Ordinal forests: Prediction and covariate importance ranking with ordinal response variables

Roman Hornung

Institute for Medical Information Processing, Biometry and Epidemiology, University of Munich

December, 21th, 2020

Introduction

- Ordinal forests (**OFs**) are a method for the prediction of ordinal outcomes using covariate information.

- They are **regression forests** that use **score values** $s_1, \dots, s_J$ in place of the class values $1, \dots, J$ of the ordinal outcome.

- **score values** $s_1, \dots, s_J$ **optimized** in such a way that the OF features an **optimal** (out-of-bag (OOB)) estimated **performance**

- concept of OF based on **latent variable assumption**: ordinal outcome assumed to be coarsened version of a latent metric variable

Algorithm

1. **Optimize score set**:
   - $s_1, \ldots, s_J$ optimized by: 1) generating **many score sets** randomly, 2) constructing an OF for each and 3) **averaging** across the ones associated with the **best** performance
   - performance measured via the out-of-bag observations of the trees in the forests using a **performance measure** $g(\cdot)$
   - **Different choices** of $g(\cdot)$ lead to **different kinds of performance**.

2. Grow a **OF** f_{final} with B_{final} trees (e.g. $B_{\text{final}} = 10^4$) using $s_1, \ldots, s_J$ as score set.

Performance measure $g(\cdot)$

- "**Equal**": Classify observations from all **classes** with the **same accuracy**.
- "**Proportional**": Classify **correctly** as **many observations** as possible (larger classes are given more weight).
- "**Oneclass**": Maximize the performance with respect to a **specific class**, disregarding the other classes.
- **NEW**: "**Probability**": Allow **conditional class probability estimation** (by using the ranked probability score).

R package `ordinalForest`

- **on CRAN**, version 2.4-1

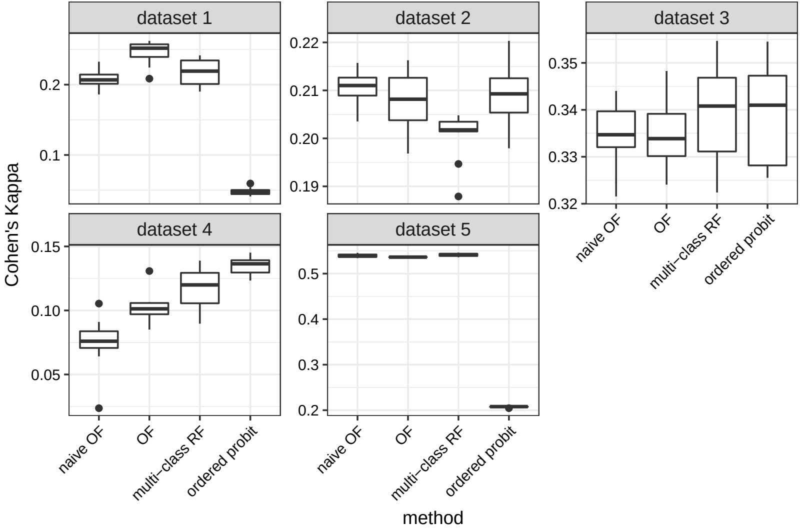
- **ordinalForest uses** (code from) R package **ranger** (fast!) to construct the regression forests

- **variable importance measure** (VIM) - two variants:
  1. based on **misclassification error rate** – importance with respect to **class point prediction**
  2. **NEW**: based on **ranked probability score** – importance with respect to **conditional class probability estimation**

