Package 'rocTree'

August 1, 2020

Title Receiver Operating Characteristic (ROC)-Guided Classification and Survival Tree

Version 1.1.1

Description Receiver Operating Characteristic (ROC)-guided survival trees and ensemble algorithms are implemented, providing a unified framework for tree-structured analysis with censored survival outcomes. A time-invariant partition scheme on the survivor population was considered to incorporate time-dependent covariates. Motivated by ideas of randomized tests, generalized time-dependent ROC curves were used to evaluate the performance of survival trees and establish the optimality of the target hazard/survival function. The optimality of the target hazard function motivates us to use a weighted average of the time-dependent area under the curve (AUC) on a set of time points to evaluate the prediction performance of survival trees and to guide splitting and pruning. A detailed description of the implemented methods can be found in Sun et al. (2019) <arXiv:1809.05627>.

Depends R (>= 3.5.0)

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

LinkingTo Rcpp, RcppArmadillo

Imports DiagrammeR (>= 1.0.0), data.tree (>= 0.7.5), graphics, stats, survival (>= 2.38), ggplot2, MASS, flexsurv, Rcpp

URL http://github.com/stc04003/rocTree

BugReports http://github.com/stc04003/rocTree/issues

NeedsCompilation yes

Author Yifei Sun [aut],
Mei-Cheng Wang [aut],
Sy Han Chiou [aut, cre]

Maintainer Sy Han Chiou <schiou@utdallas.edu>

Repository CRAN

Date/Publication 2020-08-01 09:40:02 UTC

R topics documented:

`rocTree-package` *rocTree:Receiver Operating Characteristic (ROC)-Guided Classification Survival Tree and Ensemble.*

Description

The `rocTree` package uses a Receiver Operating Characteristic (ROC) guided classification algorithm to grow prune survival trees and ensemble.

Introduction

The `rocTree` package provides implementations to a unified framework for tree-structured analysis with censored survival outcomes. Different from many existing tree building algorithms, the `rocTree` package incorporates time-dependent covariates by constructing a time-invariant partition scheme on the survivor population. The partition-based risk prediction function is constructed using an algorithm guided by the Receiver Operating Characteristic (ROC) curve. The generalized time-dependent ROC curves for survival trees show that the target hazard function yields the highest ROC curve. The optimality of the target hazard function motivates us to use a weighted average of the time-dependent area under the curve on a set of time points to evaluate the prediction performance of survival trees and to guide splitting and pruning. Moreover, the `rocTree` package also offers a novel ensemble algorithm, where the ensemble is on unbiased martingale estimating equations.

Methods

The package contains functions to construct ROC-guided survival trees and ensemble through the main function `rocTree`.

Author(s)

Maintainer: Sy Han Chiou <schiou@utdallas.edu>

Authors:

- Yifei Sun <ys3072@cumc.columbia.edu>
- Mei-Cheng Wang <mcwang@jhu.edu>

See Also

rocTree

---

export_Surv Surv *function imported from* survival

Description

This function is imported from the survival package. See Surv.

---

plot.rocTree *Plotting an* rocTree *object*

Description

Plots an rocTree object. The function returns a dgr_graph object that is rendered in the RStudio Viewer or survival/hazard estimate at terminal nodes.

Usage

```
## S3 method for class 'rocTree'
plot(
  x,
  output = c("graph", "visNetwork"),
  digits = 4,
  tree = 1L,
  rankdir = c("TB", "BT", "LR", "RL"),
  shape = "ellipse",
  nodeOnly = FALSE,
  savePlot = FALSE,
  file_name = "pic.pdf",
  file_type = "pdf",
  type = c("tree", "survival", "hazard"),
  ...
)
```

Arguments

x	an object of class "rocTree", usually returned by the rocTree function.
output	a string specifying the output type; graph (the default) renders the graph using the grViz function, and visNetwork renders the graph using the visnetwork function.
digits	the number of digits to print.
tree	is an optional integer specifying the n^{th} tree in the forest to print.

