Package 'forestError'

November 28, 2019

Type Package

Title A Unified Framework for Random Forest Prediction Error Inference

Version 0.0.0.9000

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 mean squared prediction errors, conditional biases, and conditional prediction intervals. The error distributions and associated parameters are conditional in the sense that each distribution is specific to the individual observation whose response is being predicted. More details can be found in Lu and Hardin (2019+) (in preparation).

Imports Rcpp

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

LinkingTo Rcpp

R topics documented:

`perror` *Empirical cumulative distribution functions of conditional errors*

Description

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

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.

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

```
# 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
perror(c(-4, 7))

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

---

qerror *Empirical quantile functions of conditional errors*

Description

Returns quantiles from the estimated quantile function of the 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.

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

```
# get the 0.25 and 0.8 quantiles of the error distribution associated
# with each test observation
qerror(c(0.25, 0.8))

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

---

quantForestError *Quantify random forest prediction error*

Description

Estimates the 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 = c("mspe", "bias", "interval", "p.error", "q.error"),
  alpha = 0.05, rcpp = TRUE)
```

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 the ranger package, but not if forest was created using the randomForest package, the randomForestSRC package, or the quantregForest package.
what	A vector of characters indicating what outputs are desired. Possible options are conditional mean squared prediction error estimates ("mspe"), conditional bias estimates ("bias"), conditional prediction intervals ("interval"), conservative conditional prediction intervals ("cons.interval"), the conditional error distribution functions ("p.error"), and the conditional quantile functions ("q.error").
alpha	The type-I error rate desired for the conditional prediction intervals; required if "interval" or "cons.interval" are included in what.
rcpp	A logical indicating whether the weights should be computed in C++ using the Rcpp package for reduced runtime. Recommended especially when the number of training observations, test observations, or trees is large. Defaults to TRUE.

Details

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 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; including "interval" in what will add to the data.frametwo additional columns with the lower and upper bounds of a conditional prediction interval for each test prediction; and including "cons.interval" will add to the data.frame two additional columns with the lower and upper bounds of a conservative prediction interval for each test prediction. Conservative prediction intervals are generated by breaking ties in the empirical error distribution conservatively. They may be desirable when the number of observations is small.

If "p.error" or "q.error" is included in what, 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 one or two elements of the list are the empirical CDFs (perror) and/or the empirical quantile functions (qerror) of the conditional error distributions associated with the test predictions.

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
error	The estimated conditional mean square prediction errors of the random forest predictions

bias	The estimated conditional biases of the random forest predictions
lower	The estimated lower bounds of the conditional prediction intervals for the test observations
upper	The estimated upper bounds of the conditional prediction intervals for the test observations
lowerCons	The conservatively estimated lower bounds of the conditional prediction intervals for the test observations
upperCons	The conservatively estimated upper bounds of the conditional prediction intervals for the test observations

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

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

Author(s)

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

See Also

perror, qerror

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(Xtrain, Ytrain, nodesize = 5,
                   ntree = 500, keep.inbag = TRUE)

# get conditional mean squared prediction errors, biases,
# prediction intervals, and empirical error distribution
# functions for the test observations
test.errors <- quantForestError(rf, Xtrain, Xtest, alpha = 0.05)
```

```
# get just the conditional mean squared prediction errors
# and prediction intervals for the test observations
test.errors <- quantForestError(rf, Xtrain, Xtest,
                                what = c("mspe", "interval",
                                         "cons.interval"),
                                alpha = 0.05)

# get just the conditional empirical error distribution
# functions for the test observations
test.errors <- quantForestError(rf, Xtrain, Xtest,
                                what = c("p.error", "q.error"))
```

Index