Package 'quantdr'

July 1, 2025

Type Package

Title Dimension Reduction Techniques for Conditional Quantiles

Version 1.3.2

Description An implementation of dimension reduction techniques for conditional quantiles. Nonparametric estimation of conditional quantiles is also available.

License GPL-2 | GPL-3

Encoding UTF-8

Imports KernSmooth, mvtnorm, quantreg

Depends R (>= 3.2.0)

RoxygenNote 7.3.1

Collate 'ValAR.R' 'bic_d.R' 'dr.R' 'llqrcv.R' 'llqr.R' 'cqs.R'

Suggests ggplot2, knitr, rmarkdown, testthat, pracma, MASS, PerformanceAnalytics, png, graphics, grDevices

VignetteBuilder knitr

SystemRequirements GNU make

URL https://github.com/elianachristou/quantdr

BugReports https://github.com/elianachristou/quantdr/issues

NeedsCompilation no

Author Eliana Christou [aut, cre] (ORCID: <https://orcid.org/0000-0001-5044-0969>)

Maintainer Eliana Christou <echris15@uncc.edu>

Repository CRAN

Date/Publication 2025-07-01 11:20:02 UTC

Contents

---

cqs *Central quantile subspace*

Description

cqs estimates the directions of the central quantile subspace.

Usage

```
cqs(x, y, tau = 0.5, dtau = NULL)
```

Arguments

x	A design matrix (n x p). The rows represent observations and the columns represent predictor variables.
y	A vector of the response variable.
tau	A quantile level, a number strictly between 0 and 1.
dtau	An optional dimension of the central quantile subspace. If specified, it should be an integer between 1 and p, the number of columns of the design matrix x. In the context of the algorithm, if dtau is known to be one, i.e., the assumed model is a single-index model, then the algorithm stops after estimating the initial vector and saves computational time. However, if dtau is greater than one or (more realistically) unknown, then the algorithm continues on creating more vectors.

Details

The function computes the directions that span the τth central quantile subspace, i.e., the directions that define linear combinations of the predictor x that contain all the information available on the conditional quantile function.

The function starts by estimating the initial vector, which is defined as the least-squares estimator from regressing the conditional quantile on the predictor variable x. Then, if the dimension of the central quantile subspace is one, the algorithm stops and reports that vector as the basis of the central quantile subspace. Otherwise, the algorithm continues by creating more vectors and applying an eigenvalue decomposition to extract linearly independent vectors.

Value

cqs computes the directions of the central quantile subspace and returns:

qvectors:	The estimated directions of the τth central quantile subspace.
qvalues:	The eigenvalues resulting from the eigenvalue decomposition of the matrix with column vectors that span the central quantile subspace. If dtau is one, the qvalues output is not produced.

`dtau:` Suggested dimension of the central quantile subspace. If `dtau` is specified by the user then the algorithm outputs the user-defined value. If `dtau` is not specified by the user then the algorithm outputs a suggested dimension using the modified-BIC type criterion of Zhu et al. (2010). Note that this is one suggested method to estimate the structural dimension and is not necessarily a perfect one. The user has the option to use the eigenvalues `qvalues` on other criteria, like cross-validation, and determine the estimated dimension of the subspace.

References

Zhu, L.-P., Zhu, L.-X., Feng, Z.-H. (2010) Dimension reduction in regression through cumulative slicing estimation. *Journal of the American Statistical Association*, 105, 1455-1466.

Examples

```
# estimate the directions of a single-index model
set.seed(1234)
n <- 100
p <- 10
x <- matrix(rnorm(n * p), n, p)
error <- rnorm(n)
y <- 3 * x[, 1] + x[, 2] + error
tau <- 0.5
out <- cqs(x, y, tau, dtau = 1)
out
# without specifying dtau
out <- cqs(x, y, tau)
out
out$qvectors[, 1:out$dtau]
```

`llqr` *Local linear quantile regression*

Description

`llqr` estimates the τth conditional quantile of `y` given `x` based on a local linear fit. The estimation is performed at each of the design points or, if specified, at a single observation point `x0`.

Usage

```
llqr(x, y, tau = 0.5, h = NULL, method = "rule", x0 = NULL)
```

Arguments

`x` A design matrix (n x p). The rows represent observations and the columns represent predictor variables.

