Package 'forestError'

October 13, 2022

Type Package

Title A Unified Framework for Random Forest Prediction Error Estimation

Version 1.1.0

Author Benjamin Lu and Johanna Hardin

Maintainer Benjamin Lu <b.lu@berkeley.edu>

Description Estimates the conditional error distributions of random forest predictions and common parameters of those distributions, including conditional misclassification rates, conditional mean squared prediction errors, conditional biases, and conditional quantiles, by out-of-bag weighting of out-of-bag prediction errors as proposed by Lu and Hardin (2021). This package is compatible with several existing packages that implement random forests in R.

Imports data.table, purrr

Suggests randomForest

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-08-10 19:50:02 UTC

R topics documented:

`findOOBErrors` *Compute and locate out-of-bag prediction errors*

Description

Computes each training observation's out-of-bag prediction error using the random forest and, for each tree for which the training observation is out of bag, finds the terminal node of the tree in which the training observation falls.

Usage

```
findOOBErrors(forest, X.train, Y.train = NULL, n.cores = 1)
```

Arguments

`forest`	The random forest object being used for prediction.
`X.train`	A `matrix` or `data.frame` with the observations that were used to train `forest`. Each row should be an observation, and each column should be a predictor variable.
`Y.train`	A vector of the responses of the observations that were used to train `forest`. Required if `forest` was created using `ranger`, but not if `forest` was created using `randomForest`, `randomForestSRC`, or `quantregForest`.
`n.cores`	Number of cores to use (for parallel computation in `ranger`).

Details

This function accepts classification or regression random forests built using the `randomForest`, `ranger`, `randomForestSRC`, and `quantregForest` packages. When training the random forest using `randomForest`, `ranger`, or `quantregForest`, `keep.inbag` must be set to `TRUE`. When training the random forest using `randomForestSRC`, `membership` must be set to `TRUE`.

Value

A `data.table` with the following three columns:

`tree`	The ID of the tree of the random forest
`terminal_node`	The ID of the terminal node of the tree
`node_errs`	A vector of the out-of-bag prediction errors that fall within the terminal node of the tree

Author(s)

Benjamin Lu <b.lu@berkeley.edu>; Johanna Hardin <jo.hardin@pomona.edu>

See Also

`quantForestError`

Examples

```
# load data
data(airquality)

# remove observations with missing predictor variable values
airquality <- airquality[complete.cases(airquality), ]

# get number of observations and the response column index
n <- nrow(airquality)
response.col <- 1

# split data into training and test sets
train.ind <- sample(1:n, n * 0.9, replace = FALSE)
Xtrain <- airquality[train.ind, -response.col]
Ytrain <- airquality[train.ind, response.col]
Xtest <- airquality[-train.ind, -response.col]

# fit random forest to the training data
rf <- randomForest::randomForest(Xtrain, Ytrain, nodesize = 5,
                                 ntree = 500, keep.inbag = TRUE)

# compute out-of-bag prediction errors and locate each
# training observation in the trees for which it is out
# of bag
train_nodes <- findOOBErrors(rf, Xtrain)

# estimate conditional mean squared prediction errors,
# biases, prediction intervals, and error distribution
# functions for the test observations. provide
# train_nodes to avoid recomputing that step.
output <- quantForestError(rf, Xtrain, Xtest,
                           train_nodes = train_nodes)
```

`perror` *Estimated conditional prediction error CDFs*

Description

Returns probabilities from the estimated conditional cumulative distribution function of the prediction error associated with each test observation.

Usage

```
perror(q, xs)
```

Arguments

`q`	A vector of quantiles.

xs — A vector of the indices of the test observations for which the conditional error CDFs are desired. Defaults to all test observations given in the call of quantForestError.

Details

This function is only defined as output of the quantForestError function. It is not exported as a standalone function. See the example.

Value

If either q or xs has length one, then a vector is returned with the desired probabilities. If both have length greater than one, then a data.frame of probabilities is returned, with rows corresponding to the inputted xs and columns corresponding to the inputted q.

