LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Roman Hornung

Ordinal Forests

Technical Report Number 212, 2017
Department of Statistics
University of Munich

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

Ordinal Forests

Roman Hornung[1*]

October 23, 2017

[1] Institute for Medical Information Processing, Biometry and Epidemiology, University of Munich, Munich, 81377, Germany

Abstract

The prediction of the values of ordinal response variables using covariate data is a relatively infrequent task in many application areas. Accordingly, ordinal response variables have gained comparably little attention in the literature on statistical prediction modeling. The random forest method is one of the strongest prediction methods for binary response variables and continuous response variables. Its basic, tree-based concept has led to several extensions including prediction methods for other types of response variables.

In this paper, the ordinal forest method is introduced, a random forest based prediction method for ordinal response variables. Ordinal forests allow prediction using both low-dimensional and high-dimensional covariate data and can additionally be used to rank covariates with respect to their importance for prediction.

Using several real datasets and simulated data, the performance of ordinal forests with respect to prediction and covariate importance ranking is compared to competing approaches. First, these investigations reveal that ordinal forests tend to outperform competitors in terms of prediction performance. Second, it is seen that the covariate importance measure currently used by ordinal forest discriminates influential covariates from noise covariates at least similarly well as the measures used by competitors. In an additional investigation using simulated data, several further important properties of the OF algorithm are studied.

The rationale underlying ordinal forests to use optimized score values in place of the class values of the ordinal response variable is in principle applicable to any regression method beyond random forests for continuous outcome that is considered in the ordinal forest method.

*Corresponding author. Email: hornung@ibe.med.uni-muenchen.de.

1 Introduction

In statistical applications it is sometimes of interest to predict the values of an ordinal response variable. However, to date there are relatively few prediction methods for ordinal response variables that make use of the ordinal nature of such response variables; in particular, few such methods exist that are applicable to high-dimensional covariate data. Ordinal response variables are often treated as nominal variables, applying prediction techniques for binary response variables to all paired combinations of classes of the ordinal response variable.

In this paper, the ordinal forest (OF) method is introduced, an innovative prediction method for ordinal response variables applicable to both low-dimensional and high-dimensional covariate data that makes use of the ordinal nature of the response variable. Roughly spoken, assuming a latent variable model that is also underlying the well-known ordered probit regression, the ordinal response variable is treated as a continuous variable, where the differing extents of the individual classes of the ordinal response variable are implicitly taken into account (see below for details). OFs are closely related to conventional random forests (Breiman, 2001) for continuous outcomes, termed regression forests in the following. In contrast to the latter, OFs make use of the out-of-bag (OOB) error estimates during the construction of the forest.

A straightforward forest-based prediction method for ordinal response variables would consist of simply considering a regression forest using the class values $1, \ldots, J$ of the response variable for the corresponding classes. However, this procedure is suboptimal as will be demonstrated in this paper. An important reason for the suboptimality of this approach is the fact that the extents of the classes of the ordinal response variable, from now on denoted as "class widths", differ from class to class. In the latent variable model already mentioned above that is underlying OF, these class widths are the widths of J adjacent intervals in the range of an underlying continuous response variable; these J intervals correspond to the J classes of the ordinal response variable. The single assumption in this model is that, underlying the observed ordinal variable y, there exists a particular—known or unknown—refined continuous variable y^* that determines the values of the ordinal variable. The relationship between this continuous variable y^* and y is such that the higher the value of y^* is for an observation, the higher is the class of the ordinal response variable for that observation. More precisely, if y^* falls into the jth interval of J adjacent intervals, y will take the value j. As an illustration, school grades are usually determined by the total number of points the pupils score in all exercises composing the respective test. Here, the grade and the corresponding number of points take the role of the ordinal variable y and the underlying continuous variable y^*, respectively.

If the continuous variable y^* is known, it can be used in regression tech-

niques for continuous response variables. In the context of conditional inference trees, for situations in which y^* is known, Hothorn *et al.* (2006) suggest to use—as a continuous response variable—the midpoints of the intervals of y^* that correspond to the classes of y.

The OF method is, however, designed for the common situation in which the underlying continuous variable was not measured or might not even be known. In OF, interval boundaries in y^* corresponding to the different classes of y are estimated or rather optimized by maximizing the OOB prediction performance of regression forests. Using score values that correspond to these optimized class intervals instead of using the class values $1, \ldots, J$ leads to an improvement in prediction performance as will be seen in the analyses presented in this paper. Note, however, that apart from considering optimized score values for the class values, choosing arbitrary score values for the class values does not necessarily impact the prediction accuracy notably. This is also suggested by the results of a study performed by Janitza *et al.* (2016), who considered regression forests with conditional inference trees as base learners using the class values $1, \ldots, J$ for the classes of the ordinal response variable. In order to study the robustness of their results they considered the score values $1, 2^2, \ldots, J^2$ in addition to the values $1, 2, \ldots, J$ and found no differing prediction accuracies between these two choices. Note that conditional inference trees (Hothorn *et al.*, 2006) are preferable over classical classification and regression trees (Breiman *et al.*, 1984) in the presence of categorical covariates, because in this situation only the former allow unbiased variable selection for splitting. However, for high-dimensional data using conditional inference trees in regression forests is computationally overly demanding due to the large quantity of permutation tests necessary to conduct in the case of this approach. High-dimensional data has, however, become a very common application field of random forests, in particular in the biomedical field. Therefore, in this paper classical regression forests are considered, which use regression trees (Breiman *et al.*, 1984). Although the prediction performance is increased by using score values that correspond to class intervals obtained via maximizing the OOB prediction performance instead of using the values $1, \ldots, J$, the widths of the estimated intervals are not useful for interpretation purposes: they carry no useful information on the actual class widths as will become evident in the analyses presented in this paper.

The paper is structured as follows. In section 2 the OF algorithm and how OF can be used for prediction and for ranking the importances of the covariates is described. Subsequently, using five real datasets and simulated data, in section 3 the performance of OF with respect to prediction accuracy and quality of its variable importance measure is extensively compared to that of other (forest-based) approaches. Moreover, several important properties of the OF algorithm are investigated empirically using simulated data in this section. In the discussion (section 4) the most important findings of

the paper are summarized, further related points are discussed and possibilities for future research are described.

2 Methods

2.1 Construction of OF prediction rules

Assume a sample $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i$ ($i \in \{1, \ldots, n\}$) designates the vector of covariates of observation i and $y_i \in \{1, \ldots, J\}$ correspondingly the class value of the ordinal response variable for that observation. Then an OF prediction rule is constructed as follows:

1. For $b = 1, \ldots, B_{\text{sets}}$ (e.g., $B_{\text{sets}} = 1000$):

   (a) Draw $J - 1$ instances of a $U(0, 1)$ distributed random variable and sort the resulting values. The sorted values are designated as $d_{b,2}, \ldots, d_{b,J}$. Moreover, set $d_{b,1} := 0$ and $d_{b,J+1} := 1$. Note that in the R package `ordinalForest` (see section 4) in which the OF algorithm is implemented, a more sophisticated algorithm for drawing $d_{b,2}, \ldots, d_{b,J}$ is used. See section A of Supplementary Material 1 for a description of this algorithm. It delivers a more heterogeneous collection of sets $\{d_{b,1}, \ldots, d_{b,J+1}\}$ ($b \in \{1, \ldots, B_{\text{sets}}\}$) across the iterations $1, \ldots, B_{\text{sets}}$. It is important that the collection of sets $\{d_{b,1}, \ldots, d_{b,J+1}\}$ ($b \in \{1, \ldots, B_{\text{sets}}\}$) is heterogeneous enough across the iterations $1, \ldots, B_{\text{sets}}$ to ensure that the best of the considered sets $\{d_{b,1}, \ldots, d_{b,J+1}\}$ feature an OOB prediction performance close to that of the best possible set (see also section H.1 in Supplementary Material 1).