rankdir	is a character string specifying the direction of the tree flow. The available options are top-to-bottom ("TB"), bottom-to-top ("BT"), left-to-right ("LR"), and right-to-left ("RL"); the default value is "TB".
shape	is a character string specifying the shape style. Some of the available options are "ellipse", "oval", "rectangle", "square", "egg", "plaintext", "diamond", and "triangle". The default value is "ellipse".
nodeOnly	is a logical value indicating whether to display only the node number; the default value is "TRUE".
savePlot	is a logical value indicating whether the plot will be saved (exported); the default value is "FALSE".
file_name	is a character string specifying the name of the plot when "savePlot = TRUE". The file name should include its extension. The default value is "pic.pdf"
file_type	is a character string specifying the type of file to be exported. Options for graph files are: "png", "pdf", "svg", and "ps". The default value is "pdf".
type	is an optional character string specifying the type of plots to produce. The available options are "tree" for plotting survival tree (default), "survival" for plotting the estimated survival probabilities for the terminal nodes, and "hazard" for plotting the estimated hazard for the terminal nodes. The dgr_graph options are available only when type = tree.
...	arguments to be passed to or from other methods.

See Also

See rocTree for creating rocTree objects.

Examples

```
## Not run:
data(simDat)
fit <- rocTree(Surv(Time, death) ~ z1 + z2, id = id, data = simDat, ensemble = FALSE)
## Plot tree
plot(fit)
## Plot survival estimates at terminal nodes
plot(fit, type = "survival")
## Plot hazard estimates at terminal nodes
plot(fit, type = "haz")

## End(Not run)
```

---

predict.rocTree *Predicting based on a* rocTree *model.*

Description

The function gives predicted values with a rocTree fit.

Usage

```
## S3 method for class 'rocTree'
predict(object, newdata, type = c("survival", "hazard"), control = list(), ...)
```

Arguments

object	is an rocTree object.
newdata	is an optional data frame in which to look for variables with which to predict. If omitted, the fitted predictors are used. If the covariate observation time is not supplied, covariates will be treated as at baseline.
type	is an optional character string specifying whether to predict the survival probability or the cumulative hazard rate.
control	a list of control parameters. See 'details' for important special features of control parameters. See rocTree for more details.
...	for future developments.

Value

Returns a data.frame of the predicted survival probabilities or cumulative hazard.

Examples

```
data(simDat)
fit <- rocTree(Surv(Time, death) ~ z1 + z2, id = id, data = simDat, ensemble = FALSE)

## testing data
newdat <- data.frame(Time = sort(unique(simDat$Time)),
                     z2 = runif(1))
newdat$z1 <- 1 * (newdat$Time < median(newdat$Time))
head(newdat)

## Predict survival
pred <- predict(fit, newdat)
plot(pred)

## Predict hazard
pred <- predict(fit, newdat, type = "hazard")
plot(pred)
```

---

print.rocTree *Printing an* rocTree *object*

Description

The function prints an rocTree object. It is a method for the generic function print of class "rocTree".

Usage

```
## S3 method for class 'rocTree'
print(x, digits = 5, tree = NULL, ...)
```

Arguments

x	an rocTree object.
digits	the number of digits of numbers to print.
tree	an optional integer specifying the n^{th} tree in the forest to print. The function prints the contents of an rocForest object by default, if a tree is not specified.
...	for future development.

Examples

```
data(simDat)

## Fitting a pruned survival tree
rocTree(Surv(Time, death) ~ z1 + z2, id = id, data = simDat, ensemble = FALSE)

## Fitting a unpruned survival tree
rocTree(Surv(Time, death) ~ z1 + z2, id = id, data = simDat, ensemble = FALSE,
        control = list(numFold = 0))

## Not run:
## Fitting the ensemble algorithm (default)
rocTree(Surv(Time, death) ~ z1 + z2, id = id, data = simDat, ensemble = TRUE)

## End(Not run)
```

---

rocTree *Roc-guided survival trees*

Description

Fits a "rocTree" model.

