LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Carolin Strobl, Torsten Hothorn, Achim Zeileis

Party on!
A New, Conditional Variable Importance Measure for Random Forests Available in the **party** Package

Technical Report Number 050, 2009
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Party on!

A New, Conditional Variable Importance Measure for Random Forests Available in the party Package

Carolin Strobl, Torsten Hothorn and Achim Zeileis

Recursive partitioning methods are amongst the most popular and widely used statistical learning tools for nonparametric regression and classification. Especially random forests, that can deal with large numbers of predictor variables even in the presence of complex interactions, are being applied successfully in many scientific fields. Thus, it is not surprising that there is a variety of recursive partitioning tools available in R (see `http://CRAN.R-project.org/view=MachineLearning` for an overview).

The scope of recursive partitioning methods in R ranges from the standard classification and regression trees available in **rpart** (Therneau et al., 2008) to the reference implementation of random forests (Breiman, 2001) available in **randomForest** (Liaw and Wiener, 2002, 2008). Both methods are popular in applied research, and several extensions and refinements have been suggested in the statistical literature in recent years.

One particularly important improvement was the introduction of unbiased tree algorithms, that overcome the major weak spot of the classical approaches available in **rpart** and **randomForest**: The new, unbiased trees do not artificially favor splits in variables with many categories or continuous variables.

In R such an unbiased tree algorithm is available in the `ctree` function for conditional inference trees in the **party** package (Hothorn et al., 2006). The package also provides a random forest implementation `cforest` based on unbiased trees, that enables learning unbiased forests (Strobl et al., 2007b).

Unbiased variable selection is the key to reliable prediction and interpretability in both individual trees and forests. However, while a single tree's interpretation is straightforward, in random forests an extra effort is necessary to assess the importance of each predictor in the complex ensemble of trees.

This issue is typically addressed by means of variable importance measures such as the Gini importance and the "mean decrease in accuracy" or "permutation" importance, available in **randomForest** in the `importance()` function (with `type = 2` and `type = 1`, respectively). Similarly, a permutation importance for `cforest` is available via `varimp()` in **party**.

Unfortunately, variable importance measures in random forests are subject to the same bias in favor of variables with many categories and continuous variables that affects variable selection in single trees, and also to a new source of bias induced by the resampling scheme (Strobl et al., 2007b). Both problems can be addressed in **party** to guarantee unbiased variable selection and variable importance for predictor variables of different types.

Even though this refined approach can provide reliable variable importance measures in many applications, the original permutation importance is highly misleading in the case of correlated predictors, creating a new source of bias in interpretations drawn from random forests. Therefore, Strobl et al. (2008) recently suggested a solution for this problem in the form of a new, conditional permutation importance measure. Starting from version 0.9-994, this new measure is available in the **party** package.

The rationale and usage of this new measure is outlined in the following and illustrated by means of a toy example.

Random forest variable imporance measures

The permutation importance, that is available in **randomForest** and **party**, is based on a random permutation of the predictor variables, as described in more detail below.

The alternative variable importance available in **randomForest**, the Gini importance, is based on the Gini gain criterion employed in most traditional classification tree algorithms. The Gini importance has been shown to carry forward the bias of the underlying Gini gain splitting criterion (see, e.g., Kim and Loh, 2001; Strobl et al., 2007a; Hothorn et al., 2006) when predictor variables vary in their number of categories or scale of measurement (Strobl et al., 2007b). Therefore, it is not recommended in these situations.

The permutation importance, on the other hand, is a reliable measure of variable importance for uncorrelated predictors when subsampling without replacement – instead of bootstrap sampling – and unbiased trees are used in the construction of the forest (Strobl et al., 2007b). Accordingly, the default settings for the control parameters `cforest_control` have been pre-defined to the default version `cforest_unbiased` to guarantee subsampling without replacement and unbiased individual trees in fitting random forests with the **party** package.

The rationale of the original random forest permutation importance is the following: By randomly permuting the predictor variable X_j, its original association with the response Y is broken. When the permuted variable X_j, together with the remaining non-permuted predictor variables, is used to predict the response for the out-of-bag observations, the prediction accuracy (i.e. the number of correctly classified observations in classification, or respectively the mean squared error in regression) decreases substan-

tially if the original variable X_j was associated with the response. Thus, Breiman (2001) suggests the difference in prediction accuracy before and after permuting X_j, averaged over all trees, as a measure for variable importance.

In standard implementations of random forests, such as **randomForest**, an additional scaled version of the permutation importance (often called z-score), that is achieved by dividing the raw importance by its standard error, is provided (for example by `importance(obj, type = 2, scale = TRUE)` in **randomForest**). Note, however, that the results of Strobl and Zeileis (2008) show that the z-score is not suitable for significance tests and that the raw importance has better statistical properties.

