Package: icrf (via r-universe)

November 1, 2024

Title Interval Censored Recursive Forests

Version 2.0.2

Date 2022-10-29

Depends R (>= 3.5.0), stats

Suggests survival, ggplot2

Description Implements interval censored recursive forests (ICRF) based on Cho, Jewell, and Kosorok (2021+). ICRF is a variant of random forests where the outcome variable is interval censored survival data. It can be used for usual right censored data and current status data as well. A recursion technique is used to improve accuracy and smoothed survival curves are provided.

Maintainer Hunyong Cho <hunyong.cho@gmail.com>

License GPL (>= 2)

NeedsCompilation yes

Date/Publication 2022-10-29 22:00:18 UTC

RoxygenNote 7.1.1

Author Hunyong Cho [aut, cre], Nicholas P Jewell [aut], Michael R Kosorok [aut], Leo Breiman [ctb] (Author of incldued randomForest C code), Adele Cutler [ctb] (Author of included randomForest Fortran code), Andy Liaw [ctb] (Author of included randomForest R code), Matthew Wiener [ctb] (Author of included randomForest R code), Alain Vandal Robert Gentleman [ctb] (Author of included Icens R code), Merck & Co. Inc. [cph] (Copyright holder of included randomForest R code), Alain Vandal Robert Gentleman [cph] (Copyright holder of included Icens R code), The R Foundation [cph] (Copyright holder of included ksmooth C code)

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

RemoteUrl https://github.com/cran/icrf

RemoteRef HEAD

RemoteSha 300ceee712e999918b04cce5c56ea75f9189429c

Contents

`getTree.icrf` *Extract a single tree from an icrf object*

Description

`getTree` 'extracts the structure of a tree from an `icrf` object.' Among `nfold` forests, the forest designated when implementing `icrf` will be considered. i.e., the `k`th tree of the last forest, when `returnBest = FALSE` or the tree of the best forest, when `returnBest = TRUE`, will be extracted. (Quoted statements are from `randomForest` by Liaw and Wiener unless otherwise mentioned.)

Usage

```
getTree(x, ...)

## S3 method for class 'icrf'
getTree(x, k = 1, labelVar = FALSE, ...)
```

Arguments

`x`	an `icrf` object.
`...`	not used.
`k`	'which tree to extract?'
`labelVar`	Splitting variables will be labelled with the original names when `labelVar = TRUE`. Otherwise they will be expressed as integers.

Details

'For numerical predictors, data with values of the variable less than or equal to the splitting point go to the left daughter node.'

'For categorical predictors, the splitting point is represented by an integer, whose binary expansion gives the identities of the categories that goes to left or right. For example, if a predictor has four

categories, and the split point is 13. The binary expansion of 13 is (1, 0, 1, 1) (because 13 = 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3), so cases with categories 1, 3, or 4 in this predictor get sent to the left, and the rest to the right.'

Value

'A matrix (or data frame, if labelVar=TRUE) with' (5 + number of time points) 'columns and number of rows equal to total number of nodes in the tree. The columns are:'

- left daughter: 'the row where the left daughter node is; 0 if the node is terminal'
- right daughter: 'the row where the right daughter node is; 0 if the node is terminal'
- split var: 'which variable was used to split the node; 0 if the node is terminal'
- split point: 'where the best split is; see Details for categorical predictor'
- status: 'is the node terminal' (-1) or not (-3)
- from the 6th to the last columns: the survival probability prediction for the node; each column represents the distinct time points. '0 if the node is not terminal'

Examples

```
library(survival) # for Surv()
data(rat2)
L = ifelse(rat2$tumor, 0, rat2$survtime)
R = ifelse(rat2$tumor, rat2$survtime, Inf)
# Note that this is a toy example. Use a larger ntree and nfold in practice.

set.seed(1)
rats.icrf <-
  icrf(Surv(L, R, type = "interval2") ~ dose.lvl + weight + male + cage.no,
       data = rat2, ntree = 10, nfold = 3)
getTree(rats.icrf, k = 2)
```

