Package 'DStree'

February 19, 2015

Type Package

Title Recursive Partitioning for Discrete-Time Survival Trees

Version 1.0

Date 2014-08-19

Author Peter Mayer, Denis Larocque, Matthias Schmid

Maintainer Peter Mayer <mayerptr@gmail.com>

Description Building discrete-time survival trees and bagged trees based on the functionalities of the rpart package. Splitting criterion maximizes the likelihood of a covariate-free logistic discrete time hazard model.

License GPL-2

NeedsCompilation no

Depends rpart,pec,Ecdat

Imports rpart.plot,survival, Rcpp

LazyData no

LinkingTo Rcpp

Repository CRAN

Date/Publication 2014-08-19 22:25:27

R topics documented:

---

bag *Bagging Discrete-Time Survival Trees*

Description

Function for bagging discrete-time survival trees.

Usage

```
bag(formula, data, status, nBoot = 10, minbucket = 40)
```

Arguments

`formula`	a formula with a response but no interaction terms. The response variable represents the observed survival times.
`status`	integer/string representing the column number/name of the 'status' variable in the data frame. 'status' must be a binary integer variable indicating whether the event occured (=1) or whether the event was censored/did not occur (=0).
`data`	data frame that contains all variables stated in the `formula` argument, as well as the `status` variable.
`nBoot`	an integer referring to the number of bootstrap replications.
`minbucket`	the minimum number of observations in any terminal node.

Details

The `bag` function grows `nBoot` unpruned trees from bootstrap samples. For each fitted tree the median survival time, as well as the predicted survival probabilities and hazard rates of the individuals in `data` are returned.

Value

The `bag` function returns an object of class 'DStreebag' which is a list containing the following objects:

The vector MedSurv contains the predicted, averaged median survival times. The matrices Surv and Haz contain the predicted, averaged probabilities and hazard rates, respectively, where each row refers to an individual and each column refers to an observed time point (1,2,...). The list element 'trees' contains the `nBoot` unpruned trees of class 'DStree'. The list elements 'minbucket' and 'nboot' have the same meaning as above.

References

Bou-Hamad I., Larocque D., Ben-Ameur H., Masse L. C., Vitaro F. and Tremblay R. E. (2009), Discrete-Time Survival Trees. *Canadian Journal of Statistics 37 (1), 17-32*.

Hothorn T., Lausen B., Benner A. and Radespiel-Troeger M. (2004), Bagging Survival Trees. *Statistics in Medicine 23 (1), 77-91*.

Examples

```
data(cost)
## Discretize observed days to years
d.cost <- dis.cost(cost)# Bagging Tree
pred <- bag(time~prevStroke+age+sex+alcohol+smoke,status="status",data=d.cost,nBoot=50)

# Predicted, averaged probabilities and median survival times for each individual
pred$MedSurv
pred$Surv
pred$Haz
```

`dis.cost` *Convert observed time points in Copenhagen Stroke Study data set*

Description

This function rounds up the observed number of days in the `cost` data set (contained in R package **pec**) to full years.

Usage

```
dis.cost(cost)
```

Arguments

`cost`	data set

Value

A data frame that contains the same variables as the `cost` data frame in R package **pec**. The only difference between `cost` and the output of `dis.cost` are the units of the event times. In `cost`, event times are measured in days whereas `dis.cost` results in a data frame with event times measured in years.

References

Joergensen HS, Nakayama H, Reith J, Raaschou HO, and Olsen TS. Acute stroke with atrial fibrillation. The Copenhagen Stroke Study. Stroke, 27(10):1765-9, 1996.

Mogensen UB, Ishwaran H, and Gerds TA. Evaluating random forests for survival analysis using prediction error curves. Journal of Statistical Software, 50(11), 2012.

Examples

```
data(cost)
d.cost <- dis.cost(cost)
```

DStree *Fit a discrete-time survival tree*

Description

This function builds decision trees for discrete, right-censored survival data. The fitted tree estimates the hazard and survival probabilities for every terminal node, as well as the median survival time.

Usage

```
DStree(formula, status, data, control = control, weights = NULL)
```

Arguments

formula	a formula with response but no interaction terms. The response variable represents the observed survival times.
status	integer/string representing the column number/name of the 'status' variable in the data frame. 'status' must be a binary integer variable indicating whether the event occured (=1) or whether the event was censored/did not occur (=0).
data	data frame that contains all variables stated in the formula argument, as well as the 'status' variable.
control	a list of options that control the specification of the DStree algorithm. See DStree.control.
weights	a vector of optional case weights.

Value

An object of class DStree. For details see DStree.object.

References

Bou-Hamad I., Larocque D., Ben-Ameur H., Masse L. C., Vitaro F. and Tremblay R. E. (2009), Discrete-Time Survival Trees. *Canadian Journal of Statistics 37 (1), 17-32*.

