technische universität
dortmund

fakultät
statistik

Tree Ensemble Methods for Ordinal Prediction

Dissertation zur Erlangung des Doktorgrades Dr. rer. nat. der Fakultät Statistik der Technischen Universität Dortmund

Vorgelegt von

Philip Dennis Buczak

geboren in Mettmann

Dortmund, Januar 2025

Amtierender Dekan:
Prof. Dr. Philipp Doebler

Gutachter:
Prof. Dr. Markus Pauly (Technische Universität Dortmund)
Prof. Dr. Philipp Doebler (Technische Universität Dortmund)

Tag der Prüfung:
3. April 2025

Abstract

Research questions and applications in the social and life sciences often involve ordinal response data. Student performance is assessed through ordinal grades, patients may express the perceived severity of their symptoms in ordinal levels and respondents of questionnaires may voice their political views through rating given statements. As such, the prediction of ordinal responses is relevant for many fields and can help, e.g., identifying which students may benefit from educational support systems. Traditionally, ordinal responses have been modeled through parametric models such as the proportional odds model. In light of the increasing quantities of data in these fields as well as the continued proliferation of machine learning (ML) methods, recent years saw the establishment of a new methodological stream of ordinal prediction methods based on ML. These methods promise high predictive performance for settings in which traditional parametric models may face difficulties (e.g., highly non-linear effects, high-dimensional data). However, many of these ML methods were originally not specifically tailored towards ordinal responses. Therefore, several extensions and adaptations of ML methods (particularly for tree-based methods) have been proposed to take ordinality into account. A particularly promising approach based on Random Forest (RF) is Ordinal Forest (OF; Hornung, 2019) which assigns numeric scores to the ordinal response categories and uses the scores to train a regression RF. To determine suitable score choices, OF performs a prior optimization step in which scores are optimized w.r.t. their predictive performance.

This cumulative thesis based on three articles contributes to the growing literature on ordinal prediction with RF as follows. The first article proposes the Ordinal Score Optimization Algorithm (OSOA) which is inspired by OF, but modifies its optimization procedure through a non-linear optimization algorithm. Additionally, the first article provides an encompassing comparison of RF-based ordinal prediction methods and a proportional odds model based on simulation and real data previously lacking from the literature. The second article proposes Frequency-Adjusted Borders Ordinal Forest (fabOF), a novel RF-based ordinal prediction method which improves upon OF and OSOA regarding predictive performance and computational runtime. This is achieved through a new prediction approach and a custom heuristic that avoids expensive score

optimization. The third article introduces Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF) which adapts fabOF for prediction with hierarchical data when observations are nested in clusters (e.g., students nested in classes). Both fabOF and mixfabOF are implemented in an accompanying `R` package. This thesis further contributes to the literature by extending and generalizing fabOF and mixfabOF to prediction of (hierarchical) ordinal data with arbitrary regression-based ML methods. The two resulting frameworks are showcased and achieve promising results, indicating their powerful potential for developing new ordinal prediction methods in the future.

Zusammenfassung

Forschungsfragen und Anwendungen in den Sozial- und Lebenswissenschaften führen oft zu ordinalen Daten. Beispielsweise wird die Leistung von Schüler*innen in ordinalen Noten bewertet, Patient*innen äußern die selbstempfundene Stärke ihrer Symptome in ordinalen Stufen und Befragte drücken ihre politischen Ansichten durch das Bewerten verschiedener Aussagen aus. Somit ist die Prädiktion ordinaler Daten von großer Relevanz für diese Felder und kann z.B. dabei helfen zu identifizieren, welche Schüler*innen von zusätzlichen Lehrangeboten profitieren könnten. Traditionell wurden ordinale Daten durch parametrische Modelle wie das Proportional Odds Model modelliert. Durch die zunehmend wachsenden Datenmengen und die fortschreitende Verbreitung von Machine Learning (ML) Methoden entwickelte sich in den letzten Jahren ein neuer Forschungsstrang von ML-basierten ordinalen Prädiktionsmethoden. Diese Methoden versprechen hohe prädiktive Performanz selbst in Szenarien, die traditionelle parametrische Modelle vor Herausforderungen stellen (z.B. stark nicht-linear Effekte, hochdimensionale Daten). Viele dieser ML-Methoden wurden ursprünglich jedoch nicht explizit für ordinale Daten entwickelt. Daher wurden mittlerweile zahlreiche Erweiterungen und Adaptionen von ML-Methoden (inbesondere für baumbasierte Methoden) entwickelt, um Ordinalität berücksichtigen zu können. Ein besonders vielversprechender Ansatz, der auf Random Forest (RF) fußt, ist dabei Ordinal Forest (OF; Hornung, 2019), in welchem den ordinalen Kategorien numerische Scores zugeordnet werden, welche zum Trainieren eines regressions-basierten RF verwendet werden. Um passende Scores zu finden, führt OF zunächst einen Optimierungsschritt durch, in welchem bezüglich der prädiktiven Performanz optimale Scores ermittelt werden.

Die vorliegende kumulative Dissertation, welche auf drei Artikeln basiert, trägt zu der wachsenden Literatur zu ordinaler Prädiktion mit RF wie folgt bei. Im ersten Artikel wird der Ordinal Score Optimization Algorithm (OSOA) vorgestellt, welcher von OF inspiriert wurde, jedoch dessen Optimierungsprozedur durch einen nicht-linearen Optimierungsalgorithmus modifiziert. Zusätzlich steuert der erste Artikel eine umfassende, bis dahin in der Literatur fehlende Vergleichsstudie von RF-basierten Prädiktionsmethoden und einem Proportional Odds Model auf Grundlage von simulierten und realen Daten bei. Der zweite Artikel entwickelt Frequency-Adjusted Borders Ordinal Forest

(fabOF), eine neue RF-basierte ordinale Prädiktionsmethode, welche eine Verbesserung gegenüber OF und OSOA hinsichtlich der Performanz und computationalen Laufzeit erzielen kann. Dies wird erreicht durch die Verwendung einer neuen Prädiktionsstrategie und einer Heuristik, welche die aufwändige Optimierung ersetzt. Im dritten Artikel wird Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF) vorgestellt, welcher fabOF für die Prädiktion von hierarchischen Daten, in denen Beobachtungen in Clustern gruppiert sind (z.B. Schüler*innen innerhalb von Klassen), erweitert. Sowohl fabOF als auch mixfabOF sind in einem begleitendem `R`-Paket implementiert. Diese Dissertation trägt zusätzlich zur Literatur bei, indem sie fabOF und mixfabOF für die Prädiktion von (hierarchischen) ordinalen Daten mit beliebigen regressions-basierten ML-Methoden erweitert und generalisiert. Die zwei daraus resultierenden Frameworks werden anhand von Prototypen illustriert, welche eine vielversprechende Performanz erzielen und somit das hohe Potenzial für die zukünftige Entwicklung neuer ordinaler Prädiktionsmethoden andeuten.

Acknowledgments

First of all, I would like to thank my PhD supervisor Markus Pauly for always granting me the freedom to pursue ideas that excited me and for his patience when things did not always work out. Thank you for your unyielding optimism and all the trust you have placed in me. I would also like to thank my second reviewer Philipp Doebler whom I have known and worked with since my Bachelor's studies, and who has taught me many things and provided me with many opportunities along the way. Further, I would like to kindly thank Carsten Jentsch and Andreas Groll for their time and support in forming my PhD committee.

Thank you to Daniel Horn for working with me on one of the articles in this thesis, and for everything you have taught me during my Bachelor's and Master's studies. I am grateful to the Research Center Trustworthy Data Science and Security for funding my position during the time I was working on this thesis. I am also thankful for the possibility of performing all my computational experiments on the Linux HPC cluster at TU Dortmund University (LiDO3), partially funded by the Large-Scale Equipment Initiative by the German Research Foundation (DFG) as project 27151235.

Thank you to all my friends and colleagues at the Department of Statistics. I am very grateful for the time we have spent together. Particularly, I would like to thank Susanne Brunner as well as Ina Dormuth and Lena Schmid for their wonderful friendship and the many memories I hold dear.

I am indebted to my parents Alina and Leszek Buczak and my brother Patrick Buczak for their continued and unconditional support throughout my life. Thank you to Danijela and Max Markota as well as my remaining family and friends.

Most of all, I would like to express my deepest gratitude to Marie Beisemann without whom this thesis would not have been possible. Thank you from all my heart. Not only for your endless support, but also for helping me find joy – in science as in life.

Contents

List of Publications

This cumulative thesis is based on the following three manuscripts:

Article 1: **Buczak, P.**, Horn, D., & Pauly, M. (2024). Old but gold or new and shiny? Comparing tree ensembles for ordinal prediction with a classic parametric approach. *Journal of Classification (Advance Online Publication)*, 1–27. https://doi.org/10.1007/s00357-024-09497-9

Contribution of the thesis author: The author of this thesis developed the proposed methodology with input from Dr. Horn and Prof. Dr. Pauly. He designed and carried out the simulation studies and real data comparison. He wrote the initial draft of the manuscript and finalized it based on comments from Prof. Dr. Pauly and Dr. Horn.

Article 2: **Buczak, P.** (2025). Frequency-adjusted borders ordinal forest: A novel tree ensemble method for ordinal prediction. *British Journal of Mathematical and Statistical Psychology*, 78(2), 594–616. https://doi.org/10.1111/bmsp.12375

Article 3: **Buczak, P.** (2024). Mixed-effects frequency-adjusted borders ordinal forest: A tree ensemble method for ordinal prediction with hierarchical data. *OSF preprint, version 1.1*, 1–36. https://doi.org/10.31219/osf.io/ny6we (*Currently under review with Multivariate Behavioral Research at the time of publishing this thesis*).

In the course of these projects, the author of this thesis developed the following `R` package:

(1) `fabOF` *Frequency-Adjusted Borders Ordinal Forest for Ordinal Prediction* https://github.com/phibuc/fabOF

Further publications:

- **Buczak, P.**, Chen, J.-J., & Pauly, M. (2023). Analyzing the effect of imputation on classification performance under MCAR and MAR missing mechanisms. *Entropy*, 25(3), 521. https://doi.org/10.3390/e25030521
- **Buczak, P.***, Huang, H.*, Forthmann, B., & Doebler, P. (2023). The machines take over: A comparison of various supervised learning approaches for automated scoring of divergent thinking tasks. *Journal of Creative Behavior*, 57(1), 17–36. https://doi.org/10.1002/jocb.559 (* shared first authorship)
- **Buczak, P.**, Groll, A., Pauly, M., Rehof, J., & Horn, D. (2024). Using sequential statistical tests for efficient hyperparameter tuning. *AStA Advances in Statistical Analysis*, 108, 441–460. https://doi.org/10.1007/s10182-024-00495-1 (based on Master's thesis by Buczak, P.)

Notation

The following list includes the most essential notation used throughout this thesis. In places where additional notation is needed it will be explained in the corresponding context. Vectors and matrices are printed in bold.

n	Number of observations
p	Number of predictors
k	Number of ordinal categories
m	Number of clusters (for hierarchical data)
$c_1, \ldots, c_k$	Ordinal response categories
$s_1, \ldots, s_k$	Numeric scores assigned to ordinal categories
$b_1, \ldots, b_{k+1}$	Numeric category interval borders
i	Person/observation index
r	Response category index
j	Cluster index (for hierarchical data)
Y	Generic outcome variable
Y^{ord}	Ordinal outcome variable
Y^{num}	Numeric outcome variable
Y^{class}	Nominal outcome variable
$X_1, \ldots, X_p$	Predictor/covariate variables
y_i	Observed (generic) outcome of person i
y_{ij}	Observed (generic) outcome of person i in cluster j
$\boldsymbol{y}_j$	Observed (generic) outcomes of all persons in cluster j
$\boldsymbol{x}_i$	Observed predictor values of person i

$\boldsymbol{x}_{ij}$	Observed predictor values of person i in cluster j
$\boldsymbol{X}_j$	Matrix of observed predictor values of all persons in cluster j
$\mathbb{1}_A$	Indicator function

Part I

Summary of Thesis Work

1 Introduction

Ordinal response data are commonly encountered in the social and life sciences. For example, students are awarded ordinal grades in school, survey respondents frequently express their degree of agreement towards statements on ordinal rating scales, and patients may self-report their mental or physical health status using ordinal categories (e.g., "poor", "fair", "good"). In contrast to nominal response categories, ordinal categories carry a natural order, e.g, poor $<$ fair $<$ good. However, the distances between the categories are generally not meaningfully interpretable and categories are not necessarily equidistant from another (Tutz, 2011). In principle, it is possible to use methods for nominal response data for ordinal outcomes just as it is possible to assign numeric scores to the ordinal categories such that they can be used with methods for metric outcomes. Yet, both approaches fail to completely capture the essence of the ordinal outcome and thus, developing and using methods specifically tailored for ordinal data is preferred. Compared to methods for nominal responses, ordinal methods allow for computing measures similar to metric data (e.g., rank correlations; Kendall, 1945) as well as offer more parsimonious models with less parameters that are simpler to interpret than their nominal counterparts (Agresti, 2010; Tutz, 2011). While ordinal response data often have a quantitative background and assigning numeric scores to the ordinal response categories is a viable modeling strategy, caution is advised when using these (often arbitrary) numeric scores, e.g., for fitting ordinary least-squares regression: resulting models can yield predictions below the lowest or above the highest category, and can further lead to misleading results due to floor or ceiling effects caused by the lower and upper limits of the category range (Agresti, 2010). In the context of statistical inference, metric models fitted to ordinal data may negatively impact type I and II errors (Liddell & Kruschke, 2018). For further possible pitfalls resulting from applying non-ordinal methods to ordinal response data, I refer to Agresti (2010) and Liddell and Kruschke (2018).

One of the most widely used model classes of ordinal regression methods is represented by the cumulative model (McCullagh, 1980) which assumes the ordinal response to originate from an underlying latent numeric variable that one can only observe through a set of thresholds. As a special case, the proportional odds model (McCullagh, 1980)

has enjoyed high popularity ever since due to its simplicity and intuitive interpretations (Tutz, 2022). Broadly speaking, the proportional odds model can be thought of as a series of logistic regression models holding at the same time (Tutz, 2022). In the context of ordinal prediction, the advent of machine learning (ML) methods such as Random Forest (RF; Breiman, 2001) has given rise to a parallel methodological stream in more recent years. With data increasing in quantity and complexity in many application fields including educational and behavioral sciences (see, e.g., Hilbert et al., 2021; Ulitzsch et al., 2022), parametric models are increasingly pushed towards their limits (see, e.g., Zahid & Tutz, 2013). ML methods, on the other hand, promise high predictive performance even for high-dimensional data (see, e.g., Fernández-Delgado et al., 2014). However, many classic ML methods, e.g., based on Classification and Regression Trees (CART; Breiman et al., 1984) or RF, were designed for nominal or metric outcomes. This is, for example, reflected in the splitrules used for partitioning the predictor space when training the tree models. The two most common splitrules are based on the Gini impurity for classification and on the sum of squared errors for regression (for both, see Breiman et al., 1984, or Section 2.2.1 for more detail). Neither can take the ordered categorical nature of ordinal responses into account. Consequently, several workarounds and extensions have been proposed in the literature. Contributing appropriate split criteria, Piccarreta (2007) adapted the Gini impurity for ordinal responses, while Archer (2010) and Galimberti et al. (2012) aimed to enforce the ordinal response structure through suitable misclassification costs when using the generalized Gini impurity measure.

Apart from modified splitrules, other approaches of adapting CART and RF to ordinal data have also been introduced. Frank and Hall (2001) proposed transforming the ordinal prediction task into a set of binary predictions tasks for which individual (binary) classification models can be trained. The predictions of the individual models can then be combined for obtaining ordinal predictions. Similarly, Tutz (2021) uses a set of binary RF classifiers to arrive at ordinal predictions following the logic of cumulative models. Following the approach of assigning numeric scores to the ordinal response categories, Kramer et al. (2000) used the numeric scores to train regression trees, whereas Janitza et al. (2016) explored the use of Conditional Inference Forests (Hothorn et al., 2006). A particularly promising approach was proposed by Hornung (2019) through his Ordinal Forest (OF) method. OF trains a regression RF model on the numeric scores, but instead of using arbitrary default scores, OF employs a prior optimization step in which optimal scores for the categories are determined based on the predictive performance achieved with them (Hornung, 2019). As such, OF does not require the specification of numeric scores beforehand. Apart from RF, other ML

methods have been adapted for ordinal prediction as well, e.g., boosting algorithms (Riccardi et al., 2014; Tutz & Hechenbichler, 2005), neural networks (e.g., Cao et al., 2020; Cheng et al., 2008; Shi et al., 2023) and Support Vector Machine (Chu & Keerthi, 2007; Herbrich et al., 1999). However, this thesis focuses mainly on RF as it has been shown to be particularly successful for use with tabular data (Fernández-Delgado et al., 2014; Grinsztajn et al., 2022) as will be considered in this work, all while being relatively robust regarding the choice of its hyperparameters (Probst et al., 2019). Further it is currently among the most actively researched methods for ordinal prediction originating from ML (Hornung, 2019; Janitza et al., 2016; Tutz, 2021).

The present cumulative thesis consists of three articles of which each proposes a RF-based ordinal prediction method extending upon the current methodological landscape of ordinal prediction. In the first article (Buczak et al., 2024), the Ordinal Score Optimization Algorithm (OSOA) is proposed which is modeled after OF (Hornung, 2019), but replaces its optimization procedure through a non-linear optimization algorithm. This modification was aimed at addressing the lacking ability of OF to focus on promising candidate scores during the optimization process. The first article further contributes to the literature by filling the gap for an encompassing comparison study of the two methodological streams for ordinal prediction, namely parametric models and RF-based prediction methods. To this end, the performance of a proportional odds model and several RF-based prediction methods is assessed in diverse data settings to obtain practical guidelines. The second article of this thesis (Buczak, 2025) proposes Frequency-Adjusted Borders Ordinal Forest (fabOF) which is a novel RF-based ordinal prediction method using a custom heuristic. Through its heuristic, fabOF avoids the expensive optimization procedure of OF as well as allows for more flexible modeling of the categories. In the third article (Buczak, 2024), fabOF is extended to use with hierarchical data through Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF). This extension was aimed at filling the long-existing gap for RF-based ordinal prediction methods for hierarchical data in parallel with and independently of Ordinal Mixed-Effects Random Forest (OMERF; Bergonzoli et al., 2024). Through evaluation via simulation and real data, mixfabOF is shown to yield performance benefits in the presence of medium and strong random effect variability while also outperforming its competitor OMERF. To enable widespread application and to foster future research, I have additionally implemented fabOF and mixfabOF within the `R` package `fabOF` which I made publicly available on GitHub (https://github.com/phibuc/fabOF). Despite the articles' focus on RF, the present thesis will also demonstrate that the presented methodology can well be extended to other ML algorithms. To this end, I contribute methodological extensions based on Extremely Randomized Trees (Geurts et al., 2006),

Conditional Inference Forest (Hothorn et al., 2006) and XGBoost (Chen & Guestrin, 2016). As an additional contribution, I generalize fabOF and mixfabOF to introduce two powerful frameworks that allow for extending any regression-based ML method to prediction of (hierarchical) ordinal response data. Finally, I will provide a sketch for how fabOF can also be extended to multivariate ordinal prediction, i.e, when the outcome is vector-valued and it is of interest to take possible dependencies between the outcome components into account.

The present thesis is structured as follows. In the subsequent Chapter 2, I will introduce the statistical concepts and methods that this thesis and its articles are based on as well as distill essential parts of the current literature on ordinal prediction with RF. In Chapter 3, I will summarize the three main articles of this thesis including their motivation, contributions and the insights learned from them. In Chapter 4, I will provide a brief overview of the `fabOF` `R` package and showcase its main functionalities. In Chapter 5, I will introduce the above-mentioned extensions and generalizations of fabOF and mixfabOF. Finally, I will discuss the contributions and limitations of the present thesis and its articles as well as provide avenues for further research in Chapter 6.

2 Statistical Methods and Background

This section introduces the statistical concepts used in this thesis and the corresponding articles, and provides an essential overview of the relevant methodological landscape. While the main focus of this thesis is the prediction of ordinal response variables, I will frequently also refer to frameworks and concepts from prediction in classification and regression contexts. On an abstract level and without further restricting the response type for now, the goal of statistical and machine learning is to infer a functional relationship $g : \mathcal{X} \rightarrow \mathcal{Y}$ between a set of p predictors (or covariates) $\boldsymbol{X} = (X_1, \ldots, X_p)^\top \in \mathcal{X}$ and an outcome (or target variable) $Y \in \mathcal{Y}$. For now, it will be assumed that Y is univariate. As the true function g is unknown, one can only determine an approximation $\hat{g}$. In the context of supervised learning, $\hat{g}$ is estimated based on an annotated dataset $\boldsymbol{W} \in \mathcal{W} = \{\mathcal{X} \times \mathcal{Y}\}$ containing n observation pairs of the form $(\boldsymbol{x}_i, y_i)_{i=1,\ldots,n}$, where $\boldsymbol{x}_i$ contains the predictor values and y_i the ground truth of observation $i,\ i = 1, \ldots, n$. While Y refers to a general target variable of arbitrary scale, the scale of the outcome will be of relevance in many instances in the following. To specify the respective scale more clearly, $Y^{\text{class}} \in \{c_1, \ldots, c_k\}$ will denote a nominal outcome with k categories $\{c_1, \ldots, c_k\}$, $Y^{\text{ord}} \in \{c_1, \ldots, c_k\}$ with $c_1 < c_2 < \cdots < c_k$ will denote an ordinal outcome where the k categories are ordered, and $Y^{\text{num}} \in \mathbb{R}$ will denote a numeric outcome.

The methodological overview begins with cumulative ordinal regression models which have long been used for ordinal response data. Subsequently, I will introduce ML approaches based on Classification and Regression Trees (CART; Breiman et al., 1984) and Random Forest (RF; Breiman, 2001) as another major methodological stream for ordinal prediction. Because the third article of this thesis deals with RF-based ordinal prediction in hierarchical data settings, this section will close with an introduction to extensions of tree-based methods for hierarchical data.

2.1 Cumulative Ordinal Regression Models

Traditionally, ordinal response data have been modeled through parametric ordinal regression models. Because the ordinal nature of the response can be conceptualized

in different ways, several modeling approaches have emerged in the literature. Classic ordinal regression model types include cumulative models, sequential models and adjacent-categories models (Tutz, 2011). Cumulative models assume the ordinal response to originate from an underlying latent numeric outcome which can only be observed through certain thresholds. Sequential models, on the other hand, assume that one traverses the ordinal categories in a successive manner. For example, for reaching the second category, one must have first passed through the first category. Adjacent-category models model a binary decision between two adjacent categories (Tutz, 2011).

Since cumulative models are the most widely used type of models (Tutz, 2011) and because many of the RF-based approaches introduced later are also motivated by the assumption of an underlying numeric variable, I will mainly focus on cumulative models in the following.

Cumulative models as first introduced by McCullagh (1980) model the probability that the ordinal outcome variable Y_i^{ord} takes at most category $c_r, r = 1, \ldots, k-1$, for person $i = 1, \ldots, n$, given covariate vector $\boldsymbol{x}_i$ through

$$P(Y_i^{\text{ord}} \leq c_r | \boldsymbol{x}_i) = F(\eta_{ir}) = F\left(\gamma_r + \boldsymbol{x}_i^\top \boldsymbol{\beta}\right), \tag{2.1}$$

with $-\infty < \gamma_1 < \cdots < \gamma_{k-1} < \infty$ denoting the thresholds, F denoting a strictly increasing distribution function and η_{ir} denoting the linear predictor of person i for the r-th category (Tutz, 2011). A frequent choice for the distribution function F is the logistic distribution function, yielding the proportional odds model

$$P(Y_i^{\text{ord}} \leq c_r | \boldsymbol{x}_i) = \frac{\exp(\eta_{ir})}{1 + \exp(\eta_{ir})} = \frac{\exp(\gamma_r + \boldsymbol{x}_i^\top \boldsymbol{\beta})}{1 + \exp(\gamma_r + \boldsymbol{x}_i^\top \boldsymbol{\beta})}, \; i = 1, \ldots, n; r = 1, \ldots, k-1$$

(Tutz, 2011). The parameters of the model can be estimated via maximum likelihood estimation. In `R`, proportional odds models (and other cumulative models) can be fitted e.g., using the `ordinal` package (Christensen, 2022). Proportional odds models are popular as they are relatively simple and offer intuitive interpretations. Their name derives from the model's property that comparing two populations regarding their cumulative odds $P(Y_i \leq c_r | \boldsymbol{x}_i)/P(Y_i > c_r | \boldsymbol{x}_i)$ does not depend on the categories, i.e., population effects are the same across all categories (Tutz, 2011). However, in many scenarios the proportional odds model may be too simple to sufficiently model the data at hand. Therefore, models for more complex settings have been proposed. For example, McCullagh (1980) introduced the location-scale model which additionally accounts for variance heterogeneity by including a scaling component. Tutz and Berger (2017) account for heterogeneity in their location-shift model by letting the thresholds of the

cumulative model vary across individuals. For settings in which it is likely that effects vary across categories, partial proportional or non-proportional odds models (see, e.g., Brant, 1990; Peterson & Harrell, 1990) may be a better fit. For a more detailed taxonomy of ordinal regression models, I refer to Tutz (2022).

2.2 Classification/Regression Trees and Random Forest

Apart from classic parametric models, tree-based ML methods are increasingly used for ordinal prediction as well. As noted above (see Section 1), methods such as CART (Breiman et al., 1984) and RF (Breiman, 2001) were originally not designed for ordinal responses, but nonetheless constitute pivotal model classes on which several ordinal prediction methods (including the methods proposed in this thesis) are based. Therefore, I will briefly introduce CART and RF in the following.

