

Strobl, Carolin; Boulesteix, Anne-Laure; Zeileis, Achim; Hothorn, Torsten

Working Paper

Bias in random forest variable importance measures: illustrations, sources and a solution

Discussion Paper, No. 490

Provided in Cooperation with:

Collaborative Research Center (SFB) 386: Statistical Analysis of discrete structures - Applications in Biometrics and Econometrics, University of Munich (LMU)

Suggested Citation: Strobl, Carolin; Boulesteix, Anne-Laure; Zeileis, Achim; Hothorn, Torsten (2006) : Bias in random forest variable importance measures: illustrations, sources and a solution, Discussion Paper, No. 490, Ludwig-Maximilians-Universität München, Sonderforschungsbereich 386 - Statistische Analyse diskreter Strukturen, München, <https://doi.org/10.5282/ubm/epub.1858>

This Version is available at:

<https://hdl.handle.net/10419/31116>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

```

#####
# To start the analysis the following packages must be loaded.      #
# (Please install the latest versions of the packages before use.) #
#####

require("randomForest")

require("party")

#####
# Prepare the data set:                                             #
#####

# Download the data:

arabidopsis_url <-
  "http://www.biomedcentral.com/content/supplementary/1471-2105-5-132-S1.txt"

arabidopsis <- read.table(arabidopsis_url, header = TRUE,
  sep = " ", na.string = "X")

# Remove cases with missing values and variables without variation:

arabidopsis <- subset(arabidopsis, complete.cases(arabidopsis))

arabidopsis <- arabidopsis[, !(names(arabidopsis) %in% c("X0", "loc"))]

#####
#
#           randomForest                                           #
#
#####

# Function call to create a random forest:

my_randomForest <- randomForest(edit ~ ., data = arabidopsis,
  importance = TRUE, ntree = 50,
  mtry = 3, replace = TRUE)

#####
# Options:                                                         #
#
# importance = TRUE                                               #
# calculate the variable importance                               #
#
# ntree = 50                                                       #
# number of individual classification trees grown                 #
# in the random forest                                           #
#
# mtry = 3                                                         #
# size of the random subset of predictor variables provided as   #
# splitting variables in each split of each classification tree  #
#
# replace = TRUE                                                  #
# bootstrap samples are drawn with replacement (default)        #
#####

# Function call to return variable importance:

my_varimp <- importance(my_randomForest, scale=TRUE)[,3]

```

```
#####
# Options: #
# #
# scale = TRUE #
# return scaled measure incorporating standard error (default) #
# #
# Usage: #
# Elements importance()[,1] and importance()[,2] return separate #
# variable importance measures for response class $Y = 0$ and #
# $Y = 1$. #
# importance()[,4] returns the overall improvement in the #
# ``Gini gain'' splitting criterion. #
# Note that the ``Gini gain'' splitting criterion is strongly #
# biased in favor of predictor variables with a high number of #
# categories! #
#####
```

```
#####
# #
# cforest #
# #
#####
```

Function call to create a random forest:

```
my_cforest_control <- cforest_control(teststat = "quad",
  testtype = "Univ", mincriterion = 0, ntree = 50, mtry = 3,
  replace = TRUE)
```

```
my_cforest <- cforest(edit ~ ., data = arabidopsis,
  controls = my_cforest_control)
```

```
#####
# Options: #
# #
# teststat = "quad" #
# type of the test statistic is quadratic #
# #
# testtype = "Univ" #
# computation of the distribution of the test statistic, where #
# the default values for teststat and testtype produce result #
# more similar to randomForest #
# #
# mincriterion = 0 #
# the threshold criterion value that must be exceeded for #
# splitting is set to zero to guarantee that a split is #
# conducted in each tree #
# #
# ntree = 50 #
# same as in randomForest #
# #
# mtry = 3 #
# same as in randomForest #
# #
# replace = TRUE #
# same as in randomForest #
#####
```

Function call to return variable importance:

```
varimp_cforest <- varimp(my_cforest)
```

```
varimp_cforest_unscaled <- varimp_cforest[,1]
```

```
varimp_cforest_scaled <- varimp_cforest[,1]/varimp_cforest[,2]
```