Package 'Qindex'

October 12, 2022

Type Package

Title Quantile-Based Predictors for Survival Outcome

Version 0.1.0

Date 2022-09-14

Author Misung Yi [aut, cph],
Tingting Zhan [aut, cre, cph] (<https://orcid.org/0000-0001-9971-4844>),
Inna Chervoneva [aut, cph]

Maintainer Tingting Zhan <tingtingzhan@gmail.com>

Description Select optimal quantile-based predictors for survival outcome, to include the means to handle dichotomizing the quantiles, mean-signal-intensity or other continuous markers, and to obtain estimates for dichotomized predictors by Bootstrap-based bias correction.

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.1

Encoding UTF-8

License GPL-2

Depends R (>= 4.2),

Language en-US

Imports stats, boot, rpart, survival, matrixStats

Suggests knitr

Collate '0info.R' 'pkg_data_doc.R' 'optQp.R' 'sampleQp.R' 'BBC_dichotom.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2022-09-19 08:36:11 UTC

R topics documented:

Qindex-package 2
BBC_dichotom 2
celldata 4
coxph_optQ 5
dichotom_optQ 5
evalQp 6
optQp 7
print.evalQp 7
sampleQp 8
summary.evalQp 9

Index **10**

`Qindex-package` *Qindex*

Description

Optimize quantile-based predictors for survival outcome (and includes the means to handle dichotomizing the MSI or other continuous markers).

Details

Will provide later

References

paper in review

`BBC_dichotom` *Bootstrap Bias Correction for Dichotomization*

Description

Bootstrap-based optimism correction for dichotomizing quantiles

Usage

```
BBC_dichotom(formula, data, optQ, R, seed, ...)
```

Arguments

formula	formula
data	data.frame
optQ	character scalar
R	see boot
seed	integer scalar, random seeds for generating repeated samples, see set.seed
...	..

Details

The bootstrap optimism correction procedure (1) is performed as described for a general model selection in (2). First, R bootstrap samples are drawn with replacement from the main sample. In each bootstrap sample, the recursive partitioning tree model is used to establish an objective data-driven optimal cut-point for a specific optimal quantile. The cut-point from the current bootstrap sample is used to compute the log odds-ratio (OR) and/or hazards ratio (HR) for dichotomized quantile predictor in the current bootstrap sample (**bootstrap performance**) and in the main sample (**test performance**), and the optimism in log OR/HR estimation is computed as the difference between log OR/HR for "Bootstrap performance" and for "Test performance". The median optimism estimate is computed as the median of optimism estimates over all bootstrap samples. The cutpoint for dichotomizing each selected optimal quantile is also established in the main sample and its "apparent performance" is computed as the log OR/HR for dichotomized quantile in the univariate logistic regression or Cox models. Finally, the optimism-corrected performance estimate is computed by subtracting the mean optimism estimate from the apparent performance estimate.

Value

..

Examples

```
Ki67_Qps = sampleQp(data = Ki67, endpoint = 'RECURRENCE', subjID = 'PATIENT_ID',
  Qpredictor = 'Marker', probs = seq(from = .05, to = .95, by = .05))
sapply(Ki67_Qps, FUN = class)
head(Ki67_Qps$Marker)
(Ki67_eval = evalQp(RECURRENCE ~ Marker, data = Ki67_Qps, seeds = 1:100))
lapply(Ki67_eval, FUN = dim)
summary(Ki67_eval)
Ki67_opt = optQp(Ki67_eval, n = 2L)
colnames(Ki67_opt$Marker) = paste0('Ki67_', colnames(Ki67_opt$Marker))

PR_Qps = sampleQp(data = PR, endpoint = 'RECURRENCE', subjID = 'PATIENT_ID',
  Qpredictor = 'Marker', probs = seq(from = .05, to = .95, by = .05))
head(PR_Qps$Marker)
PR_eval = evalQp(RECURRENCE ~ Marker, data = PR_Qps, seeds = 1:100)
PR_opt = optQp(PR_eval, n = 3L)
colnames(PR_opt$Marker) = paste0('PR_', colnames(PR_opt$Marker))
```

```
cellOpt0 = merge(Ki67_opt, PR_opt, by = setdiff(names(Ki67_opt), 'Marker'),
  suffixes = c('.Ki67', '.PR'))
cellOpt = within(cellOpt0, expr = {
  Marker = cbind(Marker.Ki67, Marker.PR)
  Marker.Ki67 = Marker.PR = NULL
})
dim(cellOpt)
names(cellOpt)
head(cellOpt$Marker)

mod = BBC_dichotom(RECURRENCE ~ NodeSt + Tstage, data = cellOpt,
  optQ = 'Marker', R = 100, seed = 1)
names(mod)
mod$ucox_dictom
mod$mcox_dictom
mod$median.optimism
```

`celldata` *Subset of Ki67 and PR Data*

Description

A subset of Ki67 cell data containing 623 patients and PR cell data containing 631 patients. A total of 572 patients are in common.

Usage

```
Ki67

PR
```

Format

PATIENT_ID factor, ..

tissueID character, ..

RECURRENCE Surv, ...

Marker numeric, ...

all other columns

An object of class `data.frame` with 467603 rows and 20 columns.

coxph_optQ ..

Description

..

