Package: boostrq (via r-universe)

September 2, 2024

Type Package

Title Boosting Regression Quantiles

Version 1.0.0

Description Boosting Regression Quantiles is a component-wise boosting algorithm, that embeds all boosting steps in the well-established framework of quantile regression. It is initialized with the corresponding quantile, uses a quantile-specific learning rate, and uses quantile regression as its base learner. The package implements this algorithm and allows cross-validation and stability selection.

License GPL (>= 2)

URL https://github.com/stefanlinner/boostrq

BugReports https://github.com/stefanlinner/boostrq/issues

Encoding UTF-8

LazyData true

Depends mboost, stabs, stats, parallel

Imports quantreg, checkmate

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://stefanlinner.r-universe.dev

RemoteUrl https://github.com/stefanlinner/boostrq

RemoteRef HEAD

RemoteSha f2d3480f8db336564f31dda144a11f2b47a3f862

Contents

boostrq 2
brq 4
coef.boostrq 4
cvrisk.boostrq 5
fitted.boostrq 7
mstop.boostrq 7
predict.boostrq 8
print.boostrq 9
print.summary.boostrq 10
residuals.boostrq 11
risk.boostrq 11
selected.boostrq 12
stabsel.boostrq 13
summary.boostrq 15
update.boostrq 16
[.boostrq 17

Index **18**

boostrq *Fitting a boosting regression quantiles model*

Description

Component-wise functional gradient boosting algorithm to fit a quantile regression model.

Usage

```
boostrq(
  formula,
  data,
  mstop = 100,
  nu = NULL,
  tau = 0.5,
  offset = NULL,
  weights = NULL,
  oobweights = NULL,
  risk = "inbag",
  digits = 10,
  exact.fit = FALSE
)
```

Arguments

`formula` a symbolic description of the model to be fit.

`data` a data frame (or data.table) containing the variables stated in the formula.

`mstop` number of iterations, as integer

`nu` learning rate, as numeric

`tau` quantile parameter, as numeric

`offset` a numeric vector used as offset.

`weights` (optional) a numeric vector indicating which weights to used in the fitting process (default: all observations are equally weighted, with 1).

`oobweights` an additional vector of out-of-bag weights, which is used for the out-of-bag risk.

`risk` string indicating how the empirical risk should be computed for each boosting iteration. inbag leads to risks computed for the learning sample (i.e. observations with non-zero weights), oobag to risks based on the out-of-bag (i.e. observations with non-zero oobagweights).

`digits` number of digits the slope parameter different from zero to be considered the best-fitting component, as integer.

`exact.fit` logical, if set to TRUE the negative gradients of exact fits are set to 0.

Value

A (generalized) additive quantile regression model is fitted using the boosting regression quantiles algorithm, which is a functional component-wise boosting algorithm. The base-learner can be specified via the formula object. brq (linear quantile regression) and brqss(nonlinear quantile regression) are available base-learner.

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

boosted.rq$mstop()

boosted.rq$selection.freqs()

boosted.rq$coef()

boosted.rq$risk()
```

brq — *base learner for boosting linear regression quantiles*

Description

Base-learner for linear quantile regression.

Usage

```
brq(formula, method = "fn")
```

Arguments

`formula` a symbolic description of the base learner.

`method` the algortihm used to fit the quantile regression, the default is set to "fn", referring to the Frisch-Newton inferior point method. For more details see the documentation of quantreg::rq.

Value

brq returns a string, which is used to specifiy the formula in the fitting process.

Examples

```
brq(cyl * hp)
```

coef.boostrq — *estimated coefficients of boosting regression quantiles*

Description

estimated coefficients of boosting regression quantiles

Usage

```
## S3 method for class 'boostrq'
coef(object, which = NULL, aggregate = "sum", ...)
```

Arguments

`object`	object of class boostrq
`which`	a subset of base-learners
`aggregate`	a character specifying how to aggregate coefficients of single base learners. The default returns the coefficient for the final number of boosting iterations. "cumsum" returns a list with matrices (one per base-learner) with the cumulative coefficients for all iterations. "none" returns a list of matrices where the jth columns of the respective matrix contains coefficients of the base-learner of the jth boosting iteration.v "sum_aggr" ...
`...`	additional arguments passed to callies

Value

coef extracts the regression coefficients of the fitted boostrq model.

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

coef(boosted.rq, aggregate = "cumsum")
```

`cvrisk.boostrq` *Crossvalidation for boostrq*

Description

Crossvalidation for boostrq

Usage

```
## S3 method for class 'boostrq'
cvrisk(
  object,
  folds = mboost::cv(object$weights, type = "kfold"),
  grid = 0:mstop(object),
  papply = parallel::mclapply,
  mc.preschedule = FALSE,
  fun = NULL,
  ...
)
```

Arguments

`object` a boostrq object

`folds` a matrix indicating the weights for the k resampling iterations

`grid` a vetor of stopping parameters the empirical quantile risk is to be evaluated for.