Examples

```
##Build tree
fit<- DStree(spell~ui+age+tenure+logwage,status="censor1",data=UnempDur,control=list(cp=0))
plot(fit)
```

DStree.control *Control arguments for DStree algorithm*

Description

Various paramters that control the specification of the DStree algorithm.

Usage

```
DStree.control(minsplit = 20L, minbucket = round(minsplit/3), cp = 0.005,
  maxcompete = 4L, maxdepth = 30, maxsurrogate = 0)
```

Arguments

minsplit	the minimum number of observations that must exist in a node in order for a split to be attempted.
minbucket	the minimum number of observations in any terminal 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.
cp	complexity parameter. Any split that does not decrease the overall lack of fit by a factor of cp is not attempted. This means that the overall deviance must decrease by cp at each step. The main role of this parameter is to save computing time by pruning off splits that are not worthwhile by definition. Essentially, the user informs the program that any split which does not improve the fit by cp will likely be pruned off.
maxcompete	the number of competitor splits retained in the output. It is useful to know not just which split was chosen, but which variable came in second, third, etc.
maxdepth	Set the maximum depth of any node of the final tree, with the root node counted as depth 0. Values greater than 30 DStree will give nonsense results on 32-bit machines.
maxsurrogate	the number of surrogate splits retained in the output. If this is set to zero the compute time will be reduced, since approximately half of the computational time (other than setup) is used in the search for surrogate splits.

DStree.object *DStree Object*

Description

This class of objects is returned by the DStree function to represent a fitted decision tree for right-censored survival data.

Value

`frame` data frame with one row for each node in the tree. The `row.names` of `frame` contain the (unique) node numbers that follow a binary ordering indexed by node depth. Columns of `frame` include `var`, a factor giving the names of the variables used in the split at each node (leaf nodes are denoted by the level `"<leaf>"`), `n`, the number of observations reaching the node, `wt`, the sum of case weights for observations reaching the node, `dev`, the deviance of the node, `yval`, the estimated value of the median survival time at the node, and `splits`, a two-column matrix of left and right split labels for each node. Also included in the frame are `complexity`, the complexity parameter at which this split will collapse, `ncompete`, the number of competitor splits recorded, and `nsurrogate`, the number of surrogate splits recorded. `yval2` contains in its first columns the fitted hazard probabilities and in last the fitted survival probabilities (see `surv`).

`where` an integer vector of the same length as the number of observations in the root node, containing the row number of `frame` corresponding to the leaf node that each observation falls into.

`call` an image of the call that produced the object, but with the arguments all named and with the actual formula included as the formula argument. To re-evaluate the call, say `update(tree)`.

`terms` an object of class `c("terms", "formula")` (see `terms.object`) summarizing the formula. Used by various methods, but typically not of direct relevance to users.

`splits` a numeric matrix describing the splits: only present if there are any. The row label is the name of the split variable, and columns are `count`, the number of observations (which are not missing and are of positive weight) sent left or right by the split (for competitor splits this is the number that would have been sent left or right had this split been used, for surrogate splits it is the number missing the primary split variable which were decided using this surrogate), `ncat`, the number of categories or levels for the variable (`+/-1` for a continuous variable), `improve`, which is the improvement in deviance given by this split, or, for surrogates, the concordance of the surrogate with the primary, and `index`, the numeric split point. The last column `adj` gives the adjusted concordance for surrogate splits. For a factor, the `index` column contains the row number of the csplit matrix. For a continuous variable, the sign of `ncat` determines whether the subset `x < cutpoint` or `x > cutpoint` is sent to the left.

`csplit` an integer matrix. (Only present only if at least one of the split variables is a factor or ordered factor.) There is a row for each such split, and the number of columns is the largest number of levels in the factors. Which row is given by the `index` column of the `splits` matrix. The columns record `1` if that level of the factor goes to the left, `3` if it goes to the right, and `2` if that level is not present at this node of the tree (or not defined for the factor).

`method` character string: the method used to grow the tree. Since `DStree` is based on a user-defined split function of `rpart` the method is always `"user"`.

`cptable` a matrix of information on the optimal prunings based on a complexity parameter.

`variable.importance`
: a named numeric vector giving the importance of each variable. (Only present if there are any splits.) When printed by `summary.DStree` these are rescaled to add to 100.

`numresp`
: integer number of responses; the number of levels for a factor response.

`parms, control`
: a record of the arguments supplied, which defaults filled in.

`functions`
: the `summary` and split function used.

`ordered`
: a named logical vector recording for each variable if it was an ordered factor.

`na.action`
: (where relevant) information returned by `model.frame` on the special handling of `NA`s derived from the `na.action` argument.