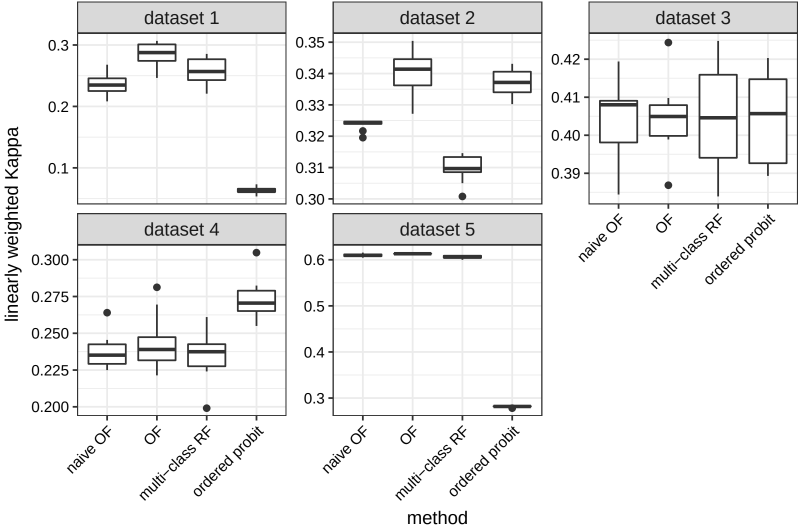
Real data analysis - study design

- **five** real **datasets**
- **four methods**: 1) OF using $1, \ldots, J$ as score values ("naive OF"), 2) OF, 3) multi-class Random Forests (RF), 4) ordered probit regression
- **goal**: Assess **prediction performance**.
- **performance metrics**: Cohen's Kappa, weighted Kappa
- **validation scheme**: 10 times repeated 10-fold **cross-validation**

Real data analysis - results: Cohen's Kappa



Real data analysis - results: linearly weighted Kappa



Simulation results

- **three considered methods**: 1) naive OF, 2) OF, 3) multi-class RF
- **prediction performance**: For all settings (with not negligibly small signal) **OF better** than naive OF and multi-class RF
- **variable importance** (based on misclassification error rate): **OF better** than multi-class RF, **no improvement over naive OF**
- **OF** particularly **effective when** the **low and high classes are smaller** (common in practice)

Outlook: Estimating conditional class probabilities and considering the ordinal scale in the ranking of the variables

- **Background**:

  1. in its **original form** (Hornung (2020a)) OF only suitable for point class prediction, **not conditional class probability estimation**

  2. Buri & Hothorn (2020): introduce Ordinal Transformation Forests (OTFs); find that **available performance measures** ("equal" and "proportional") **do not perform well** for conditional class probability estimation using OFs.

- $\Rightarrow$ **New performance measure**: Use of (negative) **ranked probability score** (RPS):

$$\text{RPS} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{J-1} \sum_{j=1}^{J} \left[\widehat{\mathbb{P}}(Y_i \leq j | \boldsymbol{X}_i = x_i) - I(Y_i \leq j) \right]^2$$

Outlook: Estimating conditional class probabilities and considering the ordinal scale in the ranking of the variables

- **Evaluated OFs for probability estimation** with the negative **RPS** using the **same simulation design** as Buri & Hothorn (2020).

  $\Rightarrow$ **Much better performance than for** performance measures "**equal**" and "**proportional**"; performance **comparable to OTFs** (depending on setting: similar, better or worse).

- **New VIM based on RPS**:

$$\mathrm{VIM}_j = \frac{1}{ntree} \sum_{i=1}^{ntree} \mathrm{RPS}_{tree,i}^{(\text{OOB-permuted j})} - \mathrm{RPS}_{tree,i}$$

\- **To Do**: Performance evaluation of new VIM...

Thank you for your attention!

References

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

Buri, M. and Hothorn, T. (2020).
Model-based random forests for ordinal regression.
International Journal of Biostatistics **16(2)**, 20190063.

Hornung, R. (2020a).
Ordinal Forests.
Journal of Classification **37**, 4–17.

Hornung, R. (2020b).
ordinalForest: Ordinal Forests: Prediction and Variable Ranking with Ordinal Target Variables.
R package version 2.4-1.
`http://cran.R-project.org/package=ordinalForest`

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.

Wright, M. and Ziegler, A. (2017).
ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R.
Journal of Statistical Software **77**, 1–17.

Performance measures $g(\cdot)$ for different purposes

Be $\text{Yind}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j)$ **Youden's index for predicting class** j, where $\boldsymbol{y}$ / $\widehat{\boldsymbol{y}}$ is the vector of true / predicted classes, then:

- equal classification performance for **all classes** $\Rightarrow$

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

- equal classification performance for **all observations** $\Rightarrow$

$$g(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \sum_{j=1}^{J} \frac{\#\{y_i = j : i \in \{1, \dots, n\}\}}{n} \text{Yind}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, j)$$