---

icrf *Interval Censored Recursive Forests (ICRF)*

Description

`icrf` implements the ICRF algorithm to estimate the conditional survival probability for interval censored survival data. (It can also be used for right-censored survival data and current status data.) `icrf` recursively builds random forests using the extremely randomized trees (ERT) algorithm and uses kernel smoothing in the time domain. This `icrf` package is built based on the `randomForest` package by Andy Liaw and Matthew Wiener. (Quoted statements are from `randomForest` by Liaw and Wiener unless otherwise mentioned.)

Usage

```
icrf(x, ...)

## Default S3 method:
icrf(
  x,
  L,
  R,
  tau = max(R[is.finite(R)]) * 1.5,
  bandwidth = NULL,
  quasihonesty = TRUE,
  initialSmoothing = TRUE,
  timeSmooth = NULL,
  xtest = NULL,
  ytest = NULL,
  nfold = 5L,
  ntree = 500L,
  mtry = ceiling(sqrt(p)),
 split.rule = c("Wilcoxon", "logrank", "PetoWilcoxon", "PetoLogrank", "GWRS", "GLR",
    "SWRS", "SLR"),
  ERT = FALSE,
  uniformERT = ERT,
  returnBest = sampsize < n,
  imse.monitor = 1,
  replace = !ERT,
  sampsize = ifelse(ERT, 0.95, 0.632) * n,
  nodesize = 6L,
  maxnodes = NULL,
  importance = FALSE,
  nPerm = 1,
  proximity,
  oob.prox = ifelse(sampsize == n & !replace, FALSE, proximity),
  do.trace = FALSE,
  keep.forest = is.null(xtest),
  keep.inbag = FALSE,
  ...
)

## S3 method for class 'formula'
icrf(
  formula,
  data = NULL,
  data.type = c("interval", "right", "currentstatus"),
  interval.label = c("L", "R"),
  right.label = c("T", "status"),
  currentstatus.label = c("monitor", "status"),
  ...,
  na.action = na.fail,
```

```
  epsilon = NULL
)

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

Arguments

x	a data frame or a matrix of predictors. x is not needed when formula is specified.
...	optional arguments to be passed to icrf.default.
L, R	the left and right end point of the interval. R should be greater than or equal to L. In case of equality, a small number epsilon (the smaller of minimum nonzero interval length and 1e-10) is added.
tau	the study end time. ([0, tau] is the window for the analysis.)
bandwidth	a positive number. The bandwidth of the kernel smoothing. For faster computing, set bandwidth = 0 for no smoothing.
quasihonesty	if TRUE, the terminal node prediction is given by the NPMLE of the interval data. If FALSE, the terminal node prediction is given by the average of the conditional probabilities (exploitative).
initialSmoothing	if TRUE, the initial survival curve used for interval-conditional survival probability estimate is smoothed using the Gaussian kernel.
timeSmooth	a numeric vector of time points at which the smoothed survival curves are estimated. It should be in an increasing order. If null, a set of distinct interval end points is used.
xtest	a dataset or matrix of predictors for the test dataset.
ytest	a true survival curve for the test set in a form of the dataframe or matrix. The number of rows is the same as xtest and each column corresponds to the time points of timeSmooth.
nfold	Number of forests to iterate. In practice, numbers between 5 and 10 is reasonable.
ntree	Number of trees to build within each forest. 'This should not be set to too small a number, to ensure that every input row gets predicted at least a few times.'
mtry	Number of candidate predictors tried at each split. The default value is sqrt(p) where p is number of variables in x.
split.rule	Splitting rules. See details. The default is "Wilcoxon", or equivalently "GWRS".
ERT	If ERT=TRUE ERT algorithm applies. If FALSE, a comprehensive greedy algorithm (Breiman's random forest algorithm) applies.
uniformERT	Only relevant when ERT=TRUE. If uniformERT=TRUE, random candidate cutpoints are selected using uniform distribution. If FALSE, random candidate cutpoints are chosen among the midpoints of two neighboring predictor values.

returnBest	If returnBest=TRUE, the survival curve estimate at the best iteration is returned. If FALSE, the estimate at the last iteration is returned. The best iteration is determined by the type of IMSE measures specified in imse.monitor. By default, returnBest=TRUE when the out-of-bag sample is available (sampsize < n).
imse.monitor	Which type of IMSE is used to monitor which fold is the best?
replace	Whether the cases are sampled with or without replacement?
sampsize	Size of random sampling.
nodesize	Each terminal node cannot be smaller than this value. 'Setting this number larger causes smaller trees to be grown (and thus take less time).'
maxnodes	Up to how many terminal nodes can a tree have? 'If not given, trees are grown to the maximum possible (subject to limits by nodesize). If set larger than maximum possible, a warning is issued.'
