quantization.
B	number of bootstrap replications for the bootstrap estimator.
tildeB	number of bootstrap replications for the choice of N.
same_N	whether to use the same value of N for each alpha (TRUE by default).
ncores	number of cores to use. Default is set to 1 (see Details below).

Details

- This function calculates estimated conditional quantiles with a method based on optimal quantization when the covariate is unvariate. For multivariate covariate, see QuantifQuantile.d2 or QuantifQuantile.d.
- The criterion for selecting the number of quantizers is implemented in this function. The user has to choose a grid testN of possible values in which N will be selected. It actually minimizes some bootstrap estimated version of the ISE (Integrated Squared Error). More precisely, for N fixed, it calculates the sum according to alpha of hatISE_N and then minimizes the resulting vector to get N_opt. However, the user can choose to select a different value of N_opt for each alpha by setting same_N=FALSE. In this case, the vector N_opt is obtained by minimizing each column of hatISE_N separately. The reason why same_N=TRUE by default is that taking N_opt according to alpha could provide crossing conditional quantile curves (rarely observed for not too close values of alpha). The function plot.QuantifQuantile illustrates the selection of N_opt. If the graph is not decreasing then increasing, the argument testN should be adapted.
- This function can use parallel computation to save time, by simply increasing the parameter ncores. Parallel computation relies on mclapply from parallel package, hence is not available on Windows unless ncores=1 (default value).

Value

An object of class QuantifQuantile which is a list with the following components:

hatq_opt	A matrix containing the estimated conditional quantiles. The number of columns is the number of considered values for x and the number of rows the size of the order vector alpha. This object can also be returned using the usual fitted.values function.
N_opt	Optimal selected value for N. An integer if same_N=TRUE and a vector of integers of length length(alpha) otherwise.
hatISE_N	The matrix of estimated ISE provided by our selection criterion for N. The number of columns is then length(testN) and the number of rows length(alpha).
hatq_N	A 3-dimensional array containing the estimated conditional quantiles for each considered value for alpha, x and N.

X	The vector of covariates.
Y	The vector of response variables.
x	The considered vector of values for x in q_alpha(x).
alpha	The considered vector of order for the quantiles.
testN	The considered grid of values for N that were tested.

References

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimation through optimal quantization*, Journal of Statistical Planning and Inference, 2015 (156), 14-30.

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimator based on optimal quantization: from theory to practice*, Submitted.

See Also

QuantifQuantile.d2 and QuantifQuantile.d for multivariate versions.

plot.QuantifQuantile, print.QuantifQuantile, summary.QuantifQuantile

Examples

```
set.seed(644972)
n <- 300
X <- runif(300,-2,2)
Y <- X^2+rnorm(n)
res <- QuantifQuantile(X,Y,testN=seq(10,25,by=5))
## Not run:
res2 <- QuantifQuantile(X,Y,testN=seq(10,30,by=5),same_N=FALSE)

data(gironde)
X <- gironde[[1]]$middleemp
Y <- gironde[[2]]$density
set.seed(642536)
res <- QuantifQuantile(X,Y,testN=seq(5,25,by=5))

## End(Not run)
```

QuantifQuantile.d *QuantifQuantile for general X*

Description

Estimation of conditional quantiles using optimal quantization when X is d-dimensional.

Usage

```
QuantifQuantile.d(X, Y, x, alpha = c(0.05, 0.25, 0.5, 0.75, 0.95),
  testN = c(35, 40, 45, 50, 55), p = 2, B = 50, tildeB = 20,
  same_N = TRUE, ncores = 1)
```

Arguments

X	matrix of covariates.
Y	vector of response variables.
x	matrix of values for x in q_alpha(x).
alpha	vector of order of the quantiles.
testN	grid of values of N that will be tested.
p	L_p norm optimal quantization.
B	number of bootstrap replications for the bootstrap estimator.
tildeB	number of bootstrap replications for the choice of N.
same_N	whether to use the same value of N for each alpha (TRUE by default).
ncores	number of cores to use. Default is set to 1 (see Details below).

