Journal of Classification
https://doi.org/10.1007/s00357-018-9302-x

Ordinal Forests

Roman Hornung[1]

Published online: 22 January 2019

Abstract
The ordinal forest method is 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. An extensive comparison study reveals that ordinal forests tend to outperform competitors in terms of prediction performance. Moreover, 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. Several further important properties of the ordinal forest algorithm are studied in additional investigations. The rationale underlying ordinal forests of using 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.

Keywords Prediction · Ordinal response variable · Covariate importance ranking · Random forest

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

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00357-018-9302-x) contains supplementary material, which is available to authorized users.

✉ Roman Hornung
hornung@ibe.med.uni-muenchen.de

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

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. Analogously to the general class of regression models for ordinal data developed by McCullagh (1980) that includes the well-known ordered probit regression, the OF method is based on the notion of a latent continuous response variable underlying the observed ordinal response variable. Roughly spoken, with OFs, 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 the underlying continuous response variable; these J intervals correspond to the J classes of the ordinal response variable. The underlying refined continuous variable y^* determines the values of the ordinal variable y. 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 techniques 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. 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 are 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 with 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 are 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

In Section 2.1.1, first, to give an initial overview, the algorithm used for constructing an OF prediction rule is described in a simplified form. Second, some aspects of specific steps of this algorithm are discussed. The algorithm is subsequently described in full detail in Section 2.1.2.

2.1.1 Sketch of the OF Algorithm

The OF algorithm consists of the following two main steps:

1. Optimization of the score set:
   As described in the "Introduction," ordinal forests are regression forests in which the class values are replaced by score values that are optimized with the aim of maximizing the (OOB) prediction performance.

   The first step in the optimization of the score set $\{s_1, \ldots, s_J\}$ is performed as follows: First, repeatedly and randomly generate a candidate score set $\{s_{b,1}, \ldots, s_{b,J}\}$; second, construct an OF as a regression forest using $\{s_{b,1}, \ldots, s_{b,J}\}$ for the class values of the target variable; and lastly, measure the OOB prediction performance according to a specific measure, called the *performance function*.

In the second step, the final score set is calculated as a summary of the score sets that featured the highest OOB prediction performance in the first step.

2. Construction of the OF as a regression forest:
   Using $\{s_1, \ldots, s_J\}$ for the class values of target variable, construct an ordinal forest as a regression forest.

In the first step of the algorithm, it is important that the collection of tried score sets is large and as heterogeneous as possible. This ensures that the best of the tried score sets are close to the optimal score set that is associated with optimal OOB prediction performance. The heterogeneity of the collection of score sets is ensured by a specific algorithm described in Section 2.1.2.

As described above, the final score set is obtained as a summary of those score sets considered in the optimization that featured the highest OOB prediction performance. In principle, it would be possible to use the score set considered in the optimization which was associated with the highest OOB prediction performance as the final score set. However, the OOB prediction performance of the corresponding OF constructed in the optimization may be high merely by chance. However, when summarizing several score sets with the highest OOB prediction performance estimates, it is unlikely that these estimates are consistently high merely due to chance.

The choice of the form of the performance function (Section 2.1.2) used in the optimization depends on the kind of performance the ordinal forest should feature. For example, in many situations, it is of interest to classify observations from each class with the same accuracy independent of class size. 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.

2.1.2 Detailed Description of the OF Algorithm

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. As described in Section 2.1, the collection of score sets tried during the optimization should be as heterogeneous as possible. As seen below, each tried score set consists of the J (transformed) midpoints of J adjacent intervals that form a division of the interval [0, 1] into J adjacent intervals. The heterogeneity of the collection of tried score sets is reached by making the corresponding collection of divisions of [0, 1] into J adjacent intervals heterogeneous.

   The following algorithm is used for obtaining this heterogeneous collection of B_{sets} divisions $\{d_{b,1}, \ldots, d_{b,J+1}\}$ ($b \in \{1, \ldots, B_{\text{sets}}\}$) of [0, 1]:

   (a) In this step, the rankings of the class widths for each of the B_{sets} sets are generated. If $J! < B_{\text{sets}}$, this is performed as follows:

      i Generate all $J!$ possible permutations of $1, \ldots, J$ and permute this set of permutations randomly. The result of the latter step is the rankings $\{r_{b,1}, \ldots, r_{b,J}\}$, $b = 1, \ldots, J!$, for the first $J!$ sets.

ii Copy each of the rankings $\{r_{b,1}, \ldots, r_{b,J}\}$, $b = 1, \ldots, J!$, $\lfloor B_{\text{sets}}/J! \rfloor - 1$ times to produce the rankings $\{r_{b,1}, \ldots, r_{b,J}\}$, $b = J! + 1, \ldots, J!\lfloor B_{\text{sets}}/J! \rfloor$, for the next $J!(\lfloor B_{\text{sets}}/J! \rfloor - 1)$ sets.

iii The last rankings $\{r_{b,1}, \ldots, r_{b,J}\}$, $b = J!\lfloor B_{\text{sets}}/J! \rfloor + 1, \ldots, B_{\text{sets}}$, are produced through random sampling from the first $J!$ rankings $\{r_{b,1}, \ldots, r_{b,J}\}$, $b = 1, \ldots, J!$.