importance	If TRUE, variable importance measure will be computed.
nPerm	How many permutations (of OOB data) to do for variable importance assessment? 'Number larger than 1 gives slightly more stable estimate, but not very effective. Currently only implemented for regression.'
proximity	If TRUE, proximity measure among the cases is calculated.
oob.prox	If TRUE, proximity is calculated only on "out-of-bag" data.
do.trace	If TRUE, intermediate outputs are printed during the tree building procedure. 'If set to some integer, then running output is printed for every do.trace trees.'
keep.forest	'If set to FALSE, the forest will not be retained in the output object. If xtest is given, defaults to FALSE.'
keep.inbag	'Should an n by ntree matrix be returned that keeps track of which samples are "in-bag" in which trees (but not how many times, if sampling with replacement)'
formula, data.type, interval.label, right.label, currentstatus.label	a formula object, with the response in a Surv 'interval2' or cbind. Alternatively, the survival outcome may be omitted in the formula and the labels relevent to the survival outcome can be entered in either interval.label, right.label, or currentstatus.label with the data.type being specified.
data	a data frame that includes the intervals and the predictor values.
na.action	'a function to specify the action to be taken if NAs are found. (NOTE: If given, this argument must be named.)'
epsilon	A small positive value needed to discriminate the left and right interval end points for the uncensored data.

Details

Four split.rule options are available: Wilcoxon, logrank, PetoWilcoxon, PetoLogrank. The aliases are GWRS, GLR, SWRS, and SLR, respectively. The first two are generalized Wilcoxon-rank-sum test and generalized log-rank test proposed in Cho et al (2020+), and the latter two are score-based Wilcoxon-rank-sum test and score-based log-rank test proposed by Peto and Peto (1972) "Asymptotically efficient rank invariant test procedures."

Value

An `icrf` class object which contains the following components in a list:

An `icrf` class object which contains the following components in a list:

- callthe original call to `icrf`
- methodThe input values of `split.rule`, `ERT`, quasihonest, `bandwith`, and the subsample ratio (= `sampsize` / `n`)
- predictedthe estimated survival curves of the training set using out-of-bag samples.
- predictedNOthe estimated survival curves of the training set using non-out-of-bag samples.
- predictedNO.Smthe smoothed survival curves of the training set using non-out-of-bag samples.
- time.pointstime points at which the survival curves are estimated.
- time.points.smoothtime points at which the smoothed survival curves are estimated.
- imse.oobIntegrated mean squared error (IMSE) measured based on the out-of-bag samples
- imse.NOIntegrated mean squared error (IMSE) measured based on the non-out-of-bag samples
- oob.timesnumber of times for which each case was 'out-of-bag'
- importancean array of three matrices where each matrix has `nfold` columns and `p` (number of predictors) rows. The importance is measured based on increase in IMSE types 1 and 2, respectively, and the node impurity.
- importanceSD'The "standard errors" of the permutation-based importance measure.' A `p` by `nfold` by 2 array corresponding to the first two matrices of the importance array.
- nfoldnumber of forests iterated over.
- ntreenumber of trees built.
- mtrynumber of candidate predictors tried at each node.
- forest'a list that contains the entire forest;' `NULL` 'if `keep.forest=FALSE`.'
- intervalsn by 2 matrix of the intervals.
- proximityif `proximity=TRUE` if `proximity=TRUE` when `icrf` is called, a matrix of proximity measures among the input (based on the frequency that pairs of data points are in the same terminal nodes). inbagif `keep.inbag=TRUE` provides a matrix of in-bag indicators for the last forest iteration. runtimestart and end times and the elapsed time. testif test set is given (through the `xtest` or additionally `ytest` arguments), this component is a list which contains the corresponding `predicted` and error measures (IMSE's). If `proximity=TRUE`, there is also a component, `proximity`, which contains the proximity among the test set as well as proximity between test and training data.

Author(s)

Hunyong Cho, Nicholas P. Jewell, and Michael R. Kosorok.

References

Cho H., Jewell N. J., and Kosorok M. R. (2020+). "Interval censored recursive forests"

See Also

predict.icrf, plot.icrf, survplot, importance.icrf

Examples