Details

- This function calculates estimated conditional quantiles with a method based on optimal quantization for any dimension for the covariate. The matrix of covariate X must have d rows (dimension). For particular cases of d =1 or 2, it is strongly recommended to use QuantifQuantile and QuantifQuantile.d2 respectively (computationally faster). The argument x must also have d rows.
- The criterion for selecting the number of quantizers is implemented in this function. The user has to choose a grid testN of possible values in which N will be selected. It actually minimizes some bootstrap estimated version of the ISE (Integrated Squared Error). More precisely, for N fixed, it calculates the sum according to alpha of hatISE_N and then minimizes the resulting vector to get N_opt. However, the user can choose to select a different value of N_opt for each alpha by setting same_N=FALSE. In this case, the vector N_opt is obtained by minimizing each column of hatISE_N separately. The reason why same_N=TRUE by default is that taking N_opt according to alpha could provide crossing conditional quantile curves (rarely observed for not too close values of alpha). The function plot.QuantifQuantile illustrates the selection of N_opt. If the graph is not decreasing then increasing, the argument testN should be adapted.
- This function can use parallel computation to save time, by simply increasing the parameter ncores. Parallel computation relies on mclapply from parallel package, hence is not available on Windows unless ncores=1 (default value).

Value

An object of class QuantifQuantile which is a list with the following components:

hatq_opt	A matrix containing the estimated conditional quantiles. The number of columns is the number of considered values for x and the number of rows the size of the order vector alpha. This object can also be returned using the usual fitted.values function.
N_opt	Optimal selected value for N. An integer if same_N=TRUE and a vector of integers of length length(alpha) otherwise.

hatISE_N	The matrix of estimated ISE provided by our selection criterion for N before taking the mean according to alpha. The number of columns is then length(testN) and the number of rows length(alpha).
hatq_N	A 3-dimensional array containing the estimated conditional quantiles for each considered value for alpha, x and N.
X	The matrix of covariates.
Y	The vector of response variables.
x	The considered vector of values for x in q_alpha(x).
alpha	The considered vector of order for the quantiles.
testN	The considered grid of values for N that were tested.

References

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimation through optimal quantization*, Journal of Statistical Planning and Inference, 2015 (156), 14-30.

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimator based on optimal quantization: from theory to practice*, Submitted.

See Also

QuantifQuantile and QuantifQuantile.d2 for particular dimensions one and two.

plot.QuantifQuantile, print.QuantifQuantile, summary.QuantifQuantile

Examples

```
## Not run:
set.seed(644925)
n <- 500
X <- runif(n,-2,2)
Y <- X^2+rnorm(n)
x <- seq(min(X),max(X),length=100)
res <- QuantifQuantile.d(X,Y,x,testN=seq(15,35,by=5))

## End(Not run)
## Not run:
set.seed(272422)
n <- 1000
X <- matrix(runif(n*2,-2,2),ncol=n)
Y <- apply(X^2,2,sum)+rnorm(n)
x1 <- seq(min(X[1,]),max(X[1,]),length=20)
x2 <- seq(min(X[2,]),max(X[2,]),length=20)
x <- matrix(c(rep(x1,20),sort(rep(x2,20))),nrow=nrow(X),byrow=TRUE)
res <- QuantifQuantile.d(X,Y,x,testN=seq(90,140,by=10),B=20,tildeB=15)

## End(Not run)
```

QuantifQuantile.d2 *QuantifQuantile for X bivariate*

Description

Estimation of conditional quantiles using optimal quantization when X is bivariate.

Usage

```
QuantifQuantile.d2(X, Y, alpha = c(0.05, 0.25, 0.5, 0.75, 0.95),
  x = matrix(c(rep(seq(min(X[1, ]), max(X[1, ]), length = 20), 20),
  sort(rep(seq(min(X[2, ]), max(X[2, ]), length = 20), 20))), nrow = 2, byrow =
  TRUE), testN = c(110, 120, 130, 140, 150), p = 2, B = 50, tildeB = 20,
  same_N = TRUE, ncores = 1)
```

Arguments

X	matrix of covariates.
Y	vector of response variables.
alpha	vector of order of the quantiles.
x	matrix of values for x in q_alpha(x).
testN	grid of values of N that will be tested.
p	L_p norm optimal quantization.
B	number of bootstrap replications for the bootstrap estimator.
tildeB	number of bootstrap replications for the choice of N.
same_N	whether to use the same value of N for each alpha (TRUE by default).
ncores	number of cores to use. Default is set to 1 (see Details below).