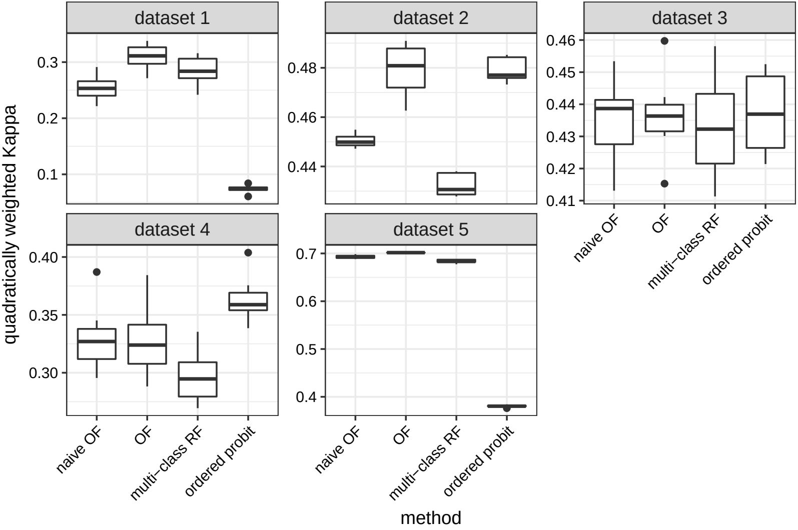
- optimal prediction performance for **class** l $\Rightarrow$

$$g(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \text{Yind}(\boldsymbol{y}, \widehat{\boldsymbol{y}}, l)$$

Real data analysis - datasets

Dataset	# observ.	# classes	class frequenc.	# covariates
mammography	412	3	234, 104, 74	5
nhanes	1914	5	198, 565, 722, 346, 83	26
supportstudy	798	5	310, 104, 57, 7, 320	15
vlbw	218	9	33, 16, 19, 15, 25, 27, 35, 36, 12	10
winequality	4893	6	20, 163, 1457, 2198, 880, 175	11

Real data analysis - results: quadratically weighted Kappa



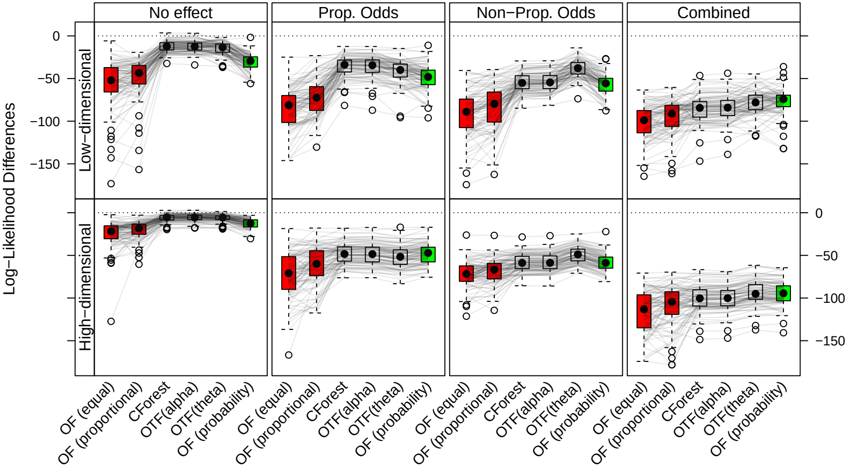
Simulation - study design

- **three methods**: 1) naive OF, 2) OF, 3) multi-class RF
- **30 settings**, 100 datasets per setting
- **setting parameters**:
  - correlations among covariates
  - sample size
  - number of covariates
  - numbers of classes of ordinal outcome
  - known true intervals considered for latent variable underlying ordinal outcome