   (b) Form a continuous response variable $\boldsymbol{z}_b = z_{b,1}, \ldots, z_{b,n}$ by replacing each class value j, $j = 1, \ldots, J$, in the ordinal response variable $\boldsymbol{y} = y_1, \ldots, y_n$ by the jth value in the score set $\boldsymbol{s}_b := \{s_{b,1}, \ldots, s_{b,J}\}$, where $s_{b,j} := \Phi^{-1}(c_{b,j})$ and $c_{b,j} := (d_{b,j} + d_{b,j+1})/2$ ($j \in \{1, \ldots, J\}$).

   (c) Grow a regression forest $f_{\boldsymbol{s}_b}$ with $B_{\text{ntreeprior}}$ trees (e.g., $B_{\text{ntreeprior}} = 100$) using $\boldsymbol{z}_b$ as response variable.

   (d) Obtain OOB predictions $\hat{z}_{b,1}, \ldots, \hat{z}_{b,n}$ of $z_{b,1}, \ldots, z_{b,n}$.

   (e) Obtain OOB predictions of $y_1, \ldots, y_n$ as follows: $\hat{y}_{b,i} = j$ if $\hat{z}_{b,i} \in]\Phi^{-1}(d_{b,j}), \Phi^{-1}(d_{b,j+1})]$ ($i \in \{1, \ldots, n\}$).

   (f) Assign a performance score $sc_b := g(\boldsymbol{y}, \widehat{\boldsymbol{y}}_b)$ to $f_{\boldsymbol{s}_b}$, where $\widehat{\boldsymbol{y}}_b := \hat{y}_{b,1}, \ldots, \hat{y}_{b,n}$ and g is a specific function, termed "performance function" in the following, the choice of which depends on context (see further down for details).

2. Be S_{best} the set of indices of the B_{bestsets} (e.g., $B_{\text{bestsets}} = 10$) regression forests constructed in 1. that feature the largest sc_b values. Then for each $j \in \{1, \ldots, J+1\}$ take the average of those $d_{b,j}$ values for which $b \in S_{\text{best}}$, resulting in a set of $J+1$ values denoted as $d_1, \ldots, d_{J+1}$.

3. Form a new continuous response variable $\boldsymbol{z} = z_1, \ldots, z_n$ by replacing each class value j, $j = 1, \ldots, J$, in the ordinal response variable $\boldsymbol{y} = y_1, \ldots, y_n$ by the jth value in the optimized score set $\{s_1, \ldots, s_J\}$, where $s_j := \Phi^{-1}(c_j)$ and $c_j := (d_j + d_{j+1})/2$ $(j \in \{1, \ldots, J\})$.

4. Grow a regression forest f_{final} with B_{ntree} trees (e.g. $B_{\text{ntree}} = 5000$) using $\boldsymbol{z}$ as response variable.

Three different variants of the performance function g (see below) are provided in the R package `ordinalForest` (see section 4), where the user chooses the most suitable version depending on the particular kind of performance the OF should feature. For example, in many situations it is of interest to classify observations from each class with the same accuracy independent of class sizes. In other situations, the main goal is to correctly classify as many observations as possible. The latter goal implies prioritizing larger classes at the expense of a lower classification accuracy with respect to smaller classes. Sometimes, specific classes are of particular importance, which should then be prioritized by the OF algorithm.

The performance function g has the following general form:

$$g(\boldsymbol{y}, \widehat{\boldsymbol{y}}) := \sum_{j=1}^{J} w_j \ \text{Yind}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j), \tag{1}$$

$$\text{where} \ \sum_j w_j = 1, \ \ \text{Yind}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j); = \text{sens}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j) + \text{spec}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j) - 1,$$

$$\text{sens}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j) := \frac{\#\{y_i = j \wedge \hat{y}_i = j : i \in \{1, \ldots, n\}\}}{\#\{y_i = j : i \in \{1, \ldots, n\}\}}, \ \text{and}$$

$$\text{spec}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j) := \frac{\#\{y_i \neq j \wedge \hat{y}_i \neq j : i \in \{1, \ldots, n\}\}}{\#\{y_i \neq j : i \in \{1, \ldots, n\}\}},$$

where '#' denotes the cardinality and $\widehat{\boldsymbol{y}} := \{\hat{y}_1, \ldots, \hat{y}_n\}$ represents an estimate of $\boldsymbol{y}$. As is apparent from the above definitions $\text{Yind}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j)$ denotes the Youden's index calculated with respect to class j. The higher the weight w_j assigned to class j is chosen, the stronger the performance of the OF with respect to distinguishing observations in class j from observations not in class j will tend to be.

In the following, three important special cases of g are presented that result from specific choices of $w_1, \ldots, w_J$:

- If observations from each class should be classified with the same accuracy, g should be specified as follows:

$$g_{\text{clequal}}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \sum_{j=1}^{J} \frac{1}{J} \ \text{Yind}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j) \tag{2}$$

- If as many observations as possible should be classified correctly, the weights of the classes should be proportional to their sizes, leading to the following choice for g:

$$g_{\text{clprop}}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \sum_{j=1}^{J} \frac{\#\{y_i = j : i \in \{1, \ldots, n\}\}}{n} \ \text{Yind}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j) \tag{3}$$

  Note that the prioritization of larger classes resulting from the use of g_{clprop} leads to a decreased classification performance for smaller classes.

- If it is merely relevant that observations in class j can be distinguished as reliably as possible from observations not in class j, g should be specified as follows:

$$g_{\text{clj}}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \text{Yind}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j) \ \ \text{(i.e., } w_j = 1) \tag{4}$$

2.2 Prediction using OF

A prediction of the value of the response variable of an independent observation i^* based on its covariate vector $\boldsymbol{x}_{i^*}$ is obtained as follows:

1. For $b = 1, \ldots, B_{\text{ntree}}$:
   (a) Apply the bth tree in f_{final} to observation i^* and obtain a prediction $\hat{z}_{i^*,b}$.
   (b) Obtain a class prediction from the bth tree: $\hat{y}_{i^*,b} = j$ if $\hat{z}_{i^*,b} \in]\Phi^{-1}(d_j), \Phi^{-1}(d_{j+1})]$.
2. Obtain a final class prediction from $\hat{y}_{i^*,1}, \ldots, \hat{y}_{i^*,B_{\text{ntree}}}$ by majority voting.