2.2.1 Classification and Regression Trees

CART (Breiman et al., 1984) are a class of non-parametric decision tree models for classification and regression tasks. The description and notation in the following are largely inspired by Hastie et al. (2009). CART aim to partition the predictor space into disjunct regions $R_1, \ldots, R_T$ in which the outcome is modeled through a constant value h_t, i.e.,

$$g_{\text{CART}}(\boldsymbol{x}_i) = \sum_{t=1}^{T} h_t \mathbb{1}_{\boldsymbol{x}_i \in R_t}$$

(Hastie et al., 2009). Starting with the entire predictor space at its root, tree models are grown by recursively splitting a given region of the predictor space along predictor X_v and split value $w \in \mathbb{R}$ into two sub-regions $R_{\text{left}}(v, w) = \{\boldsymbol{X} | X_v \leq w\}$ and $R_{\text{right}}(v, w) = \{\boldsymbol{X} | X_v > w\}$ assuming X_v is a numeric predictor. For categorical predictors, the two resulting regions are of the form $R_{\text{left}}(v, \tilde{w}) = \{\boldsymbol{X} | X_v = \tilde{w}\}$ and $R_{\text{right}}(v, \tilde{w}) = \{\boldsymbol{X} | X_v \neq \tilde{w}\}$ where $\tilde{w}$ is a categorical label (Hastie et al., 2009). Due to the resulting binary tree structure, the sub-regions of the predictor space are commonly called nodes. CART greedily determine their splits based on the training dataset by cycling through all available predictors and predictor values. For classification trees, the optimal split is based on the Gini impurity Q_t which is given by

$$Q_t = \sum_{r=1}^{k} \hat{p}_{tr} \left(1 - \hat{p}_{tr}\right),$$

where $\hat{p}_{tr}$ denotes the relative frequency of category c_r in node t with $r = 1, \ldots, k$ and $t = 1, \ldots, T$ (Hastie et al., 2009). The Gini impurity is minimal for pure nodes, i.e., when all observations belong to the same class. The optimal split variable X_{v^*} and split value w^* are determined such that the combined Gini impurity of the resulting child nodes is minimal, i.e., solving the optimization problem

$$(v^*, w^*) = \arg\min_{v,w} \left\{ \frac{n_{\text{left}}}{n_{\text{parent}}} Q_{\text{left}} + \frac{n_{\text{right}}}{n_{\text{parent}}} Q_{\text{right}} \right\},$$

where $n_{\text{left}}, n_{\text{right}}$ and n_{parent} denote the number of observations in the left child node, right child node and parent node, respectively (Hastie et al., 2009). Analogously, Q_{left} and Q_{right} denote the impurity values of the left and right child nodes as induced by split variable X_v with split value w. For regression trees, on the other hand, splits are determined such that the combined sum of squares in the resulting child nodes is minimal. The optimal choices v^* and w^*, thus, are given by

$$(v^*, w^*) = \arg\min_{v,w} \left\{ \sum_{i:\boldsymbol{x}_i \in R_{\text{left}}(v,w)} \left(y_i^{\text{num}} - \bar{y}_{R_{\text{left}}(v,w)}\right)^2 + \sum_{i:\boldsymbol{x}_i \in R_{\text{right}}(v,w)} \left(y_i^{\text{num}} - \bar{y}_{R_{\text{right}}(v,w)}\right)^2 \right\},$$

where $\bar{y}_{R_{\text{left}}(v,w)}$ and $\bar{y}_{R_{\text{right}}(v,w)}$ are the mean outcome values of all observations falling into the left and right child node, respectively (Hastie et al., 2009). The splitting procedure continues until a stopping criterion is fulfilled. Common criteria include, e.g., a minimum number of observations in a given node, a maximum tree depth or (for classification) the purity of a given node (Breiman et al., 1984). In the terminal nodes, the outcome is modeled based on the outcome values of the observations falling into the node. For classification trees, the majority class is selected, i.e, $h_t = \text{mode}(y_i^{\text{class}})_{i:\boldsymbol{x}_i \in R_t}$, while for regression, the mean outcome is used, i.e., $h_t = n_{R_t}^{-1} \sum_{i:\boldsymbol{x}_i \in R_t} y_i^{\text{num}}$ where n_{R_t} is the number of observations in $R_t, t = 1, \ldots, T$ (Hastie et al., 2009). While offering high interpretability, CART models are prone to overfitting and suffer from high variability (James et al., 2021). One possibility to approach the risk of overfitting is pruning the tree based on some cost-complexity criterion (see, e.g., Breiman et al., 1984). An implementation of CART is, e.g., available in the `rpart` package (Therneau & Atkinson, 2023). CART only constitutes one, albeit a particularly popular class of decision tree models used in supervised ML. Alternative algorithms include C4.5 (Quinlan, 1993), GUIDE (Loh, 2002), and Conditional Inference Trees (CTs; Hothorn et al., 2006). The latter, for example, differ from CART in their split selection. While CART consider all predictor variables and their realized values as potential split candidates, CTs first performs an association test between the outcome variable and the predictors at each split.

Only the predictor indicating the strongest association with the outcome is further considered for determining the optimal split point. This alternative splitting behavior aims to address CART's bias towards predictors with many possible split points (Hothorn et al., 2006).

2.2.2 Random Forest

RF (RF; Breiman, 2001) is an ensemble ML method aggregating a set of L individual tree models into a combined model. The subsequent description of RF and its notation mostly follow Hastie et al. (2009). RFs aim to amend the overfitting and variability issues of single CART models in two ways. First, the individual tree models are trained on bootstrap samples of the data. The tree models are aggregated such that for classification, the predicted class is determined via majority voting over all tree models, i.e,

$$\hat{g}_{\text{RF}}^{\text{class}}(\boldsymbol{x}_i) = \text{mode}(\hat{y}_{i,\ell}^{\text{class}})_{\ell=1,\ldots,L},$$

where $y_{i,\ell}^{\text{class}}$ is the class label predicted by the ℓ-th tree model for person $i = 1, \ldots, n$ (Hastie et al., 2009). For regression, the mean of all tree predictions is computed, i.e.,

$$\hat{g}_{\text{RF}}^{\text{num}}(\boldsymbol{x}_i) = \frac{1}{L} \sum_{\ell=1}^{L} \hat{y}_{i,\ell}^{\text{num}},$$

where $\hat{y}_{i,\ell}^{\text{num}}$ is the (numeric) outcome value predicted by the ℓ-th tree model for person $i = 1, \ldots, n$ (Hastie et al., 2009). Second, instead of taking all available predictors as split candidates into account, only a random subset of size $\texttt{mtry} \leq p$ is considered for every split. In this way, the individual tree models are de-correlated which can be shown to reduce the overall variability of RF (Hastie et al., 2009). Commonly used implementations in `R` include the `randomForest` package (Liaw & Wiener, 2002) and the `ranger` package (Wright & Ziegler, 2017). RFs enjoy popularity due to their high performance (see, e.g., Grinsztajn et al., 2022) while being relatively robust regarding their hyperparameter choices (Probst et al., 2019). However, due to aggregating over a large set of tree models, RFs lose the interpretability individual tree models provide. As a remedy, interpretable ML tools are often used to gain additional insight into the prediction behavior of RF models (see, e.g., Molnar, 2022, for an introduction to interpretable ML). A commonly used approach of studying the impact of individual predictors is computing variable importance measures (Breiman, 2001, VIMs,). VIMs quantify the importance of the individual predictors on the predictive performance of the model (Molnar, 2022). Permutation VIMs are a particular class of VIMs in which

the importance of a predictor is assessed by comparing the original model performance with the performance achieved when the values of the respective predictor are permuted (Breiman, 2001). Since the permutation voids the original relation between the predictor and the target variable, a notable performance decrease after permutation indicates that the predictor is important for prediction. On the other hand, little to no performance changes indicate that the predictor is less important (Molnar, 2022). In the presence of highly correlated predictors, however, VIMs can be biased (see, e.g., Strobl et al., 2008). To this end, Strobl et al. (2008) proposed conditional permutation VIMs which restrict the permutation procedure to better preserve the original correlation structure in the data. For more details, I refer to Strobl et al. (2008) and Debeer and Strobl (2020).

2.3 Ordinal Prediction with Random Forest

As indicated in Section 1, several adaptions and workarounds have been proposed to use CART and RF for ordinal prediction. Out of these, two methods are particularly relevant for this thesis and its articles: Ordinal Forest (OF; Hornung, 2019) and Split-based Ordinal Forest (RFSp; Tutz, 2021). OF lays the foundational framework on which the methods proposed in the three articles of this thesis stand. RFSp, on the other hand, follows a different approach, but its approach of mimicking the cumulative model through RF represents another key contribution to RF-based ordinal prediction. Both methods will be described in the following.

2.3.1 Ordinal Forest

OF (Hornung, 2019) is a RF-based method for ordinal prediction based on assigning numeric scores to the ordinal response categories. Similar to cumulative models, OF assumes the ordinal response to be a coarsened version of an underlying numeric variable. To approximate the numeric variable, OF partitions the $[0, 1]$ interval such that each ordinal category is assigned a numeric sub-interval and a representative numeric score. For category r with numeric category borders b_r and b_{r+1}, the midpoint of the respective interval $(b_r, b_{r+1}]$ is used as the numeric score s_r representing the category, i.e., $s_r = \frac{b_r+b_{r+1}}{2}, r = 1, \ldots, k$. After transforming the numeric scores and category borders using the quantile function Φ^{-1} of the standard normal distribution, the scores are used as a proxy target variable to train a regression RF. For new observations, the RF model outputs numeric predictions which can be translated back into ordinal category predictions by checking into which category interval a given numeric prediction falls.

More specifically, the numeric prediction $\hat{y}_{i,\ell}^{\text{num}}$ of the ℓ-th tree model for observation i is converted to $\hat{y}_{i,\ell}^{\text{ord}} = c_r$ if $\hat{y}_{i,\ell}^{\text{num}} \in (\Phi^{-1}(b_r), \Phi^{-1}(b_{r+1})], r = 1, \ldots, k$ (Hornung, 2019).

To determine suitable choices for the category borders, OF performs a prior optimization step. The optimization procedure randomly generates a set of different partitions of the $[0, 1]$ interval (1000 per default). Each partition is evaluated by using it to fit a regression RF and determining the model's out-of-bag (OOB) performance, i.e., for each observation only the tree models for which the given observation was not part of the training data are used for computing the performance, respectively. The predictive performance is assessed through Youden's Index J (Youden, 1950) which leverages sensitivity and specificity values. To this end, OF computes a category-specific variant J_{OF} where

$$\begin{aligned} J_{\text{OF}} &= \frac{1}{k} \sum_{r=1}^{k} \left[\text{sensitivity}_r + \text{specificity}_r - 1\right] \\ &= \frac{1}{k} \sum_{r=1}^{k} \left[\frac{\#\{i \in \{1, \ldots, n\} : y_i = c_r \wedge \hat{y}_i = c_r\}}{\#\{i \in \{1, \ldots, n\} : y_i = c_r\}} \right. \\ &\qquad \left. + \frac{\#\{i \in \{1, \ldots, n\} : y_i \neq c_r \wedge \hat{y}_i \neq c_r\}}{\#\{i \in \{1, \ldots, n\} : y_i \neq c_r\}} - 1 \right] \end{aligned}$$

(Hornung, 2019). Note that J_{OF} does not take ordinality into account as it only considers the specificity and sensitivity of the classification for each category. As such, the outcomes $y_i, i = 1, \ldots, n$, may be either nominal or ordinal, but for the latter the ordered nature of the response is not reflected. The n_{best} best performing partitions are averaged into a final partition with

$$b_r = \frac{1}{n_{\text{best}}} \sum_{a=1}^{n_{\text{best}}} b_{r,a},$$

where $b_{r,a}$ is the r-th border of the a-th partition, $r = 1, \ldots, k + 1$ and $a = 1, \ldots, n_{\text{best}}$ (Hornung, 2019). Based on the final partition, a final regression RF model is trained and returned, along with the final category borders for predicting new observations. OF is implemented in the `ordinalForest` package (Hornung, 2022). For fitting RF models, it relies on the RF implementation from the `ranger` package (Wright & Ziegler, 2017).

2.3.2 Split-based Ordinal Forest

RFSp (Tutz, 2021) is a RF-based method for ordinal prediction following the logic of cumulative models. In contrast to OF, RFSp does not rely on assigning numeric scores to the ordinal response categories. Instead, RFSp transforms the ordinal prediction task into a series of binary prediction tasks. For $r = 2, \ldots, k$, the binary prediction tasks consist of classifying observations as belonging to categories $c_1, \ldots, c_{r-1}$ or to categories $c_r, \ldots, c_k$, respectively (Tutz, 2021). For each task, a classification RF is trained. For new observations, predicted response category probabilities are determined by combining the cumulative probabilities obtained from the individual RF models. For category c_{r-1}, RFSp computes the predicted probability $\hat{P}(Y_i^{\text{ord}} = c_{r-1}|\boldsymbol{x}_i)$ as

$$\hat{P}(Y_i^{\text{ord}} = c_{r-1}|\boldsymbol{x}_i) = \hat{P}(Y_i^{\text{ord}} \geq c_{r-1}|\boldsymbol{x}_i) - \hat{P}(Y_i^{\text{ord}} \geq c_r|\boldsymbol{x}_i) \tag{2.2}$$

with $i = 1, \ldots, n$ and $r = 2, \ldots, k$ (Tutz, 2021). To this end, the condition $\hat{P}(Y_i^{\text{ord}} \geq c_{r-1}|\boldsymbol{x}_i) \geq \hat{P}(Y_i^{\text{ord}} \geq c_r|\boldsymbol{x}_i)$ must hold for all $r = 2, \ldots, k$. If this condition does not hold for a given situation, it is enforced through monotone regression tools (Tutz, 2021). The individual probabilities in Equation 2.2 can be obtained from the internally fitted RF models, i.e., $\hat{P}(Y_i^{\text{ord}} \geq c_r|\boldsymbol{x}_i)$ is obtained from the RF model that classifies observations as belonging to categories $1, \ldots, c_{r-1}$ or $c_r, \ldots, c_k$ with $r = 2, \ldots, k$ (Tutz, 2021). An implementation of RFSp is available from GitHub (https://github.com/GerhardTutz/ScoreFreeTrees). For training the RF models, the implementation relies on the `randomForest` package (Liaw & Wiener, 2002).

2.4 (Ordinal) Prediction for Hierarchical Data

In the social and life sciences, datasets often contain a hierarchical structure. For example, students are commonly nested within classes or in longitudinal study designs, multiple measurements are nested within the same person. Such structures introduce dependencies into the data that should be accounted for (Gelman & Hill, 2006). In the framework of (generalized) linear models, hierarchical structures are accounted for by mixed-effects models that distinguish between global fixed effects and cluster-specific random effects. In the classic linear mixed model (LMM), the normally distributed outcome vector $\boldsymbol{Y}_j \in \mathbb{R}^{n_j}$ of cluster j is modeled as

$$\boldsymbol{Y}_j = \boldsymbol{X}_j\boldsymbol{\beta} + \boldsymbol{Z}_j\boldsymbol{u}_j + \boldsymbol{\varepsilon}_j,\ j = 1, \ldots, m, \tag{2.3}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ denotes the fixed effects and $\boldsymbol{u}_j \in \mathbb{R}^q$ the random effects for cluster j (Verbeke & Molenberghs, 2000). Likewise for cluster j, $\boldsymbol{X}_j \in \mathbb{R}^{n_j \times p}$ denotes the matrix of fixed effect covariate values, $\boldsymbol{Z}_j \in \mathbb{R}^{n_j \times q}$ the matrix of random effect covariate values, and $\boldsymbol{\varepsilon}_j \in \mathbb{R}^{n_j}$ the vector of error terms. The model assumes that $\boldsymbol{u}_j \sim \mathcal{N}(\mathbf{0}, \boldsymbol{D})$ with $\boldsymbol{D} \in \mathbb{R}^{q \times q}$ as well as $\boldsymbol{\varepsilon}_j \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}_j)$. It is additionally often assumed that $\boldsymbol{R}_j = \sigma^2 \boldsymbol{I}_{n_j \times n_j}$. The random effects $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ and error terms $\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_m$ are further assumed to be independent from another (Fahrmeir et al., 2021). The model parameters are estimated via marginal maximum likelihood estimation (for more details, see Verbeke & Molenberghs, 2000), e.g., using the Expectation Maximization (EM) algorithm (Dempster et al., 1977). In `R`, LMMs can be fitted using, e.g., the `lme4` package (Bates et al., 2015).

2.4.1 Cumulative Mixed Model

For ordinal regression models, extensions to hierarchical data have been proposed by Hedeker and Gibbons (1994) and Tutz and Hennevogl (1996). The cumulative model as introduced in Equation 2.1 can be extended to the Cumulative Mixed Model which models the probability that the outcome Y_{ij}^{ord} of observation i in cluster j reaches at most category c_r through

$$P(Y_{ij}^{\text{ord}} \leq c_r | \boldsymbol{x}_{ij}, \boldsymbol{z}_{ij}, \boldsymbol{u}_j) = F(\eta_{ijr}) = F\left(\gamma_r + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta} + \boldsymbol{z}_{ij}^\top \boldsymbol{u}_j\right), \; r = 1, \ldots, k-1, \tag{2.4}$$

where $\boldsymbol{x}_{ij}$ and $\boldsymbol{z}_{ij}$ are the fixed and random effect component covariate values of observation i in cluster j with $i = 1, \ldots, n_j$ and $j = 1, \ldots, m$ (Tutz & Hennevogl, 1996). Similar to the cumulative model, a common choice for F is the logistic distribution function, yielding the Cumulative Logit Mixed Model (CLMM)

$$P(Y_{ij}^{\text{ord}} \leq c_r | \boldsymbol{x}_{ij}, \boldsymbol{z}_{ij}, \boldsymbol{u}_j) = \frac{\exp(\eta_{ijr})}{1 + \exp(\eta_{ijr})} = \frac{\exp\left(\gamma_r + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta} + \boldsymbol{z}_{ij}^\top \boldsymbol{u}_j\right)}{1 + \exp\left(\gamma_r + \boldsymbol{x}_{ij}^\top \boldsymbol{\beta} + \boldsymbol{z}_{ij}^\top \boldsymbol{u}_j\right)}$$

with $i = 1, \ldots, n_j$, $j = 1, \ldots, m$ and $r = 1, \ldots, k-1$ (Agresti, 2010). Specifiable random effects include random simultaneous shifts of thresholds (comparable to random intercepts), random slope effects and random thresholds (Tutz & Hennevogl, 1996). The latter type of random effects refer to varying thresholds in all clusters instead of simultaneously shifting thresholds by a random cluster-specific shift constant. For more details regarding the model estimation based on marginal maximum likelihood estimation, I refer to Tutz and Hennevogl (1996) and Agresti (2010). In `R`, cumulative mixed models can be fitted using, e.g., the `ordinal` package (Christensen, 2022) which of-

fers a selection of different link functions.

2.4.2 Tree-based Prediction for Hierarchical Data

Extensions to hierarchical data have also been proposed for regression trees and regression RF. Most notably, Hajjem et al. (2011) proposed the Mixed Effects Regression Tree (MERT; Hajjem et al., 2011) and Sela and Simonoff (2012) the Random Effects Expectation Maximization (RE-EM) tree. Conceptually, both approaches replace the linear fixed effect component $\boldsymbol{X}_j\boldsymbol{\beta}$ in Equation 2.3 by a non-linear function $f(\boldsymbol{X}_j)$, yielding the modified model

$$\boldsymbol{Y}_j = f(\boldsymbol{X}_j) + \boldsymbol{Z}_j\boldsymbol{u}_j + \boldsymbol{\varepsilon}_j,\ j = 1, \ldots, m \tag{2.5}$$

(Hajjem et al., 2011). Both methods iterate between estimating the fixed and random effect components of the modified model through a type of EM algorithm (Dempster et al., 1977). MERT and RE-EM trees differ in how they specify and estimate the fixed effect component. To estimate $f(\boldsymbol{X}_j)$, MERT trains a regression tree on the modified outcome $\tilde{\boldsymbol{y}}_j$ from which the random effects have been canceled out, i.e.,

$$\tilde{\boldsymbol{y}}_j = \boldsymbol{y}_j - \boldsymbol{Z}_j\boldsymbol{u}_j,\ j = 1, \ldots, m \tag{2.6}$$

(Hajjem et al., 2011). While RE-EM trees also train a regression tree on the modified outcome, they only use the fitted tree model to extract the resulting partition of the predictor space. The fixed effects are then estimated alongside the random effects with a LMM. For the LMM model, the fixed effects structure only consists of a variable indicating to which sub-region of the tree partition a given observation belongs (Sela & Simonoff, 2012). MERT and RE-EM trees have both been extended to RF through Mixed-Effects Random Forest (MERF; Hajjem et al., 2012) and REEMforest (Capitaine et al., 2020), respectively. Similarly, extensions of regression trees and RF for hierarchical data have been proposed for other response types (e.g., Fontana et al., 2021; Hajjem et al., 2017; Pellagatti et al., 2021; Speiser et al., 2018, 2019). Salditt et al. (2023) adapted MERT and RE-EM trees for gradient boosting.

2.4.3 Ordinal Mixed-Effects Random Forest

For ordinal response data, a similar extension to the methods presented in Section 2.4.2 was not available until Bergonzoli et al. (2024) recently proposed Ordinal Mixed-Effects Random Forest (OMERF) in a pre-print. Building on the cumulative mixed

model from Equation 2.4, OMERF operates on the level of the linear predictor η_{ijr} (where $i = 1, \ldots, n_j$, $j = 1, \ldots, m$ and $r = 1, \ldots, k-1$; cf. Equation 2.4.1) for its estimation procedure (Bergonzoli et al., 2024). Since the true thresholds as well as fixed and random effect component parameters are unknown, η_{ijr} must be initialized beforehand for the estimation procedure. To this end, OMERF fits a OF model to the ordinal response using only the fixed effect component predictors. From the OF model, cumulative predicted probabilities $\hat{\pi}_{ijr}^{\text{OF}}$ for observation i from cluster j taking at most category c_r are obtained. Using the link function F^{-1} (i.e., the inverse of the distribution function F from Equations 2.1 and 2.4), η_{ijr} is initialized as

$$\eta_{ijr}^{\text{OF}} = F^{-1}\left(\hat{\pi}_{ijr}^{\text{OF}}\right),\ i = 1, \ldots, n_j;\ j = 1, \ldots, m;\ r = 1, \ldots, k-1,$$

where F is chosen as the logistic distribution function akin to the CLMM (Bergonzoli et al., 2024). Based on the initialized η_{ijr}^{OF}, OMERF transitions into its iterative estimation procedure. Similar to MERT and RE-EM trees, the fixed effects linear predictor term $\boldsymbol{x}_{ij}^{\top}\boldsymbol{\beta}$ is replaced by a non-linear function $f(x_{ij})$ that is estimated through a RF model. To this end, a RF model is fitted to the modified linear predictor $\tilde{\eta}_{ijr} = \eta_{ijr}^{\text{OF}} - \boldsymbol{z}_{ij}^{\top}\hat{\boldsymbol{u}}_j$ where $\hat{\boldsymbol{u}}_j$ are the current random effect estimates for cluster $j = 1, \ldots, m$ (Bergonzoli et al., 2024). The newly obtained fixed effect component estimate $\hat{f}(\boldsymbol{x}_{ij})$ is then used as an offset for a CLMM model which is fitted to the original ordinal response. As such, new estimates for the random effects and thresholds are obtained. The estimation procedure iterates until the random effect estimates between two iterations do not differ by pre-specified minimum change anymore (Bergonzoli et al., 2024). An implementation of OMERF is available from GitHub (https://github.com/giuliabergonzoli/OMERF). OMERF and the methodology presented in the third article of this thesis (Buczak, 2024) were developed independently from each another.

3 Summary of the Articles

3.1 Article 1: Comparing Tree Ensembles for Ordinal Prediction with a Proportional Odds Model (*J Classif*, 2024)

The article summarized in the following is published in: **Buczak, P.**, Horn, D., & Pauly, M. (2024). Old but gold or new and shiny? Comparing tree ensembles for ordinal prediction with a classic parametric approach. *Journal of Classification (Advance Online Publication)*, 1–27. https://doi.org/10.1007/s00357-024-09497-9

3.1.1 Motivation

The aim of the first article was two-fold: First, the current literature was lacking an encompassing simulation study comparing the two methodological streams for ordinal prediction presented in Section 2. Previous works had mostly focused on illustrating their respective novel contributions without providing a systematic comparison of both methodological streams (Hornung, 2019; Janitza et al., 2016; Tutz, 2021). Out of these works, Tutz (2021) offered the most thorough comparison study by benchmarking several ML-based approaches, parametric models and combined joint ensemble learners on a number of real data examples. In his comparison, the author found that predictive performance gaps occurred primarily between the groups of parametric and ML-based methods as a whole. Within the group of ML-based approaches, the predictive performance was mostly similar (Tutz, 2021). However, the comparison study in Tutz (2021) was only based on real datasets and did not include simulation data for which the effect structures were known and manipulable. For evaluating and comparing different methods, it is generally recommended to use a combination of simulation and real data (see, e.g., Friedrich & Friede, 2024). Therefore, the first main objective of the first article of this thesis was to fill this gap by performing an encompassing comparison study of parametric and ML-based ordinal prediction methods using diverse simulation data and a range of real datasets (Buczak et al., 2024).

The second aim of the article was to investigate whether in the context of score-based

prediction with RF, the optimization of category scores as in OF (Hornung, 2019) leads to consistent improvement in predictive performance over methods relying, e.g., on the default scores $1, 2, \ldots, k$. Since the optimization procedure of OF is associated with a potentially high computational cost, answering this question could help in guiding whether or not it is worth using in practical applications. For a direct comparison, OF is compared to a naive OF version which relies on using the default scores $1, 2, \ldots, k$ and category borders $0.5, 1.5, \ldots, k + 0.5$. Furthermore, we explored potential improvements to the optimization procedure of OF. Since OF's optimization step relies on evaluating the performance of pre-generated sets of interval partitions, the optimization procedure cannot iteratively explore the search space of all possible partitions and focus on promising regions. To allow for this, we therefore additionally introduced the *Ordinal Score Optimization Algorithm* (OSOA) which follows the logic of OF, but relies on a non-linear optimization algorithm instead. I will briefly sketch OSOA in the next section (Buczak et al., 2024).

3.1.2 Ordinal Score Optimization Algorithm

OSOA follows the general flow of OF as presented in Section 2.3.1. It is described in more detail with pseudocode in Algorithm 1 as obtained from Buczak et al. (2024). Similar to OF, the general idea is to assign numeric sub-intervals to the ordinal response categories and to derive numeric category scores to be used to learn a regression RF. To arrive at an optimal set of category borders, OSOA optimizes the internal target function EVALUATEBORDERS as described in pseudocode in Algorithm 2 (Buczak et al., 2024). The target function receives a set of inner category borders (i.e., omitting the outer borders b_1 and b_{k+1} which are fixed to 0 and 1, respectively) and evaluates their viability based on the OOB performance achieved with them. To this end, the numeric category scores are determined as the midpoints of the category sub-intervals. The category scores are used as a proxy target variable for fitting a regression RF. Following OF, the scores are transformed through the quantile function of the standard normal distribution Φ^{-1}, yielding numeric values in the interval $(-\infty, \infty)$. From the RF model, numeric OOB predictions can be obtained and converted back to ordinal category predictions using the (transformed) numeric category borders. Based on the ordinal OOB predictions, the category-specific variant of Youden's Index J_{OF} as used in OF (cf. Section 2.3.1) is computed and returned. To optimize the internal target function, OSOA employs the non-linear optimization algorithm Sbplx from the `NLopt` library (Johnson, 2007) which is based on a variant of the Nelder-Mead optimization algorithm (Nelder & Mead, 1965). As the category border sets generated during the optimization process relate to the ordinal categories, only ordered sets should be considered, i.e., it must

hold that $0 < b_2 < \cdots < b_k < 1$. As the Sbplx optimizer does not allow for inequality constraints, the ordered structure of the category border sets is instead enforced through the penalization of unordered sets. The optimizer runs as long as a maximum number of iterations (`max.eval`) is reached or the performance improvement is smaller than a pre-specified threshold ε. Once the optimization terminates, OSOA fits the final RF model based on the (transformed) optimal category borders, and returns both (Buczak et al., 2024).