`y` A vector of the response variable.

tau	A quantile level, a number strictly between 0 and 1.
h	A univariate bandwidth. If not specified, the bandwidth is estimated using either "rule" or "CV". See method below for details.
method	A character string specifying the method to select the bandwidth, if it is missing. Use "rule" for the rule-of-thumb bandwidth of Yu and Jones (1994) or "CV" for the method of cross-validation.
x0	A single observation for which to perform the estimation. It needs to be a scalar (for a univariate predictor) or a vector (for a multivariate predictor). If x0 is missing, the estimation will be performed on the design matrix x.

Details

The function computes the local linear quantile regression fit for a specified quantile level τ at the design points of the matrix x or at a pre-specified point x0. The estimation is based on a standard normal kernel and a univariate bandwidth. The bandwidth, if not specified by the user, is defined using either the rule-of-thumb given by Yu and Jones (1994) or the cross-validation criterion.

The estimation applies to univariate and multivariate predictor variables. For the latter, the local linear fit uses the multivariate standard normal kernel. Note that if the estimation is performed at a pre-specified point x0, then x0 should be a scalar (for univariate predictor) or a vector (for multivariate predictor).

Value

llqr computes the local linear τth conditional quantile function of y given x and returns:

ll_est:	The estimated function value at the design points x or, if specified, at the point x0.
h:	The bandwidth for the local linear quantile regression fit. If not specified by the user, h is estimated using either the rule-of-thumb given by Yu and Jones (1994) or the cross-validation criterion.

Warning

The user needs to be careful about the bandwidth selection. When the dimension of the predictor variable is large compared to the sample size, local linear fitting meets the 'curse of dimensionality' problem. In situations like that, the bandwidth selected by the rule-of-thumb or the cross- validation criterion might be small and lead to a sparse neighborhood. This will cause the function to fail. For these cases, we advice the user to specify a bandwidth in the function. See the last example below.

References

Yu, K., and Jones, M.C. (1998), Local linear quantile regression. *Journal of the American Statistical Association*, 93, 228-237.

Examples

```
# Example 1
# estimate the function at a specific quantile level for simulated data
set.seed(1234)
```

```
n <- 100
x <- rnorm(n)
error <- rnorm(n)
y <- (x + 1)^3 + 0.1 * (x - 2)^3 + error
tau <- 0.5
plot(x, y, main = tau)
points(x, llqr(x, y, tau = tau)$ll_est, col = 'red', pch = 16)

# Example 2
# estimate the function at a point x0
set.seed(1234)
n <- 100
x <- rnorm(n)
error <- rnorm(n)
y <- (x + 1)^3 + 0.1 * (x - 2)^3 + error
tau <- 0.5
x0 <- 1
llqr(x, y, tau = tau, x0 = x0)

# Example 3
# estimate the function for different quantile levels
data(mcycle, package = "MASS")
attach(mcycle)
plot(times, accel, xlab = "milliseconds", ylab = "acceleration")
taus <- c(0.1, 0.25, 0.5, 0.75, 0.9)
for(i in 1:length(taus)) {
 fit <- llqr(times, accel, tau = taus[i])$ll_est
 lines(times, fit, lty = i)
}
legend(45, -50, c("tau=0.1","tau=0.25","tau=0.5","tau=0.75", "tau=0.9"),
    lty=1:length(taus))

# Example 4
# demonstrate a situation where the dimension of the predictor is large and
# the local linear fitting meets the 'curse of dimensionality' problem
set.seed(1234)
n <- 100
p <- 10
x <- matrix(rnorm(n * p), n, p)
error <- rnorm(n)
y <- 3 * x[, 1] + x[, 2] + error
tau <- 0.5
# use the following instead of llqr(x, y, tau = tau)
fit.alt <- llqr(x, y, tau = tau, h=1)
fit.alt
```

`llqrcv` *Cross-Validation for bandwidth selection of local linear quantile regression*

Description

llqrcv estimates the bandwidth necessary for the local linear fit of the τth conditional quantile of y given x. The estimation is performed using the Cross-Validation criterion.