Author(s)

Benjamin Lu <b.lu@berkeley.edu>; Johanna Hardin <jo.hardin@pomona.edu>

See Also

quantForestError

Examples

```
# load data
data(airquality)

# remove observations with missing predictor variable values
airquality <- airquality[complete.cases(airquality), ]

# get number of observations and the response column index
n <- nrow(airquality)
response.col <- 1

# split data into training and test sets
train.ind <- sample(1:n, n * 0.9, replace = FALSE)
Xtrain <- airquality[train.ind, -response.col]
Ytrain <- airquality[train.ind, response.col]
Xtest <- airquality[-train.ind, -response.col]
Ytest <- airquality[-train.ind, response.col]

# fit random forest to the training data
rf <- randomForest::randomForest(Xtrain, Ytrain, nodesize = 5,
                                 ntree = 500,
                                 keep.inbag = TRUE)

# estimate conditional error distribution functions
output <- quantForestError(rf, Xtrain, Xtest,
                           what = c("p.error", "q.error"))

# get the probability that the error associated with each test
```

```
# prediction is less than -4 and the probability that the error
# associated with each test prediction is less than 7
output$perror(c(-4, 7))

# same as above but only for the first three test observations
output$perror(c(-4, 7), 1:3)
```

---

qerror *Estimated conditional prediction error quantile functions*

Description

Returns quantiles of the estimated conditional error distribution associated with each test prediction.

Usage

```
qerror(p, xs)
```

Arguments

p	A vector of probabilities.
xs	A vector of the indices of the test observations for which the conditional error quantiles are desired. Defaults to all test observations given in the call of quantForestError.

Details

This function is only defined as output of the quantForestError function. It is not exported as a standalone function. See the example.

Value

If either p or xs has length one, then a vector is returned with the desired quantiles. If both have length greater than one, then a data.frame of quantiles is returned, with rows corresponding to the inputted xs and columns corresponding to the inputted p.

Author(s)

Benjamin Lu <b.lu@berkeley.edu>; Johanna Hardin <jo.hardin@pomona.edu>

See Also

quantForestError

Examples

```
# load data
data(airquality)

# remove observations with missing predictor variable values
airquality <- airquality[complete.cases(airquality), ]

# get number of observations and the response column index
n <- nrow(airquality)
response.col <- 1

# split data into training and test sets
train.ind <- sample(1:n, n * 0.9, replace = FALSE)
Xtrain <- airquality[train.ind, -response.col]
Ytrain <- airquality[train.ind, response.col]
Xtest <- airquality[-train.ind, -response.col]
Ytest <- airquality[-train.ind, response.col]

# fit random forest to the training data
rf <- randomForest::randomForest(Xtrain, Ytrain, nodesize = 5,
                                 ntree = 500,
                                 keep.inbag = TRUE)

# estimate conditional error distribution functions
output <- quantForestError(rf, Xtrain, Xtest,
                           what = c("p.error", "q.error"))

# get the 0.25 and 0.8 quantiles of the error distribution
# associated with each test prediction
output$qerror(c(0.25, 0.8))

# same as above but only for the first three test observations
output$qerror(c(0.25, 0.8), 1:3)
```

`quantForestError` *Quantify random forest prediction error*

Description

Estimates the conditional misclassification rates, conditional mean squared prediction errors, conditional biases, conditional prediction intervals, and conditional error distributions of random forest predictions.

Usage

```
quantForestError(
  forest,
  X.train,
  X.test,
```

```
  Y.train = NULL,
 what = if (grepl("class", c(forest$type, forest$family, forest$treetype), TRUE))
    "mcr" else c("mspe", "bias", "interval", "p.error", "q.error"),
  alpha = 0.05,
  train_nodes = NULL,
  return_train_nodes = FALSE,
  n.cores = 1
)
```

Arguments

forest	The random forest object being used for prediction.
X.train	A matrix or data.frame with the observations that were used to train forest. Each row should be an observation, and each column should be a predictor variable.
X.test	A matrix or data.frame with the observations to be predicted; each row should be an observation, and each column should be a predictor variable.
Y.train	A vector of the responses of the observations that were used to train forest. Required if forest was created using ranger, but not if forest was created using randomForest, randomForestSRC, or quantregForest.
what	A vector of characters indicating what estimates are desired. Possible options are conditional mean squared prediction errors ("mspe"), conditional biases ("bias"), conditional prediction intervals ("interval"), conditional error distribution functions ("p.error"), conditional error quantile functions ("q.error"), and conditional misclassification rate ("mcr"). Note that the conditional misclassification rate is available only for categorical outcomes, while the other parameters are available only for real-valued outcomes.
alpha	A vector of type-I error rates desired for the conditional prediction intervals; required if "interval" is included in what.
train_nodes	A data.table indicating what out-of-bag prediction errors each terminal node of each tree in forest contains. It should be formatted like the output of findOOBErrors. If not provided, it will be computed internally.
return_train_nodes	A boolean indicating whether to return the train_nodes computed and/or used.
n.cores	Number of cores to use (for parallel computation in ranger).