Algorithm 1 Ordinal Score Optimization Algorithm (OSOA; Buczak et al., 2024)

procedure OSOA(*max.eval*, ε)
  Assign fixed outer borders $b_1^{'} \leftarrow 0$ and $b_{k+1}^{'} \leftarrow 1$
  Assign starting inner borders $\left(\tilde{b}_2, \ldots, \tilde{b}_k\right) \leftarrow \left(\frac{1}{k}, \ldots, \frac{k-1}{k}\right)$
  Run optimizer on EVALUATEBORDERS using $\left(\tilde{b}_2, \ldots, \tilde{b}_k\right)$ as starting values until *max.eval* is reached or performance improvement is smaller than ε
  Extract optimal inner borders $(b_2^\star, b_3^\star, \ldots, b_k^\star)$
  Compute final scores $(s_1^\star, s_2^\star, \ldots, s_k^\star,) \leftarrow \left(\frac{b_1^{'}+b_2^\star}{2}, \frac{b_2^\star+b_3^\star}{2}, \ldots, \frac{b_k^\star+b_{k+1}^{'}}{2}\right)$
  Fit final RF using $(\Phi^{-1}(s_1^\star), \Phi^{-1}(s_2^\star), \ldots, \Phi^{-1}(s_k^\star))$
  return Final RF and transformed borders $\left(\Phi^{-1}(b_1^{'}), \Phi^{-1}(b_2^\star), \ldots, \Phi^{-1}(b_{k+1}^{'})\right)$
end procedure

Algorithm 2 Target function optimized internally in OSOA (Buczak et al., 2024)

procedure EVALUATEBORDERS($b_2, \ldots, b_k$)
Require: $0 < b_2 < \cdots < b_k < 1$
  Compute scores $(s_1, s_2, \ldots, s_k) \leftarrow \left(\frac{b_1^{'}+b_2}{2}, \frac{b_2+b_3}{2}, \ldots, \frac{b_k+b_{k+1}^{'}}{2}\right)$
  Fit RF using transformed scores $(\Phi^{-1}(s_1), \Phi^{-1}(s_2), \ldots, \Phi^{-1}(s_k))$
  Obtain OOB predictions from fit and assign class labels based on transformed borders $\left(\Phi^{-1}(b_1^{'}), \Phi^{-1}(b_2), \ldots, \Phi^{-1}(b_{k+1}^{'})\right)$
  Compute performance through J_{OF} using class predictions and true class labels
end procedure

3.1.3 Evaluation Through Simulation and Real Data Experiments

To achieve the two objectives of the first article, we performed an extensive comparison study using simulated and real data. As prediction methods, we considered OSOA,

(naive) OF, RFSp, CF and a proportional odds model (specified to include all covariates as main effects). We aimed to create diverse data settings by using three data generating processes (DGPs) as well as three different response category distribution patterns. The DGPs were characterized by an increasing amount of non-linear effects. Whereas the first DGP contained only linear effects, DGPs 2 and 3 gradually replaced some of the linear effects by non-linear effects. In this way, we were able to the performance of the different prediction methods under varying effect structures where we expected the proportional odds model to perform well for linear effect structures and ML-based methods to have performance advantages for the more non-linear effect structures. It further allowed us to study how robust the classic proportional odds model is to deviations from linear effect structures. Inspired by Hornung (2019), who explored the effect of different response category distributions, we introduced further variety into the data generation by using three different distribution patterns: a pattern with equally distributed categories, a pattern with prominent middle categories (referred to as wide middle pattern) and a pattern with prominent margin categories (referred to as wide margins pattern). Apart from the simulation data, we also evaluated the prediction methods on eight real datasets from different domains, e.g., psychology, (bio-)medicine and the social sciences. The datasets varied regarding their sample sizes, number of categories and distribution of response categories. A handful of these datasets have already been analyzed in Tutz (2021). Generally, our results from the simulation and real data experiments echoed the findings of Tutz (2021) that differences between the prediction methods mainly occurred between the group of RF-type methods and the proportional odds model. Our comparisons revealed three key insights. First, despite the expected performance advantage for RF-based methods in scenarios with strong non-linear effects, the proportional odds model remained quite competitive for small samples and under limited non-linear effects. Second, the RF-type methods themselves only displayed small performance differences between each other. While CF and RF slightly fell behind most of the time, OSOA, (naive) OF and RFSp often performed similarly. Third, the benefit of score optimization performed in OF and OSOA was only situational. Both methods could improve upon naive OF for simulation data with the wide middle response pattern as well as for six of the eight real datasets. However, the improvement was not consistent enough to warrant an unreserved recommendation of OF and OSOA over naive OF, especially considering their computational costs (Buczak et al., 2024).

3.1.4 Methodological Outlook

Since the simulation and real data experiments revealed only a situational benefit of score optimization as performed in OF and OSOA, the first article closed with a discussion of possible reasons for the lack of consistent improvement. Apart from questioning the reliance on Youden's Index J (which is not an ordinal measure, but only a measure of sensitivity and specificity) as the optimization target in OF and OSOA, we discussed the role that the prediction procedure used in both methods could play. Since in OF and OSOA the numeric predictions from the RF model are transformed back into ordinal categories at the tree-level, the actual score and category border values may have limited impact if the nodes of the individual trees are often pure. Therefore, we hypothesized, an alternative prediction scheme may prove beneficial where the numeric predictions from the individual trees are first aggregated and the transformation back into an ordinal response category occurs at the forest-level (Buczak et al., 2024). This consideration was investigated in the second article of this thesis (Buczak, 2025).

3.2 Article 2: Frequency-Adjusted Borders Ordinal Forest (*BJMSP*, 2025)

The article summarized in the following is published in: **Buczak, P.** (2025). Frequency-adjusted borders ordinal forest: A novel tree ensemble method for ordinal prediction. *British Journal of Mathematical and Statistical Psychology*, 78(2), 594–616. https://doi.org/10.1111/bmsp.12375

3.2.1 Motivation

The results from the first article of this thesis revealed that the optimization procedures employed in OF and OSOA yielded only situational benefits (Buczak et al., 2024). One potential reason that was discussed in the article was the prediction scheme used in OF and OSOA. Both methods predict an ordinal category for a given observation by transforming the numeric predictions obtained from the internal regression RF model. The numeric tree-level predictions are transformed into category labels using the numeric category borders. As such, the transformation into ordinal category predictions already occurs at the tree-level. In a second step, the individual category predictions are aggregated via majority voting to arrive at a final prediction. The first article of this thesis discusses that this prediction scheme may limit the impact of the actual numeric score choices. Therefore, an alternative scheme is discussed in which the tree-level numeric predictions are first aggregated by averaging. The resulting mean prediction is in turn transformed into an ordinal category prediction at the forest-level (Buczak et al., 2024). The second article of this thesis investigated the use of this alternative prediction scheme. To differentiate between the two schemes, the second article coined the terms transform-first-aggregate-after (TFAA) prediction for the original scheme in OF and OSOA, and aggregate-first-transform-after (AFTA) prediction for the new prediction scheme. Furthermore, the second article pointed out another limitation of OF and OSOA: the link between category borders and scores. For both methods, category scores are always determined as the midpoints of the category sub-intervals. However, it is not clear whether this design decision represents an optimal choice. Decoupling category borders and scores from one another may allow for more flexibility. Addressing the points above, the second article proposed Frequency-Adjusted Borders Ordinal Forest (fabOF), a novel RF-based method for ordinal prediction, which relies on the newly introduced AFTA prediction, and decouples the choice of category borders and scores. I will cover fabOF in more detail in the next section.

3.2.2 Frequency-Adjusted Borders Ordinal Forest

Sharing the score-based regression RF framework for ordinal prediction, fabOF differs conceptually from OF and OSOA in three key aspects. First, fabOF relies on AFTA prediction whereas OF and OSOA use TFAA prediction. Second, fabOF breaks the direct link between category borders and scores present in OF and OSOA. Third, fabOF employs a heuristic to derive its category borders instead of optimizing them. As such, the computational runtime is notably decreased (Buczak, 2025).

The method is described in pseudocode in Algorithm 3 as obtained from Buczak (2025). After assigning arbitrary numeric scores to the ordinal response categories (per default, the scores $1, 2, \ldots, k$ for k categories are used), a regression RF is trained using the numeric scores as the target variable. Based on the RF model, numeric OOB predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, for the observations in the training data are computed. The idea of the heuristic is to use these OOB predictions to determine the category borders such that the distribution of the predicted response categories approximately matches the distribution expected in the general population. To this end, the cumulative relative frequencies $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$ of the ordinal categories up to category k in the training data are computed. The cumulative relative frequencies are in turn used to determine the quantiles $q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}$ of the numeric OOB predictions $\hat{y}_i^{\text{num}}$ for the corresponding probabilities. The quantiles are used as the inner set of category borders $b_2, \ldots, b_k$ while the margin borders are selected as s_1 and s_k since these constitute the minimum and maximum value that can be predicted by the model, respectively. Finally, the RF model and the category borders determined through the heuristic are returned (Buczak, 2025).

Algorithm 3 Frequency-Adjusted Borders Ordinal Forest (fabOF; Buczak, 2025)

procedure FABOF
  Unless specified otherwise, use scores $(s_1, s_2, \ldots, s_k) \leftarrow (1, 2, \ldots, k)$.
  Train regression RF on numeric target $y_i^{\text{num}} \in \{s_1, s_2, \ldots, s_k\}, i = 1, \ldots, n$.
  Compute numeric OOB predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, using RF model.
  For categories up to category k, compute cumulative relative frequencies $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
  Obtain quantiles $q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}$ of OOB predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, for probabilities $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
  Assign category borders $(b_1, b_2, \ldots, b_k, b_{k+1}) \leftarrow \big(s_1, q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}, s_k\big)$.
  return RF model and category borders
end procedure

Variable Importance Measure

To aid interpretation, I additionally proposed a permutation VIM for fabOF. While Janitza et al. (2016) proposed a VIM for ordinal prediction, their VIM relies on the RPS which requires the availability of category probability predictions. Because fabOF only transforms the numeric predictions at the forest-level, predicted probabilities for the ordinal categories are not available for fabOF. Therefore, a custom solution was necessary for fabOF. Furthermore, the AFTA prediction scheme also hindered the computation of variable importance values at the tree-level as is typical for RF. For fabOF, the predictive performance based on ordinal performance measures can only be assessed at the forest-level once the aggregated numeric predictions have been transformed back into a category prediction. As a consequence, variable importance can only be computed at the forest-level. In principle, this would result in each predictor being permuted only once. To enhance the stability of the results, fabOF's permutation VIM computes the variable importance for each predictor based on a pre-specified number of replications and averages the individual importance values. In the article, each predictor is permuted 100 times. The VIM assesses the predictive performance using Cohen's weighted Kappa (Cohen, 1968) with linear weights. For more information regarding fabOF's VIM I refer to Buczak (2025).

3.2.3 Evaluation Through Simulation and an Illustrative Data Example

I evaluated fabOF's performance through a simulation study and an illustrative data example on student performance in math (Cortez, 2014). In the article, fabOF was compared to existing methods such as OF, OSOA, RFSp, RF, a CLM as well as custom modifications of OF and OSOA that replaced the original TFAA prediction scheme by AFTA prediction. For the simulation study, I used different DGPs to create diverse data settings including a combination of linear and non-linear effects as well as a linear combination of two different effect structures (Buczak et al., 2024, similar to). Additionally, two different response category distribution patterns were used.

Overall, the simulation study revealed to promising results regarding fabOF. The predictive performance of fabOF was consistently higher than the performance of existing (ordinal) prediction methods. The method closest in performance were AFTA-modified OF and to some degree AFTA-modified OSOA. As such, the simulation study also revealed the benefit that AFTA prediction can provide for existing methods such as OF and OSOA. While the effects were more subtle for OSOA, OF in particular benefited from the change in prediction scheme. AFTA-modified OSOA only slightly trailed

fabOF and was also on par in some scenarios. The benefit of the frequency-adjusted heuristic of fabOF was especially visible for data scenarios with imbalanced response category frequencies. Another benefit of the heuristic was the notably reduced computational runtime of fabOF compared to OF and OSOA. This was to be expected as fabOF only fits a single RF model and does not rely on an extensive optimization procedure. The promising findings from the simulation study were supported by the results for the illustrative data example on math achievement. For the student achievement data, fabOF reached the highest predictive performance for all three performance measures that were considered (Buczak, 2025).

Apart from the predictive performance of fabOF, I also evaluated fabOF's permutation VIM using the simulation data. The VIM was able to recover the important predictors quite well. All influential predictors were correctly identified as such and achieved higher importance values than the noise predictors whose importance values mostly fell around 0. I additionally computed variable importance for the illustrative data example and obtained results that were largely consistent with the educational research literature (see Buczak, 2025, for more details).

3.2.4 Methodological Outlook

Generally, several avenues of further research opened up based on the work from the second article of this thesis. First, the simulation and real data results revealed that changing the prediction procedure from TFAA to AFTA prediction improved the predictive performance of OF. One could further study how the decoupling of category borders and scores (as in fabOF) can also be considered within an optimization framework. To this end, the second article discussed either optimizing both at the same time which may pose an extensive optimization problem or optimizing them with an EM-type (Dempster et al., 1977) estimation procedure where scores and borders are updated iteratively while the other is fixed, respectively. Overall, since the structure of fabOF is relatively modular, one could also investigate similarly extending other regression-based ML methods to ordinal prediction by replacing the regression RF model within the fabOF framework. This will be studied in detail in Section 5 of this thesis. The second article of this thesis further discussed extending fabOF to ordinal prediction in the context of hierarchical data. As hierarchical data structures commonly emerge from applications in the social and life sciences, and these two fields at the same time frequently are the source of ordinal response data, such an extension was deemed promising (Buczak, 2025). Therefore, the third article of this thesis (Buczak, 2024) developed an ordinal prediction method for hierarchical data based on fabOF.

3.3 Article 3: Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (*OSF pre-print*, 2024)

The article summarized in the following has been submitted to *Multivariate Behavioral Research* and is under review at the time of publishing this thesis. The article has been published as a pre-print: **Buczak, P.** (2024). Mixed-Effects Frequency-adjusted borders ordinal forest: A tree ensemble method for ordinal prediction with hierarchical data. *OSF pre-print, version 1.1*, 1–36. https://doi.org/10.31219/osf.io/ny6we

3.3.1 Motivation

So far, the methods considered and proposed in the first two articles of this thesis (Buczak, 2025; Buczak et al., 2024) were concerned with ordinal prediction in non-hierarchical data settings. However, applications in the social and life sciences often yield data with hierarchical data structures. For example, students are usually nested within classes and/or schools, while patients may be nested in hospitals. In longitudinal study designs, multiple measurements are nested within the same person. If left unaccounted for, the dependencies introduced through the hierarchical structure may lead to unreliable uncertainty estimates (Gelman & Hill, 2006). While this particularly concerns questions of inference, even for prediction purposes, the knowledge that observations are grouped in clusters represents information that prediction methods may benefit from (Gelman, 2006). As described in Section 2.4, several tree-based prediction methods have been extended for use with hierarchical data of various response types. For ordinal response data, such an extension was not available for long. Therefore, I developed and proposed Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF) in the third article of this thesis. The mixfabOF method extended fabOF (Buczak, 2025) for ordinal prediction with hierarchical data based on the framework used in MERF (see Section 2.4 Hajjem et al., 2012). Over the development course of mixfabOF, Bergonzoli et al. (2024) independently proposed OMERF which similarly considered ordinal prediction under hierarchical data structures with RF, but used a different approach (see Section 2.4 for a detailed description). I will present mixfabOF in more detail in the next section.

3.3.2 Mixed-Effects Frequency-Adjusted Borders Ordinal Forest

The general flow of mixfabOF consists of the following three steps. First, numeric scores are assigned to the ordinal response categories. Second, the EM-type estima-

tion procedure employed in MERF is used. Third, based on the final RF model, category borders for the ordinal response categories are derived. The method is described in more detail in pseudocode in Algorithm 4 as obtained from Buczak (2024). After assigning the numeric scores (per default, the scores $1, 2, \ldots, k$ are used), the estimation procedure is initialized. For iteration `it`, mixfabOF fits a regression RF to the current modified (numeric) outcome vector $\tilde{\boldsymbol{y}}^{\text{num}}_{(\texttt{it})}$ from which the random effects have been removed. Having updated the fixed effects component through the RF model, the procedure turns to the random effects components and updates the estimated random effects $\hat{\boldsymbol{u}}_{j,(\texttt{it})}$, random effect covariance matrix $\hat{\boldsymbol{D}}_{(\texttt{it})}$ as well as the residual variance $\hat{\sigma}^2_{(\texttt{it})}$. These steps (cf. lines 5-13 with pseudocode in Hajjem et al., 2012) are equivalent with the estimation procedure used in MERF. Similarly, mixfabOF uses the same generalized log-likelihood (GLL) criterion to check for the convergence of the estimation procedure. The GLL criterion is given by

$$GLL(f, \boldsymbol{u}_j | \boldsymbol{y}^{\text{num}}_j) = \sum_{j=1}^{m} \Big\{ \left(\boldsymbol{y}^{\text{num}}_j - f\left(\boldsymbol{X}_j\right) - \boldsymbol{Z}_j \boldsymbol{u}_j\right)^\top \boldsymbol{R}_j^{-1} \left(\boldsymbol{y}^{\text{num}}_j - f\left(\boldsymbol{X}_j\right) - \boldsymbol{Z}_j \boldsymbol{u}_j\right) + \boldsymbol{u}_j^\top \boldsymbol{D}^{-1} \boldsymbol{u}_j + \log|\boldsymbol{D}| + \log|\boldsymbol{R}_j| \Big\}. \tag{3.1}$$

(Hajjem et al., 2012). For computing the criterion, the current estimates are used, respectively. Once the relative change in the GLL criterion falls below a threshold value δ (default value: 0.001), the estimation procedure terminates. If convergence is not achieved, the estimation procedure is stopped after a maximum number of iterations (default value: 100). Both default values were inspired by Salditt et al. (2023). After the estimation procedure, the category borders $b_1, \ldots, b_{k+1}$ for the ordinal response categories are determined. To this end, OOB predictions $\hat{f}(\boldsymbol{X}_j)_{\text{OOB}}, j = 1, \ldots, m$, are computed from the final RF model and combined with the final random effect predictions to arrive at the numeric OOB-based predictions $\hat{y}^{\text{num}}_{ij}, i = 1, \ldots, n_j, j = 1, \ldots, m$, for the training set. The iteration index has been omitted for the sake of readability. Employing the frequency-adjusted borders heuristic of fabOF, the quantiles $q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}$ of the OOB-based predictions $\hat{y}^{\text{num}}_{ij}$ for probabilities $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$ corresponding to the cumulative relative frequencies of the ordinal response categories in the training data up to category k are computed. These quantiles are used as the inner category borders $b_2, \ldots, b_k$ while the outer borders b_1 and b_{k+1} are set to $-\infty$ and ∞, respectively (Buczak, 2024).

Algorithm 4 Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF)

procedure MIXFABOF
  Unless specified otherwise, assign scores $(s_1, s_2, \ldots, s_k) \leftarrow (1, 2, \ldots, k)$.
  Create $\boldsymbol{y}_j^{\text{num}}, j = 1, \ldots, m$, by assigning scores to ordinal response categories.
  Set $\texttt{it} = 0, \hat{\boldsymbol{D}}_{(0)} = \boldsymbol{I}_{n_j \times n_j}, \hat{\boldsymbol{u}}_{j,(0)} = \boldsymbol{0}_{n_j}, j = 1, \ldots, m$.
  while $\texttt{it} \leq \texttt{max.iter}$ and `not converged` **do**
    $\texttt{it} = \texttt{it} + 1$
  Update $\tilde{\boldsymbol{y}}_{j,(\texttt{it})}^{num}, \hat{f}_{(\texttt{it})}(\boldsymbol{X}_j)$ *and* $\hat{\boldsymbol{u}}_{j,(\texttt{it})}$:
    $\tilde{\boldsymbol{y}}_{j,(\texttt{it})}^{\text{num}} = \boldsymbol{y}_j^{\text{num}} - \boldsymbol{Z}_j \hat{\boldsymbol{u}}_{j,(\texttt{it}-1)}, j = 1, \ldots, m$.
    Obtain $\hat{f}_{(\texttt{it})}(\boldsymbol{X}_j)$ by fitting a regression RF to $\tilde{\boldsymbol{y}}_{(\texttt{it})}^{\text{num}}$ with covariates $\boldsymbol{X}$.
    $\hat{\boldsymbol{u}}_{j,(\texttt{it})} = \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \boldsymbol{Z}_j^\top \hat{\boldsymbol{V}}_{j,(\texttt{it}-1)}^{-1} \left(\boldsymbol{y}_j^{\text{num}} - \hat{f}_{(\texttt{it})}(\boldsymbol{X}_j) \right), j = 1, \ldots, m,$
      where $\hat{\boldsymbol{V}}_{j,(\texttt{it}-1)}^{-1} = \boldsymbol{Z}_j \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \boldsymbol{Z}_j^\top + \hat{\sigma}_{(\texttt{it}-1)}^2 \boldsymbol{I}_{n_j \times n_j}$.
  Update $\hat{\sigma}_{(\texttt{it})}^2$ *and* $\hat{\boldsymbol{D}}_{(\texttt{it})}$:

$$\hat{\sigma}_{(\texttt{it})}^2 = \frac{1}{n} \sum_{j=1}^{m} \hat{\boldsymbol{\varepsilon}}_{j,(\texttt{it})}^\top \hat{\boldsymbol{\varepsilon}}_{j,(\texttt{it})} + \hat{\sigma}_{(\texttt{it}-1)}^2 \left(n_j - \hat{\sigma}_{(\texttt{it}-1)}^2 \text{trace} \left(\hat{\boldsymbol{V}}_{j,(\texttt{it}-1)} \right) \right),$$

$$\hat{\boldsymbol{D}}_{(\texttt{it})} = \frac{1}{m} \sum_{j=1}^{m} \left\{ \hat{\boldsymbol{u}}_{j,(\texttt{it})} \hat{\boldsymbol{u}}_{j,(\texttt{it})}^\top + \left(\hat{\boldsymbol{D}}_{(\texttt{it}-1)} - \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \boldsymbol{Z}_j^\top \hat{\boldsymbol{V}}_{j,(\texttt{it}-1)}^{-1} \boldsymbol{Z}_j \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \right) \right\},$$

      where $\hat{\boldsymbol{\varepsilon}}_{j,(\texttt{it})} = \boldsymbol{y}_j^{\text{num}} - \hat{f}_{(\texttt{it})}(\boldsymbol{X}_j) - \boldsymbol{Z}_j \hat{\boldsymbol{u}}_{j,(\texttt{it})}$.
    Check convergence using GLL criterion.
  end while
  Compute numeric OOB predictions $\hat{f}(\boldsymbol{X}_j)_{\text{OOB}}$ with final RF model, $j = 1, \ldots, m$.
  Compute OOB-based predictions $\hat{\boldsymbol{y}}_j^{\text{num}} = \hat{f}(\boldsymbol{X}_j)_{\text{OOB}} + \boldsymbol{Z}_j \hat{\boldsymbol{u}}_j, j = 1, \ldots, m$.
  For categories up to category k, compute cumulative relative frequencies $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
  Obtain prediction quantiles $q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}$ of OOB-based predictions $\hat{y}_{ij}^{\text{num}}$ for probabilities $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1},\ i = 1, \ldots, n_j, j = 1, \ldots, m$.
  Assign category borders $(b_1, b_2, \ldots, b_k, b_{k+1}) \leftarrow \left(-\infty, q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}, \infty\right)$.
  return RF model, final random effect estimates and category borders
end procedure

Variable Importance Measure for mixfabOF

For interpretability, I also implemented a permutation VIM for mixfabOF. The VIM was largely based on the permutation VIM introduced for fabOF in the second article of this thesis (Buczak, 2025). It was adapted for use with mixfabOF in two ways. First, the predictive performance takes the fixed and random effects components into account as it is based on the numeric predictions

$$\hat{y}_{ij}^{\text{num}} = \hat{f}(\boldsymbol{x}_{ij})_{\text{OOB}} + \boldsymbol{z}_{ij}^{\top}\hat{\boldsymbol{u}}_j, \;\; i = 1, \ldots, n_j; \; j = 1, \ldots, m,$$

(cf. line 15 in Algorithm 4) which are transformed into ordinal category predictions using the computed category borders. In contrast to this, Pellagatti et al. (2021) and Bergonzoli et al. (2024) only considered the fixed effects component for computing variable importance. However, I would argue that it fits the spirit of the overarching model which consists of fixed and random effects more aptly when both components are used for computing variable importance instead of only considering the fixed effects component. Second, to take the hierarchical nature further into account, mixfabOF's VIM also allows for restricting permutations to occur only within the same cluster. Similar to the permutation proposed for fabOF, Cohen's weighted Kappa (Cohen, 1968) with linear weights is used to assess the predictive performance (Buczak, 2024).

3.3.3 Evaluation Through Simulation and an Illustrative Data Example

To evaluate mixfabOF, I performed a simulation study and used an illustrative data example on math achievement. For the simulation setup, I mostly followed the simulations from Hajjem et al. (2012) and Salditt et al. (2023) and adapted them for ordinal prediction. The simulation data were characterized by a predominantly non-linear effect structure. For introducing variety into the data generation, the number of clusters, the sizes of the clusters and the magnitude of the random effect variance were varied. Similar to the other two articles of this thesis (Buczak, 2025; Buczak et al., 2024), I additionally varied the response category pattern. As benchmark prediction methods, I used a CLMM, OMERF, fabOF, OF and a regular classification RF. For fabOF, OF and RF, which do not account for hierarchical data structures, I included the cluster membership indicator as an additional predictor such that the full information is available for the respective models as well (Buczak, 2024).

Overall, the simulation results were influenced the most by the magnitude of the random effect variance. For a small random effect variance, mixfabOF, fabOF and OF were close in performance while RF was slightly trailing behind these three. The CLMM fell

behind notably due to the strongly non-linear effect structure (cf. results from Buczak et al., 2024). OMERF, on the other hand, suffered from high non-convergence rates and inferior predictive performance in all simulation conditions. For medium random effect variance, mixfabOF started to display performance advantages over the remaining methods. This gap widened further for simulation settings with high random effect variance. These promising findings were underlined when evaluating mixfabOF on a data example stemming from the Trends in International Mathematics and Science Study (TIMSS) 2019 data (Fishbein et al., 2021). For the analyzed dataset, mixfabOF achieved the highest predictive performance followed by a CLMM and fabOF. Similar to the simulation study, OMERF suffered from convergence issues. To check for correct usage of the method, I additionally performed a small simulation based on the simulation setup in Bergonzoli et al. (2024). Benchmarking the same methods as above, mixfabOF reached the best predictive performance in this additional simulation followed by fabOF and OMERF. For this additional simulation, OMERF was not affected by convergence issues (Buczak, 2024).

One possible explanation may be the number and sizes of the simulated clusters. While Bergonzoli et al. (2024) only considered simulation settings with 10 clusters with 100 observations each (of which 20% were used for the test data), my simulation considered settings of 100 clusters with 20-25 observations each (of which 10 were reserved for the test data, respectively) and 250 clusters with 50-60 observations each (of which 25 were reserved for the test data, respectively). As such, it could be the case that OMERF does not scale well for many clusters or clusters of smaller size. This is currently being investigated for the preparation of a revised manuscript.