Why conditional importance?

Empirical results (see Strobl et al., 2008, and the references therein) suggest that the original permutation importance severely overestimates the importance of correlated predictor variables.

Why this is a bad thing can easily be shown in a toy example based on a spurious correlation: The data set `readingSkills` is an artificial data set generated by means of a linear model. As the response variable it contains the hypothetical score on a test of reading skills for 200 school children. Potential predictor variables in the data set are the age of the child, whether the child is a native speaker of the test language and the shoe size of the child.

Obviously, the latter is not a sensible predictor of reading skills (and was actually simulated not to have any effect on the response) – but with respect to marginal (as opposed to partial) correlations, shoe size is highly correlated with the test score. Of course this spurious correlation is only due to the fact that both shoe size and test score are associated with the underlying variable age.

In this simple problem, a linear model, e.g., would be perfectly capable of identifying the original coefficients of the predictor variables (including the fact that shoe size has no effect on reading skills once the truly relevant predictor variable age is included in the model). However, the random forest permutation importance is mislead by the spurious correlation and assigns a rather high importance value to the nonsense-variable shoe size:

```
> library("party")
> set.seed(42)
> readingSkills.cf <- cforest(score ~ .,
+     data = readingSkills, control =
+     cforest_unbiased(mtry = 2, ntree = 50))

> set.seed(42)
> varimp(readingSkills.cf)

nativeSpeaker           age      shoeSize
     12.60238      74.54657      18.27733
```

The reason for this odd behavior can be found in the way the predictor variables are permuted in the computation of the importance measure: Strobl et al. (2008) show that the original approach, where one predictor variable X_j is permuted against both the response Y and the remaining (one or more) predictor variables $Z = X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_p$ as illustrated in Figure 1, corresponds to a pattern of independence between X_j and both Y and Z.

From a theoretical point of view, his means that a high value of the importance can be caused by a violation either of the independence between X_j and Y or of the independence between X_j and Z, even though the latter is not of interest here. For practical applications, this means that correlated predictor variables artificially appear more important than uncorrelated ones.

Y	X_j	Z
y_1	$x_{\pi_j(1),j}$	z_1
$\vdots$	$\vdots$	$\vdots$
y_i	$x_{\pi_j(i),j}$	z_i
$\vdots$	$\vdots$	$\vdots$
y_n	$x_{\pi_j(n),j}$	z_n

Figure 1: Permutation scheme for the original permutation importance.

Y	X_j	Z
y_1	$x_{\pi_{j\|Z=a}(1),j}$	$z_1 = a$
y_3	$x_{\pi_{j\|Z=a}(3),j}$	$z_3 = a$
y_{27}	$x_{\pi_{j\|Z=a}(27),j}$	$z_{27} = a$
y_6	$x_{\pi_{j\|Z=b}(6),j}$	$z_6 = b$
y_{14}	$x_{\pi_{j\|Z=b}(14),j}$	$z_{14} = b$
y_{21}	$x_{\pi_{j\|Z=b}(21),j}$	$z_{21} = b$
$\vdots$	$\vdots$	$\vdots$

Figure 2: Permutation scheme for the conditional permutation importance.

The aim to reflect only the impact of X_j in predicting the response Y, rather than its correlations with other predictor variables, can be better achieved by means of a conditional importance measure in the spirit of a partial correlation: We want to measure the association between X_j and Y given the correlation structure between X_j and the other predictor variables inherent in the data set.

To meet this aim, Strobl et al. (2008) suggest a conditional permutation scheme, where X_j is permuted only within groups of observations with $Z = z$ in order to preserve the correlation structure between X_j and the other predictor variables as illustrated in Figure 2.

With this new, conditional permutation scheme, the importance measure is able to reveal the spuri-

ous correlation between shoe size and reading skills:

```
> set.seed(42)
> varimp(readingSkills.cf, conditional =
+     TRUE)

nativeSpeaker           age      shoeSize
    10.877757     39.002710      1.487559
```

Only by means of the conditional importance it becomes clear that the covariate native speaker is actually more relevant for predicting the test score than the shoe size, whose conditional effect is negligible.

How is the conditioning grid defined technically?

Conditioning is straightforward whenever the variables to be conditioned on, Z, are categorical (cf., e.g., Nason et al., 2004). However, conditioning on continuous variables, that may offer as many different values as observations in the sample, would pruduce cells with very sparse counts – which would make permuting the values of X_j within each cell rather pointless. Thus, in order to create cells of reasonable size for conditioning, continuous variables need to be discretized.