2.3 Variable importance measure of OF

The variable importance measure (VIM) of OF for covariate j is given as:

$$VI_j = \frac{1}{B_{\text{ntree}}} \sum_{b=1}^{B_{\text{ntree}}} \text{Err}(\mathbf{y}_{OOB,b,j}, \widehat{\mathbf{y}}'_{OOB,b,j}) - \text{Err}(\mathbf{y}_{OOB,b,j}, \widehat{\mathbf{y}}_{OOB,b,j}), \tag{5}$$

where

- $\mathbf{y}_{OOB,b,j}$ denotes the vector of class values of the OOB data of tree b from the OF f_{final},
- $\widehat{\mathbf{y}}'_{OOB,b,j}$ denotes the predictions of the class values of the OOB data from tree b from f_{final} obtained after randomly permuting the values of covariate j in the OOB data of tree b,
- $\widehat{\mathbf{y}}_{OOB,b,j}$ denotes the predictions of the class values of the OOB data of tree b from f_{final} without permuting the values of covariate j, and
- $\text{Err}(\{a_1,\ldots,a_M\},\{b_1,\ldots,b_M\}) = (1/M)\sum_{m=1}^{M} I(a_m \neq b_m)$, that is, the misclassification error is (currently) used as error function in this permutation variable importance measure.

3 Empirical studies

A real data analysis using five datasets and an extensive simulation study were performed in order to compare the performance of OF to that of competing (tree-based) methods for ordinal regression. Moreover, further specific properties of the OF algorithm were studied using the simulated data, for example, the appropriateness of the choices of the default values of the hyperparameters or the influence of the class distribution on the performance. For the sake of completeness, it is pointed out that the OF algorithm was developed prior to seeing the five datasets used in the real data analysis and setting up the simulation design.

The comparison of OF with the alternative methods was performed with respect to the following two aspects: aspect 1: prediction performance; aspect 2: quality of variable importance ranking. The following methods were compared: OF, multi-class random forest (RF), and regression forests in which the class values $1,\ldots,J$ of the ordinal response variable are used as score values. The latter are referred to as "naive OFs" in the following. In the real data analysis ordered probit regression was included as a fourth method. This method was, however, not suitable in the case of the simulation study, because the simulation design featured high numbers of covariates and ordered probit regression is only suitable in situations in which there is a small ratio between the number of covariates and the number of observations.

In all analyses the performance function g_{clequal} was used in the OF algorithm. This choice was made because the classification performance should in most applications not depend on the class sizes in the data. Moreover, the following values for the hyperparameters of OF were used: $B_{\text{sets}} = 1000$, $B_{\text{bestsets}} = 10$, $B_{\text{ntreeprior}} = 100$, $B_{\text{ntree}} = 5000$ (see section 2.1), and $N_{\text{perm}} = 500$ (for N_{perm} see section A of Supplementary Material 1).

All R code written to perform and evaluate the analyses presented in this paper and in Supplementary Material 1 as well as the datasets used in the real data analysis are made available in Supplementary Material 2.

3.1 Real data analysis

3.1.1 Data

In this analysis five real datasets that were also recently considered in Janitza *et al.* (2016) were used. This paper compared multi-class RF with naive OF, both, however, using conditional inference trees as base learners. Table 1 gives an overview of the five datasets. For details on the backgrounds of the datasets see Janitza *et al.* (2016).

3.1.2 Study design

Contrary to the case of simulated data, the effect sizes of the covariates are not known for real data. Therefore, in real data analysis, the ordinal regression methods can only be compared with respect to their prediction performance, however, not with respect to the quality of the variable importance ranking (obtainable only from the forest-based approaches).

In order to avoid overoptimism which results from re-using the same data to assess prediction performance that was previously used already for learning the corresponding prediction rule, 10-fold stratified cross-validation was used. First, the values of the ordinal response variable of the left out fold were predicted for each iteration of the cross-validation. After having obtained the predictions of the values of the ordinal response variable for all left out folds, that is, for all observations, second, the quality of these predictions was measured using a performance measure. This process was repeated 10 times to obtain more reliable results. Three performance measures were used to assess the quality of the predictions: weighted Kappa using quadratic weights, weighted Kappa using linear weights (Cohen, 1968), and Cohen's Kappa (Cohen, 1960) (i.e., weighted Kappa with zero off-diagonal weights). The weighted Kappa is a metric well suited for measuring the quality of predictions of ordinal data. This is because it allows one to consider also the benefit of predictions that are close to the true class values on the ordinal scale instead of considering only predictions equal to the true values as valuable (Ben-David, 2008). Quadratic and linear weights are the most commonly used weighting schemes for weighted Kappa in practice. Compared to the case of using linear weights, when using quadratic weights more benefit to predictions that are further away from the true class values is attributed. At the same time, when using quadratic weights, fewer benefit is attributed to predictions very close to or equal to the true class values than in the case of using linear weights. By contrast, in the case of Cohen's Kappa, benefit is attributed only to predictions that are equal to the true

Dataset label	Num. of classes	Num. of observ.	Num. of covariates	Response variable with class sizes
mammography	3	412	5	Last mammography visits: 1 – never ($n = 234$); 2 – within a year ($n = 104$); 3 – over a year ($n = 74$)
nhanes	5	1914	26	Self-reported health status: 1 – excellent ($n = 198$); 2 – very good ($n = 565$); 3 – good ($n = 722$) 4 – fair ($n = 346$); 5 – poor ($n = 83$)
supportstudy	5	798	15	Functional disability: 1 – patient lived 2 months, and from an interview (taking place 2 months after study entry) there were no signs of moderate to severe functional disability ($n = 310$) 2 – patient was unable to do 4 or more activities of daily living 2 months after study entry; if the patient was not interviewed but the patients surrogate was, the cutoff for disability was 5 or more activities ($n = 104$) 3 – Sickness Impact Profile total score is at least 30 2 months after study entry ($n = 57$) 4 – patient intubated or in coma 2 months after study entry ($n = 7$) 5 – patient died before 2 months after study entry ($n = 320$)
vlbw	9	218	10	Apgar score: 1 – 1 (life-threatening) ($n = 33$); 2 – 2 ($n = 16$); 3 – 3 ($n = 19$); 4 – 4 ($n = 15$); 5 – 5 ($n = 25$); 6 – 6 ($n = 27$); 7 – 7 ($n = 35$); 8 – 8 ($n = 36$); 9 – 9 (optimal physical condition) ($n = 12$)
winequality	6	4893	11	Wine quality score: 1 – 3 (moderate quality) ($n = 20$); 2 – 4 ($n = 163$); 3 – 5 ($n = 1457$) 4 – 6 ($n = 2198$); 5 – 7 ($n = 880$); 6 – 8 (high quality) ($n = 175$)

Table 1: Overview of the datasets used in the real data analysis. The following information is given for each dataset: dataset label, number of classes of ordinal response variable, number of observations, number of covariates, description of response variable and its classes together with their sizes

class values. Thus, when using Cohen's Kappa, predictions that are not equal to the true values are attributed no benefit, regardless of how close these predictions are to the true values. To summarize, when using weighted Kappa with quadratic weights, similar benefit is attributed to predictions that are equal, similar, or only roughly similar to the true class values, when using Cohen's Kappa benefit is attributed only to predictions that are equal to the true class values and weighted Kappa with linear weights poses a compromise between the former two metrics.

The following is an illustration of the behavior of the different metrics: a prediction rule that in many cases returns accurate predictions, but in other cases results in predictions that are far from the true value would be associated with a relatively high value of Cohen's Kappa but a relatively low level of weighted Kappa.