3.3.4 Methodological Outlook

The third article of this thesis only considered random intercept models in the simulation and real data experiments. In principle, mixfabOF also supports more complex random effect structures including random slope effects. Further work could study the performance of mixfabOF for such random effect structures. As mixfabOF internally relies on a LMM, it can currently take corresponding random effect types into account. In particular, random effect structures known from cumulative mixed models such as random thresholds cannot be reflected as of yet. Future work could investigate how such types of random effects could be considered in the framework of mixfabOF. Perhaps one could explore the possibility of cluster-specific category borders. The original simulation in the third article of this thesis also lacked a simulation setting for which new observations from unknown clusters were generated. Results from Salditt et al.

(2023) indicate that for the prediction of unknown clusters, the performance advantages observed for known clusters are likely to diminish. Similar to fabOF, mixfabOF is also structured in a relatively modular fashion, i.e, one could replace the regression RF used for estimating the fixed effects component through another regression-based ML method. I will explore such extensions in more detail in Section 5.

4 Computational Implementation

The methodology presented in the second and third articles of this thesis was implemented in the `fabOF` R package available from GitHub (https://github.com/phibuc/fabOF). In this section, I will briefly showcase the use of the package based on two illustrative data examples. The first data example was used in the second article of this thesis (Buczak, 2025) and pertains to predicting the math performance of Portuguese students based on eleven predictors. The predictors include age, sex, interest in higher education, parents' education, study time, etc. The original data were first analyzed in Cortez and Silva (2008) and made publicly available in Cortez (2014). To fit a fabOF model to the data, I use the `fabOF()` function. It requires the specification of a model formula and a respective dataset. All formulas compatible with the `ranger` package (Wright & Ziegler, 2017) can be supplied. Optionally, specific numeric scores for the ordinal response categories can be provided through the `scores` argument. By default, the scores $1, \ldots, k$ for k categories will be used. Variable importance values (based on the permutation VIM introduced in the second article of this thesis) can be requested by setting `importance = TRUE`. The `importance.reps` argument steers how many replications (i.e., permutations) are used for computing the importance values. As such, this argument directly impacts the runtime of `fabOF()` should variable importance be requested. By default, 100 replications are performed for determining the variable importance. As `fabOF()` internally calls `ranger()`, arguments for `ranger()` can be provided through the `ranger.control` argument offered by `fabOF()`. To this end, a list containing named entries corresponding to `ranger()` arguments can be specified, e.g. `list(num.trees = 500, mtry = 3)`. For all `ranger()` arguments that are not specifically provided, the default settings are used. In principle, a hyperparameter tuning may be needed for obtaining optimal predictive performance. For the sake of simplicity (and since RFs are also relatively robust regarding their hyperparameter choices; Probst et al., 2019), I use the default settings for `ranger()` in the following. To showcase the permutation VIM, I additionally request the computation of variable importance values (using the default of 100 replications).

```
R> fit_fabOF <- fabOF(formula = y ~ .,
                      data = dataset,
                      importance = TRUE)
```

Having trained the model, the corresponding `print` function can be used to obtain some basic information about the model.

```
R> fit_fabOF
Frequency Adjusted Borders Ordinal Forest (fabOF)

Call:
 fabOF(formula = y ~ ., data = dataset, importance = TRUE)

Number of trees:  500
Observations:     649
Covariates:       12
Target variable:  y
Categories:       1 (n = 100)
                  2 (n = 201)
                  3 (n = 154)
                  4 (n = 112)
                  5 (n = 82)
Category scores:  1 2 3 4 5
Category borders: 1 2.171181 2.771268 3.191494 3.563543 5
```

The computed variable importance values can be accessed from the fitted object and visualized as follows, for example:

```
R> barplot(sort(fit_fabOF$variable.importance),
           horiz = TRUE, las = 1)
```

Figure 4.1 shows that the most important predictors are an interest in higher education, mother's education and study time. For a discussion of these findings, I refer to the second article of this thesis (Buczak, 2025). Generally, these results were found to be consistent with the educational research literature (see Buczak, 2025, for a more detailed overview).

For the illustrative data example above, hierarchical data structures were not considered. However, if such structures are present in the data and a non-negligible amount of random effect variation is to be expected, fitting a mixfabOF model may be more sensible. To showcase fitting a mixfabOF model, I am using the data example from the third article in this thesis (Buczak, 2024). The data originate from the 2019 Trends in International Mathematics and Science Study (TIMSS; Fishbein et al., 2021) study. I have used a subset of 2773 German fourth-grade students from 191 different schools. The aim was predicting the students' math performance based on ten predictors including age, sex and scales on liking of learning math, confidence in math, disorderly behavior during math lessons, etc. For a more detailed description of the data, I refer to the third

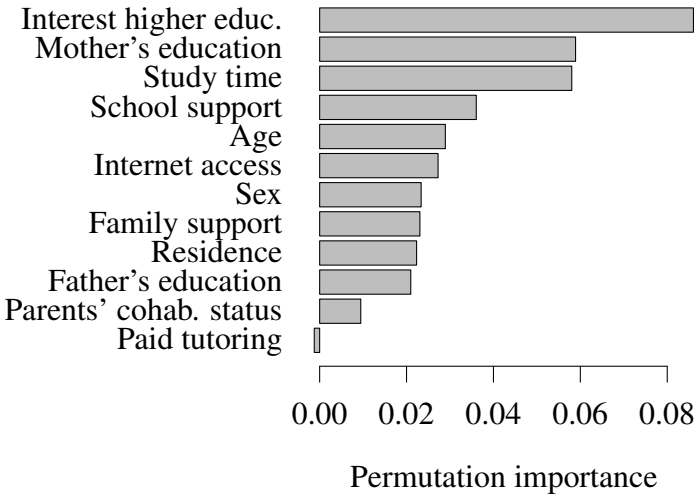

Figure 4.1: Permutation variable importance values for student performance data. *Note. Figure inspired by Buczak (2025).*

article of this thesis (Buczak, 2024).

The `mixfabOF()` function used for fitting mixfabOF models has a similar interface as the `fabOF()` function. However, due to the EM-type estimation procedure, additional considerations arise. Most importantly, the random effects structure must be specified using the `random` argument. Generally, the syntax for specifying random effects is similar to syntax used in other popular packages for mixed models, e.g., `lme4` (Bates et al., 2015). In this case, I include a random intercept on the school `id` variable which indicates the cluster membership of a given observation. Such a random intercept model can be specified through `random = ~ 1|id`. The fixed effects structure is specified using the `formula` argument. For controlling the EM-type estimation procedure, the `delta` argument modulates the threshold for the relative change in the GLL criterion (cf. Equation 3.1), while `max.iter` represents the maximum number of iterations permitted. The current default values `delta = 0.001` and `max.iter = 100` are inspired by Salditt et al. (2023). Similar to `fabOF()`, permutation variable importance values can be requested based on `importance.reps` replications. The implemented VIM respects the hierarchical data structure by considering fixed and random effect components when computing predictions and their respective performance (Buczak, 2024). To further acknowledge the hierarchical nature of the data, one can optionally limit permutations to be performed only within the same cluster by setting `permute.clusterwise = TRUE`. Note that this functionality is experimental at

this stage. More research is needed to obtain recommendations on when this permutation approach can yield benefits over the unrestricted permutation approach. As for `fabOF()`, custom category scores can be provided via the `scores` argument and the `ranger.control` argument can be used to directly provide arguments for `ranger`.

```
R> mixfabOF_fit <- mixfabOF(y ~ ASBG01 + ASDAGE + ASBGSLM
                            + ASBGDML + ASBGICM + ASBGSEC
                            + ASDG05S + ASBGSSB + ASBGSB
                            + ASBGSCM,
                           data = dataset2,
                           random = ~ 1|id,
                           importance = TRUE,
                           permute.clusterwise = TRUE)
```

The resulting object can be inspected for further information. Variable importance values can be accessed analogously to `fabOF` objects as demonstrated above. For a corresponding visualization and a discussion of the results, I refer to the third article of this thesis (Buczak, 2024).

```
R> mixfabOF_fit
Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF)

Call:
 mixfabOF(y ~ ASBG01 + ASDAGE + ASBGSLM + ASBGDML + ASBGICM + ASBGSEC
 + ASDG05S + ASBGSSB + ASBGSB + ASBGSCM, data = dataset2,
 random = ~1|id, importance = TRUE, permute.clusterwise = TRUE)

Number of trees:  500
Observations:     2773
Covariates:       10
Target variable:  y
Fixed effects:    y ~ ASBG01 + ASDAGE + ASBGSLM + ASBGDML + ASBGICM
                    + ASBGSEC + ASDG05S + ASBGSSB + ASBGSB + ASBGSCM
Random effects:   ~ 1|id
Categories:       1 (n = 354)
                  2 (n = 598)
                  3 (n = 778)
                  4 (n = 681)
                  5 (n = 362)
Category scores:  1 2 3 4 5
Category borders: -Inf 2.015757 2.696708 3.360551 4.01296 Inf
Converged:        TRUE
Iterations:       3
```

In the future, the `fabOF` package could be expanded through further functionality. For

example, one could add means of visualizing variable importance results or implement convenience functions, e.g., for hyperparameter tuning. In light of the methodological extensions of fabOF and mixfabOF in Section 5 one could also implement wrapper functions that allow for creating custom prediction methods based on the (Mixed-Effects) Frequency-Adjusted Borders Ordinal Prediction Framework (see Section 5). Moreover, one could add more functionality regarding the interpretability of fabOF and mixfabOF (see Section 6.4 for a more detailed discussion).

5 Methodological Extensions

The three articles of this thesis (Buczak, 2024, 2025; Buczak et al., 2024) have demonstrated that ordinal prediction is an interesting application field with much untapped potential. In particular, fabOF (Buczak, 2025) and mixfabOF (Buczak, 2024) achieved promising results. Apart from their predictive performance, another strength of fabOF and mixfabOF is their modular algorithmic design. On a more abstract level, fabOF leverages concepts of ordinal classification and regression by assigning numeric scores to the ordinal response categories for training a regression RF and translating the numeric predictions back into ordinal response categories through a heuristic based on the response category distribution. As such, one could replace the internal RF model by another regression-based ML model. This is relevant in so far as one cannot expect RF to be the best performing method for each data application. Being able to replace the regression RF model allows for tailoring the framework of fabOF towards potential model requirements for a given application, and also allows for benchmarking multiple different ML methods for determining the optimal method for the respective dataset.

In this section, I will therefore explore extensions of the frequency-adjusted borders heuristic of fabOF to alternative RF algorithms such as Extremely Randomized Trees (Extra-Trees; Geurts et al., 2006) and Conditional Inference Forest (Hothorn et al., 2006) as well as the popular gradient boosting algorithm XGBoost (Chen & Guestrin, 2016). I will showcase all three extensions through a small simulation experiment that demonstrates the promising potential of these prototypes. Building off these extensions, I will further demonstrate that the fabOF algorithm can be generalized to a model-agnostic variant that can be used as a blueprint for extending arbitrary regression-based ML methods to ordinal prediction. Therewith, I provide a powerful framework for developing new methodology for ordinal prediction. In a similar fashion, I will further introduce a model-agnostic variant of the mixfabOF method which opens the possibility of developing a wide array of ordinal prediction methods for hierarchical data. I will showcase the model-agnostic generalization through a prototype based on XGB.

Finally, I will touch upon another interesting avenue for extending fabOF: multivariate ordinal prediction. In some applications, it may be of interest to not only predict a single, univariate outcome per person, but a vector-valued outcome for each person.

For example, one may be interested in predicting multiple dimensions of self-reported health status or the agreement towards several items on a rating scale. To this end, I will provide a prototype sketch of how fabOF could be extended to multivariate prediction tasks. Additionally, I will discuss some of the challenges associated with such extensions that would need to be addressed when developing corresponding prediction methods in the future.

5.1 Extensions for Extra-Trees and Conditional Inference Forest

5.1.1 Motivation

While the CART-based implementation of RF used in fabOF (and mixfabOF) is a popular and versatile choice, alternative implementations may be favorable depending on the specific application at hand. If it is of particular interest to reduce the variability of the resulting models, then the Extra-Trees (Geurts et al., 2006, ET) algorithm could be considered. ET differs from the original RF algorithm (Breiman, 2001) in two ways. First, instead of bootstrapping, the observations for the individual tree models are sampled without replacement. Second, instead of selecting the best split from the randomly sampled subset of predictors, a random split value for these predictors is used. Compared to RF, ET tends to decrease variance, but increase bias (Geurts et al., 2006). Additionally, as noted in Section 2.2.1, CART-based trees have been shown to favor predictors with many possible split points or missing values during the splitting process (Hothorn et al., 2006). To amend this variable selection bias, Hothorn et al. (2006) proposed their conditional inference framework for recursive partitioning. The use of Conditional Inference Forest (CF) for ordinal prediction has been studied in Janitza et al. (2016). While supporting ordinal responses, CFs internally transform the ordinal response into a numeric variable for performing the association tests needed for the splitting procedure. However, at least for the simulation and real data considered in the first article of this thesis, CF often fell behind other RF-based learners regarding the predictive performance. Therefore, it would be interesting to study whether an extension of the frequency-adjusted borders heuristic to CF can yield improvements over the use of regular CF for ordinal prediction.

5.1.2 Extension Prototypes

For exploring the use of the frequency-adjusted borders heuristic for both ET and CF, I introduce the two prototype extensions Frequency-Adjusted Borders Extra-Trees (fa-

bET) and Frequency-Adjusted Borders Conditional Inference Forest (fabCF). They are described with pseudocode in Algorithms 5 and 6 based on the fabOF pseudocode from Buczak (2025). Conceptually, fabET and fabCF differ from fabOF only in the underlying RF model fitted to the numeric scores assigned to the ordinal response categories and used to obtain numeric predictions for training data or new observations. As such the regression RF model of fabOF is replaced by a regression ET model for fabET, and by a regression CF model for fabCF, respectively. As both ET and CF allow for computing OOB predictions for the observations in the training data, the category borders can still be computed – as in fabOF – from the OOB prediction quantiles of probabilities corresponding to the cumulative relative frequencies of the ordinal response categories in the training data (cf. Section 3.2.2). In the final step, the category borders as well as the corresponding ET or CF model are returned, respectively.

Algorithm 5 Frequency-Adjusted Borders Extra-Trees (fabET)

procedure FABET
  Unless specified otherwise, use scores $(s_1, s_2, \ldots, s_k) \leftarrow (1, 2, \ldots, k)$.
  Train regression ET on numeric target $y_i^{\text{num}} \in \{s_1, s_2, \ldots, s_k\}, i = 1, \ldots, n$.
  Compute numeric OOB predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, using ET model.
  For categories up to category k, compute cumulative relative frequencies $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
  Obtain quantiles $q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}$ of OOB predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, for probabilities $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
  Assign category borders $(b_1, b_2, \ldots, b_k, b_{k+1}) \leftarrow \left(s_1, q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}, s_k\right)$.
  return ET model and category borders
end procedure

Algorithm 6 Frequency-Adjusted Borders Conditional Inference Forest (fabCF)

procedure FABCF
  Unless specified otherwise, use scores $(s_1, s_2, \ldots, s_k) \leftarrow (1, 2, \ldots, k)$.
  Train regression CF on numeric target $y_i^{\text{num}} \in \{s_1, s_2, \ldots, s_k\}, i = 1, \ldots, n$.
  Compute numeric OOB predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, using CF model.
  For categories up to category k, compute cumulative relative frequencies $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
  Obtain quantiles $q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}$ of OOB predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, for probabilities $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
  Assign category borders $(b_1, b_2, \ldots, b_k, b_{k+1}) \leftarrow \left(s_1, q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}, s_k\right)$.
  return CF model and category borders
end procedure

5.1.3 Simulation Setup

For showcasing both prototypes, I performed a small simulation experiment based on parts of the simulation setup of the second article of this thesis (Buczak, 2025). To this end, I simulated 2500 observations of 15 standard normally distributed predictors $X_1, \ldots, X_{15}$. The predictors were simulated as uncorrelated. As in Buczak (2025), the ordinal five-category outcome was simulated from a proportional odds model where (omitting the person indices for readability) the linear predictor $\boldsymbol{x}^\top \boldsymbol{\beta}$ was determined by

$$\begin{aligned}\boldsymbol{x}^\top \boldsymbol{\beta} = &\begin{cases} 1, & x_1 \in (-1, 1] \\ -1, & x_1 \notin (-1, 1] \end{cases} + \mathbb{1}_{x_2 > 0} + 0.75 x_3 + 0.25 x_3^2 + 0.75 x_4 \\ &+ 0.25 \cdot \mathbb{1}_{x_2 > 0.5 \wedge x_4 \leq 0.5} + 0.5 x_5 + 0.5 x_6.\end{aligned}$$

As such, six of the 15 predictors were influential, while the remaining nine predictors were only noise. Similar to Buczak (2025), I have also considered two different response category distribution patterns: a pattern with equally distributed categories and a pattern with prominent middle categories. For generating these patterns, I have used the same approach as in the second article of this thesis (Buczak, 2025) to which I refer for further details. From the datasets generated as described above, $2/3$ of the data points were used for training the model while the remaining $1/3$ data points were used for evaluation. Predictive performance was assessed with Cohen's weighted Kappa (Cohen, 1968) as it is a commonly used measure for ordinal prediction (see, e.g., Ben-David, 2008; Hornung, 2019). Through different weighting schemes, deviations between true and predicted categories can be accentuated differently (see Hornung, 2019, for a more detailed discussion). Common weight choices for ordinal prediction include linear and quadratic weights (Ben-David, 2008; Hornung, 2019). As an additional performance measure, I consider Kendall's rank correlation (Kendall, 1948). As a benchmark method for fabCF, I have included regular CF (supporting ordinal responses), while for fabET I have included multi-label classification ET and a naive regression ET approach. Naive ET uses the default scores $1, 2, 3, 4, 5$ and category borders $-\infty, 1.5, 2.5, 3.5, 4.5, \infty$ (cf. naive OF from the `ordinalForest` package; Hornung, 2022). Additionally, I have included fabOF as a further benchmark method. Since all approaches are RF-type algorithms which are relatively robust (see Probst et al., 2019), I did not perform a hyperparameter tuning, but used the respective default values instead. Since this simulation was only meant to be a proof-of-concept, I only performed 100 replications.

5.1.4 Simulation Results and Outlook

Figure 5.1 shows the results for the small simulation study. It can be seen that both fabCF and fabET improved upon their counterparts, i.e., fabCF reached higher predictive performance than regular CF, while fabET improved upon (multi-label classification) ET and naive ET. The differences were particularly pronounced when the distribution of the response categories followed the wide middle pattern. Between CF and ET, the relative improvement when applying the frequency-adjusted borders heuristic was greater for ET which does not support ordinal responses compared to CF which already supports ordinal responses to a degree. The performance of naive ET shows that it is not a viable alternative to multi-label classification ET, and that further adjustment is needed as realized by the heuristic in fabET. Even though all approaches stayed notably or slightly behind fabOF in terms of predictive performance, this simulation experiment

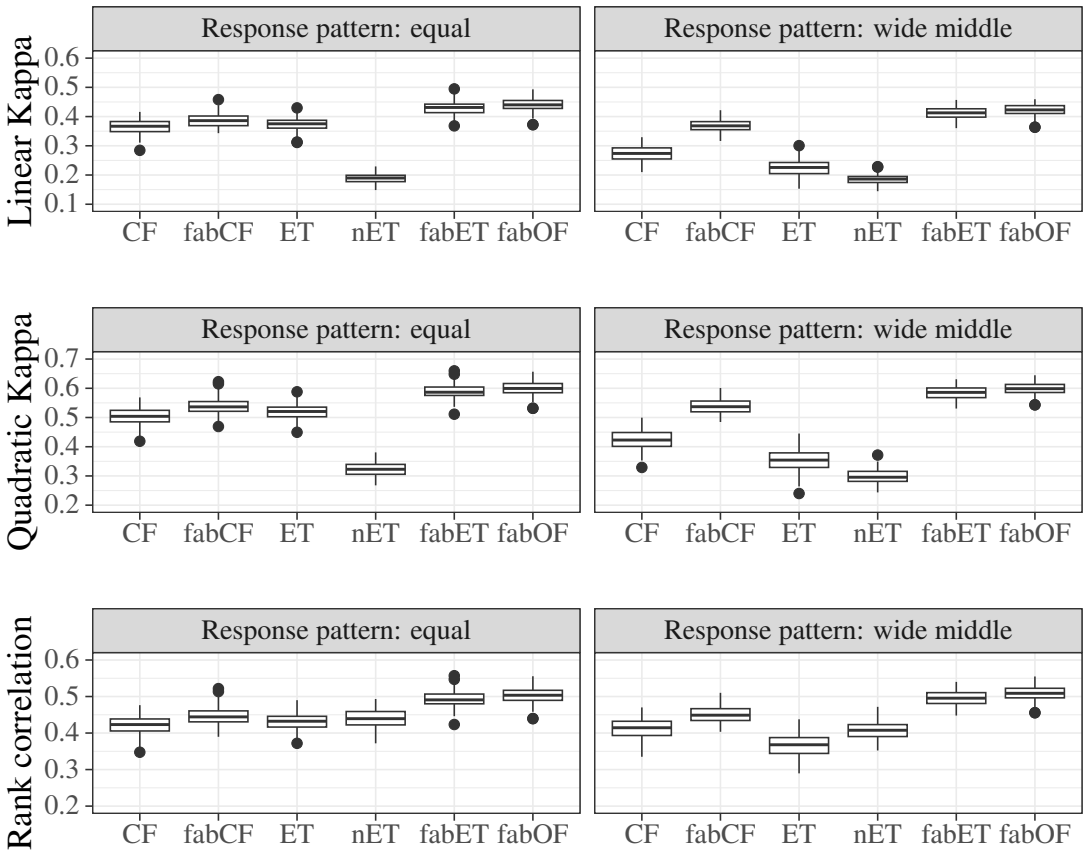

Figure 5.1: Predictive performance of fabCF and fabET extensions for different response distribution patterns. *Note. nET: naive Extra-Trees.*

demonstrates the promising potential of employing the frequency-borders heuristic for other RF-type prediction methods.

Future work could build upon these findings to study the performance of fabET and fabCF in more diverse data scenarios. This could include expanding the simulation settings, introducing further data generating processes as well as benchmarking the predictive performance on a variety of real datasets. In particular, it would be of interest to study whether the beneficial properties for which ET (bias/variance trade-off) and CF (avoidance of variable selection bias) are favored in certain data scenarios carry over to fabET and fabCF, respectively. This could help in working out their use cases and establishing their position next to fabOF.

5.2 Extension for XGBoost

5.2.1 Motivation

The previous section provided a proof-of-concept for extending the frequency-adjusted borders heuristic to other RF-type algorithms. However, the modularity of the fabOF method also allows for employing ML methods apart from RF-type algorithms. Depending on the data application at hand, other ML methods may lead to higher predictive performance. Another powerful class of ML models are boosting methods which are briefly presented in the following (influenced by the description in Hastie et al., 2009). Boosting methods aim to iteratively combine models of a weak learner (e.g., a regression tree) in an additive manner to arrive at an improved ensemble model $g_{\text{Boost}}(\boldsymbol{x})$ with

$$g_{\text{boost}}(\boldsymbol{x}_i) = \sum_{\ell=1}^{L} \zeta_\ell H\left(\boldsymbol{x}_i; \theta_\ell\right), \quad i = 1, \ldots, n,$$

where $\zeta_\ell \in \mathbb{R}$ denotes the coefficient of the ℓ-th base learner model $H\left(\boldsymbol{x}_i; \theta_\ell\right)$ with parameters $\theta_\ell \in \Theta, \ell = 1, \ldots, L$ (Hastie et al., 2009). The base learner models are added iteratively such that a pre-specified loss function $\mathcal{L}\left(y_i, g(\boldsymbol{x}_i)\right)$ is optimized. A common loss function for regression tasks is, e.g., the squared error loss function $\mathcal{L}_{\text{sq}}\left(y_i, g(\boldsymbol{x}_i)\right) = \left(y_i - g(\boldsymbol{x}_i)\right)^2, \ i = 1, \ldots, n$ (Hastie et al., 2009). To limit the computational complexity of the optimization, the coefficients and model parameters of already added base models are fixed and only the coefficient ζ_ℓ and model parameters θ_ℓ of the model to be added are optimized (also known as forward stagewise additive modeling; Hastie et al., 2009). For the ℓ-th step, the coefficient ζ_ℓ and base learner

model parameters θ_ℓ are determined by

$$(\zeta_\ell, \theta_\ell) = \arg\min_{\zeta,\theta} \sum_{i=1}^{n} \mathcal{L}\left(y_i, g_{\ell-1}(\boldsymbol{x}_i) + \zeta H\left(\boldsymbol{x}_i; \theta\right)\right), \quad \ell = 1, \ldots, L$$

(Hastie et al., 2009). The model is then updated through

$$g_\ell(\boldsymbol{x}_i) = g_{\ell-1}(\boldsymbol{x}_i) + \zeta_\ell H\left(\boldsymbol{x}_i; \theta_\ell\right), \; i = 1, \ldots, n; \; \ell = 1, \ldots, L$$

(Hastie et al., 2009). A major class of boosting algorithms are gradient boosting algorithms (Friedman, 2001) which use the gradient descent method for the stagewise optimization procedure. Since the gradient of the loss function is only defined at the training data points, a base learner model (e.g., a regression tree) is trained on the so-called pseudo-residuals $\xi_{i\ell}$ with

$$\xi_{i\ell} = -\left[\frac{\partial \mathcal{L}\left(y_i, g(\boldsymbol{x}_i)\right)}{\partial g(\boldsymbol{x}_i)}\right]_{g=g_{\ell-1}}, \; i = 1, \ldots, n; \; \ell = 1, \ldots, L,$$

to approximate the negative gradient, while the corresponding step length is determined via line search (Hastie et al., 2009). The number of added base learner models L, i.e., the number of boosting iterations, is a crucial parameter of boosting algorithms. If L is set too high, the risk of overfitting increases (Hastie et al., 2009). For further reducing the risk of overfitting, Friedman (2001) suggested the use of a learning rate parameter $\nu \in (0, 1]$ which limits the impact of newly added tree models. Typically, optimal choices for L and ν are determined through a hyperparameter tuning. For more information on gradient boosting, I refer to Friedman (2001) and Hastie et al. (2009). A particularly high performing and popular implementation of gradient boosting is XGBoost (XGB; Chen & Guestrin, 2016). XGB additionally introduces several measures aimed at reducing the risk of overfitting: employing a regularized loss function, shrinking the influence of newly added trees (learning rate) and subsampling predictors similar to RF (Chen & Guestrin, 2016). Furthermore, several techniques are used to make the splitting process of the tree models faster and more memory-efficient. For further details, I refer to Chen and Guestrin (2016). XGB has been shown to achieve high predictive performance on a variety of datasets and could often outperform RF and neural networks for tabular data (see, e.g., Grinsztajn et al., 2022; Shwartz-Ziv & Armon, 2022). Therefore, extending the frequency-adjusted borders heuristic of fabOF to XGB seems promising as it would potentially allow for harnessing the predictive power of XGB for ordinal prediction. I will propose and evaluate a corresponding prototype method in the following.