`papply` (parallel) apply function, defaults to mclapply. To run sequentially (i.e. not in parallel), one can use lapply.

`mc.preschedule` preschedule tasks if are parallelized using mclapply (default: FALSE)? For details see mclapply.

`fun` if fun is NULL, the out-of-sample risk is returned. fun, as a function of object, may extract any other characteristic of the cross-validated models. These are returned as is.

`...` additional arguments passed to callies

Value

Cross-validated Boosting regression quantiles

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

set.seed(101)

cvk.out <-
cvrisk(
 boosted.rq,
 grid = 0:mstop(boosted.rq),
 folds = mboost::cv(boosted.rq$weights, type = "kfold", B = 5)
)

cvk.out

plot(cvk.out)

mstop(cvk.out)

boosted.rq[mstop(cvk.out)]
```

fitted.boostrq *fitted values of boosting regression quantiles*

Description

fitted values of boosting regression quantiles

Usage

```
## S3 method for class 'boostrq'
fitted(object, ...)
```

Arguments

`object`	object of class boostrq
`...`	additional arguments passed to callies

Value

fitted returns the fitted values of the fitted boostrq model.

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

fitted(boosted.rq)
```

mstop.boostrq *Current number of iterations of boostrq*

Description

Current number of iterations of boostrq

Usage

```
## S3 method for class 'boostrq'
mstop(object, ...)
```

Arguments

`object`	a boostrq object
`...`	additional arguments passed to callies

Value

current number of boosting iterations

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

mstop(boosted.rq)
```

`predict.boostrq` *Model predictions for boosting regression quantiles*

Description

Model predictions for boosting regression quantiles

Usage

```
## S3 method for class 'boostrq'
predict(object, newdata = NULL, which = NULL, aggregate = ”sum”, ...)
```

Arguments

`object`	a boostrq object
`newdata`	a data.frame (or data.table) including all covariates contained in the baselearners
`which`	a subset of base-learners
`aggregate`	a character specifying how to aggregate coefficients of single base learners. The default returns the coefficient for the final number of boosting iterations. "cumsum" returns a list with matrices (one per base-learner) with the cumulative coefficients for all iterations. "none" returns a list of matrices where the jth columns of the respective matrix contains coefficients of the base-learner of the jth boosting iteration.
`...`	additional arguments passed to callies

Value

predictions for the new data

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

predict.data <- data.frame(hp = 165, cyl = 6, am = 1, wt = 3.125)

predict(boosted.rq, newdata = predict.data)
```

print.boostrq *printing boosting regression quantiles*

Description

printing boosting regression quantiles

Usage

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

Arguments

x	object of class boostrq
...	additional arguments passed to callies

Value

print shows a dense representation of the boostrq model fit.

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

boosted.rq
```

print.summary.boostrq *Print result summaries for a boostrq object*

Description

Print result summaries for a boostrq object

Usage

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

Arguments

x	a summary.boostrq object
...	additional arguments passed to callies

Value

printing the result summaries for a boostrq object including the print-information, estimated coefficients, and selection frequencies

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

summary(boosted.rq)
```

residuals.boostrq *residuals of boosting regression quantiles*

Description

residuals of boosting regression quantiles

Usage

```
## S3 method for class 'boostrq'
residuals(object, ...)
```

Arguments

object	object of class boostrq
...	additional arguments passed to callies

Value

residuals returns the residuals of the fitted boostrq model.

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

residuals(boosted.rq)
```

risk.boostrq *Empirical Quantile Risk of boostrq Object*

Description

Empirical Quantile Risk of boostrq Object

Usage

```
## S3 method for class 'boostrq'
risk(object, ...)
```

Arguments

`object`	a boostrq object
`...`	additional arguments passed to callies

Value

numeric vector containing the respective empirical quantile risk of the different boosting iterations.

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

risk(boosted.rq)
```

`selected.boostrq` *Extract indices of selected base learners*

Description

Extract indices of selected base learners

Usage

```
## S3 method for class 'boostrq'
selected(object, ...)
```

Arguments

`object`	a boostrq object
`...`	additional arguments passed to callies

Value

an index vector indicating the selected base learner in each iteration

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

selected(boosted.rq)
```

stabsel.boostrq *Stability Selection for boosting regression quantiles*

Description

Stability Selection for boosting regression quantiles

Usage

