Package 'obliqueRSF'

November 2, 2018

Title Oblique Random Forests for Right-Censored Time-to-Event Data

Version 0.1.0

Description
Oblique random survival forests incorporate linear combinations of input variables into random survival forests (Ishwaran, 2008 <DOI:10.1214/08-AOAS169>). Regularized Cox proportional hazard models (Simon, 2016 <DOI:10.18637/jss.v039.i05>) are used to identify optimal linear combinations of input variables.

Depends R (>= 3.5.0)

Imports Rcpp, pec, data.table, stats, missForest, purrr, glmnet, survival, dplyr, rlang, prodlim, ggthemes, tidyr, ggplot2, scales

License GPL-3

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

NeedsCompilation yes

Author Byron Jaeger [aut, cre]

Maintainer Byron Jaeger <bcjaeger@uab.edu>

Repository CRAN

Date/Publication 2018-11-02 17:40:03 UTC

R topics documented:

---

obliqueRSF *Oblique Random Survival Forests*

Description

Oblique random survival forest are ensembles for right-censored survival data that incorporate linear combinations of input variables into random survival forests (see Ishwaran et al., 2008 <doi:10.1214/08-AOAS169>). Regularized Cox proportional hazard models (see Simon et al., 2016 <doi:10.18637/jss.v039.i05>) identify optimal linear combinations of input variables in each recursive partitioning step while building survival trees (see Bou-hamad et al., 2011 <doi: 10.1214/09-SS047>).

Author(s)

Byron C. Jaeger <bcjaeger@uab.edu>

ORSF *Grow an oblique random survival forest (ORSF)*

Description

Grow an oblique random survival forest (ORSF)

Usage

```
ORSF(data, alpha = 0.5, ntree = 100, time = "time",
  status = "status", eval_times = NULL, features = NULL,
  min_events_to_split_node = 5, min_obs_to_split_node = 10,
  min_obs_in_leaf_node = 5, min_events_in_leaf_node = 1, nsplit = 25,
  max_pval_to_split_node = 0.5, mtry = ceiling(sqrt(ncol(data) - 2)),
  dfmax = mtry, use.cv = FALSE, verbose = TRUE, random_seed = NULL)
```

Arguments

data	The data used to grow the forest.
alpha	The elastic net mixing parameter. A value of 1 gives the lasso penalty, and a value of 0 gives the ridge penalty. If multiple values of alpha are given, then a penalized model is fit using each alpha value prior to splitting a node.
ntree	The number of trees to grow.
time	A character value indicating the name of the column in the data that measures time.
status	A character value indicating the name of the column in the data that measures participant status. A value of zero indicates censoring and a value of 1 indicates that the event occurred.

`eval_times` A numeric vector holding the time values where ORSF out-of-bag predictions should be computed and evaluated.

`features` A character vector giving the names of columns in the data set that will be used as features. If NULL, then all of the variables in the data apart from the time and status variable are treated as features. None of these names should contain special characters or spaces.

`min_events_to_split_node`
The minimum number of events required to split a node.

`min_obs_to_split_node`
The minimum number of observations required to split a node.

`min_obs_in_leaf_node`
The minimum number of observations in child nodes.

`min_events_in_leaf_node`
The minimum number of events in child nodes.

`nsplit` The number of random cut-points assessed for each variable.

`max_pval_to_split_node`
The maximum p-value corresponding to the log-rank test for splitting a node. If the p-value exceeds this cut-point, the node will not be split.

`mtry` Number of variables randomly selected as candidates for splitting a node. The default is the square root of the number of features.

`dfmax` Maximum number of variables used in a linear combination for node splitting.

`use.cv` if TRUE, cross-validation is used to identify optimal values of lambda, a hyper-parameter in penalized regression. if FALSE, a set of candidate lambda values are used. The set of candidate lambda values is built by picking the maximum value of lambda such that the penalized regression model has k degrees of freedom, where k is between 1 and mtry.

`verbose` If verbose=TRUE, then the ORSF function will print output to console while it grows the tree.

`random_seed` If a number is given, then that number is used as a random seed prior to growing the forest. Use this seed to replicate a forest if needed.

Value

An oblique random survival forest.