Usage

```
llqrcv(x, y, tau = 0.5)
```

Arguments

x	A design matrix (n x p). The rows represent observations and the columns represent predictor variables.
y	A vector of the response variable.
tau	A quantile level, a number strictly between 0 and 1.

Details

A grid of bandwidth values is created and the local linear fit is estimated using all the data points except for one point, which is used to make the prediction. This procedure is repeated n times, where n is the number of observations. Then, the bandwidth is selected as the one with the smallest average error.

When the dimension of the predictor variable is large compared with the sample size, local linear fitting meets the 'curse of dimensionality' problem. In situations like that, the grid bandwidth values might be too small and cause the function to fail. For these cases, we advice the user to directly use the llqr function of the package and specify a bandwidth in the function.

Value

llqrcv returns the optimal bandwidth selected using Cross-Validation criterion for the local linear fit of the τth conditional quantile of y given x.

Examples

```
set.seed(1234)
n <- 100
x <- rnorm(n)
error <- rnorm(n)
y <- x^2 + error
tau <- 0.5
llqrcv(x, y, tau = tau)
```

ValAR *Value-at-Risk estimation using the central quantile subspace*

Description

ValAR estimates the one-step ahead τth Value-at-Risk for a vector of returns.

Usage

```
ValAR(y, p, tau, movwind = NULL, chronological = TRUE)
```

Arguments

y	A vector of returns (1 x n).
p	An integer for the number of past observations to be used as predictor variables. This will form the n x p design matrix.
tau	A quantile level, a number strictly between 0 and 1. Commonly used choices are 0.01, 0.025, and 0.05.
movwind	An optional integer number for the moving window. If not specified, a default value of min(250, n - p) will be used. If specified, the moving window should be an integer between p and n - p. Typical values for moving windows correspond to one or two years of return values. If the user wants to use all observations to fit the model, then the moving window should be equal to n - p. Note that, the number n - p comes from the fact that the full data set starts from the (p + 1)th observation since we use the last p observations for prediction.
chronological	A logical operator to indicate whether the returns are in standard chronological order (from oldest to newest). The default value is TRUE. If the returns are in reverse chronological order, the function will rearrange them.

Details

The function calculates the τth Value-at-Risk of the next time occurrence, i.e., that number such that the probability that the returns fall below its negative value is τ. The parameter τ is typically chosen to be a small number such as 0.01, 0.025, or 0.05. By definition, the negative value of the τth Value-at-Risk is the τth conditional quantile. Therefore, the estimation is performed using a local linear conditional quantile estimation. However, prior to this nonparametric estimation, a dimension reduction technique is performed to select linear combinations of the predictor variables.

Specifically, the user provides a vector of returns y (usually log-returns) and an integer p for the number of past observations to be used as the predictor variables. The function then forms the m x p design matrix x, where m is the number of used observations. For example, m can be n - p if the user wants to use all observations, or m can be equal to the moving window (default value is min(250, n - p)). Value-at-Risk is then defined as the negative value of the τth conditional quantile of y given x. However, to aid the nonparametric estimation of the τth conditional quantile, the cqs function is applied to estimate the fewest linear combinations of the predictor x that contain all the information available on the conditional quantile function. Finally, the llqr function is applied

to estimate the local linear conditional quantile of y using the extracted directions as the predictor variables.

For more details on the method and for an application to the Bitcoin data, see Christou (2020). Also, see Christou and Grabchak (2019) for a thorough comparison between the proposed methodology and commonly used methods.

Value

ValAR returns the one-step ahead τth Value-at-Risk.

References

Christou, E. (2020) Central quantile subspace. *Statistics and Computing*, 30, 677–695.

Christou, E., Grabchak, M. (2019) Estimation of value-at-risk using single index quantile regression. *Journal of Applied Statistics*, 46(13), 2418–2433.

Examples

```
# estimate the one-step ahead Value-at-Risk with default moving window
data(edhec, package = "PerformanceAnalytics")
y <- as.vector(edhec[, 1]) # Convertible Arbitrage
p <- 5 # use the 5 most recent observations as predictor variables
tau <- 0.05
ValAR(y, p, tau) # the data is already in standard chronological order

# compare it with the historical Value-at-Risk calculation
PerformanceAnalytics::VaR(y, 0.95, method = 'historical')
```

Index