Note that Cohen's Kappa and the weighted Kappa depend on the class sizes and the number of classes (Jakobsson and Westergren, 2005). This does, however, not pose a problem in the analysis performed in this paper, because the different methods are compared with each other for a given dataset (or given simulation setting in the case of the simulation, see section 3.2). Weighted Kappa and Cohen's Kappa take values between zero and one, where higher values indicate a better performance in terms of the respective metric.

3.1.3 Results

Figure 1 shows the values of the linearly weighted Kappa obtained for each of the five datasets. For the sake of clarity and because the linearly weighted Kappa poses a compromise between the quadratically weighted Kappa and Cohen's Kappa, the values obtained for the latter two metrics are presented in Supplementary Material 1 (Supplementary Figures 1 and 2). Correspondingly, the descriptions of the results will first focus on the linearly weighted Kappa and subsequently important differences in the results obtained for the quadratically weighted Kappa and Cohen's Kappa will be discussed briefly. For the sake of brevity, "linearly weighted Kappa" will occasionally be denoted as "Kappa" for short in cases where there is no chance of confusion.

For two of the five datasets, OF performs notably better than naive OF in terms of the linearly weighted Kappa. For the remaining three datasets the values are similar between these two methods. Nevertheless, the means over the 10 cross-validation iterations are higher for OF than for naive OF in the cases of all five datasets. OF is also better than multi-class RF for those two datasets for which OF is clearly better than naive OF. For the other three datasets the means over the cross-validation iterations are similar between OF and multi-class RF. For the dataset "supportstudy" the means over the cross-validation iterations are almost virtually identical between OF and multi-class RF (OF: 0.40450, multi-class RF: 0.40454) and for the

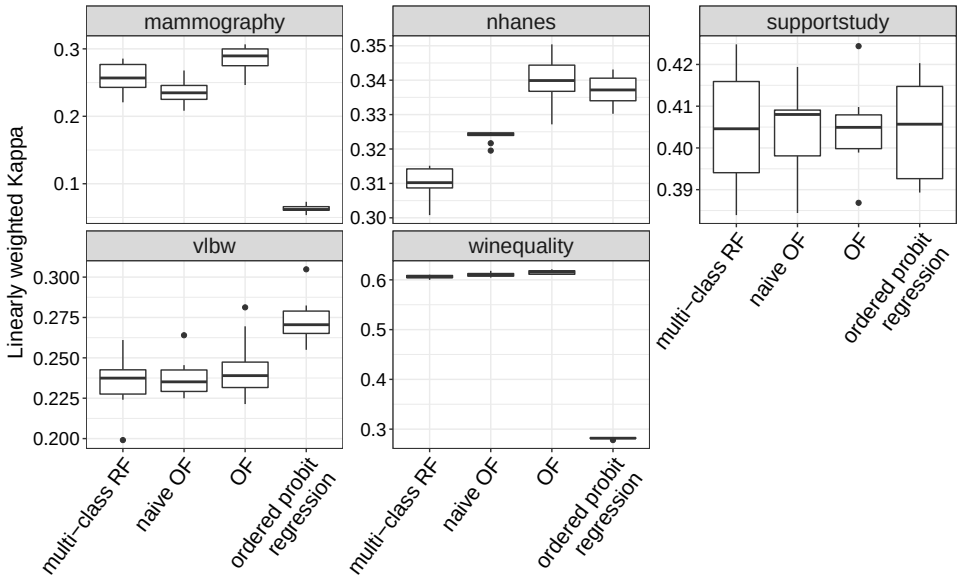

Figure 1: Values of linearly weighted Kappa for each of the five datasets and each of the four methods considered. Each boxplot shows the values obtained for the individual repetitions of the 10-fold stratified cross-validation.

datasets "vlbw" and "winequality" the means are slightly higher for OF. Ordered probit regression performs very similar to OF for two datasets, better than OF for one dataset, and much worse than OF and the other methods for the remaining two datasets. For the latter two datasets there were convergence problems in the case of ordered probit regression. Moreover, one cross-validation iteration resulted in an error for one of these datasets ("winequality"), which is why for this dataset, in the case of ordered probit regression, the results of only nine instead of 10 repetitions of the 10-fold stratified cross-validation are available.

The results are very similar for the quadratically weighted Kappa (Supplementary Figure 1). While the Kappa values are generally higher for quadratic weights than for linear weights, the improvement of OF over multi-class RF tends to be stronger for the quadratically weighted Kappa. This observation might be interpreted as follows: In contrast to multi-class RF, OF takes the ordinal nature of the response variable into account. Therefore, OF can be expected to deliver less often predictions that are far from the true class value on the ordinal scale than does multi-class RF.

In the case of Cohen's Kappa (Supplementary Figure 2), for which, as mentioned above, benefit is attributed only to predictions that are equal to the true class values, OF performs less well in comparison to multi-class RF: For two datasets, OF performs slightly better than multi-class RF and for

three datasets OF performs slightly worse. Thus, while OF can be expected to deliver more often predictions that are close to the true class values than multi-class RF, OF might at the same time be less performant with respect to predicting the exact values of the true class values in comparison to multi-class RF.

3.2 Simulation study

The real data analysis had the aim of comparing OF to alternatives with respect to prediction performance. In terms of the linearly weighted Kappa, OF outperformed multi-class RF for some datasets, where for other datasets the two methods performed comparably well.

Using simulated data it was possible to study various further properties of the OF algorithm. The simulation study had several aims:

1. Validate the findings on the prediction performance of OF compared to that of naive OF and multi-class RF that were obtained in the real data analysis.

2. Compare the variable importance measure of OF with that of naive OF and multi-class RF with respect to the abilities of these measures to discern influential variables from non-influential variables.

3. Investigate whether the class widths estimated by OF carry useful information on the true class widths.

4. Investigate how the class distribution influences the prediction performance of OF relative to that of naive OF.

5. Study whether the different variants of the performance function in the OF algorithm are actually associated with the specific kinds of prediction performance they are intended for.

6. Evaluate whether the default values chosen for the hyperparameters of OF are actually appropriate and study the robustness of the results obtained using OF with respect to changes in these default hyperparameter values.

3.2.1 Simulation design

Data with continuous response was simulated and the resulting continuous values were subsequently coarsened to obtain the class values of the ordinal response variable. This is in accordance with the latent variable model underlying OF, in which it is assumed that the values of the observed ordinal response variable are determined by a corresponding latent continuous response variable.

Janitza *et al.* (2016) whose work had a related scope to that of this paper (see again section 3.1.1) performed an extensive simulation study in their paper as well. They considered simulation settings with independent covariates, settings with sophisticated correlation patterns between influential covariates and settings with high-dimensional covariate data.

The simulation design used in the present paper is based on that of Janitza *et al.* (2016). Concerning the simulation of the covariates, the same settings were considered as in Janitza *et al.* (2016). However, the response was simulated differently in the present paper. Janitza *et al.* (2016) considered mixtures of pairs of proportional odds models where the corresponding two mixture components differed with respect to the influences of the covariates. As the simulation design considered in the present paper is not based on a mixture model but features a latent continuous response variable, in the present paper the first mixture components only were considered in each setting.

