

# Package ‘RFinfer’

August 29, 2016

**Type** Package

**Title** Inference for Random Forests

**Version** 0.2.0

**Date** 2016-06-09

**Author** Cole Brokamp

**Maintainer** Cole Brokamp <cole.brokamp@gmail.com>

**Description** A set of add on tools for the 'randomForest' package.

**Depends** R (>= 3.1.2), randomForest (>= 4.6-12)

**Imports** Matrix, party, pbapply

**License** GPL

**LazyData** TRUE

**RoxygenNote** 5.0.1

**Suggests** knitr, rmarkdown, ggplot2, nhanesA

**VignetteBuilder** knitr

**URL** <https://github.com/cole-brokamp/RFinfer>

**BugReports** <https://github.com/cole-brokamp/RFinfer/issues>

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-06-10 18:27:02

## R topics documented:

rfPredVar . . . . .	2
<b>Index</b>	<b>4</b>

---

 rfPredVar

*rfPredVar*


---

### Description

Generate predictions and prediction variances from a random forest based on the infinitesimal jack-knife.

### Usage

```
rfPredVar(random.forest, rf.data, pred.data = rf.data, CI = FALSE,
          tree.type = "rf", prog.bar = FALSE)
```

### Arguments

random.forest	A random forest trained with <code>keep.inbag=TRUE</code> . See details for more information.
rf.data	The data used to train rf
pred.data	The data used to predict with the forest; defaults to <code>rf.data</code> if not given
CI	Should 95% confidence intervals based on the CLT be returned along with predictions and prediction variances?
tree.type	either 'ci' for conditional inference tree or 'rf' for traditional CART tree
prog.bar	should progress bar be shown? (only applicable when <code>tree.type='ci'</code> )

### Details

The random forest trained with `keep.inbag=TRUE` is supplied only for the purpose of defining the resampling scheme. The function builds a new random forest based on the `tree.type` setting. However, the resamples are maintained identically to the supplied random forest. This allows for direct comparison of the tree methods without having to account for variation in resampling.

Currently, the CI methods are much more computationally intensive because there is no C implementation of the CI random forest method that indicates the number of times that each sample is included in each resample. In order to carry out our simulations using  $\hat{V}_{I,J}^B$ , we had to use a pure R implementation of CI random forests. This is different for CART random forests, where a C implementation already exists in the `randomForest` package. However, it should be noted that the difference in computational times is due to the random forest creation step, not the implementation of  $\hat{V}_{I,J}^B$ . This should not be an issue in the future when a C implementation of CI random forests is created.

Note: This function does not use the default predict method for forests produced by `cforest`. The predictions here are the direct averages of all tree predictions, instead of using the observation weights. Therefore, predictions from this function will likely differ from `predict.cforest` when using subsampling.

This function currently only works with regression forests – not classification forests.

**Value**

A data frame with the predictions and prediction variances (and optionally 95% confidence interval)

**Examples**

```
library(randomForest)
data(airquality)
d <- na.omit(airquality)
rf <- randomForest(Ozone ~ ., data=d, keep.inbag=TRUE, sampsize=30, replace=FALSE, ntree=500)
rfPredVar(rf, rf.data=d, CI=TRUE, tree.type='rf')
```

# Index

rfPredVar, 2