If $J! \geq B_{\text{sets}}$, the generation of the rankings of the class widths for each set is performed as follows:

i Permute $1, \ldots, J$ randomly once to produce the ranking $\{r_{1,1}, \ldots, r_{1,J}\}$ for the first set.

ii For $b = 2, \ldots, B_{\text{sets}}$: Draw N_{perm} (e.g., $N_{\text{perm}} = 500$) random permutations of $1, \ldots, J$ denoted as $\{r^*_{l,1}, \ldots, r^*_{l,J}\}$, $l = 1, \ldots, N_{\text{perm}}$. Determine that permutation from $\{r^*_{l,1}, \ldots, r^*_{l,J}\}$, $l = 1, \ldots, N_{\text{perm}}$ that features the greatest quadratic distance to $\{r_{b-1,1}, \ldots, r_{b-1,J}\}$, that is $\text{argmax}_{\{r^*_{l,1}, \ldots, r^*_{l,J}\}} \sum_{j=1}^{J} (r^*_{l,j} - r_{b-1,j})^2$ and use this permutation as the ranking for the bth set (see Appendix A.8.1 in the Supplementary Online Materials for a discussion of the influence of the value of N_{perm}).

(b) In this step, the sets $\{d_{b,1}, \ldots, d_{b,J+1}\}$ for all iterations $b = 1, \ldots, B_{\text{sets}}$ are generated.

For $b = 1, \ldots, B_{\text{sets}}$:

i 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 $d^*_{b,J+1} := 1$.

ii Re-order the intervals of the $[0, 1]$ partition $\{d^*_{b,1}, \ldots, d^*_{b,J+1}\}$ in such a way that the jth interval, $j = 1, \ldots, J$, is that interval out of $]d^*_{b,1}, d^*_{b,2}], \ldots,]d^*_{b,J}, d^*_{b,J+1}]$ that features the $r_{b,j}$th smallest width and use this re-ordered partition as the bth set $\{d_{b,1}, \ldots, d_{b,J+1}\}$.

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

(a) 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 $s_{b,j} := \Phi^{-1}(c_{b,j})$, $c_{b,j} := (d_{b,j} + d_{b,j+1})/2$ ($j \in \{1, \ldots, J\}$), and Φ^{-1} denotes the quantile function of the standard normal distribution.

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

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

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

(e) Assign a performance score $sc_b := g(\boldsymbol{y}, \widehat{\boldsymbol{y}}_b)$ to f_{s_b}, where $\widehat{\boldsymbol{y}}_b := \hat{y}_{b,1}, \ldots, \hat{y}_{b,n}$ and g is the performance function (see further down for details).

3. 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 $d_1, \ldots, d_{J+1}$.

4. 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 $c_j := (d_j + d_{j+1})/2$ $(j \in \{1, \ldots, J\})$.
5. 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.

The performance function g has the following general form:

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

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

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

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

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 plurality voting (i.e., use the class value that is predicted most often by the trees).

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 $N_{\text{perm}} = 500$.

All R code written to perform and evaluate the analyses presented in this paper and in Appendix A in the Supplementary Online Materials as well as the datasets used in the real data analysis are made available in Appendix B in the Supplementary Online Materials.

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, but 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. Second, after having obtained the predictions of the values of the ordinal response variable for all left out folds, that is, for all observations, 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).

The weighted Kappa is given by:

$$\kappa_w := \frac{\sum_{i=1}^{J}\sum_{j=1}^{J} w_{ij}\, p_{oij} - \sum_{i=1}^{J}\sum_{j=1}^{J} w_{ij}\, p_{cij}}{1 - \sum_{i=1}^{J}\sum_{j=1}^{J} w_{ij}\, p_{cij}}, \tag{6}$$

where p_{oij} is the observed proportion of cases for which class i is true and class j is predicted, p_{cij} is the proportion of cases for which class i is true and class j is predicted that is expected just by chance, and w_{ij} are so-called agreement weights. For the linearly weighted Kappa, the agreement weights are given by $w_{ij} = 1 - |i - j|/(J - 1)$, for the quadratically weighted Kappa by $w_{ij} = 1 - |i - j|^2/(J - 1)^2$ and for Cohen's Kappa by $w_{ij} = 1$ if $i = j$ and $w_{ij} = 0$ if $i \neq j$.