First, to obtain the values of the latent continuous response variable the values of the linear predictors of the respective first mixture components were calculated and, second, Gaussian noise with unit variance was added. In the calculation of the values of the linear predictors the values of the coefficients in the linear prediction functions were the same as those in the first mixture components in Janitza *et al.* (2016). Finally, the values of the latent continuous response variable were coarsened to obtain the class values of the ordinal response variable. In this step, two scenarios for the intervals corresponding to the different classes of the ordinal response variable were considered: scenario 1: intervals of equal width (starting / ending at the 0.001 quantile / 0.999 quantile of the marginal (normal) distribution of the continuous response variable); scenario 2: intervals of random width. In the case of scenario 2 the borders of the intervals were redrawn in iteration it of the simulation in the following way: 1) Draw $J-1$ instances of a $U(0,1)$ distributed random variable and sort these values. Denote the sorted values as $d_{\text{it},2}, \ldots, d_{\text{it},J}$; 2) Calculate $l_{\text{it},j} = Q(d_{\text{it},j})$, $j = 2, \ldots, J$, where $Q(\cdot)$ denotes the (approximated) quantile function of the marginal distribution of the latent continuous response variable. Set $l_{\text{it},1} = -\infty$ and $l_{\text{it},J+1} = +\infty$. Finally, define $]l_{\text{it},1}, l_{\text{it},2}], \ldots,]l_{\text{it},J}, l_{\text{it},J+1}]$ as the intervals used for simulation iteration it. The quantile functions $Q(\cdot)$ were approximated by sample quantiles obtained from a simulated dataset of size 50,000 for each of the three different scenarios "correlated", "independent", and "highdim" considered for the linear predictor (see next paragraph).

The following three setting parameters were varied:

- covariates and sample size:

    1. "correlated_n200": 65 covariates (15 with effect), partly correlated covariates, 200 observations

2. "independent_n200": 65 covariates (15 with effect), independent covariates, 200 observations
3. "correlated_n400": 65 covariates (15 with effect), partly correlated covariates, 400 observations
4. "independent_n400": 65 covariates (15 with effect), independent covariates, 400 observations
5. "highdim": 1015 covariates (15 with effect), partly correlated covariates, 200 observations

- type of intervals in latent continuous response variable:
  1. equal class widths
  2. random class widths
- number of classes of ordinal response variable:
  1. "nclass = 3": 3 classes
  2. "nclass = 6": 6 classes
  3. "nclass = 9": 9 classes

All combinations of the above parameter values were considered, which led to 30 ($5 \times 2 \times 3$) simulation settings. For each setting, 100 training datasets were generated and for each of these a corresponding independent test dataset of size 10,000 was generated to evaluate the prediction performances of the three considered methods.

3.2.2 Results

Prediction performance As in the case of the real data analysis, for clarity, the results obtained for the linearly weighted Kappa are presented in the paper and those obtained for the quadratically weighted Kappa and Cohen's Kappa are presented in Supplementary Material 1.

Equal class widths Figure 2, Supplementary Figure 3, and Supplementary Figure 4 show the results obtained for all settings with equal class widths for the linearly weighted Kappa, for the quadratically weighted Kappa, and for Cohen's Kappa. Again we will first focus on the results obtained for the linearly weighted Kappa and subsequently, if applicable, notable differences for the results obtained for the other two metrics will be explored. For all settings with equal class widths, OF features notably higher values of the linearly weighted Kappa than naive OF. Moreover, for most settings, naive OF performs better than multi-class RF. The exceptions are all settings with "nclass = 3". For these settings naive OF performs

equally well as multi-class RF with the exception of the setting with high-dimensional covariate data, where naive OF performs worse than multi-class RF. The latter setting is also the only setting for which OF performs slightly worse than multi-class RF. However, for this setting the Kappa values are generally very small.

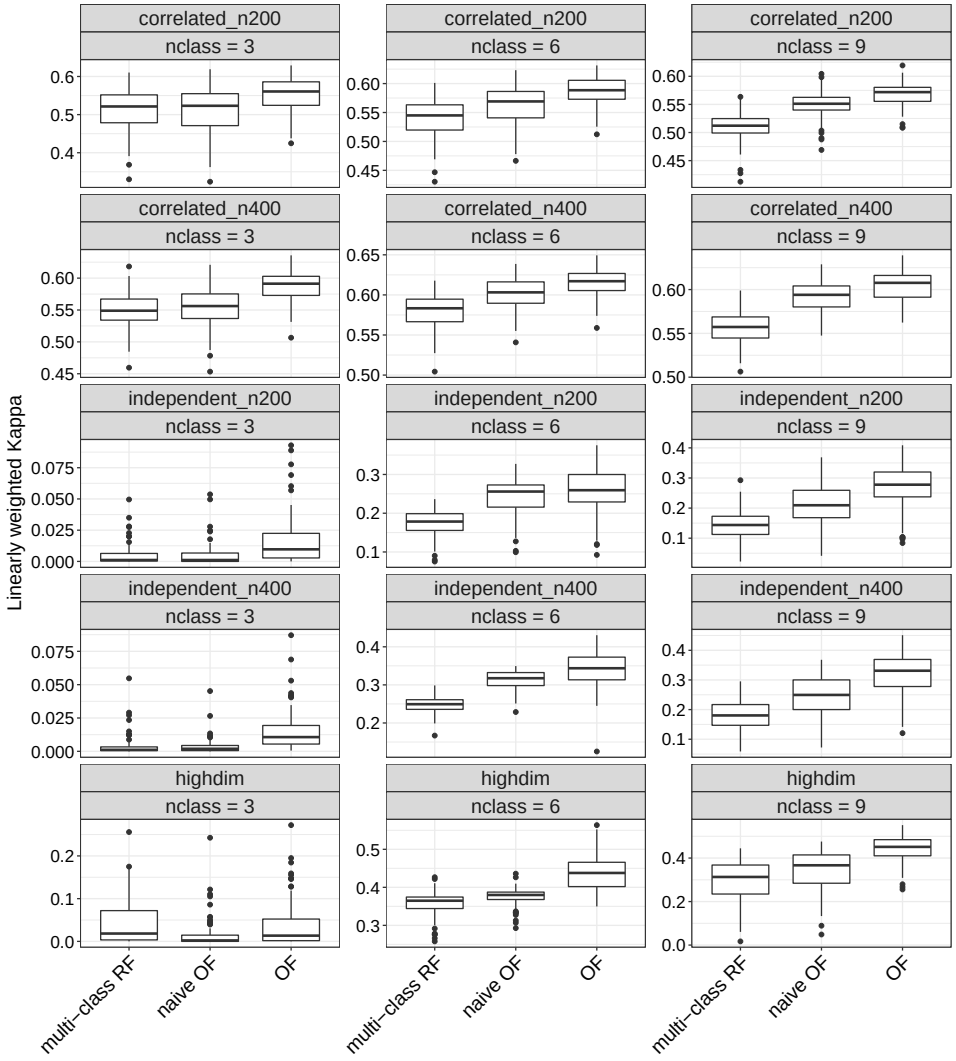

Figure 2: Values of linearly weighted Kappa for each simulation setting with equal class widths and each of the three methods considered. Each boxplot shows the values obtained on the corresponding test dataset for each of the 100 simulation iterations.