```
# rats data example.
# The type of this dataset is current status data.
# Note that this is a toy example. Use a larger ntree and nfold in practice.
data(rat2)


 set.seed(2)
# 1. formula (currentstatus)
rats.icrf <-
  icrf(~ dose.lvl + weight + male + cage.no, data = rat2,
       data.type = "currentstatus", currentstatus.label = c("survtime", "tumor"),
       returnBest = TRUE, ntree=10, nfold=3)

# 2. formula containing the interval
# Alternatively, create the interval endpoints and use the Surv object.
L = ifelse(rat2$tumor, 0, rat2$survtime)
R = ifelse(rat2$tumor, rat2$survtime, Inf)
library(survival) # for Surv function
icrf(Surv(L, R, type = "interval2") ~ dose.lvl + weight + male + cage.no, data = rat2,
     ntree=10, nfold=3)

# Or, 3. formula (interval)
rat2b <- cbind(rat2, L = L, R = R)
set.seed(1)
icrf( ~ dose.lvl + weight + male + cage.no, data = rat2b,
     data.type = "interval", interval.label = c("L", "R"),
     ntree=10, nfold=3)

# 4. default method
set.seed(1)
icrf(rat2[, c("dose.lvl", "weight", "male", "cage.no")], L = L, R = R,
     ntree=10, nfold=3)
```

---

importance.icrf *'Extract variable importance measure'*

Description

'This is the extractor function for variable importance measures as produced by' `icrf`. (Quoted statements are from `randomForest` by Liaw and Wiener unless otherwise mentioned.)

Usage

```
importance(x, ...)

## Default S3 method:
importance(x, ...)

## S3 method for class 'icrf'
importance(x, type = NULL, ...)
```

Arguments

x	'an object of class' icrf
...	'not used'
type	either 1, 2, 3, or any combination of them, 'specifying the type of importance measure' (1 = mean increase in IMSE1, 2 = mean increase in IMSE2, 3 = mean decrease in node impurity). If not specified, all available types of importances are returned.

Details

'Here are the definitions of the variable importance measures. The first two measures are computed from permuting OOB data: For each tree, the prediction error on the out-of-bag portion of the data is recorded' (IMSE1 and IMSE2). 'Then the same is done after permuting each predictor variable.' 'The difference between the two are then averaged over all trees' The normalization by the standard deviation of the differences is not supported in this version. The third measure 'is the total decrease in node impurities from splitting on the variable, averaged over all trees.' 'For regression, it is measured by residual sum of squares.'

Value

An array of importance measure matrices, one row for each predictor variable.' Each column corresponds to the forest iteration. Each matrix corresponds to the type of the measure.

Author(s)

Hunyong Cho, Nicholas P. Jewell, and Michael R. Kosorok.

Cho H., Jewell N. J., and Kosorok M. R. (2020+). "Interval censored recursive forests"

See Also

icrf, varImpPlot

Examples