As a straightforward discretization strategy for random forests, Strobl et al. (2008) suggest to define the conditioning grid by means of the partition of the feature space induced by each individual tree. This grid can be used to conditionally permute the values of X_j within cells defined by combinations of Z, where Z can contain potentially large sets of covariates of different scales of measurement.

The main advantages of this approach are that this partition has already been learned from the data during model fitting, that it can contain splits in categorical, ordered and continuous predictor variables, and that it can thus serve as an internally available means for discretizing the feature space. For ease of computation, the conditioning grid employed in `varimp` uses all cutpoints as bisectors of the sample space (the same approach is followed by Nason et al., 2004).

The set of variables Z to be conditioned on should contain all variables that are correlated with the current variable of interest X_j. In the `varimp` function, this is assured by the small default value 0.2 of the `threshold` argument: By default, all variables whose correlation with X_j meets the condition 1 - p-value > 0.2 are used for conditioning. A larger value of `threshold` would have the effect that only those variables that are strongly correlated with X_j would be used for conditioning, but would also lower the computational burden.

Note that the same permutation tests that are used for split selection in the tree building process (Hothorn et al., 2006) are used here to measure the association between X_j and the remaining covariates.

A short recipe for fitting random forests and computing variable importance measures with R

To conclude, we would like to summarize the application of the conditional variable importance and general issues in fitting random forests with R. Depending on certain characteristics of your data set, we suggest the following approaches:

- If all predictor variables are of the same type (for example: all continuous or all unordered categorical with the same number of categories), use either `randomForest` (**randomForest**) or `cforest` (**party**). While `randomForest` is computationally faster, `cforest` is safe even for variables of different types.

  For predictor variables of the same type, the Gini importance `importance(obj, type = 2)` or the permutation importance `importance(obj, type = 1)` available for `randomForest` and the permutation importance `varimp(obj)` available for `cforest` are all adequate importance measures.

- If the predictor variables are uncorrelated but of different types (for example: different scales of measurement, different numbers of categories), use `cforest` (**party**) with the default option `controls = cforest_unbiased` and the permutation importance `varimp(obj)`.

- If the predictor variables are correlated, use `cforest` (**party**) with the default option `controls = cforest_unbiased` and the conditional permutation importance `varimp(obj, conditional = TRUE)`.

General remarks:

- Note that the default settings for `mtry` differ in `randomForest` and `cforest`: In `randomForest` the default setting for classification, e.g., is `floor(sqrt(ncol(x)))`, while in `cforest` it is fixed to the value 5 for technical reasons.

- Always check whether you get the same results with a different random seed before interpreting the variable importance ranking!

  If the ranking of even the top scoring predictor variables depends on the choice of the random seed, increase the number of trees (argument `ntree` in `randomForest` and `cforest_control`).

And an outlook:

- The current version of `varimp` cannot deal with missing values, so you will have to omit or impute them beforehand – but missing value handling is one of the top entries on our "to-do list"...

Bibliography

L. Breiman. Random forests. *Machine Learning*, 45(1): 5–32, 2001.

T. Hothorn, K. Hornik, and A. Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, 2006.

H. Kim and W. Loh. Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, 96(454):589–604, 2001.

A. Liaw and M. Wiener. Classification and regression by randomForest. *R News*, 2(3):18–22, 2002.

A. Liaw and M. Wiener. *randomForest: Breiman and Cutler's Random Forests for Classification and Regression*, 2008. URL http://CRAN.R-project.org/package=randomForest. R package version 4.5-28.

M. Nason, S. Emerson, and M. Leblanc. CARTscans: A tool for visualizing complex models. *Journal of Computational and Graphical Statistics*, 13(4):1–19, 2004.

C. Strobl and A. Zeileis. Danger: High power! – Exploring the statistical properties of a test for random forest variable importance. In *Proceedings of the 18th International Conference on Computational Statistics, Porto, Portugal*, 2008.

C. Strobl, A.-L. Boulesteix, and T. Augustin. Unbiased split selection for classification trees based on the Gini index. *Computational Statistics & Data Analysis*, 52(1):483–501, 2007a.

C. Strobl, A.-L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8:25, 2007b.

C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC Bioinformatics*, 9:307, 2008.

T. M. Therneau, B. Atkinson, and B. D. Ripley. rpart: Recursive partitioning. 2008. URL http://CRAN.R-project.org/package=rpart. R package version 3.1-41.

Carolin Strobl
Department of Statistics
Ludwig-Maximilians-Universität
Munich, Germany
carolin.strobl@stat.uni-muenchen.de