`names`
: a string vector of size two, which denotes the column names of the observed time points and of the `status` variable.

`wt`
: a numeric vector of equal length as the number of rows of the dataset, which denotes the optional case weigths, defined in the `weights` argument.

Optional components include the model frame (`model`), the matrix of predictors (`x`) and the response variable (`y`) used to construct the `DStree` object.

Structure

The above components must be included in a legitimate `DStree` object.

See Also

`DStree`.

`plot.DStree` *Plot a DStree Object*

Description

This function plots a DStree object on the current graphic device. It visualizes the fitted tree as well as the estimated survival/hazard probabilites for every terminal node.

Usage

```
## S3 method for class 'DStree'
plot(x, prob = "surv", select = NULL, ...)
```

Arguments

`x`
: a fitted object of class `"DStree"`

`prob`
: a string that indicates which probability should be plotted. `"surv"` draws survival probabilities, `"haz"` draws hazard probabilities.

select	a vector of strings that indicate terminal leaves for which the survival/hazard probabilities should be drawn. If not specified probabilities for every terminal leaf are drawn.
...	extra prp arguments to modify the appearance of the tree (from R package **rpart.plot**)

Examples

```
data(cost)
# Discretize observed days to years
d.cost<-dis.cost(cost)

# Grow tree
fit <- DStree(time~prevStroke+age+sex+alcohol+smoke,status="status",data=d.cost)

#Plot tree and survival probabilities
plot(fit)
#survival probabilities are in the first plot

#Plot tree and hazard probabilities for terminal leaves 4 and 15
plot(fit,prob="haz",select=c("2","7"))
```

`predict.DStree` *Predictions from a fitted DStree Object*

Description

Returns the predicted median survival time, hazard and survival probabilities from a fitted `DStree` object.

Usage

```
## S3 method for class 'DStree'
predict(object, data, ...)
```

Arguments

object	a fitted object of class "DStree"
data	data frame containing the values at which predictions are required. The predictors referred to in the right side of `formula(object)` as well as the 'status' variable must be present by name in `data`. If missing, the fitted values are returned.
...	further arguments passed to or from other methods.

Value

A named list with the following elements: The vector MedSurv contains the predicted median survival times of the observations in data. The matrices Surv and Haz contain the predicted probabilities and hazard rates, respectively, where each row refers to an individual and each column refers to an observed time point (1,2,...).

References

Bou-Hamad I., Larocque D.,Ben-Ameur H., Masse L. C., Vitaro F. and Tremblay R. E. (2009), Discrete-Time Survival Trees. *Canadian Journal of Statistics 37 (1), 17-32*.

Examples

```
data(cost)
# Convert observed days to years
d.cost<-dis.cost(cost)

# Train Data
Train<-d.cost[1:300,]
# Test Data
Test<-d.cost[301:518,]

fit <- DStree(time~prevStroke+age+sex+alcohol+smoke,status="status",data=d.cost)

#Predictions from Test Data
predict(fit,Test)
```

---

predict.DStreebag *Prediction from bagged Discrete-Time Survival Trees*

Description

Predict the median survival time, hazard and survival probabilities from a DStreebag object.

Usage

```
## S3 method for class 'DStreebag'
predict(object, data, ...)
```

Arguments

object	a fitted object of class DStreebag
data	data frame containing the values at which predictions are required. The predictors referred to in the right side of formula(object) as well as the 'status' variable must be present by name in newdata.
...	further arguments passed to or from other methods.

Value

A named list with the following elements: The vector MedSurv contains the predicted median survival times of the observations in data. The matrices Surv and Haz contain the predicted probabilities and hazard rates, respectively, where each row refers to an individual and each column refers to an observed time point (1,2,...).

References

Bou-Hamad I., Larocque D.,Ben-Ameur H., Masse L. C., Vitaro F. and Tremblay R. E. (2009), Discrete-Time Survival Trees. *Canadian Journal of Statistics 37 (1), 17-32*.

Hothorn T., Lausen B., Benner A. and Radespiel-Troeger M. (2004), Bagging Survival Trees. *Statistics in Medicine 23 (1), 77-91*.

Examples

```
data(cost)
## Discretize observed days to years
d.cost<-dis.cost(cost)

# Bagging Tree
bag <- bag(time~prevStroke+age+sex+alcohol+smoke,status="status",data=d.cost[1:330,],nBoot=10)
pred <- predict(bag,newdata=d.cost[331:518,])
pred$MedSurv
```

---

prune.DStree *Prune a fitted survival tree*

Description

The prune function evaluates and prunes a survival tree that has been fitted by DStree. Different criteria can be used for evaluation (e.g. Brier Score or AIC). The result of the prune function is the optimal subtree (of class DStree) with regard to the chosen criterium, as well as various performance measures that were obtained from the subtrees during analysis. The returned performance measures are the Brier Score, the deviance, and an information criterion defined by gamma.