```
# rats data example.
# Note that this is a toy example. Use a larger ntree and nfold in practice.
data(rat2)

set.seed(1)
```

```
rats.icrf <-
  icrf(~ dose.lvl + weight + male + cage.no, data = rat2,
       data.type = "currentstatus", currentstatus.label = c("survtime", "tumor"),
       returnBest = TRUE, ntree=10, nfold=3)
importance(rats.icrf)
```

`measure` *Prediction error measures*

Description

This function measures the prediction errors including the IMSE (integrated mean squared error) of type 1 and 2, the integrated absolute error, and the supremum absolute error. When the true survival curve is unknown but the observed interval is available, IMSE is used. When the true survival curve is known, the integrated and supremum absolute errors are used.

Usage

```
measure(
  surv.hat,
  timepoints,
  tau,
  method = c("all", "imse", "int.error"),
  L = NULL,
  R = NULL,
  surv.true = NULL
)
```

Arguments

`surv.hat`	the estimated survival curve matrix with rows representing the observations and the columns representing the time points at which the survival curve is estimated.
`timepoints`	a vector of time points at which the survival curve is estimated.
`tau`	the study end time. ([0, `tau`] is the window for the analysis.)
`method`	Which measure will be used? Either `imse`, `int.error` (`int.error` returns both integrated and supremum absolute errors), or `all` (both) should be entered.
`L`, `R`	the left and right interval endpoints. These are required when `method == "imse"` or `"all"`.
`surv.true`	the true survival curve matrix with rows representing the observations and the columns representing the time points at which the survival curve is evaluated. This is required when `method == "int.error"` or `"all"`.

Details

For details of the error measures, see Cho H., Jewell N. J., and Kosorok M. R. (2020+). "Interval censored recursive forest"

Value

A vector of prediction errors:

- `imse.type1` and `imse.type2` when `method == "imse"`
- `int.error` and `sup.error` when `method == "int.error"`
- `imse.type1`, `imse.type2`, `int.error`, and `sup.error` when `method == "all"`

Author(s)

Hunyong Cho hunycho@live.unc.edu, based on the code and the documents of `randomForest` by Andy Liaw and Matthew Wiener.

References

Cho H., Jewell N. J., and Kosorok M. R. (2020+). "Interval censored recursive forests"

Examples

```
# rats data example.
# Note that this is a toy example. Use a larger ntree and nfold in practice.
library(survival)  # for Surv()
data(rat2)
L = ifelse(rat2$tumor, 0, rat2$survtime)
R = ifelse(rat2$tumor, rat2$survtime, Inf)

set.seed(1)
rats.icrf <-
  icrf(Surv(L, R, type = "interval2") ~ dose.lvl + weight + male + cage.no,
       data = rat2, ntree = 10, nfold = 3)

measure(rats.icrf$predicted.Sm, timepoints = rats.icrf$time.points,
        tau = rats.icrf$tau, method = "imse", L = L, R = R)
```

`plot` *icrf IMSE rate plot*

Description

'Plot the error rates or MSE of a randomForest object' (Quoted statements are from `randomForest` by Liaw and Wiener unless otherwise mentioned.)

Usage

```
## S3 method for class 'icrf'
plot(x, type = "l", main = deparse(substitute(x)), oob = FALSE, ...)
```

Arguments

x	an object of icrf class.
type	'type of plot.'
main	'main title of the plot.'
oob	Whether the out-of-bag error should be returned?
...	'other graphical parameters.'

Value

The IMSE (integrated mean squared error) of the icrf object is invisibly returned. 'If the object has a non-null test component, then the returned object is a matrix where the first' (two) column is the IMSE measure (types 1 and 2), 'and the second column is for the test set.' The rows represent the forest iterations.

Note

'If the x has a non-null test component, then the test set errors are also plotted.'

Author(s)

Hunyong Cho, Nicholas P. Jewell, and Michael R. Kosorok.

References

Cho H., Jewell N. J., and Kosorok M. R. (2020+). "Interval censored recursive forests"

Examples