Examples

```
data("pbc",package='survival')
pbc$status[pbc$status>=1]=pbc$status[pbc$status>=1]-1
pbc$id=NULL
fctrs<-c('trt','ascites','spiders','edema','hepato','stage')
for(f in fctrs)pbc[[f]]=as.factor(pbc[[f]])
pbc=na.omit(pbc)

orsf=ORSF(data=pbc,ntree=5)
```

pdplot *Plot partial variable dependence using an oblique random survival forest*

Description

Plot partial variable dependence using an oblique random survival forest

Usage

```
pdplot(object, xvar, xlab = NULL, xvar_units = NULL, xvals = NULL,
  nxpts = 10, ytype = "nonevent", event_lab = "death",
  nonevent_lab = "survival", fvar = NULL, flab = NULL,
  flvls = NULL, time_units = "years", xlvls = NULL,
  sub_times = NULL, separate_panels = TRUE, color_palette = "Dark2")
```

Arguments

object	an ORSF object (i.e. object returned from the ORSF function)
xvar	a string giving the name of the x-axis variable
xlab	the label to be printed describing the x-axis variable
xvar_units	the unit of measurement for the x-axis variable. For example, age is usually measured in years.
xvals	a vector containing the values that partial dependence will be computed with.
nxpts	instead of specifying xvals, you can specify how many points on the x-axis you would like to plot predicted responses for, and a set of nxpts equally spaced percentile values from the distribution of xvar will be used.
ytype	String. Use 'event' if you would like to plot the probability of the event, and 'nonevent' if you prefer to plot the probability of a non-event.
event_lab	string that describes the event
nonevent_lab	string that describes a non-event.
fvar	a string indicating a variable to facet the plot with
flab	a label describing the facet variable.
flvls	the labels to be printed describing the facet variable. For a facet variable with k categories, flab should be a vector with k labels, given in the same order as the levels of the facet variable.
time_units	the unit of time, e.g. days, since baseline.
xlvls	A character vector with descriptions of each category in the x-variable. This is only relevant if x is categorical.
sub_times	a vector of times to compute predicted survival probabilities. Note that the eval_times from the ORSF object are used to compute predictions, and sub_times must be a subset of those times.

separate_panels
: true or false. If true, the plot will display predictions in two separate panels, determined by the facet variable.

color_palette
: Palette to use for colors in the figure. Options are Diverging (BrBG, PiYG, PRGn, PuOr, RdBu, RdGy, RdYlBu, RdYlGn, Spectral), Qualitative (Accent, Dark2, Paired, Pastel1, Pastel2, Set1, Set2, Set3), Sequential (Blues, BuGn, BuPu, GnBu, Greens, Greys, Oranges, OrRd, PuBu, PuBuGn, PuRd, Purples, RdPu, Reds, YlGn, YlGnBu, YlOrBr, YlOrRd), and viridis.

Value

A ggplot2 object showing partial dependence according to the oblique random survival forest object.

Examples

```
## Not run:
data("pbc",package='survival')
pbc$status[pbc$status>=1]=pbc$status[pbc$status>=1]-1
pbc$time=pbc$time/365.25
pbc$id=NULL
fctrs<-c('trt','ascites','spiders','edema','hepato','stage')
for(f in fctrs)pbc[[f]]=as.factor(pbc[[f]])
pbc=na.omit(pbc)

orsf=ORSF(data=pbc, eval_time=1:10,ntree=30)

pdplot(object=orsf, xvar='bili', xlab='Bilirubin',
       xvar_units='mg/dl', sub_times=10)

## End(Not run)
```

---

predict.orsf *Compute predictions using an oblique random survival forest.*

Description

Compute predictions using an oblique random survival forest.

Usage

```
## S3 method for class 'orsf'
predict(object, newdata, times, ...)
```

Arguments

object
: An object fitted using the ORSF function.

newdata
: A data frame containing observations to predict.

times
: A vector of times in the range of the response variable, e.g. times when the response is a survival object, at which to return the survival probabilities.

...
: Other arguments passed to or from other functions.

Value

A matrix of survival probabilities containing 1 row for each observation and 1 column for each value in times.

Examples

```
data("pbc",package='survival')
pbc$status[pbc$status>=1]=pbc$status[pbc$status>=1]-1
pbc$id=NULL
fctrs<-c('trt','ascites','spiders','edema','hepato','stage')
for(f in fctrs)pbc[[f]]=as.factor(pbc[[f]])
pbc=na.omit(pbc)

orsf=ORSF(data=pbc,ntree=5)
times=seq(365, 365*4,length.out = 10)

predict(orsf,newdata=pbc[c(1:5),],times=times)
```

---

predictSurvProb.orsf *Compute predictions using an oblique random survival forest.*

Description

Compute predictions using an oblique random survival forest.