For all three methods, the Kappa values are very small in the settings with independent covariates and at the same time "nclass = 3". A natural

reason for the bad performance for settings with only three classes could be that for these settings the latent continuous response variable is coarsened to a very strong degree leading to a strong loss of signal. However, in cases of settings with correlated covariates the Kappa values are high also for settings with "nclass = 3". The reason why we observe high Kappa values for the correlated data but not for the uncorrelated data when considering three classes will be explained in the following. The correlation matrices of the influential variables in the settings with correlated covariates are structured as follows: 6 of the 15 influential variables have a common correlation of 0.8 among each other, while the remaining 9 variables are uncorrelated. Moreover, all regression coefficients are positive. As a consequence there are frequent tuples of covariate values that are either all low or all high for the same observation, which then features a low or a high value of the linear predictor that corresponds to class 1 or class 3, respectively. Thus, tuples of covariate values that are either all small or all high are associated with class 1 or class 3, respectively. As these tuples occur both in the training data and test data and as they are a characteristic pattern in the data that is easily learned by the regression methods, the resulting prediction rules perform comparably well with respect to predicting the classes 1 and 3. For the scenario with independent covariates by contrast, the methods almost always predict the large class in the middle. For this scenario, the relation between the values of the linear predictor and the covariate values is more complex, because the covariates all behave independently of each other. Therefore in the setting with independent covariates there is a greater loss of signal through the strong coarsening of the latent continuous response variable for "nclass = 3".

For the settings with high-dimensional covariate data, the improvement of OF over naive OF is the strongest. With the exception of the setting with independent covariates and at the same time "nclass = 3" the Kappa values are higher for the larger training set sizes, as expected. The results for the two training sets sizes do, however, differ hardly in terms of the performances of the three methods relative to each other. Apart from the observations made above, there seems to be no consistent influence of the number of classes on the performances of the methods relative to each other.

The results obtained for the quadratically weighted Kappa (Supplementary Figure 3) are very similar to that obtained for the linearly weighted Kappa (apart from the fact that the values are larger for the quadratically weighted Kappa).

For Cohen's Kappa (Supplementary Figure 4) the results are overall similar to that obtained for the linearly weighted Kappa. However, for Cohen's Kappa there is slightly less improvement of OF over naive OF than in the case of the linearly weighted Kappa except in the cases of the settings with "nclass = 3". Keeping in mind that for Cohen's Kappa benefit is attributed only to predictions equal to the true class values, whereas for

weighted Kappa a similar benefit is attributed to predictions in the vicinity of the true class values, this can be interpreted in the following way: in the presence of equal class widths, there is more improvement of OF over naive OF with respect to approximately true predictions than there is with respect to exact predictions.

Random class widths Figure 3, Supplementary Figure 5, and Supplementary Figure 6 show the results obtained for the settings with random class widths.

In general, the differences between the three methods are much smaller than in the settings with equal class widths. Nevertheless, for almost all settings there is a small improvement of OF over naive OF. Moreover, for all settings with number of classes larger than three, there is a small improvement of naive OF over multi-class RF.

For the settings with independent covariates and the settings with high-dimensional covariate data the variances of the Kappa values are much higher than in the settings with equal class widths. This is not surprising as the individual sets of class widths generated for each dataset in the settings with random class widths can be expected to be associated with differing prediction performances for all three methods.

It is, however, surprising that the improvements of OF over naive OF are smaller for the random class widths than they were for the equal class widths. By contrast, at first sight, it would seem more natural to assume that there is no or only a slight improvement of OF over naive OF in the settings with equal class widths. In naive OF the distances between the score values used for the classes of the ordinal response variable are the same independently of the level of the classes. That is, the distance between the first and the second score is the same as that between the second and the third score and so on. Accordingly, in the settings with equal class widths, the neighboring interval midpoints have the same distances to each other. Therefore, it seems natural to assume that for these settings optimizing the interval borders to maximize prediction accuracy as performed by OF is unnecessary and that it is, by contrast, sufficient to use the score values $1, \ldots, J$, that is, to use naive OF. Thus, the finding that OF shows a considerable improvement over naive OF in the presence of equal class widths is counter-intuitive at first sight. In section "Influence of the class distribution on the performance of OF" the reason for this fact will be explored.

For the settings with high-dimensional covariates the improvement of OF over naive OF is stronger than for the other settings.

The results obtained for the quadratically weighted Kappa (Supplementary Figure 5) are again very similar to those obtained for the linearly weighted Kappa.

For Cohen's Kappa (Supplementary Figure 6) again the results are simi-

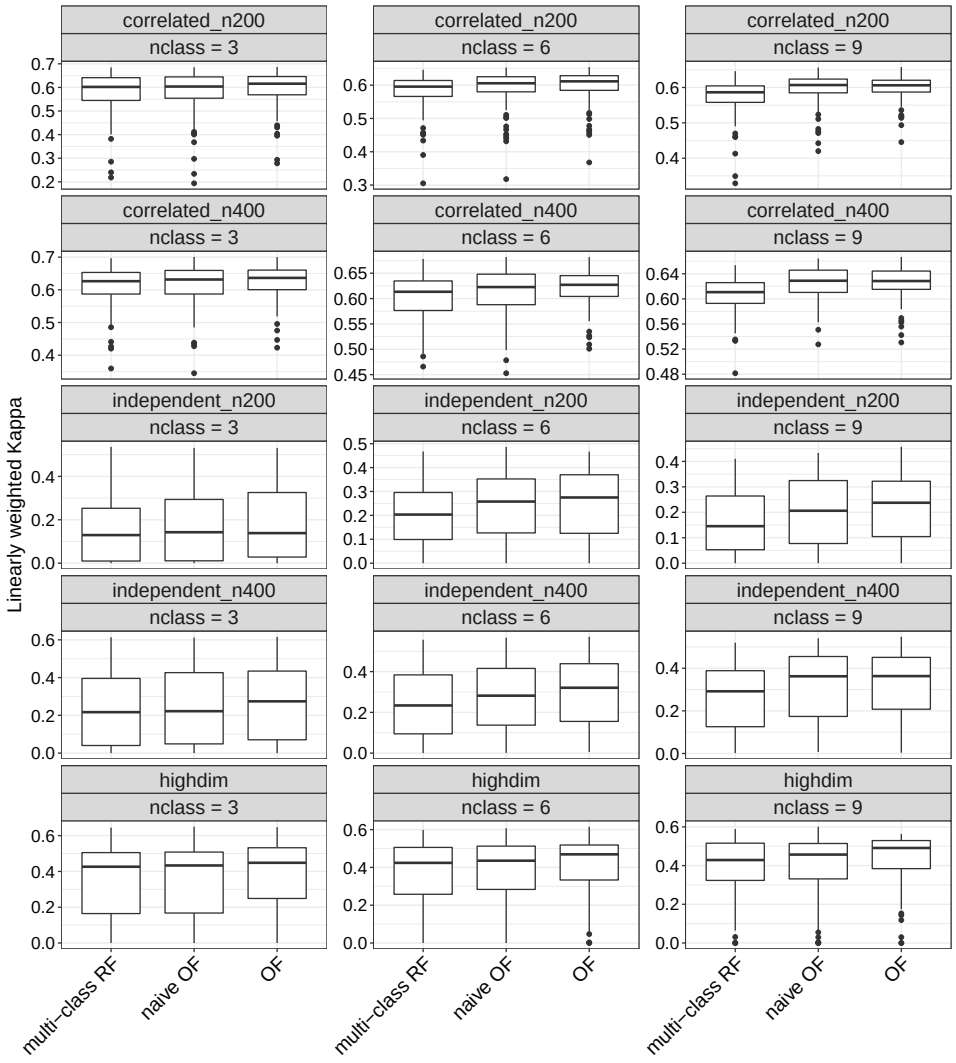

Figure 3: Values of linearly weighted Kappa for each simulation setting with random class widths and each of the three methods considered. Each boxplot shows the values obtained on the corresponding test dataset for each of the 100 simulation iterations.

lar to those obtained for the weighted Kappa. However, the improvement of OF over naive OF for the settings with high-dimensional covariates and at the same time number of classes greater than three is much smaller for Cohen's Kappa than for the linearly weighted Kappa. Thus, for these settings the same conclusion can be drawn as in the case of the equal class widths for all settings with number of classes greater than three: the improvement of OF over naive OF is greater with respect to approximately true predictions

than with respect to exact predictions.