```
# rats data example
# Note that this is a toy example. Use a larger ntree and nfold in practice.
data(rat2)
set.seed(1)
samp <- sample(1:dim(rat2)[1], 200)
rats.train <- rat2[samp, ]
rats.test <- rat2[-samp, ]
# Note that this is a toy example. Use a larger ntree and nfold in practice.
set.seed(2)


 rats.icrf.small <-
   icrf(~ dose.lvl + weight + male + cage.no, data = rat2,
        data.type = "currentstatus", currentstatus.label = c("survtime", "tumor"),
        returnBest = TRUE, ntree=10, nfold=3)
 plot(rats.icrf.small)
```

predict.icrf *icrf predictions*

Description

Prediction method of test data using interval censored recursive forest. (Quoted statements are from randomForest by Liaw and Wiener unless otherwise mentioned.)

Usage

```
## S3 method for class 'icrf'
predict(
  object,
  newdata,
  predict.all = FALSE,
  proximity = FALSE,
  nodes = FALSE,
  smooth = TRUE,
  ...
)
```

Arguments

object	an object of icrf class generated by the function icrf.
newdata	'a data frame or matrix containing new data. (Note: If not given, the predicted survival estimate of the training data set in the object is returned.)'
predict.all	'Should the predictions of all trees be kept?'
proximity	'Should proximity measures be computed?'
nodes	'Should the terminal node indicators (an n by ntree matrix) be returned? If so, it is in the "nodes" attribute of the returned object.'
smooth	Should smoothed curve be returned?
...	'not used currently.'

Value

A matrix of predicted survival probabilities is returned where the rows represent the observations and the columns represent the time points. 'If predict.all=TRUE, then the returned object is a list of two components: aggregate, which is the vector of predicted values by the forest, and individual, which is a matrix where each column contains prediction by a tree in the forest.' The forest is either the last forest or the best forest as specified by returnBest argument in icrf function.

'If proximity=TRUE, the returned object is a list with two components: pred is the prediction (as described above) and proximity is the proximitry matrix.'

'If nodes=TRUE, the returned object has a "nodes" attribute, which is an n by ntree matrix, each column containing the node number that the cases fall in for that tree.'

Author(s)

Hunyong Cho, Nicholas P. Jewell, and Michael R. Kosorok.

References

Cho H., Jewell N. J., and Kosorok M. R. (2020+). "Interval censored recursive forests"

Examples

```
# rats data example
# Note that this is a toy example. Use a larger ntree and nfold in practice.
library(survival)  # for Surv()
data(rat2)
set.seed(1)
samp <- sample(1:dim(rat2)[1], 200)
rats.train <- rat2[samp, ]
rats.test <- rat2[-samp, ]
L = ifelse(rats.train$tumor, 0, rats.train$survtime)
R = ifelse(rats.train$tumor, rats.train$survtime, Inf)

 set.seed(2)
 rats.icrf.small <-
   icrf(survival::Surv(L, R, type = "interval2") ~ dose.lvl + weight + male + cage.no,
        data = rats.train, ntree = 10, nfold = 3, proximity = TRUE)

 # predicted survival curve for the training data
 predict(rats.icrf.small)
 predict(rats.icrf.small, smooth = FALSE) # non-smoothed

 # predicted survival curve for new data
 predict(rats.icrf.small, newdata = rats.test)
 predict(rats.icrf.small, newdata = rats.test, proximity = TRUE)