5.2.2 Extension Prototype

For extending the frequency-adjusted borders heuristic to XGB, I introduce Frequency-Adjusted Borders XGBoost (fabXGB) as described with pseudocode (based on pseudocode from Buczak, 2025) in Algorithm 7. Generally, fabXGB follows the flow of fabOF, but replaces the internal regression RF model by a regression XGB model. In contrast to the fabET and fabCF prototypes, fabXGB requires further adjustment as XGB does not offer the possibility of computing OOB predictions. To avoid the possible risk of overfitting, fabOF's heuristic was designed to determine its category borders based on the OOB predictions such that it mimicks the use of an external dataset. If OOB predictions are not available, one would therefore need to set aside a validation subset of the data beforehand if the category borders are to be based on data not used for training the respective model. However, neither the XGB model nor the category borders could make use of the full information in the original data. This appears particularly impractical for small datasets. The alternative approach would be to rely on the training data predictions to determine the class borders instead. While potentially being more prone to overfitting, this solution appears more feasible when data is limited and one aims to make use of the full information contained in the data. Another minor difference between fabXGB and fabOF lies in the selection of the outer category borders b_1 and b_{k+1}. While for the latter, s_1 and s_k were viable choices, the XGB model could output values smaller than s_1 or greater than s_k. Therefore, fabXGB relies on $-\infty$ and ∞ as outer borders, respectively.

Algorithm 7 Frequency-Adjusted Borders XGBoost (fabXGB)

procedure FABXGB
  Unless specified otherwise, use scores $(s_1, s_2, \ldots, s_k) \leftarrow (1, 2, \ldots, k)$.
  Train regr. XGB model on numeric target $y_i^{\text{num}} \in \{s_1, s_2, \ldots, s_k\}, i = 1, \ldots, n$.
  Compute numeric training predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, using XGB model.
  For categories up to category k, compute cumulative relative frequencies $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
  Obtain quantiles $q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}$ of predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, for probabilities $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
  Assign category borders $(b_1, b_2, \ldots, b_k, b_{k+1}) \leftarrow \left(-\infty, q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}, \infty\right)$.
  return XGB model and category borders
end procedure

5.2.3 Simulation Setup

To showcase fabXGB, I have performed a simulation experiment similar to Section 5.1 in which I compared the predictive perfomance of fabXGB, multi-label classification XGB, a naive regression XGB (i.e., a regression XGB model with category scores $1, 2, \ldots, k$ and category borders $0.5, 1.5, \ldots, k+0.5$) and fabOF. To this end, I have generated data according to the same data generating process as described in Section 5.1.3. However, as XGB requires the specification of several hyperparameters, I have additionally performed a hyperparameter tuning. For the list of hyperparameters considered for tuning as well as their respective search spaces, I refer to Table 6.1 in the Appendix. The hyperparameters were tuned using a 3-fold cross-validation on the training data. From 500 randomly generated hyperparameter configurations, the configuration leading the the highest cross-validated linearly weighted Kappa value was selected as the optimal configuration. The optimal hyperparameter settings were then used to retrain the model on the entire training data such that it could be evaluated on the test data. For the sake of comparison, I have applied the same tuning procedure to fabOF (see Table 6.1 in the Appendix for respective hyperparameters and search spaces). For both fabXGB and fabOF, I used the default values of the implementations from the `xgboost` (Chen et al., 2024) and the `ranger` (Wright & Ziegler, 2017) packages, respectively. In particular, the squared error was used as the loss function for XGB's regression tree models. The simulation study was based on 100 replications.

5.2.4 Simulation Results and Outlook

Figure 5.2 shows that fabXGB outperformed XGB and naive XGB for all three performance measures. The differences between between fabXGB and naive XGB were most pronounced for Cohen's weighted Kappa and less for Kendall's rank correlation, while the disparities between fabXGB and XGB were mostly consistent across the three measures. Furthermore, fabXGB also achieved slightly higher predictive performance than fabOF for the data considered here. As such, this simulation experiment (along with the experiments for fabET and fabCF) demonstrates the promising potential of extending the frequency-borders heuristic to other ML methods. Particularly the results for fabXGB indicate that the heuristic can well be extended to methods beyond RF. To foster future research, I will therefore propose a model-agnostic extension variant of the frequency-adjusted borders heuristic in the next section. Regarding fabXGB specifically, future work could investigate the performance of fabXGB more thoroughly based on expanded simulation settings and real datasets. Since XGB requires the specification

of many hyperparameters, future work could also investigate how sensitive fabXGB reacts to the choice of given hyperparameters, and whether, e.g., the transformation of the numeric predictions into ordinal response categories somewhat affects the impact of hyperparameter choices compared to regular XGB.

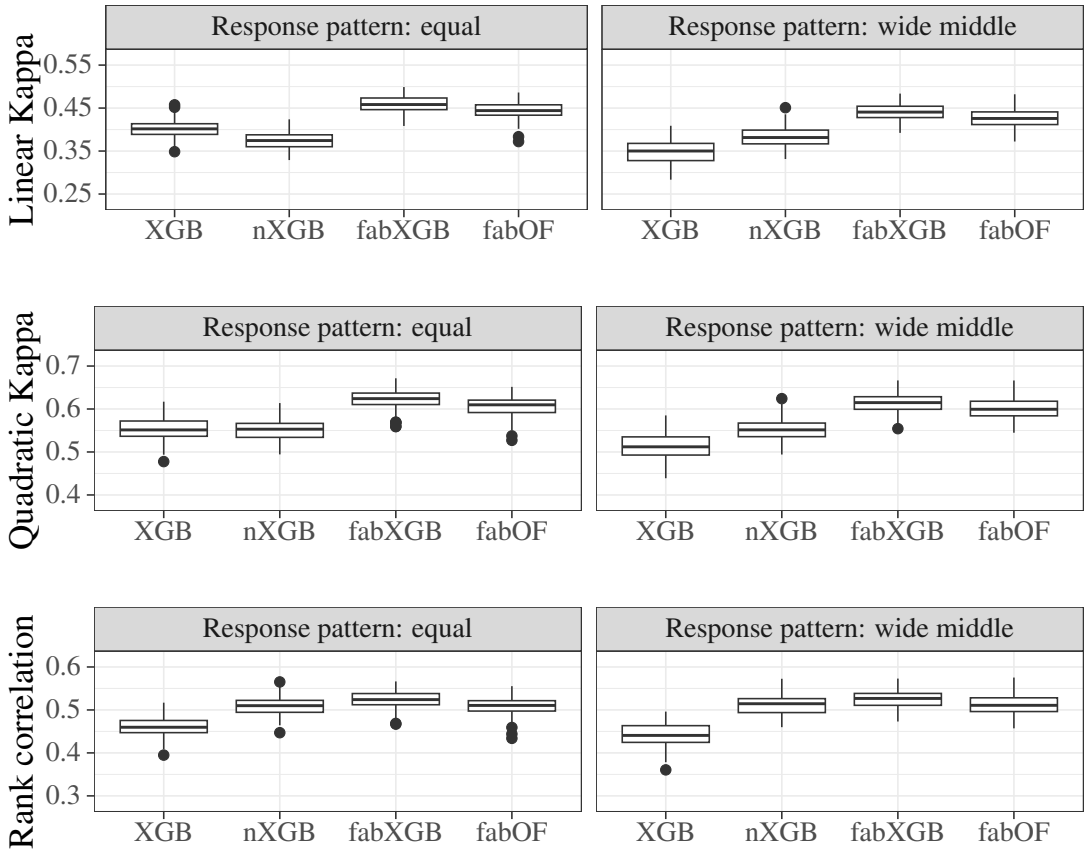

Figure 5.2: Predictive performance of fabXGB extension for different response distribution patterns. *Note. nXGB: naive XGBoost.*

5.3 Frequency-Adjusted Borders Ordinal Prediction Framework

The previous sections have demonstrated that fabOF's modular design can be successfully adapted for developing ordinal prediction methods based on ML methods other than RF. Motivated by the promising results of the fabET, fabCF and fabXGB prototypes, I therefore propose the Frequency-Adjusted Borders Ordinal Prediction Framework, a model-agnostic framework that offers a blueprint for using the frequency-adjusted borders heuristic with any regression-based ML method. The framework underlines the broad applicability of the frequency-borders heuristic and can serve as a reference for the development of a wide variety of ordinal prediction methods in future work. It is described with pseudocode in Algorithm 8 inspired by fabOF's pseudocode in Buczak (2025). The framework follows the logic of fabOF and the extensions in-

Algorithm 8 Frequency-Adjusted Borders Ordinal Prediction Framework

1: Unless specified otherwise, use scores $(s_1, s_2, \ldots, s_k) \leftarrow (1, 2, \ldots, k)$.
2: Train regression ML model on numeric target $y_i^{\text{num}} \in \{s_1, s_2, \ldots, s_k\}, i = 1, \ldots, n$.
3: Compute numeric predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, using ML model.
4: For categories up to category k, compute cumulative relative frequencies $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
5: Obtain quantiles $q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}$ of predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, for probabilities $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
6: Assign category borders $(b_1, b_2, \ldots, b_k, b_{k+1}) \leftarrow \big(-\infty, q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}, \infty\big)$.
7: **return** ML model and category borders

troduced above, but leaves the internal ML model unspecified. Further, the type of predictions used for determining the category borders is also left unspecified. When available (as, e.g., in RF-type methods) OOB predictions can be used and would be recommended. Otherwise, one can use the training predictions instead or rely on a separate set of data. Future work would need to assess how prone to overfitting the strategy of using the training predictions is. While the results from the small simulation for fabXGB indicated promising results, a more encompassing simulation study is needed in the future. Moreover, future research could employ the Frequency-Adjusted Borders Ordinal Prediction Framework for other regression-based ML methods than the ones considered in this thesis, e.g., support vector regression or neural networks.

Additionally, future work could also consider implementing custom VIMs akin to the permutation VIM proposed for fabOF (see Buczak, 2025). To this end, one could use the model-agnostic blueprint provided by Fisher et al. (2019) that served as the inspiration for fabOF's VIM. As such, future ordinal prediction models developed based

on the Frequency-Adjusted Borders Ordinal Prediction Framework could be enhanced with additional functionality regarding the models' interpretability.

5.4 Mixed-Effects Frequency-Adjusted Borders Ordinal Prediction Framework

In a similar fashion to how the the Frequency-Adjusted Borders Ordinal Prediction Framework extended the frequency-adjusted borders heuristic to arbitrary regression-based ML learners, one can likewise generalize mixfabOF (Buczak, 2024). Since mixfabOF was designed with a comparable degree of modularity as fabOF, the internal regression RF model used to estimate the fixed effect component can well be replaced by another ML method. This would allow for developing a wide variety of ordinal prediction methods for hierarchical data which in turn helps in finding the best fitting method for the data application at hand, e.g., regarding predictive performance, the variance/bias trade-off or variable selection bias. To this end, I propose the Mixed-Effects Frequency-Adjusted Borders Ordinal Prediction Framework. The model-agnostic framework is described through pseudocode in Algorithm 9 as inspired by mixfabOF's pseudocode in Buczak (2024). It differs from the original pseudocode of mixfabOF (cf. Algorithm 4) by replacing the regression RF model with an arbitrary regression-based ML model as well as leaving the type of predictions used for the category borders unspecified. Similar to the Frequency-Adjusted Borders Ordinal Prediction Framework (Section 5.3), one can either use OOB predictions (if available), training predictions or predictions obtained from a separate set of data. For further discussion of this matter, I refer to Sections 5.2.2 and 5.3. The Mixed-Effects Frequency-Adjusted Borders Ordinal Prediction Framework can be used to guide future research of novel ordinal prediction methods that can take hierarchical data structures into account. It should be noted that similar to mixfabOF, the estimation of the random effects component takes place at the numeric level, i.e., where the ordinal response categories are represented through numeric scores. Consequently, the specifiable random effects are currently limited to random effects known from a LMM. In particular, random effects originating from the ordinal regression literature such as random thresholds are not supported as of yet (cf. discussion in Section 6).

As a proof-of-concept, I have implemented a prototype of Mixed-Effects Frequency-Borders XGBoost (mixfabXGB) which is described through pseudocode in Algorithm 11 as inspired by pseudocode from Buczak (2024) in the Appendix. The mixfabXGB prototype follows the Mixed-Effects Frequency-Adjusted Borders Ordinal Prediction

Algorithm 9 Mixed-Effects Frequency-Adjusted Borders Ordinal Prediction Framework

1: Unless specified otherwise, assign scores $(s_1, s_2, \ldots, s_k) \leftarrow (1, 2, \ldots, k)$.
2: Create $\boldsymbol{y}_j^{\text{num}}, j = 1, \ldots, m$, by assigning scores to ordinal response categories.
3: Set $\texttt{it} = 0, \hat{\boldsymbol{D}}_{(0)} = \boldsymbol{I}_{n_j \times n_j}, \hat{\boldsymbol{u}}_{j,(0)} = \boldsymbol{0}_{n_j}, j = 1, \ldots, m$.
4: **while** $\texttt{it} \leq \texttt{max.iter}$ and `not converged` **do**
5: $\quad \texttt{it} = \texttt{it} + 1$
6: *Update* $\tilde{\boldsymbol{y}}_{j,(it)}^{num}, \hat{f}_{(it)}(\boldsymbol{X}_j)$ *and* $\hat{\boldsymbol{u}}_{j,(it)}$:
7: $\quad \tilde{\boldsymbol{y}}_{j,(\texttt{it})}^{\text{num}} = \boldsymbol{y}_j^{\text{num}} - \boldsymbol{Z}_j \hat{\boldsymbol{u}}_{j,(\texttt{it}-1)}, j = 1, \ldots, m$.
8: $\quad$ Obtain $\hat{f}_{(\texttt{it})}(\boldsymbol{X}_j)$ by fitting a regression ML model to $\tilde{\boldsymbol{y}}_{(\texttt{it})}^{\text{num}}$ and covariates $\boldsymbol{X}$.
9: $\quad \hat{\boldsymbol{u}}_{j,(\texttt{it})} = \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \boldsymbol{Z}_j^\top \hat{\boldsymbol{V}}_{j,(\texttt{it}-1)}^{-1} \left(\boldsymbol{y}_j^{\text{num}} - \hat{f}_{(\texttt{it})}(\boldsymbol{X}_j) \right), j = 1, \ldots, m,$

$$\text{where } \hat{\boldsymbol{V}}_{j,(\texttt{it}-1)}^{-1} = \boldsymbol{Z}_j \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \boldsymbol{Z}_j^\top + \hat{\sigma}_{(\texttt{it}-1)}^2 \boldsymbol{I}_{n_j \times n_j}.$$

10: *Update* $\hat{\sigma}_{(it)}^2$ *and* $\hat{\boldsymbol{D}}_{(it)}$:

$$\hat{\sigma}_{(\texttt{it})}^2 = \frac{1}{n} \sum_{j=1}^{m} \hat{\boldsymbol{\varepsilon}}_{j,(\texttt{it})}^\top \hat{\boldsymbol{\varepsilon}}_{j,(\texttt{it})} + \hat{\sigma}_{(\texttt{it}-1)}^2 \left(n_j - \hat{\sigma}_{(\texttt{it}-1)}^2 \text{trace} \left(\hat{\boldsymbol{V}}_{j,(\texttt{it}-1)} \right) \right),$$

$$\hat{\boldsymbol{D}}_{(\texttt{it})} = \frac{1}{m} \sum_{j=1}^{m} \left\{ \hat{\boldsymbol{u}}_{j,(\texttt{it})} \hat{\boldsymbol{u}}_{j,(\texttt{it})}^\top + \left(\hat{\boldsymbol{D}}_{(\texttt{it}-1)} - \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \boldsymbol{Z}_j^\top \hat{\boldsymbol{V}}_{j,(\texttt{it}-1)}^{-1} \boldsymbol{Z}_j \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \right) \right\},$$

$$\text{where } \hat{\boldsymbol{\varepsilon}}_{j,(\texttt{it})} = \boldsymbol{y}_j^{\text{num}} - \hat{f}_{(\texttt{it})}(\boldsymbol{X}_j) - \boldsymbol{Z}_j \hat{\boldsymbol{u}}_{j,(\texttt{it})}.$$

11: $\quad$ Check convergence using GLL criterion.
12: **end while**
13: Compute numeric fixed effects predictions $\hat{f}(\boldsymbol{X}_j)$ with final ML model, $j = 1, \ldots, m$.
14: Compute numeric predictions $\hat{\boldsymbol{y}}_j^{\text{num}} = \hat{f}(\boldsymbol{X}_j) + \boldsymbol{Z}_j \hat{\boldsymbol{u}}_j, j = 1, \ldots, m$.
15: For categories up to category k, compute cumulative relative frequencies $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.
16: Obtain prediction quantiles $q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}$ of numeric predictions $\hat{y}_{ij}^{\text{num}}$ for probabilities $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1},\ i = 1, \ldots, n_j, j = 1, \ldots, m$.
17: Assign category borders $(b_1, b_2, \ldots, b_k, b_{k+1}) \leftarrow \left(-\infty, q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}, \infty\right)$.
18: **return** ML model, final random effect estimates and category borders

Framework and uses a regression XGB model as the internal ML model. Similar to fabXGB, the numeric training predictions are used for obtaining the category borders. I will showcase the mixfabXGB prototype through a small simulation study in the following.

5.4.1 Simulation Setup

The proof-of-concept simulation study for mixfabXGB was based on the simulation framework from Hajjem et al. (2012), Salditt et al. (2023) as well as the third article of this thesis (Buczak, 2024). To this end, I generated data for 100 clusters where the training data contained between 25 and 35 observations for each cluster, while the test data contained a fixed count of 25 observations per cluster. I simulated nine standard normally distributed predictors $X_1, \ldots, X_9 \sim \mathcal{N}(0, 1)$ with $\text{Cor}(X_v, X_{v'}) = 0.4 \; \forall v, v' = 1, \ldots, 9$ where $v \neq v'$ (omitting person and cluster indices for readability in this instance). The numeric outcome for person i in cluster j was simulated according to the random intercept population model

$$y_{ij}^{\text{num}} = f(\boldsymbol{x}_{ij}) + u_j + \varepsilon_{ij},$$

where $\boldsymbol{x}_{ij}$ denotes the covariate values and $f(\boldsymbol{x}_{ij})$ the fixed effects component for person i in cluster j, u_j the random intercept of cluster j and ε_{ij} the corresponding error term with $i = 1, \ldots, n_j, j = 1, \ldots, m$. The fixed effect component was generated through

$$f(\boldsymbol{x}_{ij}) = 2x_{1,ij} + x_{2,ij}^2 + 4 \cdot \mathbb{1}_{x_{3,ij}>0} + 2 \log\left(|x_{1,ij}|\right) x_{3,ij}$$

(Hajjem et al., 2011). Inspired by Salditt et al. (2023), the random intercepts $u_j, j = 1, \ldots, m$, were drawn from a normal distribution specified by with expected mean $\mu_u = 0$ and variance $\sigma_u^2 = \frac{\text{ICC}}{1-\text{ICC}}$. Values for ICC (intraclass correlation) were varied between 0.25 and 0.5 to investigate medium and large random effect variability. The error terms $\varepsilon_{ij}, i = 1, \ldots, n_j, j = 1, \ldots, m$, were drawn from a standard normal distribution. The numeric outcome was transformed into an ordinal response by binning the values into five ordinal categories. Similar to the simulation setups above, I investigated two different response category distribution patterns: a pattern with equally distributed categories and a pattern with prominent middle categories (wide middle). For more details, on the specific binning values and how these patterns were created, I refer to the third article of this thesis (Buczak, 2024). Contrary to the simulation of in Buczak (2024), the cluster indicator variable was not included as a predictor since XGB only supports numeric predictors. While workarounds such as dummy encoding would have

been possible, this would have greatly inflated the number of predictors. As the purpose of this simulation study was mainly to provide a proof-of-concept, I have therefore refrained from such workarounds to limit the computational burden. I compared mixfabXGB with a multilabel classification XGB model, fabXGB and mixfabOF. Similar to the simulation study in Section 5.2, I performed a hyperparameter tuning based on a 3-fold CV using the training data. The data were split at the cluster-level such that the folds were guaranteed to contain observations from all clusters. I considered the same hyperparameters and respective search spaces as in Section 5.2 (cf. Table 6.1 in the Appendix).

5.4.2 Simulation Results and Outlook

Since the response category distribution pattern did not notably impact the results, I am only showing results for equally distributed categories in the following. For the remaining results, I refer to Figure 6.1 the Appendix. Figure 5.3 shows that mixfabXGB could successfully improve upon XGB and fabXGB in the presence of medium (ICC $= 0.25$) and more so for high (ICC $= 0.5$) random effect variability. Compared to mixfabOF, mixfabXGB reached similar predictive performance values. These are promising findings as they indicate the widely applicable potential of extending any regression-based ML method to ordinal prediction with hierarchical data. Future work could use the Mixed-Effects Frequency-Adjusted Borders Ordinal Prediction Framework as a blueprint for developing analogous methods based on, e.g., ET and CF as considered above, or further ML methods such as support vector regression, neural networks, etc. Similarly, one could enhance the interpretability of such methods by implementing custom permutation VIMs inspired by mixfabOF's permutation VIM. Regarding mixfabXGB in particular, future work could assess its performance in more extensive simulation and real data experiments as well as investigate the impact of the hyperparameters of the underlying XGB model (cf. fabXGB's future work outlook in Section 5.2.4.

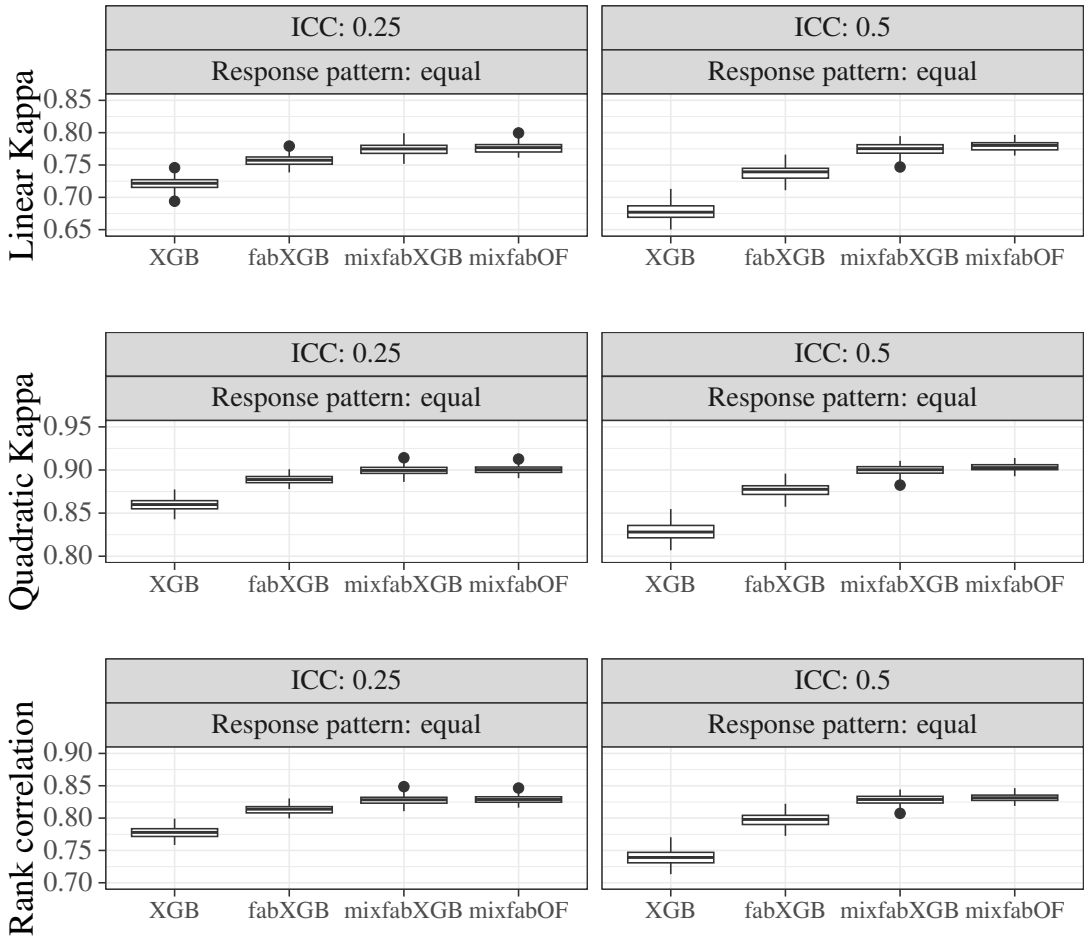

Figure 5.3: Predictive performance of mixfabXGB extension for equally distributed response categories.

5.5 Multivariate Frequency-Adjusted Borders Ordinal Forest

So far, all methods proposed in this thesis are concerned with predicting a univariate ordinal response. In some applications, however, it may be of interest to predict an outcome vector $\boldsymbol{Y} = \left(Y^{(1)}, \ldots, Y^{(d)}\right)^\top$ of dimensionality $d \geq 2$. For example, one may be interested in predicting the performance of students in multiple related subjects (e.g., mathematics and natural sciences) or the self-reported health status along several dimensions (e.g., different symptoms of depression). In such cases, it is plausible that the individual outcomes are correlated and therefore, it may be of interest to capture these dependencies through a multivariate modeling and prediction approach. In practice, multivariate prediction tasks are also often addressed by fitting d univariate models which separately predict the individual outcome components. While this represents a simple solution to the multivariate prediction problem, especially when multivariate extensions of a given method do not exist, the dependencies between the outcome components are ignored. Which of these approaches, i.e., multivariate prediction or component-wise univariate prediction, yields better predictive performance usually depends on the data. For RF-type methods, Schmid et al. (2023) compared both approaches for several data situations. In their simulation study, the authors found that for multivariate outcomes multivariate approaches mostly performed at least as well as component-wise univariate approaches and often better (Schmid et al., 2023). Only when the data generating processes differed between the components, component-wise univariate approaches outperformed multivariate approaches. For the real datasets considered by the authors, the results were more mixed (Schmid et al., 2023).

One possible way of extending fabOF to the multivariate case is sketched in Algorithm 10. This prototype will be referred to as Multivariate Frequency-Adjusted Borders Ordinal Forest (multifabOF) in the following. For each ordinal outcome component, the ordinal categories are represented by numeric scores, respectively, leading to a multivariate numeric outcome vector for each observation. Based on the numeric outcome vectors, a multivariate RF model is trained. Multivariate regression trees and RF have been studied, e.g., in De'ath (2002) and Segal and Xiao (2011). Implementations in `R` include the `randomForestSRC` (Ishwaran & Kogalur, 2024) and `MultivariateRandomForest` (Rahman, 2017) packages. The RF model is used to compute OOB predictions $\hat{\boldsymbol{y}}_i^{\text{num}}, i = 1, \ldots n$, for all observations. To determine the category borders, the frequency-adjusted borders heuristic is employed for each ordinal outcome component separately. As such, each ordinal outcome component is associated with a separate set of category borders $b_1^{(o)}, \ldots, b_{k_o+1}^{(o)}$ that can be used for obtaining predictions for new observations.