Variable importance The VIMs of OF, naive OF, and multi-class RF were compared with respect to their abilities to discern influential covariates from non-influential noise covariates. A good VIM should tend to attribute higher VI score values to influential covariates than to noise covariates. A suitable measure used in Janitza *et al.* (2016) to assess a variable ranking in this respect is obtained as follows: Consider the covariates as observations, where the influential covariates are "diseased" and the noise variables are "healthy", and the VI score values as score values of a medical test and calculate the area under the ROC curve (AUC) for this scenario. This AUC value can be interpreted as the estimated probability that an influential variable has a higher VI score value than a noise variable.

Supplementary Figures 7 and 8 show the AUC values obtained for the VI rankings of OF, naive OF, and multi-class RF for each simulated dataset in each simulation setting (see section 3.2.1). While the AUC values are very similar between the three methods for almost all settings with "nclass = 3", they tend to be higher for OF and naive OF than for multi-class RF for all settings with "nclass = 6" and "nclass = 9". Correspondingly, in their simulation Janitza *et al.* (2016) found higher AUC values for naive OF using conditional inference trees than for multi-class RF, also using conditional inference trees. While we thus observe relevant differences between the AUC values obtained for OF and naive OF on the one hand and multi-class RF on the other, the differences between the AUC values obtained for OF and naive OF are minimal and inconsistent.

Estimation of class widths Beyond utilizing OF to predict the values of an ordinal response variable using covariate information and to rank the importances of the covariates, it would be desirable to obtain information on the true class widths using OF. As the range of the underlying latent continuous response variable is not known, it is certainly not possible to make inference on the absolute class widths. However, the magnitudes of the estimated class widths relative to each other might carry information on the magnitudes of the actual class widths relative to each other.

In section E of Supplementary Material 1 an extensive analysis of the relation between the true class widths underlying the simulated datasets (see section 3.2.1) and the corresponding class widths estimated by OF is presented.

This analysis reveals the following: 1) For the low and high classes on the ordinal scale there is a rather strong relation between true and estimated class widths. This relation is, however, often negative and depends on the setting; 2) The low and high classes on the ordinal scale tend to be associated with large estimated class widths; 3) More generally: the closer the classes

are to the center of the class value range the smaller their estimated class widths tend to be.

Summarizing, it can be concluded that OF cannot be used to make inference on the magnitudes of the class widths relative to each other.

Influence of the class distribution on the performance of OF In section F of Supplementary Material 1 an analysis is presented in which it was investigated whether there are specific kinds of class distributions for which OF performs particularly well in comparison to naive OF. In this analysis the true partitions of $[0, 1]$ considered in the simulation in the scenario with random class widths (see section 3.2.1) for the individual datasets were contrasted with the corresponding performances of OF relative to those of naive OF.

It is seen from this analysis that the improvement of OF over naive OF is stronger if there are large classes around the center of the class value range and, at the same time, the low and the high classes tend to be small. This explains the good performance of OF in comparison to that of naive OF in the simulation in the case of the equal class widths: The class distributions with equal class widths are associated with larger classes around the center of the class value range and increasingly smaller classes for lower and higher classes, respectively. The latter is due to the fact that the latent continuous response variable is normally distributed (see section 3.2.1).

Influence of the choice of the performance function used in the OF algorithm In section 2.1 three different variants of the performance functions were introduced. The choice for one of these variants depends on the kind of performance the OF should feature. In section G of Supplementary Material 1 a simulation study is presented using which it was investigated in how far these three different variants are actually associated with the specific kinds of prediction performance they are intended for. The data was simulated as described in section 3.2.1, using, however, only 50 instead of 100 pairs of training and test datasets for each setting.

In this simulation, all of the three variants of the performance functions were indeed associated with the specific kinds of prediction performance they are intended for. Nevertheless, frequently, the differences between the results when applying the different performance functions were not large.

Hyperparameters in the OF algorithm: appropriateness of their default values and robustness of the results with respect to the choices of their values As seen in section 2.1, the OF algorithm depends on several hyperparameters. These are:

1. number B_{sets} of score sets tried prior to the calculation of the optimized score set

2. number B_{bestsets} of score sets with largest sc_b values, $b \in \{1, \ldots, B_{\text{sets}}\}$, that are used to calculate the optimized score set

3. number $B_{\text{ntreeprior}}$ of trees in the regression forests $f_{\boldsymbol{s}_b}$, $b \in \{1, \ldots, B_{\text{sets}}\}$, that are constructed for each of the score sets tried

4. number B_{ntree} of trees in the OF f_{final}

5. number N_{perm} of permutations of the class width ordering to try for the 2th to the B_{sets}th score set considered prior to the calculation of the optimized score set (see section A of Supplementary Material 1)

In section H of Supplementary Material 1, first, a detailed heuristic discussion on the influences of these hyperparameters of the OF algorithm on its performance is provided. Second, the results of a small simulation study are presented which was conducted to assess the appropriateness of the default hyperparameter values and the robustness of the OF performance with respect to the choices of the values of the hyperparameters. The results from this simulation study suggest that the chosen default values are indeed in a reasonable range. Nevertheless, for ultra-high dimensional data it might be necessary to choose a higher value for $B_{\text{ntreeprior}}$ than the considered default value 100. The OF performance was, moreover, seen to be quite robust to varying the values of the hyperparameters. The sensitivity of the results was greater with respect to the choice of the combination of the values of B_{sets} and B_{bestsets} and to that of the value of $B_{\text{ntreeprior}}$ than to the choices of the values of B_{ntree} and N_{perm}.

4 Discussion

In terms of prediction performance, OF tended to outperform both naive OF (i.e., OF with score values fixed to $1, \ldots, J$) and multi-class RF in both the real-data analysis and in the simulation.

The variable importance measures of both OF and naive OF outperformed that of multi-class RF with respect to their abilities to discern influential covariates from non-influential noise covariates. However, the variable importance measures of OF and naive OF performed comparably well. The variable importance measure of OF currently uses the misclassification error as an error measure (see section 2.3). Considering an error measure that is based on the respective performance function used in each case might lead to an improved variable importance measure.

In section "Influence of the class distribution on the performance of OF" we saw that OF performs particularly well in comparison to naive OF if the middle classes are larger than the low and high classes. This pattern of the distribution of the class sizes can be expected to be common in practice: the low and high classes are on the margins of the class value range, that is, the

extreme ends of the ordinal scale, which is why they tend to be represented by less observations than the classes in the middle.