Usage

```
## S3 method for class 'orsf'
predictSurvProb(object, newdata, times, ...)
```

Arguments

object	A fitted model from which to extract predicted survival probabilities
newdata	A data frame containing predictor variable combinations for which to compute predicted survival probabilities.
times	A vector of times in the range of the response variable, e.g. times when the response is a survival object, at which to return the survival probabilities.
...	Additional arguments that are passed on to the current method.

Value

A matrix of survival probabilities containing 1 row for each observation and 1 column for each value in times.

Examples

```
## Not run:
data("pbc",package='survival')
pbc$status[pbc$status>=1]=pbc$status[pbc$status>=1]-1
pbc$id=NULL
fctrs<-c('trt','ascites','spiders','edema','hepato','stage')
for(f in fctrs)pbc[[f]]=as.factor(pbc[[f]])
pbc=na.omit(pbc)

orsf=ORSF(data=pbc,ntree=30)
times=seq(365, 365*4,length.out = 10)

predict(orsf,newdata=pbc[c(1:5),],times=times)

## End(Not run)
```

`print.orsf` *Grow an oblique random survival forest (ORSF)*

Description

Grow an oblique random survival forest (ORSF)

Usage

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

Arguments

`x`	an ORSF object (i.e. the object returned from the ORSF function)
`...`	additional arguments passed to print

Value

A printed summary of the oblique random survival forest.

Examples

```
## Not run:
data("pbc",package='survival')
pbc$status[pbc$status>=1]=pbc$status[pbc$status>=1]-1
pbc$id=NULL
fctrs<-c('trt','ascites','spiders','edema','hepato','stage')
for(f in fctrs)pbc[[f]]=as.factor(pbc[[f]])
pbc=na.omit(pbc)

orsf=ORSF(data=pbc,ntree=30)
print(orsf)
```

```
## End(Not run)
```

`theme_Publication` *Plot variable dependence using an oblique random survival forest*

Description

Plot variable dependence using an oblique random survival forest

Usage

```
theme_Publication(base_size = 16)
```

Arguments

`base_size`	how big to make the text

`vdplot` *Plot variable dependence using an oblique random survival forest*

Description

Plot variable dependence using an oblique random survival forest

Usage

```
vdplot(object, xvar, include.hist = TRUE, include.points = FALSE,
  ptsize = 0.75, ytype = "nonevent", event_lab = "death",
  nonevent_lab = "survival", fvar = NULL, flab = NULL,
  time_units = "years", xlab = xvar, xvar_units = NULL,
  xlvls = NULL, sub_times = NULL, se.show = FALSE)
```

Arguments

`object`	an ORSF object (i.e. object returned from the ORSF function)
`xvar`	a string giving the name of the x-axis variable
`include.hist`	if true, a histogram showing the distribution of values for the x-axis variable will be included at the bottom of the plot.
`include.points`	if true, the predictions for each observation are plotted along with a smoothed population estimate. Note that points are always included if xvar is categorical.
`ptsize`	only relevant if include.points = TRUE. The size of the points in the plot are determined by this numeric value.
`ytype`	String. Use 'event' if you would like to plot the probability of the event, and 'nonevent' if you prefer to plot the probability of a non-event.

event_lab	string that describes the event
nonevent_lab	string that describes a non-event.
fvar	(optional) a string indicating a variable to facet the plot with
flab	the labels to be printed describing the facet variable. For a facet variable with k categories, flab should be a vector with k labels, given in the same order as the levels of the facet variable.
time_units	the unit of time, e.g. days, since baseline.
xlab	the label to be printed describing the x-axis variable
xvar_units	the unit of measurement for the x-axis variable. For example, age is usually measured in years.
xlvls	a character vector giving the labels that correspond to categorical xvar. This does not need to be specified if xvar is continuous.
sub_times	the times you would like to plot predicted values for. If left unspecified, the ORSF function will use all of the times in oob_times.
se.show	if true, standard errors of the population estimate will be included in the plot.

Value

A ggplot2 object

Examples

```
## Not run:
data("pbc",package='survival')
pbc$status[pbc$status>=1]=pbc$status[pbc$status>=1]-1
pbc$time=pbc$time/365.25
pbc$id=NULL
fctrs<-c('trt','ascites','spiders','edema','hepato','stage')
for(f in fctrs)pbc[[f]]=as.factor(pbc[[f]])
pbc=na.omit(pbc)

orsf=ORSF(data=pbc, eval_time=5, ntree=30)

vdplot(object=orsf, xvar='bili', xlab='Bilirubin', xvar_units='mg/dl')

## End(Not run)
```

Index