Details

This function accepts classification or regression random forests built using the randomForest, ranger, randomForestSRC, and quantregForest packages. When training the random forest using randomForest, ranger, or quantregForest, keep.inbag must be set to TRUE. When training the random forest using randomForestSRC, membership must be set to TRUE.

The predictions computed by ranger can be parallelized by setting the value of n.cores to be greater than 1.

The random forest predictions are always returned as a data.frame. Additional columns are included in the data.frame depending on the user's selections in the argument what. In particular,

including "mspe" in what will add an additional column with the conditional mean squared prediction error of each test prediction to the data.frame; including "bias" in what will add an additional column with the conditional bias of each test prediction to the data.frame; including "interval" in what will add to the data.frame additional columns with the lower and upper bounds of conditional prediction intervals for each test prediction; and including "mcr" in what will add an additional column with the conditional misclassification rate of each test prediction to the data.frame. The conditional misclassification rate can be estimated only for classification random forests, while the other parameters can be estimated only for regression random forests.

If "p.error" or "q.error" is included in what, or if return_train_nodes is set to TRUE, then a list will be returned as output. The first element of the list, named "estimates", is the data.frame described in the above paragraph. The other elements of the list are the estimated cumulative distribution functions (perror) of the conditional error distributions, the estimated quantile functions (qerror) of the conditional error distributions, and/or a data.table indicating what out-of-bag prediction errors each terminal node of each tree in the random forest contains.

Value

A data.frame with one or more of the following columns, as described in the details section:

pred	The random forest predictions of the test observations
mspe	The estimated conditional mean squared prediction errors of the random forest predictions
bias	The estimated conditional biases of the random forest predictions
lower_alpha	The estimated lower bounds of the conditional alpha-level prediction intervals for the test observations
upper_alpha	The estimated upper bounds of the conditional alpha-level prediction intervals for the test observations
mcr	The estimated conditional misclassification rate of the random forest predictions

In addition, one or both of the following functions, as described in the details section:

perror	The estimated cumulative distribution functions of the conditional error distributions associated with the test predictions
qerror	The estimated quantile functions of the conditional error distributions associated with the test predictions

In addition, if return_train_nodes is TRUE, then a data.table called train_nodes indicating what out-of-bag prediction errors each terminal node of each tree in forest contains.

Author(s)

Benjamin Lu <b.lu@berkeley.edu>; Johanna Hardin <jo.hardin@pomona.edu>

See Also

perror, qerror, findOOBErrors

Examples

```
# load data
data(airquality)

# remove observations with missing predictor variable values
airquality <- airquality[complete.cases(airquality), ]

# get number of observations and the response column index
n <- nrow(airquality)
response.col <- 1

# split data into training and test sets
train.ind <- sample(c("A", "B", "C"), n,
                    replace = TRUE, prob = c(0.8, 0.1, 0.1))
Xtrain <- airquality[train.ind == "A", -response.col]
Ytrain <- airquality[train.ind == "A", response.col]
Xtest1 <- airquality[train.ind == "B", -response.col]
Xtest2 <- airquality[train.ind == "C", -response.col]

# fit regression random forest to the training data
rf <- randomForest::randomForest(Xtrain, Ytrain, nodesize = 5,
                                 ntree = 500,
                                 keep.inbag = TRUE)

# estimate conditional mean squared prediction errors,
# biases, prediction intervals, and error distribution
# functions for the observations in Xtest1. return
# train_nodes to avoid recomputation in the next
# line of code.
output1 <- quantForestError(rf, Xtrain, Xtest1,
                            return_train_nodes = TRUE)

# estimate just the conditional mean squared prediction errors
# and prediction intervals for the observations in Xtest2.
# avoid recomputation by providing train_nodes from the
# previous line of code.
output2 <- quantForestError(rf, Xtrain, Xtest2,
                            what = c("mspe", "interval"),
                            train_nodes = output1$train_nodes)

# for illustrative purposes, convert response to categorical
Ytrain <- as.factor(Ytrain > 31.5)

# fit classification random forest to the training data
rf <- randomForest::randomForest(Xtrain, Ytrain, nodesize = 3,
                                 ntree = 500,
                                 keep.inbag = TRUE)

# estimate conditional misclassification rate of the
# predictions of Xtest1
output <- quantForestError(rf, Xtrain, Xtest1)
```

Index