The variant of the performance function g to be used has to be chosen by the user, where this choice depends on the specific kind of performance the OF should feature. In section 2.1, three variants of this performance function were provided: g_{clequal}, g_{clprop}, and g_{clj}. In section "Influence of the performance function used in the OF algorithm" we saw that each of these three variants is actually associated with the specific kind of prediction performance it is intended for. However, the differences in results obtained when using the three different performance functions were in general not very large. In particular, g_{clprop} was only slightly superior to g_{clequal} in terms of the metric for which it should perform best from a theoretical point of view (for details see section G, Supplementary Material 1). However, g_{clequal} was clearly superior to g_{clprop} in terms of the metric for which this performance function should perform best. Given that g_{clequal}, therefore, did not perform considerably worse than g_{clprop} in any of the settings studied and was at the same time superior to g_{clprop} in many of the settings, it is reasonable to recommend using g_{clequal} as default performance function in situations in which it is not clear which specific kind of performance the OF should feature.

The OF algorithm features several hyperparameters. However, in section "Hyperparameters in the OF algorithm" we saw that its prediction performance is quite robust to the choices of the values of these hyperparameters, which is why it should not be necessary to optimize them in most cases. Similarly, in a work in progress by Probst *et al.* (prep) it is seen that the random forest algorithm seems to be quite robust to the choice of the values of its hyperparameters: using a large quantity of open source datasets Probst *et al.* (prep) show that for random forests there is quite little improvement by optimizing the values of the hyperparameters, in particular when compared to other machine learning approaches. Independent of the robustness of the OF algorithm to the choices of the values of its hyperparameters, the analysis discussed in section "Hyperparameters in the OF algorithm" also suggests that the default values used for the hyperparameters in the analyses presented throughout this paper have reasonable orders of magnitude.

The main concept of OF is to use optimized score values in place of the class values of the ordinal response variable in the vein of a latent variable model that is also underlying classical ordered probit regression. This concept is in principle applicable to any regression method for continuous outcome. The corresponding algorithm could be performed analogously to the OF algorithm, with the following differences: 1) In step 1 (c) the respective regression method would be repeatedly fitted to bootstrap samples using $\boldsymbol{z}_b$ as response variable and in step 1 (d) the OOB predictions of this bootstrapped prediction rule would be calculated; 2) In step 4 the regression method would be fitted to the data using $\boldsymbol{z}$ as response variable.

However, in the presence of high-dimensional covariate data this algorithm would be too computationally intensive in general. In this algorithm, the regression method has to be fitted very often and fitting a regression method to high-dimensional data is computationally intensive in most cases. However, regression forests can be constructed very fast also in the presence of high-dimensional data using the R package `ranger` (Wright and Ziegler, 2017), which is why the computational expense of the OF algorithm is reasonable also for such data in general. Ultra-high dimensional data, nevertheless, could pose a problem. Another problematic setting is datasets with very large numbers of observations, where applying the OF algorithm using its default hyperparameter values could be difficult to infeasible, because for such data the construction of the regression forests takes considerably longer. For data with a very large number of observations that, at the same time, features low-dimensional covariate data, an easy possibility to reduce the computational burden considerably without affecting the precision notably is to choose a small value for $B_{\text{ntreeprior}}$ (e.g., $B_{\text{ntreeprior}} = 10$). The reason why the precision of the OF is not considerably reduced by choosing a small $B_{\text{ntreeprior}}$ in this situation is that in the case of a large number of observations, the individual trees in the regression forests are more precise. Thus, a smaller number of trees in the regression forests f_{s_b}, $b = 1, \ldots, B_{\text{sets}}$, is necessary to obtain reliable sc_b values.

A related concept to using optimized score values in place of class values of the ordinal response variable is considered by Casalicchio et al. in a work in progress. They consider the following approach: 1) Fit a regression model for continuous outcome to the data using the score values $1, \ldots, J$ for the values of the ordinal response variable; 2) For obtaining predictions of the class values of new observations, assign an observation to class j ($j \in \{1, \ldots, J\}$), if its predicted value $\hat{y}$ is contained in the interval $]a_j, a_{j+1}]$, where $a_1 = -\infty$, $a_{J+1} = +\infty$, and the values $a_2, \ldots, a_J$ are chosen so that they minimize the cross-validation error of the predictions of the class values. Casalicchio et al. plan to compare the prediction performances obtained using this approach for various regression methods for continuous outcome.

As seen in this paper, OF is a well-performing prediction method for ordinal response variables. In addition to the purpose of predicting the values of the ordinal response variable, OF can be used to rank the covariates according to their importance for prediction. The estimated class widths resulting as a by-product of the OF algorithm do not, however, contain any useful information on the actual class widths. The OF algorithm is implemented in the R package `ordinalForest` that is available on CRAN in version 2.1 (Hornung, 2017).

Acknowledgements The author thanks Giuseppe Casalicchio for proofreading and comments and Jenny Lee for language corrections. This work was supported by the German Science Foundation (DFG-Einzelförderung BO3139/6-1 to Anne-Laure Boulesteix).

Supplementary Material

Supplementary Material 1: PDF file with further contents referred to in the paper; url: `http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/070_drittmittel/hornung/of_suppfiles/suppmat1_hornungtr.pdf`

This PDF file contains the following sections:

- A Algorithm used in R package `ordinalForest` for generating the class width sets
- B Supplementary Figures: real data analysis - (weighted) Kappa values
- C Supplementary Figures: simulation - (weighted) Kappa values
- D Supplementary Figures: simulation - AUC values obtained for variable importance measures
- E Estimation of class widths
- F Influence of the class distribution on the performance of OF
- G Influence of the choice of the performance function used in the OF algorithm
- H Hyperparameters in the OF algorithm: appropriateness of their default values and robustness of the results with respect to the choices of their values

Supplementary Material 2: R code and datasets used in the real data analysis; url: `http://www.ibe.med.uni-muenchen.de/organisation/mitarbeiter/070_drittmittel/hornung/of_suppfiles/suppmat2_hornungtr.zip`

All R code written to perform and evaluate the analyses presented in this paper and in Supplementary Material 1 as well as the datasets used in the real data analysis

References

Ben-David, A. (2008). Comparison of classification accuracy using Cohen's Weighted Kappa. *Expert Systems with Applications*, **34**, 825–832.

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

Breiman, L., Friedman, J. H., Olshen, R. A., and Ston, C. J. (1984). *Classification and Regression Trees*. Wadsworth International Group, Monterey, CA.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, **20**, 37–46.

Cohen, J. (1968). Weighed kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, **70**, 213–220.

Hornung, R. (2017). *ordinalForest: Ordinal Forests: Prediction and Variable Ranking with Ordinal Target Variables*. R package version 2.1.

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

Jakobsson, U. and Westergren, A. (2005). Statistical methods for assessing agreement for ordinal data. *Scandinavian Journal of Caring Sciences*, **19**, 427–431.

Janitza, S., Tutz, G., and Boulesteix, A.-L. (2016). Random forest for ordinal responses: prediction and variable selection. *Computational Statistics and Data Analysis*, **96**, 57–73.

Probst, P., Bischl, B., and Boulesteix, A.-L. (in prep.). Tunability and importance of hyperparameters of machine learning algorithms.

Wright, M. N. and Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, **77**(1), 1–17.