Usage

```
rocTree(
  formula,
  data,
  id,
  subset,
  ensemble = TRUE,
  splitBy = c("dCON", "CON"),
  control = list()
)
```

Arguments

formula	is a formula object, with the response on the left of a '~' operator, and the terms on the right. The response must be a survival object returned by the 'Surv' function.
data	is an optional data frame in which to interpret the variables occurring in the 'formula'.
id	is an optional vector used to identify the longitudinal observations of subject's id. The length of 'id' should be the same as the total number of observations. If 'id' is missing, each row of 'data' represents a distinct observation from a subject and all covariates are treated as a baseline covariate.
subset	is an optional vector specifying a subset of observations to be used in the fitting process.
ensemble	is an optional logical value. If TRUE (default), ensemble methods will be fitted. Otherwise, the survival tree will be fitted.
splitBy	is a character string specifying the splitting algorithm. The available options are 'CON' and 'dCON' corresponding to the splitting algorithm based on the total concordance measure or the difference in concordance measure, respectively. The default value is 'dCON'.
control	a list of control parameters. See 'details' for important special features of control parameters.

Details

The argument "control" defaults to a list with the following values:

tau is the maximum follow-up time; default value is the 90th percentile of the unique observed survival times.

maxTree is the number of survival trees to be used in the ensemble method (when ensemble = TRUE).

maxNode is the maximum node number allowed to be in the tree; the default value is 500.

numFold is the number of folds used in the cross-validation. When numFold > 0, the survival tree will be pruned; when numFold = 0, the unpruned survival tree will be presented. The default value is 10.

h is the smoothing parameter used in the Kernel; the default value is tau / 20.

minSplitTerm is the minimum number of baseline observations in each terminal node; the default value is 15.

minSplitNode is the minimum number of baseline observations in each splitable node; the default value is 30.

disc is a logical vector specifying whether the covariates in formula are discrete (TRUE) or continuous (FALSE). The length of disc should be the same as the number of covariates in formula. When not specified, the rocTree() function assumes continuous covariates for all.

K is the number of time points on which the concordance measure is computed. A less refined time grids (smaller K) generally yields faster speed but a very small K is not recommended. The default value is 20.

Value

An object of S4 class "rocTree" representig the fit, with the following components:

References

Sun Y. and Wang, M.C. (2018+). ROC-guided classification and survival trees. *Technical report*.

See Also

See print.rocTree and plot.rocTree for printing and plotting an rocTree, respectively.

Examples

```
data(simDat)

## Fitting a pruned survival tree
rocTree(Surv(Time, death) ~ z1 + z2, id = id, data = simDat, ensemble = FALSE)

## Fitting a unpruned survival tree
rocTree(Surv(Time, death) ~ z1 + z2, id = id, data = simDat, ensemble = FALSE,
        control = list(numFold = 0))

## Not run:
## Fitting the ensemble algorithm (default)
rocTree(Surv(Time, death) ~ z1 + z2, id = id, data = simDat, ensemble = TRUE)

## End(Not run)
```

---

simDat *Simulated dataset for demonstration*

Description

A simulated data frame with variables

id subjects identification

Time observation times

death event/death indicator; 1 = an event (death) is recorded

z1 baseline covariate generated from a standard uniform distribution

z2 baseline covariate generated from a standard uniform distribution (independent from z1

Format

A data frame with 5050 rows and 5 variables.

Details

The sample dataset is generated by set.seed(1); simu(100,0,1.3). See simu for details of the simu function.

simu *Function to generate simulated data used in the manuscript.*

Description

This function is used to generate simulated data under various settings. Let Z be a p-dimensional vector of possible time-dependent covariates and β be the vector of regression coefficient. The survival times (T) are generated from the hazard function specified as follow:

Scenario 1.1 Proportional hazards model:

$$\lambda(t|Z) = \lambda_0(t)e^{-0.5Z_1+0.5Z_2-0.5Z_3...+0.5Z_{10}},$$

where $\lambda_0(t) = 2t$.

Scenario 1.2 Proportional hazards model with noise variable:

$$\lambda(t|Z) = \lambda_0(t)e^{2Z_1+2Z_2+0Z_3+...+0Z_{10}},$$

where $\lambda_0(t) = 2t$.

