A Tutorial on Applying Novel Variable Importance Measures for Ordinal Response Data

Silke Janitza

December 2, 2014

This tutorial illustrates the use of novel variables importance measures (VIMs) for random forest presented in the paper by Janitza et al. (2014). Note that the novel VIMs are implemented in the statistical software R (R Core Team; 2013) for the package party (`http://cran.r-project.org/web/packages/party/index.html`). Before you begin please make sure that the package party is installed.

1 Obtaining the Implementation

The source code implementing the VIMs for the statistical software R is available at the website `http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/070_drittmittel/janitza/rf_ordinal/index.html`. The same website also provides the R file which contains all commands presented in this tutorial. The functions can be loaded in the workspace by executing the commands

```
> library(party)
> source("http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/070_drittmittel/janitza/rf_ordinal
```

The functions `varimpRPS`, `varimpMAE` and `varimpMSE` can now be applied to an object of class `cforest` consisting of ordinal regression trees (for response coded as ordered factor variable) or classification trees (for response coded as unordered factor variable).

2 Applying the Novel Variable Importance Measures

The VIMs can be applied the same way as the function `varimp` from the party package. Note that the function `varimp` computes the standard error rate based importance values if the `cforest` object was constructed using ordinal regression trees or classification trees. If the `cforest` object was constructed using regression trees, the use of `varimp` yields MSE-based importance values.

We use the Mammography Data in this tutorial for our illustrations. This data is part of the R package TH.data. After loading the data into the workspace we first get an overview of the data and the response variable `ME`.

```
> library(TH.data)
> data("mammoexp", package = "TH.data")
> head(mammoexp)
```

```
             ME             SYMPT PB HIST BSE           DECT
1         Never          Disagree  7   No Yes Somewhat likely
2         Never             Agree 11   No Yes     Very likely
3         Never          Disagree  8  Yes Yes     Very likely
4 Within a Year          Disagree 11   No Yes     Very likely
5   Over a Year Strongly Disagree  7   No Yes     Very likely
6         Never          Disagree  7   No Yes     Very likely
```

```
> str(mammoexp$ME)
```

```
 Ord.factor w/ 3 levels "Never"<"Within a Year"<..: 1 1 1 2 3 1 3 1 1 2 ...
```

```
> table(mammoexp$ME)
```

```
        Never Within a Year   Over a Year
          234           104            74
```

The response to be predicted is the Mammography Experience (`ME`) which is a 3-category ordinal response variable. Variables used for prediction are related to the attitude towards mammography based on a study questionnaire. The help page gives detailed information on the data and can be accessed by typing

```
> ?mammoexp
```

2.1 Fitting a Random Forest

For our illustrations we use both, ordinal regression trees and classification trees, though we often obtained slightly better results when using ordinal regression trees in the studies described in Janitza et al. (2014). Therefore we recommend using ordinal regression trees rather than classification trees if the response levels have an ordering.

Before we start fitting random forest and computing VIs, we set a seed in order to make our studies reproducible:

```
> set.seed(1234)
```

2.1.1 Using Ordinal Regression Trees

First we fit a random forest consisting of ordinal regression trees. For this purpose we transfer an *ordered factor variable* to the `cforest` function. The `cforest` function internally checks the type of the response (numeric, categorical nominal or categorical ordinal) and uses ordinal regression trees if the response is coded as an ordinal factor variable. By executing the commands

```
> is.factor(mammoexp$ME)
```

```
[1] TRUE
```

```
> is.ordered(mammoexp$ME)
```

```
[1] TRUE
```

```
> head(mammoexp$ME)
```

```
[1] Never         Never         Never         Within a Year Over a Year   Never
Levels: Never < Within a Year < Over a Year
```

we notice that the response is already specified as an ordered factor variable, with correct ordering of the levels. In many applications this is not the case. Then one can easily convert the variable to the right data type. It is important to specify response levels in the correct order.

```
> mammoexp$ME <- factor(mammoexp$ME, ordered = TRUE,
+                       levels = c("Never", "Within a Year", "Over a Year"))
```

Having made sure that the response has the correct data type, we can grow a random forest consisting of ordinal regression trees:

```
> RF_ordinal <- cforest(ME ~ ., data = mammoexp,
+                       control = cforest_unbiased(ntree = 1000))
```

We can check if ordinal regression trees were used by typing

```
> RF_ordinal@responses@is_ordinal

  ME
TRUE

> RF_ordinal@responses@is_nominal

   ME
FALSE
```

Note that ordinal regression trees make use of scores $s(1) < s(2) < \ldots < s(k)$ for categories $r = 1, \ldots, k.$, The scores reflect the distances between the categories. If scores are not explicitly specified (as in the code above), default scores, $s(r) = r$, are used, which assume that adjacent response levels are equally spaced. In general the studies in Janitza et al. (2014) suggest that the specific values for the scores do not have a significant impact. However, if there is any prior knowledge on the nature of the scores, we propose using this knowledge in the specification of scores. If we expect, for example, a greater "distance" between the response classes 'Never' and 'Within a Year' than between the classes 'Within a Year' and 'Over a Year', we might want to use a score vector which reflects this. In the following we fit a second random forest, this time using the scores $s('Never') = 1, s('WithinaYear' = 3), s('OveraYear') = 4$ which suggest that the difference between the latter two classes is half as big as the distance between the former two classes.

```
> RF_ordinal_alt_scores <- cforest(ME ~ ., data = mammoexp,
+                                  scores = list(ME = c(1, 3, 4)),
+                                  control = cforest_unbiased(ntree = 1000))
```

