

# Package ‘forestError’

August 7, 2019

**Type** Package  
**Title** Individualized Prediction Errors and Intervals for Random Forests  
**Version** 0.0.0.9000  
**Author** Benjamin Lu and Johanna Hardin  
**Maintainer** Benjamin Lu <b.lu@berkeley.edu>  
**Description** Estimates conditional mean squared prediction errors, conditional prediction intervals, and conditional error distributions for random forest predictions. The errors, intervals, and distributions 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	1
qerror	2
quantForestError	3
<b>Index</b>	<b>7</b>

---

perror	<i>Empirical cumulative distribution functions of conditional errors</i>
--------	--

---

## 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 <code>quantForestError</code> .

**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	<i>Quantify random forest prediction error</i>
------------------	--

---

**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**

<code>forest</code>	The random forest object being used for prediction.
<code>X.train</code>	A matrix or <code>data.frame</code> with the observations that were used to train <code>forest</code> ; each row should be an observation, and each column should be a predictor variable.
<code>X.test</code>	A matrix or <code>data.frame</code> with the observations to be predicted; each row should be an observation, and each column should be a predictor variable.
<code>Y.train</code>	A vector of the responses of the observations that were used to train <code>forest</code> . Required if <code>forest</code> was created using the <code>ranger</code> package, but not if <code>forest</code> was created using the <code>randomForest</code> package, the <code>randomForestSRC</code> package, or the <code>quantregForest</code> package.
<code>what</code>	A vector of characters indicating what outputs are desired. Possible options are conditional mean squared prediction error estimates (" <code>mspe</code> "), conditional prediction intervals (" <code>interval</code> "), conservative conditional prediction intervals (" <code>cons.interval</code> "), the conditional error distribution functions (" <code>p.error</code> "), and the conditional quantile functions (" <code>q.error</code> ").
<code>alpha</code>	The type-I error rate desired for the conditional prediction intervals; required if " <code>interval</code> " or " <code>cons.interval</code> " are included in <code>what</code> .
<code>rcpp</code>	A logical indicating whether the weights should be computed in C++ using the <code>Rcpp</code> package for reduced runtime. Recommended especially when the number of training observations, test observations, or trees is large. Defaults to <code>TRUE</code> .

**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 "`interval`" in `what` will add to the `data.frame` two 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:

<code>pred</code>	The random forest predictions of the test observations
<code>error</code>	The estimated conditional mean square prediction errors of the random forest predictions
<code>lower</code>	The estimated lower bounds of the conditional prediction intervals for the test observations



[illegible]

# Index

forestError (quantForestError), [3](#)

perror, [1](#), [5](#)

qerror, [2](#), [5](#)

quantForestError, [2](#), [3](#), [3](#)