Usage

```
## S3 method for class 'DStree'
prune(tree, data, gamma = 2, which, ...)
```

Arguments

tree	fitted model of class "DStree". This object is assumed to be the result of some function that produces an object with the same named components as that returned by the DStree function.
data	optional data frame that is used to evaluate the fit of the tree. The predictors referred to in formula(tree) as well as the 'status' variable must be present by name in data. If missing, the data frame that was used to grow the tree is used.

gamma	optional positive integer value that determines the user defined information criterion. Setting gamma = 2 corresponds to the AIC criterium.
which	An optional string that determines which performance criteria should be computed from the subtrees. One of "CRIT", "DEV" or "BRIER". If which is missing the "BRIER" score is chosen.
...	further arguments passed to or from other methods.

Details

The subtrees are the cost-minimzing subtrees in terms of deviance for given complexity parameters of the fitted tree. See Therneau et al (2013) p.12-13.

Value

prune returns one DStree object and four vectors of length equal to the number of subtrees:

- nsplit number of splits for every subtree
- CRIT value of the user defined information criterion (underlying formula: *CRIT* = *deviance* + *gamma* * |*terminal leaves*| * |*time periods*|).
- DEV deviance
- BRIER Integrated Brier Score, see Hothorn et al. (2004)
- pruned.fit optimal subtree regarding the choosen criterium specified in which

References

Hothorn T., Lausen B., Benner A. and Radespiel-Troeger M. (2004), Bagging Survival Trees. *Statistics in medicine 23 (1), 77-91*.

Therneau T. and Atkinson E., An introduction to recursive partitioning using the RPART routines, *Technical Report 61, Section of Biostatistics, Mayo Clinic, Rochester.*

Examples

```
data(cost)
## Discretize observed days to years
d.cost <- dis.cost(cost)

##Build tree
tree <- DStree(time~prevStroke+age+sex+alcohol+smoke,status="status",data=d.cost)

# Determine subtree with minimum AIC
prunedtree <- prune(tree,d.cost,which="CRIT")
prunedtree$prunedfit

# Visualize AIC/Deviance of subtrees
plot(prunedtree$nsplit,prunedtree$CRIT)
plot(prunedtree$nsplit,prunedtree$DEV)
```

snip *Cost-complexity Pruning of a DStree Object*

Description

Determines a nested sequence of subtrees of the supplied DStree object by recursively snipping off the least important splits, based on the complexity parameter (cp).

Usage

```
snip(tree, cp)
```

Arguments

tree	a fitted object of class "DStree"
cp	complexity parameter to which the DStree object will be trimmed

Value

A new DStree object that is trimmed to the value cp.

Examples

```
data(cost)
d.cost <- dis.cost(cost)

fit <- DStree(time~prevStroke+age+sex+alcohol+smoke,status="status",data=d.cost)

sfit<-snip(fit,cp=0.02)
plot(sfit) #plot smaller DStree object
```

summary.DStree *Summarize a Fitted DStree Object*

Description

Returns a detailed listing of a fitted DStree object.

Usage

```
## S3 method for class 'DStree'
summary(object, cp = 0, digits = getOption("digits"), file, ...)
```

Arguments

object	fitted model object of class "DStree". This is assumed to be the result of some function that produces an object with the same named components as that returned by the DStree function.
digits	Number of significant digits to be used in the result.
cp	trim nodes with a complexity of less than cp from the listing.
file	write the output to a given file name. (Full listings of a tree are often quite long).
...	arguments to be passed to or from other methods.

Details

The function prints the call, the table shown by printcp, the variable importance (summing to 100) and details for each node.

See Also

summary, DStree.object, printcp.

Examples

```
data(UnempDur)
tree <- DStree(spell~ui+age+tenure+logwage,status="censor1",data=UnempDur)
summary(tree)
```

---

surv *Print fitted Survival and Hazard Probabilites of a DStree Object*

Description

This function prints the fitted survival and hazard probabilities of every terminal node of a DStree object.

Usage

```
surv(object)
```

Arguments

object	fitted model object of class "DStree". This is assumed to be the result of some function that produces an object with the same named components as that returned by the DStree function.

Value

Two matrices containing Survival and Hazard probabilities, where each row denotes fitted probabilities in the respective terminal leaves. The first column refers to the probability of the first observed timepoint, the second column to the second timepoint, etc.

Examples

```
# Grow tree
tree <- DStree(spell~ui+age,status="censor1",data=UnempDur)

# Print fitted probabilites
surv(tree)
```

Index