Algorithm 10 Multivariate Frequency-Adjusted Borders Ordinal Forest (multifabOF)

procedure MULTIFABOF
  Unless specified otherwise, use scores $\left(s_1^{(o)}, s_2^{(o)}, \ldots, s_{k_o}^{(o)}\right) \leftarrow (1, 2, \ldots, k_o)$.
  Create multivariate numeric target $\boldsymbol{y}_i^{\text{num}}, i = 1, \ldots, n$, based on assigned scores.
  Train multivariate regression RF using $\boldsymbol{y}_i^{\text{num}}, i = 1, \ldots, n$, as target.
  Compute numeric OOB predictions $\hat{\boldsymbol{y}}_i^{\text{num}}, i = 1, \ldots n$, using RF model.
  for o in $1, \ldots, d$ **do**
    For categories up to category k_o, compute cumulative relative frequencies $\hat{\pi}_1^{(o)}, \ldots, \hat{\pi}_{k_o-1}^{(o)}$ for outcome component o.
    Obtain prediction quantiles $q_{\hat{\pi}_1}^{(o)}, \ldots, q_{\hat{\pi}_{k_o-1}}^{(o)}$ of component predictions $\hat{y}_i^{\text{num},(o)}, i = 1, \ldots n$, for probabilities $\hat{\pi}_1^{(o)}, \ldots, \hat{\pi}_{k_o-1}^{(o)}$.
    Assign outcome component category borders
    $$\left(b_1^{(o)}, b_2^{(o)}, \ldots, b_{k_o}^{(o)}, b_{k_o+1}^{(o)}\right) \leftarrow \left(-\infty, q_{\hat{\pi}_1}^{(o)}, \ldots, q_{\hat{\pi}_{k_o-1}}^{(o)}, \infty\right).$$
  end for
  return RF model and category borders
end procedure

For future work, the proposed multifabOF prototype could be implemented and evaluated via simulation and on real data. In particular, it should be compared with a prediction approach that fits d fabOF models to each of the d ordinal response vector components. A key consideration for implementing multifabOF will likely be the selection of the splitrule of the multivariate RF model. Different distance measures have been studied to quantify the impurity of nodes for multivariate responses, e.g., Euclidean distance (Segal & Xiao, 2011), Mahalanobis distance (Larsen & Speckman, 2004), Earth Mover and Mallows distance (both D'Ambrosio et al., 2017). Future work could investigate which of these splitrules is particularly suitable for multifabOF. Additionally, future work could explore further adjustments to the frequency-adjusted borders heuristic to better reflect the multivariate nature of the outcomes considered in this section. While the multifabOF prototype relies on a multivariate RF that can take dependencies between outcome components into account, the category borders are not determined in a strictly multivariate fashion as separate category borders are computed for each ordinal component. Therefore, one could investigate how the frequency-adjusted borders heuristic can be reworked such that the category borders are also determined in a multivariate manner, i.e., simultaneously for all ordinal outcome components. Once successfully implemented, multifabOF could then be generalized to develop a model-agnostic framework for multivariate ordinal prediction in the spirit of the frameworks

presented in Sections 5.3 and 5.4.

Nevertheless, the component-wise computation of category borders may lend itself for adaptation in the context of mixed-outcome prediction where only $d' < d$ components of the vector-valued outcome are ordinal while the remaining $d - d'$ components are of another type (e.g., numeric, categorical). Mixed-outcome prediction tasks have been investigated, e.g., in Dine et al. (2009) and Saha et al. (2017). In the case where the remaining outcome components are numeric for example, one could modify Algorithm 10 such that after fitting the multivariate RF model (lines 6-10 in pseudocode), category borders are only determined for the d' ordinal components. For predicting new observations, the numeric prediction vector obtained from the multivariate RF model is handled such that only the corresponding d' ordinal components are transformed back into the respective ordinal categories through their individual set of category borders, respectively. For more complex outcome combinations, future work could explore whether existing approaches for mixed-outcome prediction such as Dine et al. (2009) could be adapted for multifabOF.

6 Discussion

The prediction of ordinal responses is an ever-occurring task in the life and social sciences. Recent years saw the evolvement of a novel methodological stream of ordinal prediction methods based on ML (e.g., Hornung, 2019; Janitza et al., 2016; Tutz, 2021) which complements the more traditional stream of parametric models such as the proportional odds model (McCullagh, 1980). The present thesis consisting of three articles contributed to the literature on ordinal prediction by introducing three novel tree-based prediction methods: the Ordinal Score Optimization Algorithm (OSOA; Buczak et al., 2024), Frequency-Adjusted Borders Ordinal Forest (fabOF; Buczak, 2025) and Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF; Buczak, 2024). Further, the first article of this thesis (Buczak et al., 2024) filled the need for an extensive comparison study of tree ensemble methods for ordinal prediction and a parametric model based on simulation and real data. All three methods proposed in the articles of this thesis follow the underlying idea of Ordinal Forest (OF; Hornung, 2019) of representing the ordinal response categories through category-specific intervals and numeric category scores. Using the numeric scores as a proxy outcome, a regression RF model is fit whose (numeric) predictions can be translated back to ordinal response categories via the category borders. While OF takes the ordinal nature of the responses better into account than multi-label classification RF, it is also characterized by an extensive optimization procedure for deriving suitable category borders and scores as well as a rigid link between category borders and scores. The methods proposed in the three articles of this thesis operate within the same framework as OF, but aim to progressively improve upon its limitations. Whereas OSOA modifies OF's optimization procedure through a non-linear optimization algorithm, fabOF removes the need for an extensive optimization altogether and decouples the category borders and scores for more flexibility. Finally, mixfabOF extends fabOF to ordinal prediction with hierarchical data. I will discuss each of these methods as well as the generalizations and further extensions additionally developed over the course of this thesis in more detail in the following.

6.1 Ordinal Score Optimization Algorithm and Comparison of Tree Ensembles with Parametric Model

The OSOA method proposed in the first article of this thesis (Buczak et al., 2024) modified OF's optimization procedure. Whereas the original optimization procedure of OF first randomly generates a large number of category border sets and evaluates the predictive performance achieved when using them. While OF tries to include diverse category border candidates, the optimization procedure cannot react to the predictive performance achieved by the individual category border sets and can only determine which of the pre-generated candidates performed best. By employing a non-linear optimization algorithm instead, the aim of OSOA was to modify the optimization procedure of OF such that the optimizer could explore the solution space and focus on promising regions. Having hoped that OSOA would consequently find better solutions which in turn yield better predictive performance, we instead found that OF and OSOA achieved mostly similar performance (Buczak et al., 2024). Furthermore, the benefit of the extensive optimization procedures in OF and OSOA was found to be situational as they could not consistently outperform a naive OF model which relied on the numeric scores $1, 2, \ldots, k$ with category borders $0.5, 1.5, \ldots, k + 0.5$. A possible reason for the lacking benefit of the score optimization could be the prediction scheme of OF and OSOA which already transforms the numeric predictions into ordinal predictions at the tree-level and then aggregates the ordinal predictions at the forest-level. This may have limited the impact of the numeric score choices if the terminal nodes in the underlying regression trees were mostly pure, i.e., containing almost only observations from the same ordinal category and thus with the same numeric score (Buczak et al., 2024). As a remedy, An alternative prediction scheme was studied in the second article of this thesis (Buczak, 2025). Another reason of the situational optimization benefit may lie in the optimization criterion used for assessing the predictive performance of the candidate category border sets. OF and OSOA both optimize Youden's J (Youden, 1950) which only takes the specificity and sensitivity of the classification behavior per response category into account. As such, it is not a truly ordinal criterion. In particular, it does not award category predictions closer to the true category with higher performance than category predictions far from the true category. Therefore, future work should explore the use of other optimization criteria which take the ordinality of the response into account (Buczak et al., 2024).

Apart from proposing OSOA, the first article further contributed to the literature through an encompassing comparison study of tree-based ensemble methods and a proportional odds model for ordinal prediction. To this end, different data generating processes with

varying compositions of linear and non-linear effects as well as varying response category distribution patterns were employed. While tree ensemble methods outperformed the proportional odds model for strong non-linear effects and for large sample sizes as expected, the proportional odds model was also found to be quite competitive for small non-linear effects and small sample sizes. For linear effects only, the proportional odds model notably outperformed the tree ensemble methods as expected. The insights from our comparison also allowed us to contribute general guidelines for approaching ordinal prediction in practice (particularly when computational resources are limited) that we hope prove useful for practitioners and researchers alike (Buczak et al., 2024).

6.2 Frequency-Adjusted Borders Ordinal Forest

The first article of this thesis indicated the need for further methodological development and additionally laid out which aspects required further attention, particularly regarding the prediction scheme used in OF and OSOA. Building on these insights, the second article of this thesis (Buczak, 2025) proposed fabOF which differs from OF and OSOA as follows. First, it introduces an alternative prediction scheme. While OF and OSOA obtained predictions for new observations by transforming the numeric predictions of the internal RF model into ordinal response category predictions at the tree level and aggregating them into a combined category prediction at the forest level, fabOF reverses the order of transformation and aggregation. As such, the numeric predictions from the internal regression RF model are first aggregated to obtain a combined numeric prediction which is transformed into an ordinal response category at the forest level. The former approach was referred to transform-first-aggregate-after (TFAA) prediction, while the latter was referred to aggregate-first-transform-after (AFTA) prediction (Buczak, 2025).

Apart from changing the prediction scheme, fabOF further decoupled the category borders and numeric scores. Whereas OF and OSOA linked the borders and scores such that the category scores were always selected as the midpoints of the category intervals, respectively, fabOF seperates borders and scores. This allows for more flexible modeling of the category borders and scores (Buczak, 2025). Instead of employing a computationally expensive optimization procedure for determining its category borders and scores, fabOF relies on a frequency-based heuristic. To this end, default scores of $1, 2, \ldots, k$ are assigned to the k ordinal categories, and used to fit a regression RF model. The category borders are chosen as the quantiles of the numeric OOB predictions obtained from the RF model for probabilities corresponding to the cumulative relative frequencies of the ordinal response categories in the training data (Buczak, 2025).

Through simulation and an illustrative data example, I could demonstrate that fabOF can improve upon existing ordinal prediction methods in predictive performance as well as computational runtime. The runtime advantage over methods such as Split-based Ordinal Forest (Tutz, 2021) which fits $k - 1$ RF models as well as OF and OSOA which perform an expensive optimization is explained by the fact that fabOF only needs to fit a single RF model. Regarding the performance advantage, a contributing factor is the replacement of TFAA prediction through AFTA prediction. When similarly using OF with AFTA instead of TFAA prediction, I observed consistent improvement in predictive performance (Buczak, 2025). Perhaps one explanation for this is that aggregating the numeric predictions instead of transforming them at the tree-level already helps in discerning finer differences between individual observations. From a methodological perspective, the numeric forest-level prediction may also be associated with less variability than the individual numeric tree-level predictions. As such, it could prove beneficial to not transform the numeric predictions into ordinal categories until the forest-level. The simulation and real data experiments also revealed that fabOF's frequency-adjusted borders heuristic contributes to the observed performance advantages. This could be explained by the fact that fabOF's category borders are directly informed by the empirical distribution of the ordinal response categories. As such, the heuristic proved particularly useful for settings in which the response categories were not distributed equally (Buczak, 2025).

6.3 Mixed-Effects Frequency-Adjusted Borders Ordinal Forest

In the third article of this thesis (Buczak, 2024), I extended fabOF to the case of ordinal prediction for hierarchical data, i.e., when observations are nested within clusters. As hierarchical data structures are a common occurrence in the social and life sciences, developing a suitable ordinal prediction based on fabOF promised the potential of substantial applicability. The proposed mixfabOF method combines the Mixed-Effects Random Forest (MERF; Hajjem et al., 2012) approach with the logic of fabOF. To this end, the ordinal response categories are represented by the numeric default scores $1, \ldots, k$ and used within an Expectation Maximization (EM; see Dempster et al., 1977) estimation procedure. The procedure alternates between estimating the fixed effects component (through a regression RF) and the random effects component while assuming the other as known, respectively. Based on the final RF fit and the final random effects estimates, numeric predictions for the training data are computed and used to obtain category borders in a similar manner as in fabOF (Buczak, 2024).

Through simulated and real data, I demonstrated that mixfabOF could improve upon

fabOF and other methods for (hierarchical) ordinal prediction in the presence of medium and strong random effect variability. This finding revealed the benefit of exploiting the information contained in the hierarchical structure of such data for ordinal prediction (Buczak, 2024). Since the frequency-adjusted borders heuristic of fabOF could be combined with MERF in a quite modular fashion, the third article also demonstrated the versatility of fabOF's heuristic. Future work could study the performance of mixfabOF for more complex random effect structures than the random intercept model considered in the third article of this thesis. For example, a natural choice would be to use a random slope model as well. While mixfabOF should likely not run into issues when including random slopes, the underlying LMM does not allow for random effect structures that are unique to ordinal regression models such as cluster-specific thresholds as described in Tutz and Hennevogl (1996). To capture the spirit of such effect structures, one could explore allowing for cluster-specific category borders (Buczak, 2024). In the context of raters, for example, such an extension could potentially discover if two raters have different latent thresholds for assigning certain ratings as the same numeric prediction could be transformed into different ordinal categories based on the rater-specific category borders. However, the challenge of such an extension would be the number of observations required to estimate the category borders sufficiently well. Longitudinal studies which often only include a limited amount of observations per person may contain too few observations for profiting from such an extension. Thus, cluster-specific category borders may perhaps be more useful in settings with large clusters such as ecological momentary assessment (EMA) studies (see Shiffman et al., 2008, for an introduction to EMA). Such studies collect continuous assessments (e.g., hourly) of persons over a period of time. Therefore, EMA studies could be a viable context for exploring the modeling of category-specific category borders in future work.

6.4 Computational Implementation and Interpretability

To make fabOF and mixfabOF available for researchers and practitioners, I have implemented both in the `R` package `fabOF` which is available from GitHub (https://github.com/phibuc/fabOF). The package currently contains functionality for fitting both models, computing predictions and displaying basic information about model objects (via the print function). Additionally, I have implemented permutation variable importance measures (VIMs) for both methods based on Cohen's weighted Kappa with linear weights (Cohen, 1968). As interpretability is often of crucial interest, e.g., in psychological fields, variable importance can help amending the lack of interpretability of RF models (Henninger et al., 2023). The evaluation of fabOF's VIM in the second article

of this thesis (Buczak, 2025) revealed satisfying performance as it consistently discovered influential predictors over noise predictors. A similar evaluation of mixfabOF's VIM has not been performed yet, but is planned for the future. For mixfabOF's VIM, I have included the possibility of permuting predictor values only within the same cluster. The goal of this feature was to better reflect the hierarchical data structure. To the best of my knowledge, a similar approach has not been employed yet for hierarchical RF extensions. For the illustrative data example of the third article, I did not find notable differences between regular (i.e., unrestricted) and clusterwise permutation. Future work could compare the two permutation approaches in more detail. It appears plausible that differences between the two could emerge, e.g., in the presence of random slope effects, i.e., when there are cluster-specific effects on the predictors instead of the intercept only.

Future work could also consider enhancing the interpretability of fabOF and mixfabOF further. The VIMs implemented for fabOF and mixfabOF are both unconditional permutation VIMs, i.e., they do not place restrictions on the internal permutations process. In the case of highly correlated predictors, however, unconditional permutation VIMs can be biased (Strobl et al., 2008). As a remedy, Strobl et al. (2008) proposed conditional permutation VIMs which aim to preserve the original correlation structure of the predictors by permuting only within certain subspaces of the predictor space. As data from the life and social sciences commonly contain correlated predictors, the development of conditional permutation VIMs for fabOF and mixfabOF could be a goal of future work. As an alternative to conditional permutation VIMs, one could also consider, e.g., the Conditional Predictive Impact (CPI) as proposed in Watson and Wright (2021). The CPI allows for conditional independence testing by measuring the contribution of predictors to the predictive performance in relation to a complementary predictor subset (termed "knockoff" data) which acts as a sort of control group (Watson & Wright, 2021). This could aid in assessing the impact of individual predictors as well as adding means of uncertainty quantification for the methods proposed in this thesis (see also Section 6.6).

6.5 Methodological Extensions

The modular nature of the frequency-adjusted borders heuristic employed in fabOF did not only allow extending fabOF (Buczak, 2025) to hierarchical data through mixfabOF (Buczak, 2024), but further laid the foundation for developing more general frameworks for ordinal prediction. To this end, I introduced the Frequency-Adjusted Borders Ordinal Prediction Framework as well as the Mixed-Effects Frequency-Adjusted Borders

Ordinal Prediction Framework. Both frameworks generalize fabOF and mixfabOF to allow for replacing the internal regression RF model with an arbitrary regression-based ML method, respectively. As such, these frameworks offer a blueprint for quickly developing a wide array of ordinal prediction methods in the future. It is a well-known paradigm of machine learning that no method performs best at all tasks ("no free lunch"; Wolpert & Macready, 1997) and that the optimal model is always data-dependent. To this end, the possibility of specifying the internal ML model at will allows for tailoring the prediction model to best meet the requirements of a given application. Regarding fabOF, I introduced three prototypes based on Extra-Trees (ET; Geurts et al., 2006), Conditional Inference Forest (CF; Hothorn et al., 2006) and XGBoost (XGB; Chen & Guestrin, 2016) that each addressed different dimensions one may be interested in: balancing the bias/variance trade-off, avoiding variable selection bias or potential increases in predictive power (depending on the data at hand). To showcase these prototypes, I performed a small simulation study in which the three prototypes outperformed their original non-ordinal counterpart and their naive regression variants using scores $1, 2, \ldots, k$ with category borders $0.5, 1.5, \ldots, k + 0.5$. While more thorough evaluation is needed in future work, these early results indicate the powerful potential of the Frequency-Adjusted Borders Ordinal Prediction Framework.

Additionally, I have implemented a prototype for Mixed-Effects Frequency-Adjusted Borders Ordinal Prediction Framework based on XGB. The results revealed that the prototype could achieve higher predictive performance in the presence of medium and strong random effect variability when compared to non-hierarchical counterparts. These promising findings invite further research into the Mixed-Effects Frequency-Adjusted Borders XGB prototype as well as the development of other ordinal prediction methods in the same vein. Such developments could equip researchers and practitioners with a larger repertoire of viable tools for ordinal prediction of hierarchical data. In educational fields, for example, improving the capabilities of data-driven prediction could have a beneficial impact on informing policy design and implementing student support structures (Costa-Mendes et al., 2020; van der Scheer & Visscher, 2017).

Finally, I have sketched an extension of fabOF to the multivariate case, i.e., for vector-valued outcomes. Developing such an extension in future work still needs to overcome a few challanges, e.g., the selection of the splitrule for the multivariate RF model, how category borders can be determined in a multivariate fashion instead of separately for each outcome component or adapting the extension for mixed-outcome prediction. However, it is certainly worth pursuing as appropriate methodology for multivariate ordinal prediction can help taking the full information of multivariate data, i.e., potential dependencies between the individual outcome components into account. Depending on

the data at hand, this can yield increases in predictive performance (see, e.g., Schmid et al., 2023).

6.6 Limitations

While the methodology presented in this thesis displayed promising predictive performance, its current lack of uncertainty quantification represents a limitation that needs to be addressed in the future. One possibility to quantify uncertainty is through the computation of prediction intervals as have been studied for RF, e.g., in Zhang et al. (2019) and Ramosaj (2021). For adapting such prediction intervals to (mix)fabOF, one could explore computing numeric prediction intervals for the internal regression RF model and using the category borders to translate the prediction interval into a set of ordinal response categories that meet the targeted uncertainty level. Alternatively, one could investigate the use of conformal predictions (see, e.g., Vovk et al., 2022). Further, all methods presented in the main articles of this thesis rely on RF. While tree-based methods such as RF can be successful for many prediction tasks (see Fernández-Delgado et al., 2014; Grinsztajn et al., 2022), the first article of this thesis has also revealed that for effect structures mainly characterized by linear effects, parametric models such as the proportional odds model can often outperform RF-based approaches. This is especially likely for smaller datasets as are commonly encountered in psychology and the social sciences (see, e.g, Shen et al., 2011). Regarding the evaluation of the methods in this thesis, the simulated datasets were mainly generated from proportional odds models or from binning linear regression outcomes. While these are common approaches for simulating ordinal data (see, e.g., Hornung, 2019; Janitza et al., 2016), alternative data generating models, e.g., including category-specific effects such as partial proportional odds models (Peterson & Harrell, 1990) or further types of heterogeneity as in location-scale (McCullagh, 1980) or location-shift (Tutz & Berger, 2022) models, should also be investigated in future work. Furthermore, the main articles of this thesis did not include a hyperparameter tuning for the RF-based prediction methods. While this is in line with previous research from the field (see Hornung, 2019; Janitza et al., 2016; Tutz, 2021) and RF is known to be relatively robust regarding hyperparameter choices (Probst et al., 2019), a hyperparameter tuning should generally be preferred (if the computational resources permit) in order to obtain optimal predictive performance. However, combining the optimization approaches in methods such as OF, OSOA and OMERF with an additional hyperparameter tuning would have greatly increased the computational complexity. Due to the smaller number of replications, this may have been less severe for the real data experiments. Nonetheless, I have still refrained from a

hyperparameter tuning for the real datasets for reasons of consistency. In practice, the benefit of a hyperparameter tuning will depend on the data at hand. For example, the RF-based methods from the first article of this thesis likely would have benefitted from tuning `mtry` in simulation settings with many noise predictors. In the case of the additional results obtained when evaluating the XGB-based extension of fabOF (for which a hyperparameter tuning of fabOF was performed as well), the impact of the hyperparameter tuning for fabOF was limited. A further limitation of this thesis is that it only considered individual prediction methods. In an effort to combine the capabilities of different model classes (e.g., parametric models and RF-based methods), Tutz (2021) proposed creating joint ensembles of multiple prediction methods which are trained separately and whose predicted category probabilities are aggregated according to a weighting scheme that rewards high predictive performance for the given data. As such ensembles have several degrees of freedom (e.g., regarding the choice of learners or the computation of weights) and since fabOF, mixfabOF and all their extensions were not designed to output predicted probabilities for the ordinal categories, ensemble learners in the spirit of Tutz (2021) were not considered in this thesis.

6.7 Outlook and Conclusion

The presented extensions have indicated the versatility of the frequency-border heuristic and its potential for wide ranging application. While the thesis and its included articles focused on employing the heuristic in contexts where predictive performance was the main goal, further application contexts are conceivable as well. For example, RF is also used for imputing missing values, e.g., in missForest (Stekhoven & Bühlmann, 2011) and MICE Random Forest (van Buuren, 2018). Broadly speaking, these methods employ RF for predicting the missing values of a predictor based on the remaining predictors. Both have been studied in more detail and compared to other imputation methods, e.g., in Thurow et al. (2021), Ramosaj et al. (2022) and Buczak et al. (2023). In future work, one could explore replacing classic RF models with fabOF for imputing missing values in ordinal predictors. Another field for which an application of the frequency-adjusted borders heuristic of fabOF could be explored and is already planned by Pauly and colleagues is causal ML. Causal ML methods such as causal forest (Wager & Athey, 2018) aim to estimate heterogeneous treatment effects within the potential outcomes framework (Rubin, 1974). Common use cases include, e.g., assessing the effect of a medical treatment or an educational intervention, and how it potentially differs for certain subgroups of the population. Typically, however, treatment effects are estimated for continuous outcomes using the (conditional) average treatment effect

as an estimand which is not meaningful for ordinal outcomes (Volfovsky et al., 2015). Volfovsky et al. (2015) discuss possible estimands for ordinal outcomes on the observed and latent scale. Future work could explore how they can be leveraged to develop an ordinal causal forest model.

To conclude, this thesis along with its three articles contributed to the literature on ordinal prediction with three RF-based prediction methods and an encompassing comparison study of tree ensemble methods and a proportional odds model (Buczak, 2024, 2025; Buczak et al., 2024). Two of the proposed methods have displayed particularly promising predictive performance (Buczak, 2024, 2025) and were shown to be extendable to a wide range of other classic ML algorithms in this thesis. Combined with the many avenues for further research sketched above, the present thesis demonstrated that there remains much untapped potential the field of ordinal prediction. The thesis further provided guidance on how some of this potential can be unlocked, and it additionally provided tools in the form of two general ordinal prediction frameworks that can directly advance the field in the near feature.

Bibliography

Agresti, A. (2010). *Analysis of ordinal categorical data* (2nd ed.). Hoboken, NJ: Wiley. https://doi.org/10.1002/9780470594001

Archer, K. J. (2010). rpartOrdinal: An R package for deriving a classification tree for predicting an ordinal response. *Journal of Statistical Software*, *34*(7), 1–17. https://doi.org/10.18637/jss.v034.i07

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Ben-David, A. (2008). Comparison of classification accuracy using Cohen's weighted kappa. *Expert Systems with Applications*, *34*(2), 825–832. https://doi.org/10.1016/j.eswa.2006.10.022

Bergonzoli, G., Rossi, L., & Masci, C. (2024). Ordinal mixed-effects random forest [pre-print version 1]. https://doi.org/10.48550/ARXIV.2406.03130

Brant, R. (1990). Assessing proportionality in the proportional odds model for ordinal logistic regression. *Biometrics*, *46*(4), 1171–1178. https://doi.org/10.2307/2532457

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 123–140. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. New York, NY: Chapman and Hall/CRC. https://doi.org/10.1201/9781315139470

Buczak, P. (2024). Mixed-effects frequency-adjusted borders ordinal forest: A tree ensemble method for ordinal prediction with hierarchical data [pre-print version 1.1]. https://doi.org/10.31219/osf.io/ny6we

Buczak, P. (2025). Frequency-adjusted borders ordinal forest: A novel tree ensemble method for ordinal prediction. *British Journal of Mathematical and Statistical Psychology*, *78*(2), 594–616. https://doi.org/10.1111/bmsp.12375

Buczak, P., Chen, J.-J., & Pauly, M. (2023). Analyzing the effect of imputation on classification performance under MCAR and MAR missing mechanisms. *Entropy*, *25*(3), 521. https://doi.org/10.3390/e25030521

Buczak, P., Horn, D., & Pauly, M. (2024). Old but gold or new and shiny? Comparing tree ensembles for ordinal prediction with a classic parametric approach. *Journal of Classification (Advance Online Publication)*, 1–27. https://doi.org/10.1007/s00357-024-09497-9

Cao, W., Mirjalili, V., & Raschka, S. (2020). Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, *140*, 325–331. https://doi.org/10.1016/j.patrec.2020.11.008

Capitaine, L., Genuer, R., & Thiébaut, R. (2020). Random forests for high-dimensional longitudinal data. *Statistical Methods in Medical Research*, *30*(1), 166–184. https://doi.org/10.1177/0962280220946080

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y., & Yuan, J. (2024). *xgboost: Extreme gradient boosting* [R package version 1.7.8.1]. https://CRAN.R-project.org/package=xgboost

Cheng, J., Wang, Z., & Pollastri, G. (2008). A neural network approach to ordinal regression. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 1279–1284. https://doi.org/10.1109/ijcnn.2008.4633963

Christensen, R. H. B. (2022). *ordinal: Regression models for ordinal data* [R package version 2022.11-16]. https://CRAN.R-project.org/package=ordinal

Chu, W., & Keerthi, S. S. (2007). Support vector ordinal regression. *Neural Computation*, *19*(3), 792–815. https://doi.org/10.1162/neco.2007.19.3.792

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, *70*(4), 213–220. https://doi.org/10.1037/h0026256

Cortez, P., & Silva, A. M. G. (2008). Using data mining to predict secondary school student performance, 1–8. https://api.semanticscholar.org/CorpusID:16621299

Cortez, P. (2014). Student performance [Dataset]. Retrieved from https://archive.ics.uci.edu/dataset/320/student+performance.