```
## S3 method for class 'boostrq'
stabsel(
  x,
  cutoff,
  q,
  PFER,
  grid = 0:mstop(x),
  folds = stabs::subsample(x$weights, B = B),
  B = ifelse(sampling.type == "MB", 100, 50),
  assumption = "unimodal",
  sampling.type = "SS",
  papply = parallel::mclapply,
  verbose = TRUE,
  ...
)
```

Arguments

x a fitted model of class "boostrq"

cutoff cutoff between 0.5 and 1. Preferably a value between 0.6 and 0.9 should be used

q number of (unique) selected componenents (base-learners) that are selected in each subsample.

PFER upper bound for the per-family error rate. This specifies the amount of falsely selected base-learners, which is tolerated.

`grid` a numeric vector of the form 0:m.

`folds` a weight matrix with number of rows equal to the number of observations. Usually one should not change the default here as subsampling with a fraction of 1/2 is needed for the error bounds to hold.

`B` umber of subsampling replicates. Per default, we use 50 complementary pairs for the error bounds of Shah & Samworth (2013) and 100 for the error bound derived in Meinshausen & Buehlmann (2010). As we use B complementray pairs in the former case this leads to 2B subsamples.

`assumption` Defines the type of assumptions on the distributions of the selection probabilities and simultaneous selection probabilities. Only applicable for sampling.type = "SS". For sampling.type = "MB" we always use code"none".

`sampling.type` use sampling scheme of of Shah & Samworth (2013), i.e., with complementarty pairs (sampling.type = "SS"), or the original sampling scheme of Meinshausen & Buehlmann (2010).

`papply` (parallel) apply function, defaults to mclapply. To run sequentially (i.e. not in parallel), one can use lapply.

`verbose` logical (default: TRUE) that determines wether warnings should be issued.

`...` additional arguments passed to callies

Value

An object of class stabsel.

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl) + brq(hp) + brq(am) + brq(wt) + brq(drat),
 data = mtcars,
 mstop = 600,
 nu = 0.1,
 tau = 0.5
)

stabsel_parameters(
 q = 3,
 PFER = 1,
 p = 5,
 sampling.type = "SS",
 assumption = "unimodal"
)


set.seed(100)
brq.stabs <-
stabsel(
 x = boosted.rq,
 q = 3,
 PFER = 1,
 sampling.type = "SS",
 assumption = "unimodal"
)

brq.stabs
```

summary.boostrq *Result summaries for a boostrq object*

Description

Result summaries for a boostrq object

Usage

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

Arguments

`object`	a boostrq object
`...`	additional arguments passed to callies

Value

result summaries for a boostrq object including the print-information, estimated coefficients, and selection frequencies

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

summary(boosted.rq)
```

`update.boostrq` *Update and Re-fit a boostrq model*

Description

Update and Re-fit a boostrq model

Usage

```
## S3 method for class 'boostrq'
update(object, weights, oobweights, risk, ...)
```

Arguments

`object` a boostrq object

`weights` (optional) a numeric vector indicating which weights to used in the fitting process (default: all observations are equally weighted, with 1).

`oobweights` an additional vector of out-of-bag weights, which is used for the out-of-bag risk.

`risk` string indicating how the empirical risk should be computed for each boosting iteration. inbag leads to risks computed for the learning sample (i.e. observations with non-zero weights), oobag to risks based on the out-of-bag (i.e. observations with non-zero oobagweights).

`...` additional arguments passed to callies

Value

a re-fitted boostrq model

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

update(
boosted.rq,
weights = c(rep(1, 30), 0, 0),
oobweights = c(rep(0, 30), 1,1),
risk = "oobag"
)
```

[.boostrq *Updating number of iterations*

Description

Updating number of iterations

Usage

```
## S3 method for class 'boostrq'
x[i, return = TRUE, ...]
```

Arguments

x	a boostrq object
i	desired number of boosting iterations
return	TRUE, if the result should be returned
...	additional arguments passed to callies

Value

a boostrq object with the updated number of iterations

Examples

```
boosted.rq <-
boostrq(
 formula = mpg ~ brq(cyl * hp) + brq(am + wt),
 data = mtcars,
 mstop = 200,
 nu = 0.1,
 tau = 0.5
)

boosted.rq[500]
```

Index

[.boostrq, 17

boostrq, 2
brq, 4

coef.boostrq, 4
cvrisk.boostrq, 5

fitted.boostrq, 7

mstop.boostrq, 7

predict.boostrq, 8
print.boostrq, 9
print.summary.boostrq, 10

residuals.boostrq, 11
risk.boostrq, 11

selected.boostrq, 12
stabsel.boostrq, 13
summary.boostrq, 15

update.boostrq, 16