 # time can be extracted using attr()
 newpred = predict(rats.icrf.small, newdata = rats.test)
 attr(newpred, "time")

 newpred2 = predict(rats.icrf.small, newdata = rats.test, proximity = TRUE)
 attr(newpred2$predicted, "time")
```

`rat2` *Rat tumor data*

Description

This is the 'Rat tumor data' data from Dinse and Lagakos (1984). 112 female and 207 male rats. This data can be considered as a current status data where the time (T) from birth to the onset of the tumor is the main variable of interest but is never observed, but can only be guessed by the set of the death time (survtime) and the tumor indicator (tumor). Quotes are from Dinse and Lagakos (1984).

Usage

```
data(rat2)
```

Format

A data frame with 319 rows and 6 variables:

dose.lvl 'dose level of PBB (coded 0-5).'

weight 'initial weight in grams.'

cage.no 'number of the cage tier.'

survtime 'survival time (age) in weeks.'

tumor 'response indicator (1 = hyperplasia present, 0 = hyperplasia absent).'

male The gender (0 = female, 1 = male).

References

Dinse, G. E., & Lagakos, S. W. (1983). Regression analysis of tumour prevalence data. Journal of the Royal Statistical Society: Series C (Applied Statistics), 32(3), 236-248.

Dinse, G. E., & Lagakos, S. W. (1984). Correction: Regression analysis of tumour prevalence data. Journal of the Royal Statistical Society: Series C (Applied Statistics), 33(1), 79-80.

---

survplot *'Plotting individual survival curves'*

Description

Plotting individual survival curves.

Usage

```
survplot(
  x,
  i,
  smooth = TRUE,
  timepoints = NULL,
  title = "Estimated survival curve",
  suppress.inf.time = TRUE,
  ...
)
```

Arguments

x	an object of class icrf or a survival matrix with rows representing subjects and columns representing times
i	subject index
smooth	which curve of an icrf object to be plotted? smoothed or non-smoothed. Ignored when x is a matrix.
timepoints	A numeric vector. needed when the time attribute is missing.
title	Title of the plot.
suppress.inf.time	Do not draw the curve at timepoints being infinity?
...	'Other graphical parameters to be passed on to' ggplot.

Value

'Invisibly,' the vector of survival probabilities that are plotted.

Author(s)

Hunyong Cho, Nicholas P. Jewell, and Michael R. Kosorok.

References

Cho H., Jewell N. J., and Kosorok M. R. (2020+). "Interval censored recursive forests"

Examples

```
# rats data example.
# Note that this is a toy example. Use a larger ntree and nfold in practice.
data(rat2)

 set.seed(1)
 rats.icrf <-
   icrf(~ dose.lvl + weight + male + cage.no, data = rat2,
        data.type = "currentstatus", currentstatus.label = c("survtime", "tumor"),
        returnBest = TRUE, ntree=10, nfold=3)
 survplot(rats.icrf, c(1,3,5))
```

`treesize.icrf` *Size of trees in an* `icrf` *ensemble*

Description

'Size of trees (number of nodes)' in the returned forest of `icrf`. The returned forest depends on the `returnBest` argument of the `icrf` function; It is either the last forest, when `returnBest = FALSE` or the the best forest, when `returnBest = TRUE`. (Quoted statements are from `randomForest` by Liaw and Wiener unless otherwise mentioned.)

Usage

```
treesize(x, ...)

## S3 method for class 'icrf'
treesize(x, terminal = TRUE, ...)
```

Arguments

`x`	an object of class `icrf`, 'which contains a forest component.'
`...`	'not used.'
`terminal`	'count terminal nodes only (`TRUE`) or all nodes (`FALSE`)'

Value

'A vector containing number of nodes for the trees' in the icrf object.

Note

The `icrf` 'object must contain the forest component; i.e., created with' `icrf(..., keep.forest=TRUE)`.

Author(s)

Hunyong Cho, Nicholas P. Jewell, and Michael R. Kosorok.

References

Cho H., Jewell N. J., and Kosorok M. R. (2020+). "Interval censored recursive forests"

Examples