2.1.2 Using Classification Trees

In order to fit a random forest that consists of classification trees one has to define the response variable as an *unordered factor variable*.

```
> mammoexp$ME <- factor(mammoexp$ME, ordered = FALSE)
```

By specifying the response as unordered factor variable, the `cforest` function automatically uses classification trees.

```
> RF_classification <- cforest(ME ~ ., data = mammoexp,
+                              control = cforest_unbiased(ntree = 1000))
```

Again, we can check if classification trees were used:

```
> RF_classification@responses@is_nominal

  ME
TRUE
```

2.2 Computing Variable Importance

Having created an R object of class `cforest` that either makes use of ordinal regression trees or classification trees, we can compute variable importances using the VIMs described in Janitza et al. (2014). We compute importances based on the random forest objects `RF_ordinal` and `RF_ordinal_alt_scores` from Section 2.1.1 for illustrative purposes. One can compute importances based on `RF_classification` in exactly the same way. Though, we want to stress that the application of the RPS-, MAE- and MSE-based VIMs to classification trees does only make sense if the considered response variable has an inherent ordering. If the response variable is nominal (i.e., there is no meaningful ordering of the levels), we strongly advice against the use of these measures, since results will not be meaningful.

The following code computes the variable importances by the classical error rate based VIM, the RPS-based VIM, the MAE-based VIM and the MSE-based VIM. We compute the importances based on the random forest object `RF_ordinal`.

```
> ER_VI <- varimp(RF_ordinal) # error rate based variable importance (standard measure)
> RPS_VI <- varimpRPS(RF_ordinal) # RPS-based variable importance (novel VIM)
> MAE_VI <- varimpMAE(RF_ordinal) # MAE-based variable importance (novel VIM)
> MSE_VI <- varimpMSE(RF_ordinal) # MSE-based variable importance (existing VIM,
>                                 #   but has not been used for ordinal response data)
```

Note that the MAE- and the MSE-based VIMs make use of the scores $s(r), r = 1,\ldots,k$. These can be specified in the functions `varimpMAE` and `varimpMSE`.

```
> MAE_VI_alt_scores <- varimpMAE(RF_ordinal_alt_scores, scores = c(1, 3, 4))
> MSE_VI_alt_scores <- varimpMSE(RF_ordinal_alt_scores, scores = c(1, 3, 4))
```

If scores are not specified, the default scores, $s(r) = r$, are always used. Note that the specification of scores for computing the VIMs does technically not depend on the specification of the scores in the `cforest` function used for fitting the random forest. This means that it is technically feasible to define different scores for tree construction and VIM computation.

Figure 1 showing the variable importances by the four different VIMs, is produced by the following commands:

```
> par(mfrow = c(1, 4))
> barplot(ER_VI, ylab = "Error rate based variable importance", las = 2)
> barplot(RPS_VI, ylab = "RPS-based variable importance", las = 2)
> barplot(MAE_VI, ylab = "MAE-based variable importance", las = 2)
> barplot(MSE_VI, ylab = "MSE-based variable importance", las = 2)
> mtext(side = 3, text = "Variable importance by different measures", outer = TRUE, line = -2)
```

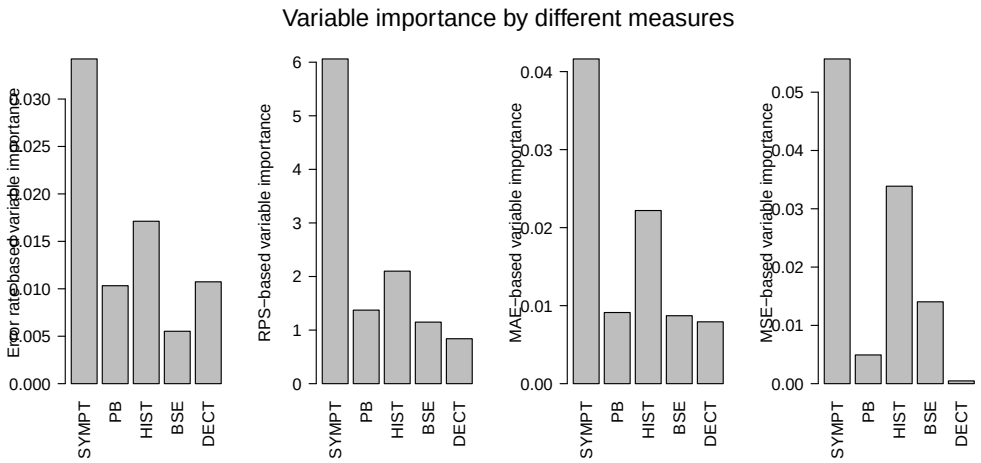

Figure 1: Variable importance obtained for the Mammography Data when using the standard error rate based importance (first plot), the novel RPS-based importance (second plot), the novel MAE-based importance (third plot) and the MSE-based importance (fourth plot). All measures were computed from ordinal regression trees with equally spaced scores.

References

Janitza, S., Tutz, G. and Boulesteix, A.-L. (2014). Random forests for ordinal response data: prediction and variable selection, *Technical Report 174*, Department of Statistics, University of Munich.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *http://www.R-project.org/*