Costa-Mendes, R., Oliveira, T., Castelli, M., & Cruz-Jesus, F. (2020). A machine learning approximation of the 2015 Portuguese high school student grades: A hybrid approach. *Education and Information Technologies*, *26*(2), 1527–1547. https://doi.org/10.1007/s10639-020-10316-y

D'Ambrosio, A., Aria, M., Iorio, C., & Siciliano, R. (2017). Regression trees for multivalued numerical response variables. *Expert Systems with Applications*, *69*, 21–28. https://doi.org/10.1016/j.eswa.2016.10.021

De'ath, G. (2002). Multivariate regression trees: A new technique for modeling species–environment relationships. *Ecology*, *83*(4), 1105–1117. https://doi.org/10.1890/0012-9658(2002)083[1105:mrtant]2.0.co;2

Debeer, D., & Strobl, C. (2020). Conditional permutation importance revisited. *BMC Bioinformatics*, *21*(1), 307. https://doi.org/10.1186/s12859-020-03622-2

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Dine, A., Larocque, D., & Bellavance, F. (2009). Multivariate trees for mixed outcomes. *Computational Statistics & Data Analysis*, *53*(11), 3795–3804. https://doi.org/10.1016/j.csda.2009.04.003

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. D. (2021). *Regression: Models, methods and applications* (2nd ed.). Berlin, Germany: Springer. https://doi.org/10.1007/978-3-662-63882-8

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, *15*(90), 3133–3181.

Fishbein, B., Foy, P., & Yin, L. (2021). TIMSS 2019 user guide for the international database. Retrieved from. https://timssandpirls.bc.edu/timss2019/international-database/

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, *20*(177), 1–81.

Fontana, L., Masci, C., Ieva, F., & Paganoni, A. M. (2021). Performing learning analytics via generalised mixed-effects trees. *Data*, *6*(7), 74. https://doi.org/10.3390/data6070074

Frank, E., & Hall, M. (2001). A simple approach to ordinal classification. In L. De Raedt & P. Flach (Eds.), *Lecture Notes in Computer Science: Vol. 2167. Machine learning: ECML 2001* (pp. 145–156). Berlin, Germany: Springer. https://doi.org/10.1007/3-540-44795-4_13

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232.

Friedrich, S., & Friede, T. (2024). On the role of benchmarking data sets and simulations in method comparison studies. *Biometrical Journal*, *66*(1), 2200212. https://doi.org/10.1002/bimj.202200212

Galimberti, G., Soffritti, G., & Maso, M. D. (2012). Classification trees for ordinal responses in R: The rpartScore package. *Journal of Statistical Software*, *47*(10), 1–25. https://doi.org/10.18637/jss.v047.i10

Gelman, A. (2006). Multilevel (hierarchical) modeling: What it can and cannot do. *Technometrics*, *48*(3), 432–435. https://doi.org/10.1198/004017005000000661

Gelman, A., & Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. New York, NY: Cambridge University Press. https://doi.org/10.1017/cbo9780511790942

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems 35 (NeurIPS 2022) - Track on datasets and benchmarks*.

Hajjem, A., Bellavance, F., & Larocque, D. (2011). Mixed effects regression trees for clustered data. *Statistics & Probability Letters*, *81*(4), 451–459. https://doi.org/10.1016/j.spl.2010.12.003

Hajjem, A., Bellavance, F., & Larocque, D. (2012). Mixed-effects random forest for clustered data. *Journal of Statistical Computation and Simulation*, *84*(6), 1313–1328. https://doi.org/10.1080/00949655.2012.741599

Hajjem, A., Larocque, D., & Bellavance, F. (2017). Generalized mixed effects regression trees. *Statistics & Probability Letters*, *126*, 114–118. https://doi.org/10.1016/j.spl.2017.02.033

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.). New York, NY: Springer. https://doi.org/10.1007/978-0-387-84858-7

Hedeker, D., & Gibbons, R. D. (1994). A random-effects ordinal regression model for multilevel analysis. *Biometrics*, *50*(4), 933. https://doi.org/10.2307/2533433

Henninger, M., Debelak, R., Rothacher, Y., & Strobl, C. (2023). Interpretable machine learning for psychological research: Opportunities and pitfalls. *Psychological Methods (Advance Online Publication)*. https://doi.org/10.1037/met0000560

Herbrich, R., Graepel, T., & Obermayer, K. (1999). Support vector learning for ordinal regression. *9th International Conference on Artificial Neural Networks: ICANN '99*, 97–102. https://doi.org/10.1049/cp:19991091

Hilbert, S., Coors, S., Kraus, E., Bischl, B., Lindl, A., Frei, M., Wild, J., Krauss, S., Goretzko, D., & Stachl, C. (2021). Machine learning for the educational sciences. *Review of Education*, *9*(3), e3310. https://doi.org/10.1002/rev3.3310

Hornung, R. (2019). Ordinal forests. *Journal of Classification*, *37*(1), 4–17. https://doi.org/10.1007/s00357-018-9302-x

Hornung, R. (2022). *ordinalForest: Ordinal forests: Prediction and variable ranking with ordinal target variables* [R package version 2.4-3]. https://CRAN.R-project.org/package=ordinalForest

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, *15*(3), 651–674. https://doi.org/10.1198/106186006X133933

Ishwaran, H., & Kogalur, U. (2024). *randomForestSRC: Fast unified random forests for survival, regression, and classification* [R package version 3.3.1]. https://cran.r-project.org/package=randomForestSRC

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R*. New York, NY: Springer. https://doi.org/10.1007/978-1-0716-1418-1

Janitza, S., Tutz, G., & Boulesteix, A.-L. (2016). Random forest for ordinal responses: Prediction and variable selection. *Computational Statistics & Data Analysis*, *96*, 57–73. https://doi.org/10.1016/j.csda.2015.10.005

Johnson, S. G. (2007). The NLopt nonlinear-optimization package [Software]. Retrieved from https://github.com/stevengj/nlopt.

Kendall, M. G. (1945). The treatment of ties in ranking problems. *Biometrika*, *33*(3), 239–251. https://doi.org/10.1093/biomet/33.3.239

Kendall, M. G. (1948). *Rank correlation methods*. London, UK: Griffin.

Kramer, S., Widmer, G., Pfahringer, B., & de Groeve, M. (2000). Prediction of ordinal classes using regression trees. In Z. W. Raś & S. Ohsuga (Eds.), *Lecture Notes in Computer Science: Vol. 1932. Foundations of Intelligent Systems. ISMIS 2000* (pp. 426–434). Berlin, Germany: Springer. https://doi.org/10.1007/3-540-39963-1_45

Larsen, D. R., & Speckman, P. L. (2004). Multivariate regression trees for analysis of abundance data. *Biometrics*, *60*(2), 543–549. https://doi.org/10.1111/j.0006-341x.2004.00202.x

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22. https://CRAN.R-project.org/doc/Rnews/

Liddell, T. M., & Kruschke, J. K. (2018). Analyzing ordinal data with metric models: What could possibly go wrong? *Journal of Experimental Social Psychology*, *79*, 328–348. https://doi.org/10.1016/j.jesp.2018.08.009

Loh, W.-Y. (2002). Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 361–386.

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society: Series B (Methodological)*, *42*(2), 109–127. https://doi.org/10.1111/j.2517-6161.1980.tb01109.x

Molnar, C. (2022). *Interpretable machine learning: A guide for making black box models explainable* (2nd ed.). https://christophm.github.io/interpretable-ml-book

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, *7*, 308–313. https://doi.org/10.1093/comjnl/7.4.308

Pellagatti, M., Masci, C., Ieva, F., & Paganoni, A. M. (2021). Generalized mixed-effects random forest: A flexible approach to predict university student dropout. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *14*(3), 241–257. https://doi.org/10.1002/sam.11505

Peterson, B., & Harrell, F. E. (1990). Partial proportional odds models for ordinal response variables. *Applied Statistics*, *39*(2), 205–217. https://doi.org/10.2307/2347760

Piccarreta, R. (2007). Classification trees for ordinal variables. *Computational Statistics*, *23*(3), 407–427. https://doi.org/10.1007/s00180-007-0077-5

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, *9*(3), 1–15. https://doi.org/10.1002/widm.1301

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann Publishers Inc.

Rahman, R. (2017). *MultivariateRandomForest: Models multivariate cases using random forests* [R package version 1.1.5]. https://cran.r-project.org/package=MultivariateRandomForest

Ramosaj, B. (2021). Interpretable machines: Constructing valid prediction intervals with random forests [pre-print version 1]. https://doi.org/10.48550/ARXIV.2103.05766

Ramosaj, B., Tulowietzki, J., & Pauly, M. (2022). On the relation between prediction and imputation accuracy under missing covariates. *Entropy*, *24*(3), 386. https://doi.org/10.3390/e24030386

Riccardi, A., Fernandez-Navarro, F., & Carloni, S. (2014). Cost-sensitive AdaBoost algorithm for ordinal regression based on extreme learning machine. *IEEE Transactions on Cybernetics*, *44*(10), 1898–1909. https://doi.org/10.1109/tcyb.2014.2299291

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and non-randomized studies. *Journal of Educational Psychology*, *66*(5), 688–701. https://doi.org/10.1037/h0037350

Saha, B., Gupta, S., Phung, D., & Venkatesh, S. (2017). A framework for mixed-type multioutcome prediction with applications in healthcare. *IEEE Journal of Biomedical and Health Informatics*, *21*(4), 1182–1191. https://doi.org/10.1109/jbhi.2017.2681799

Salditt, M., Humberg, S., & Nestler, S. (2023). Gradient tree boosting for hierarchical data. *Multivariate Behavioral Research*, *58*(5), 911–937. https://doi.org/10.1080/00273171.2022.2146638

Schmid, L., Gerharz, A., Groll, A., & Pauly, M. (2023). Tree-based ensembles for multi-output regression: Comparing multivariate approaches with separate univariate ones. *Computational Statistics & Data Analysis*, *179*, 107628. https://doi.org/10.1016/j.csda.2022.107628

Segal, M., & Xiao, Y. (2011). Multivariate random forests. *WIREs Data Mining and Knowledge Discovery*, *1*(1), 80–87. https://doi.org/10.1002/widm.12

Sela, R. J., & Simonoff, J. S. (2012). RE-EM trees: A data mining approach for longitudinal and clustered data. *Machine Learning*, *86*, 169–207. https://doi.org/10.1007/s10994-011-5258-3

Shen, W., Kiger, T. B., Davies, S. E., Rasch, R. L., Simon, K. M., & Ones, D. S. (2011). Samples in applied psychology: Over a decade of research in review. *Journal of Applied Psychology*, *96*(5), 1055–1064. https://doi.org/10.1037/a0023322

Shi, X., Cao, W., & Raschka, S. (2023). Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, *26*(3), 941–955. https://doi.org/10.1007/s10044-023-01181-9

Shiffman, S., Stone, A. A., & Hufford, M. R. (2008). Ecological momentary assessment. *Annual Review of Clinical Psychology*, *4*(1), 1–32. https://doi.org/10.1146/annurev.clinpsy.3.022806.091415

Shwartz-Ziv, R., & Armon, A. (2022). Tabular data: Deep learning is not all you need. *Information Fusion*, *81*, 84–90. https://doi.org/10.1016/j.inffus.2021.11.011

Speiser, J. L., Wolf, B. J., Chung, D., Karvellas, C. J., Koch, D. G., & Durkalski, V. L. (2018). BiMM tree: A decision tree method for modeling clustered and longitudinal binary outcomes. *Communications in Statistics - Simulation and Computation*, (4), 1004–1023. https://doi.org/10.1080/03610918.2018.1490429

Speiser, J. L., Wolf, B. J., Chung, D., Karvellas, C. J., Koch, D. G., & Durkalski, V. L. (2019). BiMM forest: A random forest method for modeling clustered and longitudinal binary outcomes. *Chemometrics and Intelligent Laboratory Systems*, *185*, 122–134. https://doi.org/10.1016/j.chemolab.2019.01.002

Stekhoven, D. J., & Bühlmann, P. (2011). Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, *28*(1), 112–118. https://doi.org/10.1093/bioinformatics/btr597

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, *9*, 307. https://doi.org/10.1186/1471-2105-9-307

Therneau, T., & Atkinson, B. (2023). *rpart: Recursive partitioning and regression trees* [R package version 4.1.23]. https://CRAN.R-project.org/package=rpart

Thurow, M., Dumpert, F., Ramosaj, B., & Pauly, M. (2021). Imputing missings in official statistics for general tasks – our vote for distributional accuracy. *Statistical Journal of the IAOS*, *37*(4), 1379–1390. https://doi.org/10.3233/sji-210798

Tutz, G., & Berger, M. (2017). Separating location and dispersion in ordinal regression models. *Econometrics and Statistics*, *2*, 131–148. https://doi.org/10.1016/j.ecosta.2016.10.002

Tutz, G. (2011). *Regression for categorical data*. New York, NY: Cambridge University Press. https://doi.org/10.1017/cbo9780511842061

Tutz, G. (2021). Ordinal trees and random forests: Score-free recursive partitioning and improved ensembles. *Journal of Classification*, *39*(2), 241–263. https://doi.org/10.1007/s00357-021-09406-4

Tutz, G. (2022). Ordinal regression: A review and a taxonomy of models. *WIREs Computational Statistics*, *14*(2), e1545. https://doi.org/10.1002/wics.1545

Tutz, G., & Berger, M. (2022). Sparser ordinal regression models based on parametric and additive location-shift approaches. *International Statistical Review*, *90*(2), 306–327. https://doi.org/10.1111/insr.12484

Tutz, G., & Hechenbichler, K. (2005). Aggregating classifiers with ordinal response structure. *Journal of Statistical Computation and Simulation*, *75*(5), 391–408. https://doi.org/10.1080/00949650410001729481

Tutz, G., & Hennevogl, W. (1996). Random effects in ordinal regression models. *Computational Statistics & Data Analysis*, *22*(5), 537–557. https://doi.org/10.1016/0167-9473(96)00004-7

Ulitzsch, E., Ulitzsch, V., He, Q., & Lüdtke, O. (2022). A machine learning-based procedure for leveraging clickstream data to investigate early predictability of failure on interactive tasks. *Behavior Research Methods*, *55*(3), 1392–1412. https://doi.org/10.3758/s13428-022-01844-1

van Buuren, S. (2018). *Flexible imputation of missing data* (2nd ed.). New York, NY: Chapman and Hall/CRC. https://doi.org/10.1201/9780429492259

van der Scheer, E. A., & Visscher, A. J. (2017). Effects of a data-based decision-making intervention for teachers on students' mathematical achievement. *Journal of Teacher Education*, *69*(3), 307–320. https://doi.org/10.1177/0022487117704170

Verbeke, G., & Molenberghs, G. (2000). *Linear mixed models for longitudinal data*. New York, NY: Springer. https://doi.org/10.1007/978-1-4419-0300-6

Volfovsky, A., Airoldi, E. M., & Rubin, D. B. (2015). Causal inference for ordinal outcomes [pre-print version 1]. https://doi.org/10.48550/ARXIV.1501.01234

Vovk, V., Gammerman, A., & Shafer, G. (2022). *Algorithmic learning in a random world* (2nd ed.). Cham, CH: Springer. https://doi.org/10.1007/978-3-031-06649-8

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*(523), 1228–1242. https://doi.org/10.1080/01621459.2017.1319839

Watson, D. S., & Wright, M. N. (2021). Testing conditional independence in supervised learning algorithms. *Machine Learning*, *110*(8), 2107–2129. https://doi.org/10.1007/s10994-021-06030-6

Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893

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. https://doi.org/10.18637/jss.v077.i01

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer*, *3*(1), 32–35. https://doi.org/10.1002/1097-0142(1950)3:1<32::aid-cncr2820030106>3.0.co;2-3

Zahid, F. M., & Tutz, G. (2013). Proportional odds models with high-dimensional data structure. *International Statistical Review*, *81*(3), 388–406. https://doi.org/10.1111/insr.12032

Zhang, H., Zimmerman, J., Nettleton, D., & Nordman, D. J. (2019). Random forest prediction intervals. *The American Statistician*, *74*(4), 392–406. https://doi.org/10.1080/00031305.2019.1585288

Part II

Articles

Article 1

Buczak, P., Horn, D., & Pauly, M. (2024). Old but gold or new and shiny? Comparing tree ensembles for ordinal prediction with a classic parametric approach. *Journal of Classification (Advance Online Publication)*, 1–27.
https://doi.org/10.1007/s00357-024-09497-9

Journal of Classification
https://doi.org/10.1007/s00357-024-09497-9

ORIGINAL RESEARCH

Old but Gold or New and Shiny? Comparing Tree Ensembles for Ordinal Prediction with a Classic Parametric Approach

Philip Buczak[1,2] · **Daniel Horn**[1] · **Markus Pauly**[1,2]

Accepted: 20 November 2024

Abstract
Ordinal data are frequently encountered, e.g., in the life and social sciences. Predicting ordinal outcomes can inform important decisions, e.g., in medicine or education. Two methodological streams tackle prediction of ordinal outcomes: Traditional parametric models, e.g., the proportional odds model (POM), and machine learning-based tree ensemble (TE) methods. A promising TE approach involves selecting the best performing from sets of randomly generated numeric scores assigned to ordinal response categories (ordinal forest; Hornung, 2019). We propose a new method, the ordinal score optimization algorithm, that takes a similar approach but selects scores through non-linear optimization. We compare these and other TE methods with the computationally much less expensive POM. Despite selective efforts, the literature lacks an encompassing simulation-based comparison. Aiming to fill this gap, we find that while TE approaches outperform the POM for strong non-linear effects, the latter is competitive for small sample sizes even under medium non-linear effects.

Keywords Ordinal prediction · Proportional odds model · Random forest · Score optimization

1 Introduction

Ordinal response data are often encountered in biomedical and psychological applications, e.g., when assessing persons' health status, rating of a set of choices or agreement towards given statements. Despite the common assignment of numeric values, ordinal responses are categorical variables whose categories are not necessarily equidistant, but (in contrast to nominal responses) carry a natural order. Typically, ordinal responses have been modeled through parametric models such as the cumulative model (particularly through its special

Philip Buczak
buczak@statistik.tu-dortmund.de

Daniel Horn
dhorn@statistik.tu-dortmund.de

Markus Pauly
pauly@statistik.tu-dortmund.de

1 Department of Statistics, TU Dortmund University, Dortmund 44227, Germany

2 Research Center Trustworthy Data Science and Security, UA Ruhr, Dortmund 44227, Germany

Published online: 09 December 2024

case, the proportional odds model). For a general overview of the cumulative and further parametric models, we refer to Tutz (2022).

Apart from parametric models, there has also been an increasing use of non-parametric machine learning (ML) methods based on recursive partitioning, e.g., classification and regression trees (CART; Breiman et al., 1984) as well as ensemble methods such as random forest (RF; Breiman, 2001). While not accounting for the nature of ordinal responses inherently, several variations of trees and RF tailored towards ordinal prediction have been proposed. Piccarreta (2007) extended the Gini-Simpson split criterion to the ordinal case, while Archer (2010) and Galimberti et al. (2012) used the generalized Gini impurity with misclassification costs to incentivize respecting the ordered structure of the response. Buri and Hothorn (2020) proposed model-based RFs for detecting changes in proportional odds. Several further approaches are based on assigning numeric scores to the ordinal response categories, e.g., Kramer et al. (2000) used regression trees based on the numeric scores, while Janitza et al. (2016) used conditional inference forest (CF; Hothorn et al., 2006). Despite also using numeric scores, Hornung (2019) avoids pre-specification of scores in their ordinal forest (OF) by using a two-stage approach based on regression RF where in the first step the numeric scores to be assigned are optimized w.r.t. the predictive performance achieved when using them.

Another possibility of avoiding the predicament of score assignment is re-formulating the ordinal prediction task as a series of binary prediction tasks and aggregating the predictions of the individual binary models into a combined prediction for the ordinal prediction task (Frank & Hall, 2001). Similarly, Tutz (2021) proposed split-based ordinal RF (RFSp), a score-free framework for using classifiers such as RF in a binarized fashion aimed at resembling parametric models.

As such, one can identify two streams of methodology for ordinal classification: traditional parametric models, such as the cumulative model, and the adaptation of modern tree ensemble (TE) methods that have displayed high predictive power in other application contexts (see, e.g., Grinsztajn et al., 2022), but are also characterized by an increased computational complexity. While the recent literature on ordinal classification focused increasingly on the latter stream, the attention attributed to parametric models has slightly fallen off as they are often overlooked when evaluating ML-based prediction approaches (Tutz, 2021). This shortcoming in the literature impedes the development of recommendations for researchers and practitioners as to which method may be preferable for their given prediction application. While the works mentioned above benchmarked their methods to some capacity, there is a lack of an encompassing simulation-based comparison of the ordinal TE methods with parametric methods in different scenarios to assess under which circumstances using TE methods over a parametric model leads to improved predictive performance and, as such, is worth the increased computational complexity. Furthermore, there is no extensive comparison of the individual TE methods available in the current literature that would help obtaining more general guidelines among the set of TE methods. In particular, it would be of interest to study whether the approach of employing a computationally demanding score optimization procedure as in OF justifies its computation costs when compared to other TE methods such as regression RFs with non-optimized scores, classification RF, CF, and RFSp.

The aspects above have only been partially covered in the existing literature on ordinal classification. Janitza et al. (2016) only compared score-based CF with a regular classification RF. They did not find notable performance differences in their study with simulated and real

datasets. Furthermore, they also did not identify a difference regarding the choice of numeric scores. However, they have only compared the default scores $1, 2, \ldots, k$ for k categories with the squared scores $1^2, 2^2, \ldots, k^2$. Hornung (2019) compared their OF with a naive OF variant using only default scores, classical RF, and a cumulative model (with probit link) on five real datasets. The parametric model performed best for one dataset, while it lagged severely behind for two datasets and was competitive for the remaining two datasets. Regarding the TE methods, for two out of the five datasets, OF could outperform naive OF and classical RF, while for the other three datasets, the three TE methods performed similarly. In an additional simulation study, Hornung (2019) further compared OF, naive OF, and RF, however, without including a parametric model. Regarding the benefit of optimizing scores, they found that OF could improve the most over naive OF in scenarios where the middle categories were distinctly more populated than the margin categories. The most complete comparison, to our knowledge, was performed by Tutz (2021) who compared parametric models, RFSp, OF, CF as well as ensembles comprised of combinations of these methods. In their study, the author found that performance gaps usually only occurred between the group of parametric models and the group of TE methods, while the different TE approaches performed similarly when compared only among themselves. However, these comparisons were only conducted on real datasets where one cannot directly control and manipulate the data generating processes. This makes deducting more general recommendations difficult because the actual effect structure present in the data is unknown. As such, it is generally recommended to use both simulation and real data for evaluating and comparing different methods (see, e.g., Friedrich & Friede, 2024).

We contribute to the literature above in two ways. First, we perform an extensive simulation study including differing data generating processes with increasingly non-linear effects and varying class distribution patterns to obtain recommendations for researchers and practitioners as to when a parametric model or TE methods are preferable. Further, we study whether optimizing numeric scores within a RF framework is worth the associated computational burden. To our knowledge, OF is currently the only method that optimizes its numeric scores. However, its optimization procedure separates the generation of score sets from their evaluation, i.e., all candidate score sets are first generated and evaluated afterwards. This means that the optimization procedure cannot react to the performance of any given score set, whereas iterative optimization approaches could take the performance of previously evaluated candidate score sets into account and specifically focus on exploring promising regions. To this end, we additionally add to the literature on ordinal classification with score-based RFs by proposing the ordinal score optimization algorithm (OSOA). Similar to OF, OSOA optimizes the numeric scores of the ordinal categories. However, it aims to enhance the optimization procedure of OF by employing a non-linear optimization algorithm based on the popular Nelder-Mead method (Nelder & Mead, 1965).

After introducing the investigated methods in Sect. 2, we describe the setup of our simulation study as well as present our results in Sect. 3. Further, we analyze the performance of all methods for real data examples in Sect. 4. We close with a discussion of our findings, potential avenues for future research, and a set of practical recommendations regarding the different methods in Sect. 5.

2 Methods

In the following, we consider an ordinal classification problem where the aim is to predict a response Y with ordinal categories $1, 2, \ldots, k$. We assume that a dataset with n observations

and p covariates is available that further contains the ground truth. We will now present the methods considered for our comparison as well as introduce our newly proposed OSOA. All methods are implemented in R (R Core Team, 2023). We will refer to their individual implementations in the respective method description.

Cumulative Model

The cumulative model (McCullagh, 1980) is a parametric model which assumes that the observed ordinal response manifests an underlying latent variable that is continuous and can only be observed via thresholds that define the response categories. It models the probability $P(Y \leq r|\boldsymbol{x})$ that the ordinal outcome variable Y for an observation with covariate vector $\boldsymbol{x}$ takes at most category $r = 1, \ldots, k$ as

$$P(Y \leq r|\boldsymbol{x}) = F\left(\gamma_r + \boldsymbol{x}^\top \boldsymbol{\beta}\right),$$

where $-\infty < \gamma_1 < \cdots < \gamma_k = \infty$ denote the thresholds and F is a strictly increasing distribution function (Tutz, 2022). A common choice for F is the logistic function resulting in the proportional odds model (Tutz, 2022), i.e.,

$$P(Y \leq r|\boldsymbol{x}) = \frac{\exp(\gamma_r + \boldsymbol{x}^\top \boldsymbol{\beta})}{1 + \exp(\gamma_r + \boldsymbol{x}^\top \boldsymbol{\beta})} \tag{1}$$

which will also be used in this work. For our simulation, we fitted the proportional odds model using the `clm` function from the `ordinal` package (Christensen, 2022). As such, we will be referring to proportional odds models in the context of ordinal prediction as CLM (cumulative link model) in the remainder of this work.

Random Forest

RF (Breiman, 2001) is a ML ensemble method combining a multitude of classification or regression trees into an aggregated model. In this work, RF was used as a standalone method for multi-label classification as well as a building block of further methods introduced below such as OF, OSOA, and RFSp. Popular implementations of RF in `R` include the `ranger` package (Wright & Ziegler, 2017) and the `randomForest` package (Liaw & Wiener, 2002).