Table 1 Overview of the datasets used in the real data analysis

Dataset label	No. of classes	Sample size	No. 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 patient's 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)

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 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 0 and 1, 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 Appendix A in the Supplementary Online Materials (Supplementary Figs. 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 those 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 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

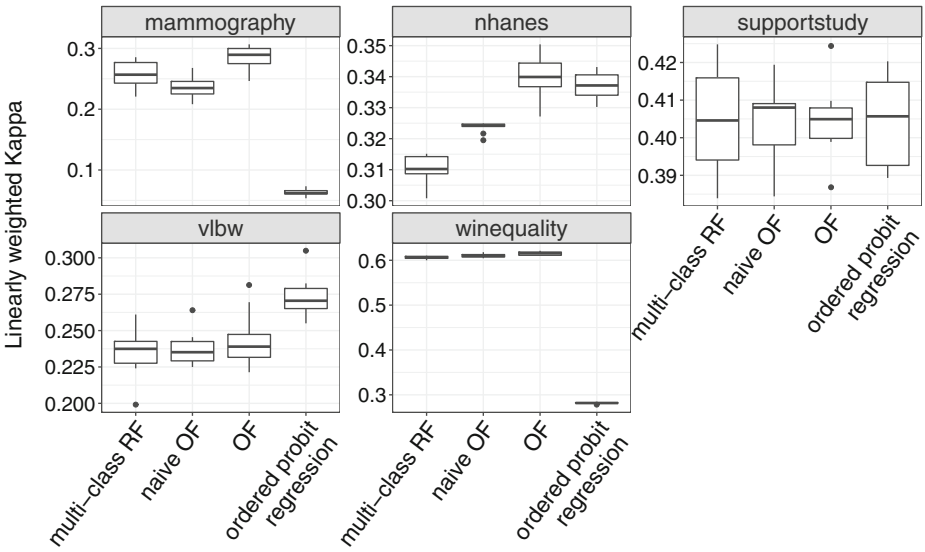

Fig. 1 Values of linearly weighted Kappa for each of the five datasets and each of the four methods considered; each boxplot shows the values obtained for the individual repetitions of the 10-fold stratified cross-validation

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 Fig. 1). While the Kappa values are generally higher for quadratic weights than those 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 Fig. 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 detailed aims and the design of the main simulation and of several additional simulation analyses are presented in the Supplementary Online Materials (Appendix A.2). Detailed results and discussions of the latter are also provided in the Supplementary Online Materials. In the following, the main findings of the simulation study are presented and important points related to these results are discussed. For details, see the corresponding sections of the Supplementary Online Materials.

OF outperformed both naive OF and multi-class RF in the majority of settings (Appendix A.3). The improvement of OF over naive OF was stronger for settings in which there were large classes around the center of the class value range and the low and the high classes tended to be small (Appendix A.6).

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 (Appendix A.4). 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 extensive analyses (Appendix A.5), it was revealed that the estimated class widths cannot be used in any way for making insightful inference on the magnitudes of the actual class widths (relative to each other).

All three variants of the performance functions (Section 2.1.2) were indeed associated with the specific kinds of prediction performance they were intended for in an additional simulation analysis presented in Appendix A.7. Nevertheless, frequently, the differences between the results when applying the different performance functions were not 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 Appendix A.7). 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} 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. Appendix A.8.1 provides a detailed heuristic discussion on the influences of these hyperparameters on the prediction performance of OF. In Appendix A.8.2, a simulation analysis is presented to assess the appropriateness of the default hyperparameter values (used in the R package `ordinalForest`) and the robustness of the OF prediction performance with respect to the choices of the hyperparameter values. This analysis suggests that the chosen default values are indeed in a reasonable range and that OF 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 by Probst et al. (2018), 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. (2018) 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. 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.

4 Discussion

The simulation study revealed 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 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 2 (b) (Section 2.1.2), the respective regression method would be repeatedly fitted to bootstrap samples using z_b as response variable, and in step 2 (c) the OOB predictions of this bootstrapped prediction rule would be calculated; (2) In step 5 the regression method would be fitted to the data using z as response variable. Note that this algorithm is not confined to parameteric approaches to ordinal regression in contrast to the the model class defined by McCullagh (1980). However, in the presence of high-dimensional covariate data, this algorithm would be (currently) 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 2.2 (Hornung 2018).

Acknowledgments 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).

References

Ben-David, A. (2008). Comparison of classification accuracy using Cohen's weighted Kappa. *Expert Systems with Applications*, *34*(2), 825–832.

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

Breiman, L., Friedman, J.H., Olshen, R.A., Ston, C.J. (1984). *Classification and regression trees*. Monterey: Wadsworth International Group.

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*(4), 213–220.

Hornung, R. (2018). ordinalForest: Ordinal Forests: Prediction and Variable Ranking with Ordinal Target Variables, R package version 2.2.

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

Jakobsson, U., & Westergren, A. (2005). Statistical methods for assessing agreement for ordinal data. *Scandinavian Journal of Caring Sciences*, *19*(4), 427–431.

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

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society Series B*, *42*(2), 109–142.

Probst, P., Bischl, B., Boulesteix, A.L. (2018). Tunability: importance of hyperparameters of machine learning algorithms. arXiv:1802.09596.

Wright, M.N., & 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.