```
# rats data example.
# Note that this is a toy example. Use a larger ntree and nfold in practice.
data(rat2)

 set.seed(1)
 rats.icrf <-
   icrf(~ dose.lvl + weight + male + cage.no, data = rat2,
```

```
       data.type = "currentstatus", currentstatus.label = c("survtime", "tumor"),
       returnBest = TRUE, ntree=10, nfold=3)
treesize(rats.icrf)
```

`varImpPlot.icrf` *'Variable Importance Plot'*

Description

'Dotchart of variable importance as measured by' `icrf`. (Quoted statements are from `randomForest` by Liaw and Wiener unless otherwise mentioned.)

Usage

```
varImpPlot(x, ...)

## S3 method for class 'icrf'
varImpPlot(
  x,
  sort = TRUE,
  n.var = min(30, nrow(x$importance)),
  type = NULL,
  forest = NULL,
  main = deparse(substitute(x)),
  ...
)
```

Arguments

`x`	'an object of class' `icrf`
`...`	'Other graphical parameters to be passed on to `dotchart`.'
`sort`	'Should the variables be sorted in decreasing order of importance?'
`n.var`	'How many variables to show? (Ignored if sort=FALSE.)'
`type`	'arguments to be passed on to importance'
`forest`	The forest for which the importance is plotted. If `NULL`, the best forest is plotted.
`main`	'plot title'

Value

'Invisibly, the importance of the variables that were plotted.'

Author(s)

Hunyong Cho, Nicholas P. Jewell, and Michael R. Kosorok.

References

Cho H., Jewell N. J., and Kosorok M. R. (2020+). "Interval censored recursive forests"

Examples

```
# rats data example.
# Note that this is a toy example. Use a larger ntree and nfold in practice.
data(rat2)

 set.seed(1)
 rats.icrf <-
   icrf(~ dose.lvl + weight + male + cage.no, data = rat2,
        data.type = "currentstatus", currentstatus.label = c("survtime", "tumor"),
        returnBest = TRUE, ntree=10, nfold=3)
 varImpPlot(rats.icrf)
```

`varUsed.icrf` *'Variables used in' an* `icrf` *ensemble*

Description

'Find out which predictor variables are actually used in' the returned forest of the `icrf`. The returned forest depends on the `returnBest` argument of the `icrf` function; It is either the last forest, when `returnBest = FALSE` or the the best forest, when `returnBest = TRUE`. (Quoted statements are from `randomForest` by Liaw and Wiener unless otherwise mentioned.)

Usage

```
varUsed(x, ...)

## S3 method for class 'icrf'
varUsed(x, by.tree = FALSE, count = TRUE, ...)
```

Arguments

`x`	'an object of class' `icrf`.
`...`	not used.
`by.tree`	'Should the list of variables used be broken down by trees in the forest?'
`count`	'Should the frequencies that variables appear in trees be returned?'

Value

'A vector containing number of nodes for the trees' in the icrf object.

'If `count=TRUE` and `by.tree=FALSE`, an integer vector containing frequencies that variables are used in the forest. If `by.tree=TRUE`, a matrix is returned, breaking down the counts by tree (each column corresponding to one tree and each row to a variable).'

'If `count=FALSE` and `by.tree=TRUE`, a list of integer indices is returned giving the variables used in the trees, else if `by.tree=FALSE`, a vector of integer indices giving the variables used in the entire forest.'

Author(s)

Hunyong Cho, Nicholas P. Jewell, and Michael R. Kosorok.

References

Cho H., Jewell N. J., and Kosorok M. R. (2020+). "Interval censored recursive forests"

Examples

```
# rats data example.
# Note that this is a toy example. Use a larger ntree and nfold in practice.
data(rat2)

 set.seed(1)
 rats.icrf <-
   icrf(~ dose.lvl + weight + male + cage.no, data = rat2,
        data.type = "currentstatus", currentstatus.label = c("survtime", "tumor"),
        returnBest = TRUE, ntree=10, nfold=3)
 varUsed(rats.icrf)
```

Index