# Package 'forestError'

July 29, 2019

License GPL-3 Encoding UTF-8 LazyData true RoxygenNote 6.1.1 LinkingTo Rcpp  R topics documented:  perror qerror quantForestError  3	Type Package		
Author Benjamin Lu and Johanna Hardin  Maintainer Benjamin Lu <b. lu@berkeley.edu="">  Description Estimates both conditional mean squared prediction errors and conditional prediction intervals for random forest predictions. The prediction errors and intervals are conditional in the sense that each error and interval 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 qerror quantForestError  Index</b.>	Title Individualized Pred	iction Errors and Intervals for Random Forests	
Maintainer Benjamin Lu <b. lu@berkeley.edu="">  Description Estimates both conditional mean squared prediction errors and conditional prediction intervals for random forest predictions. The prediction errors and interval 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</b.>	Version 0.0.0.9000		
Description Estimates both conditional mean squared prediction errors and conditional prediction intervals for random forest predictions. The prediction errors and intervals are conditional in the sense that each error and interval 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 qerror quantForestError  Index	Author Benjamin Lu and	l Johanna Hardin	
conditional prediction intervals for random forest predictions. The prediction errors and intervals are conditional in the sense that each error and interval 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 qerror quantForestError  Index	Maintainer Benjamin Lu	ı <b.lu@berkeley.edu></b.lu@berkeley.edu>	
License GPL-3 Encoding UTF-8 LazyData true RoxygenNote 6.1.1 LinkingTo Rcpp  R topics documented:  perror qerror quantForestError  Index	conditional predicti prediction errors an and interval is spec- being predicted. Mo	on intervals for random forest predictions. The d intervals are conditional in the sense that each error life to the individual observation whose response is	
Encoding UTF-8  LazyData true  RoxygenNote 6.1.1  LinkingTo Rcpp  R topics documented:  perror 1 qerror 2 quantForestError 3  Index 6	Imports Rcpp		
LazyData true  RoxygenNote 6.1.1  LinkingTo Rcpp  R topics documented:  perror	License GPL-3		
RoxygenNote 6.1.1  LinkingTo Rcpp  R topics documented:  perror qerror quantForestError  Index	<b>Encoding</b> UTF-8		
LinkingTo Rcpp  R topics documented:  perror	LazyData true		
R topics documented:  perror	RoxygenNote 6.1.1		
perror	LinkingTo Rcpp		
qerror	R topics docume	nted:	
	qerror		2 3
perror Empirical cumulative distribution functions of conditional errors	Index		6
	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.

2 qerror

#### Usage

```
perror(p, xs = 1:n.test)
```

# Arguments

р	Vector	of o	quantile
Ρ	100101	01 1	qualitie

xs 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 p 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 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>

# **Examples**

```
# get the probability that the error associated with each test prediction # is less than -5 and the probability that the error associated with each # test prediction is less than 5 perror(c(-5, 5))

# same as above but only for the first three test observations perror(c(-5, 5), 1:3)
```

gerror

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(q, xs = 1:n.test)
```

# **Arguments**

q Vector of probabilities

Indices of the test observations for which the conditional error CDFs are desired.

Defaults to all test observations given in the call of quantForestError.

quantForestError 3

#### 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 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>

#### **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 prediction intervals, and conditional error distributions of random forest predictions.

# Usage

```
quantForestError(forest, X.train, X.test, Y.train = NULL,
  what = c("mspe", "interval", "cons.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 or the randomForestSRC package.
what	A vector of characters indicating what outputs are desired. Possible options are conditional mean squared prediction error estimates ("mspe"), conditional prediction intervals ("interval"), conservative conditional prediction intervals ("cons.interval"), the conditional error distribution functions ("p.error"), and the conditional quantile functions ("q.error").

4 quantForestError

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 dataframe. Additional columns are included in the dataframe 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 dataframe; including "interval" in what will add two additional columns with the lower and upper boounds of a conditional prediction interval for each test prediction to the dataframe; and including "cons.interval" will add to the dataframe 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 dataframe 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 (gerror) of the conditional error distributions associated with the test predictions.

#### Value

A dataframe 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

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 inter-

vals for the test observations

upperCons The conservatively estimated upper bounds of the conditional prediction inter-

vals 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 distri-

butions 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>

quantForestError 5

#### **Examples**

```
# load data
data(airquality)
# remove observations with missing predictor variable values
airquality <- airquality[complete.cases(airquality), ]</pre>
# get number of observations and the response column index
n <- nrow(airquality)</pre>
response.col <- 1
# split data into training and test sets
train.ind <- sample(1:n, n * 0.9, replace = FALSE)</pre>
Xtrain <- airquality[train.ind, -response.col]</pre>
Ytrain <- airquality[train.ind, response.col]</pre>
Xtest <- airquality[-train.ind, -response.col]</pre>
Ytest <- airquality[-train.ind, response.col]</pre>
# fit random forest to the training data
rf <- randomForest(Xtrain, Ytrain, nodesize = 10,</pre>
                    ntree = 500, keep.inbag = TRUE)
# get conditional mean squared prediction errors, prediction
# intervals, and empirical error distribution functions for
# the test observations
test.preds <- quantForestError(rf, Xtrain, Xtest)</pre>
# get just the conditional mean squared prediction errors
# and prediction intervals for the test observations
test.preds <- quantForestError(rf, Xtrain, Xtest,</pre>
                                what = c("mspe", "interval",
                                          "cons.interval"))
# get just the conditional empirical error distribution
# functions for the test observations
test.preds <- quantForestError(rf, Xtrain, Xtest,</pre>
                                what = c("p.error", "q.error"))
```

# Index

```
\begin{array}{l} \text{perror, 1} \\ \\ \text{qerror, 2} \\ \\ \text{quantForestError, 3} \end{array}
```