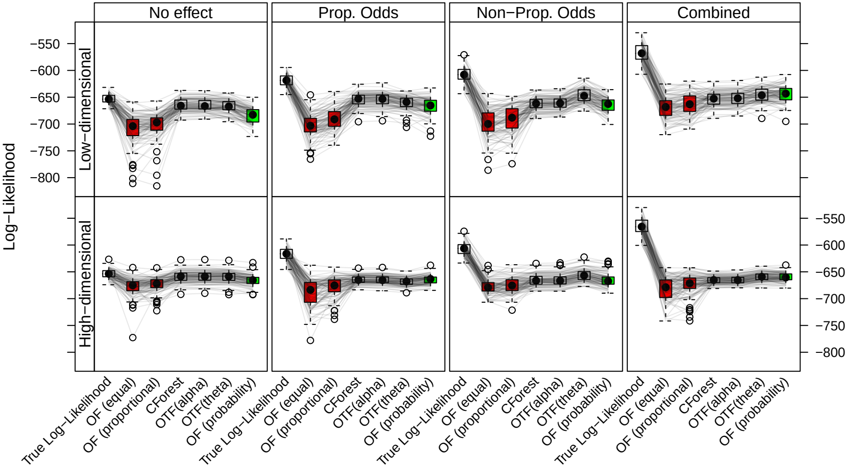
Simulation - results II

- **partition** $[0, b_1],]b_1, b_2], \dots]b_{J-1}, 1]$ of $[0, 1]$ corresponding to the optimized scores $s_1, \dots, s_J$ **NO meaningful indicator of actual “class widths”**

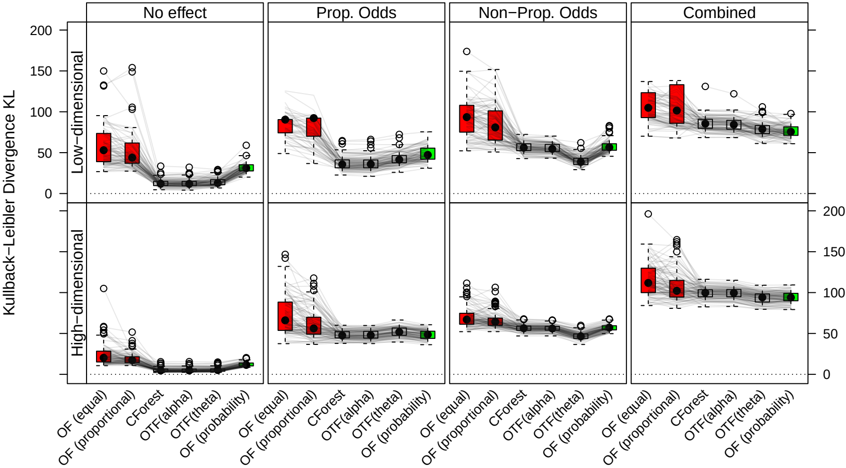
Performance regarding class probability estimation: Predicted out-of-sample log-likelihood minus true log-likelihood



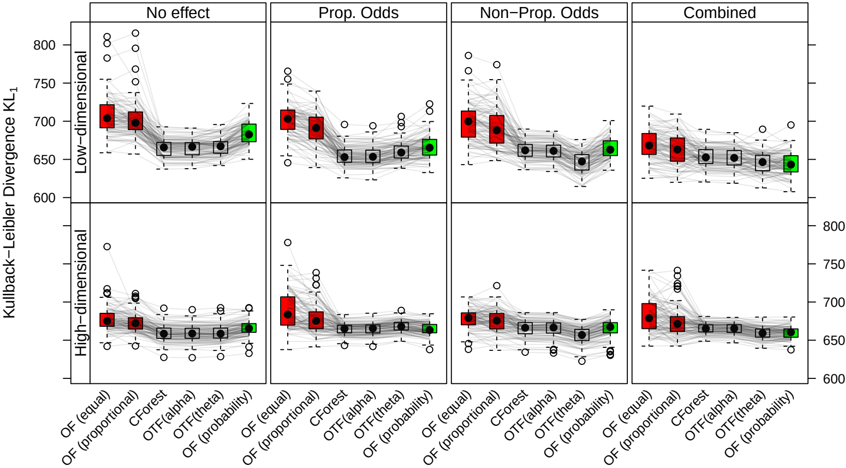
Performance regarding class probability estimation: Predicted out-of-sample log-likelihood



Performance regarding class probability estimation: Kullback-Leibler divergence - $\mathbb{P}(Y = y|\boldsymbol{X} = x)$ vs. $\widehat{\mathbb{P}}(Y = y|\boldsymbol{X} = x)$



Performance regarding class probability estimation: Kullback-Leibler divergence - Y vs. $\widehat{\mathbb{P}}(Y = y|\boldsymbol{X} = x)$



Performance regarding point prediction: Kullback-Leibler divergence - $\widehat{Y}$ vs. $\mathbb{P}(Y = y|\boldsymbol{X} = x)$