Usage

```
coxph_optQ(
  formula,
  data,
  optQ,
  dQ = dichotom_optQ(edp = eval(formula[[2L]], envir = data), optQ = data[[optQ]])
)
```

Arguments

formula	..
data	..
optQ	character scalar
dQ	returned object from dichotom_optQ

Value

coxph_optQ returns a numeric vector of the concatenated coefficients of **univariable model** (which is returned in attribute 'ucox') and coefficients **multivariable model** (which is returned in attribute 'mcox').

dichotom_optQ *Helper function dichotom_optQ*

Description

..

Usage

```
dichotom_optQ(edp, optQ, control = rpart.control(minbucket = 40))
```

Arguments

edp	Surv object, the endpoint
optQ	matrix of optimal quantiles
control	see rpart

Value

dichotom_optQ returns a logical matrix of the same dimension and dimension names as argument `optQ`, as dichotomized using the first node of rpart as threshold. The thresholds (per column) are returned as an numeric vector in attribute `'thres'`.

`evalQp` *Optimal Quantile Predictor*

Description

From the sequence of sample quantiles, selects the optimal quantile that has highest predictive value for a given survival outcome

Usage

```
evalQp(formula, data, seeds, pct_train = 0.8, minbucket.p = 0.5)
```

Arguments

`formula`	formula, currently only supports one Surv endpoint and one predictor of the markers. See details of parameter `data`.
`data`	data.frame, with at least • one Surv column as the survival outcome • one matrix column as the predictor of quantile sequence
`seeds`	integer vector of random seeds for generating repeated samples, see set.seed
`pct_train`	numeric scalar, proportion of the training set, default `.8`
`minbucket.p`	(for rpart function) the minimum number of observations in any terminal **leaf** node. If only one of `minbucket` or `minsplit` is specified, the code either sets `minsplit` to `minbucket*3` or `minbucket` to `minsplit/3`, as appropriate.

Value

evalQp returns a evalQp object, which is a list with two matrix elements `thresholds` and `coefs`.

- `thresholds` is ...
- `coefs` is ...

Examples

```
# see ?optQp
```

`optQp` *optQp*

Description

optQp

Usage

```
optQp(x, n = 3L, ...)
```

Arguments

`x`	evalQp object
`n`	integer scalar
`...`	additional parameters, not currently in use

Value

optQp returns ..

Examples

```
# see ?BBC_dichotom
```

`print.evalQp` *Print evalQp Object*

Description

..

Usage

```
## S3 method for class 'evalQp'
print(x, n = 3L, ...)
```

Arguments

`x`	..
`n`	..
`...`	..

Details

print.evalQp simply calls summary.evalQp ..

Value

print.evalQp does not have a returned value.

`sampleQp` *Cluster-specific sample quantiles for clustered data*

Description

For a user-supplied sequence of percentiles, calculates sample quantiles in each independent cluster of observations.

Usage

```
sampleQp(
  data,
  endpoint = "RECURRENCE",
  subjID = "PATIENT_ID",
  sampleID,
  Qpredictor = "Marker",
  probs = seq(from = 0.05, to = 0.95, by = 0.05),
  aggregate_sampleID_quantile = c("median", "mean", "min", "Q1", "Q3", "max"),
  ...
)
```

Arguments

`data`	data.frame
`endpoint`	character scalar, column name for the Surv endpoint
`subjID`	character scalar, column name for subject/patient ID
`sampleID`	(optional) character scalar, column name for `sampleID`, which is nested within `subjID`. When `sampleID` is missing, the analysis is performed with only one-level cluster of `subjID`
`Qpredictor`	character scalar, column name of the predictor variable
`probs`	numeric vector, probabilities of the quantiles
`aggregate_sampleID_quantile`	character vector, will be implemented in the next release
`...`	additional parameters, currently not in use ..

Value

sampleQp returns data set with one row per subjID, Np columns with Q(p), p=1,..., Np, and additional subject level variables designated to be kept in the data set. The code should allow for multiple clusters (clusterID) per subject (subjID) and output the mean or other summary statistic (according to user input for type of summary statistic) Q(p) for every p. (*) Let us first make use of summary.default, which cover all options I can think about for now

Examples

```
# see ?BBC_dichotom
```

---

summary.evalQp *Summary Information of evalQp Object*

Description

Summary method for optQp Object

Usage

```
## S3 method for class 'evalQp'
summary(object, FUN = median, ...)
```

Arguments

object	evalQp object
FUN	median (default) or mean
...	..

Value

summary.evalQp summarize ...

Index

∗ **datasets**
    celldata, 4
∗ **package**
    Qindex-package, 2

BBC_dichotom, 2
boot, *3*

celldata, 4
character, *3–5*, *8*
coxph_optQ, *5*, 5

data.frame, *3*, *6*, *8*
dichotom_optQ, *5*, 5, *6*

evalQp, *6*, 6, *7*, *9*

factor, *4*
formula, *3*, *6*

integer, *3*, *6*, *7*

Ki67 *(*celldata*)*, 4

list, *6*
logical, *6*

matrix, *5*, *6*
mean, *9*
median, *9*

numeric, *4–6*, *8*

optQp, *7*, 7, *9*

PR *(*celldata*)*, 4
print.evalQp, 7, *8*

Qindex-package, 2
quantile, *8*

rpart, *5*, *6*

sampleQp, 8, *9*
set.seed, *3*, *6*
summary.default, *9*
summary.evalQp, *8*, *9*, 9
Surv, *4–6*, *8*