Scenario 1.3 Proportional hazards model with nonlinear covariate effects:

$$\lambda(t|Z) = \lambda_0(t)e^{[2\sin(2\pi Z_1)+2|Z_2-0.5|]},$$

where $\lambda_0(t) = 2t$.

Scenario 1.4 Accelerated failure time model:

$$\log(T) = -2 + 2Z_1 + 2Z_2 + \epsilon,$$

where ϵ follows $N(0, 0.5^2)$.

Scenario 1.5 Generalized gamma family:

$$T = e^{\sigma\omega},$$

where $\omega = \log(Q^2 g)/Q$, g follows Gamma(Q^{-2}, 1), $\sigma = 2Z_1, Q = 2Z_2$.

Scenario 2.1 Dichotomous time dependent covariate with at most one change in value:

$$\lambda(t|Z(t)) = \lambda_0(t)e^{2Z_1(t)+2Z_2},$$

where $Z_1(t)$ is the time-dependent covariate: $Z_1(t) = \theta I(t \geq U_0) + (1-\theta)I(t < U_0)$, ,$\theta$ is a Bernoulli variable with equal probability, and U_0 follows a uniform distribution over $[0, 1]$.

Scenario 2.2 Dichotomous time dependent covariate with multiple changes:

$$\lambda(t|Z(t)) = e^{2Z_1(t)+2Z_2},$$

where $Z_1(t) = \theta[I(U_1 \leq t < U_2) + I(U_3 \leq t)] + (1-\theta)[I(t < U_1) + I(U_2 \leq t < U_3)]$, θ is a Bernoulli variable with equal probability, and $U_1 \leq U_2 \leq U_3$ are the first three terms of a stationary Poisson process with rate 10.

Scenario 2.3 Proportional hazard model with a continuous time dependent covariate:

$$\lambda(t|Z(t)) = 0.1e^{Z_1(t)+Z_2},$$

where $Z_1(t) = kt + b$, k and b are independent uniform random variables over $[1, 2]$.

Scenario 2.4 Non-proportional hazards model with a continuous time dependent covariate:

$$\lambda(t|Z(t)) = 0.1 \cdot [1 + \sin\{Z_1(t) + Z_2\}],$$

where $Z_1(t) = kt + b$, k and b follow independent uniform distributions over $[1, 2]$.

Scenario 2.5 Non-proportional hazards model with a nonlinear time dependent covariate:

$$\lambda(t|Z(t)) = 0.1 \cdot [1 + \sin\{Z_1(t) + Z_2\}],$$

where $Z_1(t) = 2kt \cdot \{I(t > 5) - 1\} + b$, k and b follow independent uniform distributions over $[1, 2]$. The censoring times are generated from an independent uniform distribution over $[0, c]$, where c was tuned to yield censoring percentages of 25

Usage

```
simu(n, cen, scenario, summary = FALSE)

trueHaz(dat)

trueSurv(dat)
```

Arguments

`n`	an integer value indicating the number of subjects.
`cen`	is a numeric value indicating the censoring percentage; three levels, 0%, 25%, 50%, are allowed.
`scenario`	can be either a numeric value or a character string. This indicates the simulation scenario noted above.
`summary`	a logical value indicating whether a brief data summary will be printed.
`dat`	is a data.frame prepared by `simu`.

Value

`simu` returns a `data.frame`. The returned data.frame consists of columns:

id is the subject id.

Y is the observed follow-up time.

death is the death indicator; death = 0 if censored.

z1–z10 is the possible time-independent covariate.

k, b, U are the latent variables used to generate $Z_1(t)$ in Scenario 2.1 – 2.5.

The returned data.frame can be supply to `trueHaz` and `trueSurv` to generate the true cumulative hazard function and the survival function, respectively.

Examples

```
set.seed(1)
simu(10, 0.25, 1.2, TRUE)

set.seed(1)
simu(10, 0.50, 2.2, TRUE)
```

Index