Details

- This function calculates estimated conditional quantiles with a method based on optimal quantization when the covariate is bivariate. The matrix of covariate X must have two rows (dimension). For other dimensions, see QuantifQuantile or QuantifQuantile.d. The argument x must also have two rows.
- The criterion for selecting the number of quantizers is implemented in this function. The user has to choose a grid testN of possible values in which N will be selected. It actually minimizes some bootstrap estimated version of the ISE (Integrated Squared Error). More precisely, for N fixed, it calculates the sum according to alpha of hatISE_N and then minimizes the resulting vector to get N_opt. However, the user can choose to select a different value of N_opt for each alpha by setting same_N=FALSE. In this case, the vector N_opt is obtained by minimizing each column of hatISE_N separately. The reason why same_N=TRUE by default is that taking N_opt according to alpha could provide crossing conditional quantile curves (rarely observed for not too close values of alpha). The function plot.QuantifQuantile illustrates the selection of N_opt. If the graph is not decreasing then increasing, the argument testN should be adapted.

- This function can use parallel computation to save time, by simply increasing the parameter ncores. Parallel computation relies on mclapply from parallel package, hence is not available on Windows unless ncores=1 (default value).

Value

An object of class QuantifQuantile which is a list with the following components:

hatq_opt	A matrix containing the estimated conditional quantiles. The number of columns is the number of considered values for x and the number of rows the size of the order vector alpha. This object can also be returned using the usual fitted.values function.
N_opt	Optimal selected value for N. An integer if same_N=TRUE and a vector of integers of length length(alpha) otherwise.
hatISE_N	The matrix of estimated ISE provided by our selection criterion for N before taking the mean according to alpha. The number of columns is then length(testN) and the number of rows length(alpha).
hatq_N	A 3-dimensional array containing the estimated conditional quantiles for each considered value for alpha, x and N.
X	The matrix of covariates.
Y	The vector of response variables.
x	The considered vector of values for x in q_alpha(x).
alpha	The considered vector of order for the quantiles.
testN	The considered grid of values for N that were tested.

References

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimation through optimal quantization*, Journal of Statistical Planning and Inference, 2015 (156), 14-30.

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimator based on optimal quantization: from theory to practice*, Submitted.

See Also

QuantifQuantile and QuantifQuantile.d for other dimensions.

plot.QuantifQuantile, print.QuantifQuantile, summary.QuantifQuantile

Examples

```
## Not run:
#(a few seconds to execute)
set.seed(164964)
n <- 1000
X <- matrix(runif(n*2,-2,2),ncol=n)
Y <- apply(X^2,2,sum)+rnorm(n)
res <- QuantifQuantile.d2(X,Y,testN=seq(90,140,by=10),B=20,tildeB=15)
res2 <- QuantifQuantile.d2(X,Y,testN=seq(90,150,by=10),B=20,tildeB=15,same_N=FALSE)

## End(Not run)
```

summary.QuantifQuantile *Summary of QuantifQuantile results*

Description

This function displays a summary of QuantifQuantile results.

Usage

```
## S3 method for class 'QuantifQuantile'
summary(object, ...)
```

Arguments

object	An object of class QuantifQuantile, which is the result of the QuantifQuantile, QuantifQuantile.d2 or QuantifQuantile.d functions.
...	Not used.

Details

This function prints the estimated conditional quantiles q_alpha(x) for each x and alpha considered, as an array, and also the selected tuning parameter N_opt.

Author(s)

Isabelle Charlier, Davy Paindaveine, Jerome Saracco

References

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimation through optimal quantization*, Journal of Statistical Planning and Inference, 2015 (156), 14-30.

Charlier, I. and Paindaveine, D. and Saracco, J., *Conditional quantile estimator based on optimal quantization: from theory to practice*, Submitted.

See Also

QuantifQuantile, QuantifQuantile.d2 and QuantifQuantile.d

plot.QuantifQuantile, print.QuantifQuantile

Examples

```
set.seed(644972)
n <- 300
X <- runif(300,-2,2)
Y <- X^2+rnorm(n)
res <- QuantifQuantile(X,Y,testN=seq(10,25,by=5))
summary(res)
```

Index

choice.grid, 2

gironde, 3

mclapply, *7*, *9*, *12*

par, *4*
parallel, *7*, *9*, *12*
plot.QuantifQuantile, 4, *6–13*
print.QuantifQuantile, 5, *8*, *10*, *12, 13*

QuantifQuantile, *2*, *4*, *6*, 6, *9–13*
QuantifQuantile.d, *2*, *4*, *6–8*, 8, *11–13*
QuantifQuantile.d2, *2*, *4*, *6–10*, 11, *13*

rgl, *4*

summary.QuantifQuantile, *6*, *8*, *10*, *12*, 13