Conditional Inference Forest

In contrast to RF, CF relies on conditional inference trees (CTs; Hothorn et al., 2006) as its base component. CTs determine splits by performing permutation tests to test the association between the outcome variable and a given covariate. The covariate with the strongest association is selected as the split variable, and the concrete split value is computed in a second step. Through their conditional inference framework based on permutation tests, CTs support nominal, ordinal, and metric response types. For ordinal outcomes, numeric scores are mapped to the ordinal categories. Per default, the scores $1, 2, \ldots, k$ are used. CFs have been investigated for use in ordinal classification in detail in Janitza et al. (2016). They are implemented in the `partykit` package (Hothorn & Zeileis, 2015).

Split-Based Ordinal Random Forest

RFSp (Tutz, 2021) is based on reformulating the ordinal response problem as considered in the cumulative model into a series of binary response models that hold simultaneously. To this end, $k-1$ binary classification RFs are trained where each aims to classify whether observations belong to categories $1, \ldots, r-1$ or to categories $r, \ldots, k$, respectively, with $r = 2, ..., k$. The cumulative probabilities are then computed by aggregating the probabilities obtained from the individual RF models. This allows for using RF while following the logic of cumulative models and without having to rely on numeric scores. For our simulations, we used the implementation from https://github.com/GerhardTutz/ScoreFreeTrees. There, the individual RF models are trained using the RF implementation from `randomForest` (Liaw & Wiener, 2002).

Ordinal Forest

OF (Hornung, 2019) is a score-based RF method for ordinal prediction. Though relying on numeric scores for the regression forest internally, scores for the ordinal response categories do not have to be specified beforehand as the method tries to find optimal scores in an advance optimization step. To this end, OF generates partitions of the [0, 1] interval into k sub-intervals corresponding to each ordinal category. The associated numeric scores are in turn determined as the midpoint of the respective sub-intervals and are used to fit a regression RF (using the implementation from `ranger`; Wright & Ziegler, 2017). Prediction of unseen data is achieved by obtaining the numeric predictions from the RF fit and checking in which class they fall based on the respective borders of the class intervals. The different partitions are evaluated w.r.t. the out-of-bag (OOB) performance achieved when using them. While the OF implementation in the `ordinalForest` package (Hornung, 2022) offers a choice of performance measures, a balanced version of Youden's Index J (Youden, 1950) is used by default where for a binary classification task

$$J = \frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}} - 1 = \text{sensitivity} + \text{specificity} - 1.$$

Here, TP denotes the number of true positives, TN the number of true negatives, FP the number of false positives, and FN the number of false negatives. In the balanced case, Youden's J is computed for each class and aggregated as a simple average. Thus, all classes have the same weight irrespective of their individual sizes. The best performing scores are combined into a single, final score set by averaging which is then used to fit the final RF model. For studying the benefit of score optimization, we have also included a naive OF variant in our simulations that fits a regression forest to the default scores $1, 2, \ldots, k$ with class borders $0.5, 1.5, \ldots, \frac{2k+1}{2}$.

Ordinal Score Optimization Algorithm

Similar to OF, OSOA also assumes that the observed k response categories are a coarsened version of an underlying numeric variable. To approximate this latent variable, OSOA follows OF in partitioning the [0, 1] interval into class-specific sub-intervals $[b_r, b_{r+1})$, $r = 1, \ldots, k$,

where each sub-interval is defined by its class borders b_r and b_{r+1} with $0 = b_1 < b_2 < \cdots < b_{k+1} = 1$. Each ordinal response category is represented by the midpoint of its respective sub-interval, i.e., by the numeric score $s_r = \frac{b_r + b_{r+1}}{2}$, $r = 1, \ldots, k$. As in OF, these numeric scores are used to train a regression RF. To determine optimal choices for the class borders and thus for the class scores, OSOA follows OF in performing an optimization procedure, but employs a different optimization approach. Because OF relies on pre-generating partitions of the [0, 1] interval and then assessing their optimality, the procedure cannot react to the performance of specific partitions during the optimization process. As such, it cannot iteratively explore the space of possible partitions and focus on promising regions. To address this shortcoming, we propose OSOA which uses a non-linear optimization algorithm for finding its class borders and scores. The method is described in pseudocode in Algorithm 1. The general idea is to optimize the helper function `evaluateBorders` (Algorithm 2) that takes a set of inner class borders $b_2, \ldots, b_k$ (the outer borders b_1 and b_{k+1} are fixed as 0 and 1, respectively) as input and returns the OOB performance achieved with them. Internally, `evaluateBorders` derives the numeric scores for the response categories from the provided class borders and fits a regression RF using the numeric scores as the target variable and the corresponding covariates as predictors. As in OF, the numeric scores are first transformed with the quantile function of the standard normal distribution Φ^{-1}. For fitting RFs, we are using the implementation from the `ranger` package (Wright & Ziegler, 2017). From the RF fit, OOB predictions are obtained and in turn converted into class labels by using the transformed class borders. Finally, the predicted class labels derived from the OOB predictions can be compared with the true class labels to compute the balanced version of Youden's J used in OF. The `evaluateBorders` function can be optimized using any derivative-free non-linear optimization algorithm. In this work, we have used the Sbplx algorithm from the `NLopt` library (Johnson, 2007). The Sbplx algorithm is based on the Subplex algorithm by Rowan (1990) which is a variant of the Nelder-Mead algorithm (Nelder & Mead, 1965). Since our algorithm follows Hornung (2019) in determining an optimal partition of the [0, 1] interval, we also restrict candidate class borders during the optimization through a lower bound of 0 and an upper bound of 1. As the class borders relate to the ordinal categories, they need to be sorted such that they match the order of the original categories. Hence, only sorted borders should be considered. This can either be enforced through inequality constraints if they are supported by the given optimizer or by disincentivizing unsorted solutions via penalization in the evaluation step. As starting values for the optimization, we are using $\frac{1}{k}, \ldots, \frac{k-1}{k}$, i.e., a partition with equally wide class intervals.

The optimizer will run until a pre-specified termination condition is fulfilled such as reaching a maximum number of function evaluations `max.eval` or failing to exceed a minimum performance improvement ε. For our simulations, we set `max.eval` = 300 and $\varepsilon = 1 \times 10^{-4}$. Smaller values for ε would allow for finding finer differences, but in turn, negatively impact the runtime, while larger values for ε would speed up the optimization process, but lead to a potentially more imprecise result. Optimal settings for ε and `max.eval` depend on the given application context and should be chosen w.r.t. the desired preciseness and the computational power available. The values selected here were chosen for showcasing the method and were not optimized further. Once the optimization algorithm determines a solution, the respective scores can be used to fit the final RF model and the final borders which are both returned. For unseen data, predicted (numeric) values can be obtained from the model's individual trees and converted into class labels using the (transformed) borders. The overall class prediction for a given observation can be determined by majority voting. This prediction procedure is identical to the procedure employed in OF (cf. Hornung, 2022).

Algorithm 1 Ordinal Score Optimization Algorithm.

procedure OSOA(`max.eval`, ε)

Assign fixed outer borders $b_1 \leftarrow 0$ and $b_{k+1} \leftarrow 1$

Assign starting inner borders $\left(\tilde{b}_2, \ldots, \tilde{b}_k\right) \leftarrow \left(\frac{1}{k}, \ldots, \frac{k-1}{k}\right)$

Run optimizer on `evaluateBorders` using $\left(\tilde{b}_2, \ldots, \tilde{b}_k\right)$ as starting values until `max.eval` is reached or the performance improvement is smaller than ε

Extract optimal inner borders $\left(b_2^\star, b_3^\star, \ldots, b_k^\star\right)$

Compute final scores $\left(s_1^\star, s_2^\star, \ldots, s_k^\star, \right) \leftarrow \left(\frac{b_1+b_2^\star}{2}, \frac{b_2^\star+b_3^\star}{2}, \ldots, \frac{b_k^\star+b_{k+1}}{2}\right)$

Fit final RF using $\left(\Phi^{-1}(s_1^\star), \Phi^{-1}(s_2^\star), \ldots, \Phi^{-1}(s_k^\star)\right)$

return Final RF and transformed borders $\left(\Phi^{-1}(b_1), \Phi^{-1}(b_2^\star), \ldots, \Phi^{-1}(b_{k+1})\right)$

end procedure

Algorithm 2 Target function optimized internally in OSOA.

procedure EVALUATEBORDERS($b_2, \ldots, b_k$)

Require: $0 < b_2 < \cdots < b_k < 1$

Compute scores $(s_1, s_2, \ldots, s_k) \leftarrow \left(\frac{b_1+b_2}{2}, \frac{b_2+b_3}{2}, \ldots, \frac{b_k+b_{k+1}}{2}\right)$

Fit RF using transformed scores $\left(\Phi^{-1}(s_1), \Phi^{-1}(s_2), \ldots, \Phi^{-1}(s_k)\right)$

Obtain OOB predictions from fit and assign class labels based on transformed borders $\left(\Phi^{-1}(b_1), \Phi^{-1}(b_2), \ldots, \Phi^{-1}(b_{k+1})\right)$

Compute balanced Youden's J using class predictions and true class labels

end procedure

3 Simulation Study

3.1 Simulation Setup

To compare the predictive performance of the different prediction methods, we have performed a simulation study in which we generated datasets of varying characteristics. All data were generated from proportional odds models as is a common choice for simulating ordinal data (e.g., Janitza et al., 2016; Hornung, 2019) and aligns with the ordinal prediction methods compared in this work, most of which implicitly or explicitly assume proportional odds settings. We varied the number of observations, n, between 250, 750, and 2500. As many ordinal data applications originate from (bio-)medicine, psychology or the social sciences, we aimed to keep a mix of medium sample sizes to create realistic scenarios (see, e.g., Shen et al., 2011, for a review of sample sizes in psychology). The number of covariates, p, was either 10 or 35. While the former setting reflects common application scenarios (cf. the real data examples studied further below), the latter setting represents a compromise between including a higher number of covariates and not putting the CLM which is known to suffer from high dimensionality (Zahid & Tutz, 2013) too much at a disadvantage. We would expect values such as 35 to be more commonly encountered when analyzing, e.g., large-scale assessment studies (see, e.g., Immekus et al., 2022). As increasing the number of covariates to 35 only introduced further noise variables, the two settings also distinguished between a setting with a high signal-to-noise-ratio (SNR) and a setting with a low SNR. The

number of categories k was varied between 3, 5, and 7, as these commonly occur, e.g., in questionnaires using Likert scale items. Out of the p covariates generated, seven had an effect on the outcome. The influential covariates consisted of five normally distributed variables $X_1, \ldots, X_5 \sim \mathcal{N}(0, 1)$ and two binary variables $X_6, X_7 \sim \text{Bin}(1, 0.5)$. The remaining $p-7$ covariates were normally distributed noise, i.e., $X_8, \ldots, X_p \sim \mathcal{N}(0, 1)$. All covariates were simulated as uncorrelated. This design choice was made to limit the scope of the simulation study in which we placed more focus on different effect structures and category response distributions as explained below. For our simulation design, we were loosely inspired by Janitza et al. (2016). However, apart from the normally distributed covariates, we additionally included binary covariates and further included non-linear effects, whereas Janitza et al. (2016) only studied linear effects. We simulated the outcome using the following three different data generating processes (DGPs):

$$\text{DGP 1: } \boldsymbol{x}^\top \boldsymbol{\beta} = 3x_1 + x_2 + 2x_3 + x_4 + x_5 + x_6 + x_7,$$

$$\text{DGP 2: } \boldsymbol{x}^\top \boldsymbol{\beta} = 3x_1 + x_2 + 2x_3 + \begin{cases} 3, & x_4 \in (-1, 1] \\ -1, & x_4 \notin (-1, 1] \end{cases} + 2 \times \mathbb{1}_{x_3 \leq 0.5 \wedge x_5 > 0.5} + x_6 + x_7,$$

$$\text{DGP 3: } \boldsymbol{x}^\top \boldsymbol{\beta} = 3x_1 + x_2 + 2x_3 + x_3^2 + \begin{cases} 3, & x_4 \in (-1, 1] \\ -1, & x_4 \notin (-1, 1] \end{cases} + 2 \times \mathbb{1}_{x_3 \leq 0.5 \wedge x_5 > 0.5} + x_6 + x_7.$$

The three DGPs were characterized by an increasing amount of non-linear effects. While for DGP 1 all effects were linear, DGP 2 replaced two linear effects by non-linear effects. DGP 3 added an additional quadratic effect. Introducing an increasing amount of non-linear effects allowed for investigating to which point the parametric model still sufficed compared to the TE methods and at which point the TE methods started to outperform the parametric model and should, hence, be preferred. Apart from varying the DGPs, we followed Hornung (2019) in simulating the data according to different class distribution patterns as the author found this to have impacted the methods they have studied. To extend the approach of using different class distributions to more ordinal classification methods as well as to create more diverse scenarios (Hornung, 2019, originally used equally distributed and randomly distributed classes), we included three different class distribution patterns in our simulation. First, a pattern where the classes were distributed approximately equally. Second, a pattern where the middle categories were more populated than the margin categories. Third, a pattern where the margin categories were more populated than the middle categories. Table 1 contains an overview of the relative frequencies per class that we targeted for the different class distribution patterns. They were derived by attaching linear weights to the categories depending on the distribution pattern, e.g., for the pattern "wide middle" with five categories, the weights for categories were 1, 2, 3, 2, and 1. After dividing by the sum of all weights, one arrives at 0.11, 0.22, 0.33, 0.22, and 0.11. The class distributions were obtained by selecting specific values for the thresholds $\gamma_1, \ldots, \gamma_k$ that approximately resulted in the intended relative frequencies for each combination of DGP, number of categories, and class distribution pattern in a dataset of size 100 000. Since this was a heuristic approach, the true class probabilities did not necessarily match the values in Table 1 exactly, hence the term "targeted relative frequencies." The threshold values chosen for each scenario are listed in Table 3 in Appendix A. Using the simulated linear predictor values and respective thresholds in Eq. 1, the respective cumulative probabilities were computed, transformed into class probabilities, and used for generating class labels from a multinomial distribution.

Table 1 Targeted relative frequencies $\pi_r, r = 1, \ldots, k$ w.r.t. distribution pattern and number of categories k

Pattern	k	π_1	π_2	π_3	π_4	π_5	π_6	π_7
Equal	3	0.33	0.33	0.33	–	–	–	–
	5	0.20	0.20	0.20	0.20	0.20	–	–
	7	0.14	0.14	0.14	0.14	0.14	0.14	0.14
Wide middle	3	0.25	0.50	0.25	–	–	–	–
	5	0.11	0.22	0.33	0.22	0.11	–	–
	7	0.06	0.13	0.19	0.25	0.19	0.13	0.06
Wide margins	3	0.40	0.20	0.40	–	–	–	–
	5	0.27	0.18	0.09	0.18	0.27	–	–
	7	0.21	0.16	0.11	0.05	0.11	0.16	0.21

Fully crossing the settings of DGP, class distribution pattern, number of observations, covariates, and categories resulted in 162 conditions for evaluating the seven methods considered here. We performed 1000 replications, respectively. To evaluate the classification performance, we split the generated dataset into a training set that contained $\frac{2}{3}$ of the observations for fitting the model and a test set with $\frac{1}{3}$ of the observations for validating the model. The data partitions were determined by class-stratified sampling. As performance measures, we used the weighted Kappa coefficient κ_w (Cohen, 1968) with linear and quadratic weights as well as Kendall's τ_B (Kendall, 1945) and Spearman's tie-corrected ρ (Kendall, 1948). All of these measures are specifically suited for assessing ordinal predictions. Cohen's weighted Kappa with linear and quadratic weights has frequently been used for evaluating ordinal classification performance (see, e.g., Hornung, 2019; Ben-David, 2008). It is given by

$$\kappa_w = \frac{\sum_{r=1}^{k}\sum_{s=1}^{k} w_{rs}\, p^o_{rs} - \sum_{r=1}^{k}\sum_{s=1}^{k} w_{rs}\, p^c_{rs}}{1 - \sum_{r=1}^{k}\sum_{s=1}^{k} w_{rs}\, p^c_{rs}},$$

where p^o_{rs} is the observed proportion of instances for which r is the true category and s the predicted category, while p^c_{rs} is the analogous proportion that is expected by chance (Cohen, 1968). The respective weights are denoted by w_{rs}, where for linear weights, $w^{\text{lin}}_{rs} = 1 - \frac{|r-s|}{k-1}$, and quadratic weights, $w^{\text{quad}}_{rs} = 1 - \frac{|r-s|^2}{(k-1)^2}$ is chosen. The different weights represent different strategies for penalizing the class distances between predicted and true categories. For linear weights, instances where the predicted categories are equal or close to the true categories are associated with higher weights. For quadratic weights, relatively more weight is attributed to predictions further away from the true category and less weight to predictions close to the true category as compared to linear weights (Hornung, 2019).

To limit the computational burden of the simulation study, we did not perform a hyperparameter tuning for the RF-based methods and used default values instead as RF has been shown to be relatively robust regarding parameter choices (Probst et al., 2019). This decision is in line with previous works from the field (Janitza et al., 2016; Hornung, 2019; Tutz, 2021). For consistency, we have set the number of trees to 500 for all RF methods as this is a common default, e.g., in the `ranger` and `randomForest` packages, hence, overriding the default number of trees (for the final forest) in the `ordinalForest` package that is 5000. In all other cases, the default values remained unchanged.

3.2 Simulation Results

In the following, the results from the simulation study will be presented. From the 1 134 000 simulation runs (across all 162 conditions, 7 prediction methods, and 1000 replications), we had to exclude 138 runs that led to non-computable correlation values. All of these runs occurred for $n = 250$ when the wide middle class pattern was used. In 135 of these runs, CF was used as the learner, while naive OF (referred to as nOF in the following result figures) was used in the remaining three. The correlation values were not computable due to the model predicting the same class for all observations from the test set resulting in non-existent variability. As such, these runs were excluded from the analysis. Since for all DGPs and performance measures, the findings were consistent across all numbers of categories considered, we are only showing the results for $k = 5$ categories. The results for $k = 3$ and $k = 7$ categories are included in the Online Supplement. For the same reason, we are only showing results for the weighted Kappa with linear weights and Kendall rank correlation scores because the weighted Kappa with quadratic weights and Spearman rank correlation scores led to similar findings, respectively. We also included results for the latter two performance measures in the Online Supplement.

Results for DGP 1

Figure 1 shows the linearly weighted Kappa values the seven methods achieved for DGP 1 which only included linear effects. It can be seen that the CLM outperformed the TE methods in all scenarios. With an increasing number of observations, the RF-based learners and CF were able to catch up slightly, but still lagged behind the CLM notably. Overall, increasing the number of observations reduced the variability of the Kappa values. An increase in the number of variables led to a performance decrease that seems to have affected the TE methods more than the CLM. When only comparing the TE learners, their performance was mostly similar. For class distributions of equal size and wide margins, only minor differences between the RF-based learners and CF emerged. Only when the class distribution was characterized by a wide middle, the performances became more discernible. For this pattern, RFSp mostly

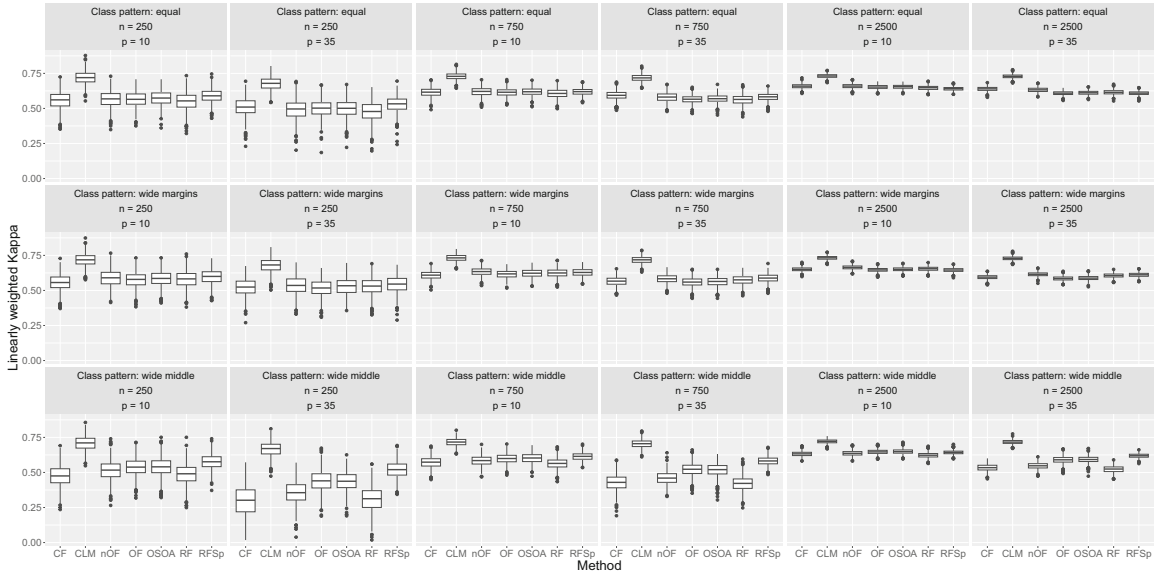

Fig. 1 Linearly weighted Kappa values achieved by prediction methods on data generated according to DGP 1 with $k = 5$ categories, and varying sample sizes, dimensions, and class patterns

performed best among the TE methods. Its performance advantage diminished, however, with an increasing number of observations. For $n = 2500$ and $p = 10$, the remaining TE methods achieved similar Kappa values again. On the other hand, CF and classical RF lagged behind for most of the wide middle scenarios. Between OF, naive OF, and OSOA, OF and OSOA performed similarly, while naive OF generally achieved slightly lower Kappa values. When studying the other two class distributions patterns, however, naive OF was on par and even slightly ahead of OF in some scenarios.

When looking at the Kendall rank correlation scores for DGP 1 (Fig. 6 in Appendix B), similar findings resulted. The CLM achieved the highest correlation scores in all scenarios. The TE methods performed mostly similar when classes were distributed equally or with wide margins. When the middle categories were more populated than the marginal categories, more differences between the TE methods became visible especially when the number of observations was 250 or 750 and the number of variables was high. In these cases, RFSp performed best among the TE learners and CF. Compared to using weighted Kappa with linear weights, CF did not lag behind as much in the wide middle scenarios and was mostly on par with the other TE methods. Further, OF could not achieve noticeable gains over the naive OF. The two OF methods performed mostly similar in all scenarios along with OSOA.

Results for DGP 2

While DGP 1 only included linear effects, DGP 2 replaced some of these linear effects and introduced non-linear effects instead. Figure 2 shows the linearly weighted Kappa values achieved by the seven learners for DGP 2. Compared to DGP 1, the performance of the learners was more similar and not as dominated by the CLM anymore. While for low observation counts, the CLM was still ahead of the other learners, the differences were not as strong. For an increasing number of observations, the CLM fell slightly behind most TE methods when the number of variables was low. When the number of variables was high, the CLM benefitted from the performance loss suffered by the TE learners. While the results for the equal and wide margins class distribution pattern display similar trends, the wide middle pattern discerns more differences between the learners, especially for a high number of

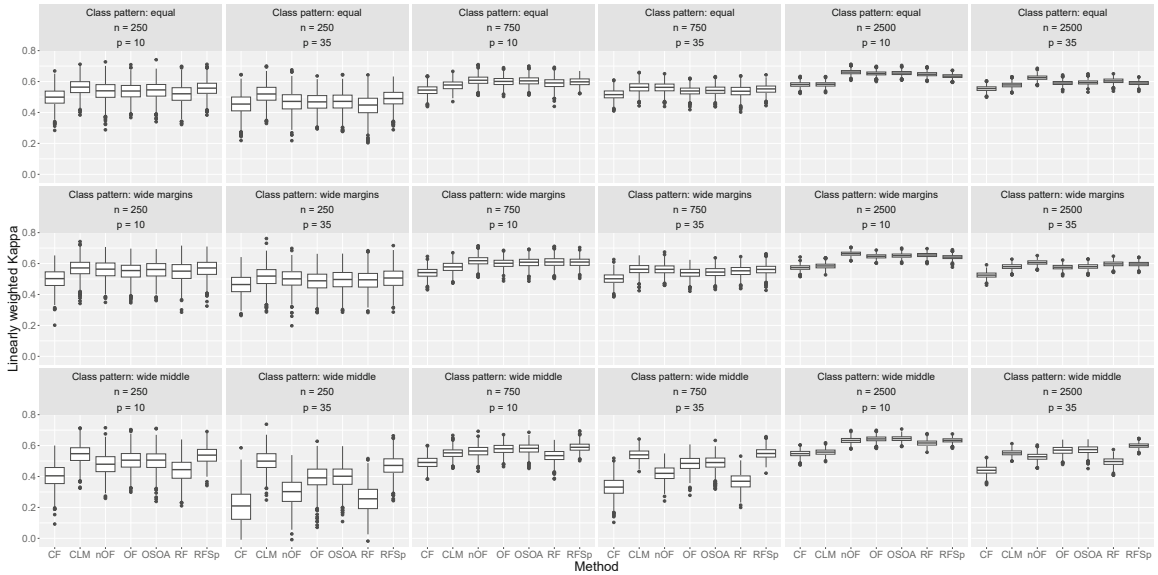

Fig. 2 Linearly weighted Kappa values achieved by prediction methods on data generated according to DGP 2 with $k = 5$ categories, and varying sample sizes, dimensions, and class patterns

variables. RFSp performed consistently well in all wide middle scenarios (even under a high number of variables). The CLM was competitive in cases where the number of observations was either low or the number of variables was high. OF performed well for higher number of observations, slightly but consistently outperforming naive OF in the wide middle scenarios. On the other hand, naive OF performed slightly better than OF for the other two class distribution patterns. For equally distributed classes, naive OF was among the best performing methods for observation counts greater than 250. OSOA was close in performance to OF in all scenarios. While the differences between the RF-based methods were overall rather subtle, the performance of CF fell off in a number of scenarios, especially for class distributions with a wide middle.

Figure 7 (Appendix B) showing the Kendall rank correlation scores achieved by the seven learners echoes the findings for the weighted Kappa with linear weights. For the most part, the CLM and all RF-based learners performed similarly. Using CF usually led to the lowest correlation scores. In spite of the existence of non-linear effects, the CLM was competitive in cases where observation counts were 250 or 750. For $n = 2500$, the CLM fell behind for all class distribution patterns in the case of $p = 10$ covariates. For $p = 35$, the CLM was competitive again since it did not suffer as much from the increase in dimensionality as the TE methods. Overall, the differences between the different learners are even less pronounced than when using the weighted Kappa with linear weights as the performance measure. Regarding OF and naive OF, OF could only outperform naive OF in the wide middle scenario with 250 observations and 35 covariates. In the remaining scenarios, naive OF was either on par with OF or very slightly ahead as in the case of classes that had wide margins or were distributed equally. OSOA matched the performance of the former two methods in all scenarios without deviating notably in any direction.

Results for DGP 3

Figure 3 shows the performance of the seven learners as measured by the weighted Kappa with linear weights for DGP 3 where an additional quadratic effect was introduced. It can be seen that while the CLM was still competitive for small sample sizes ($n = 250$), it fell slightly

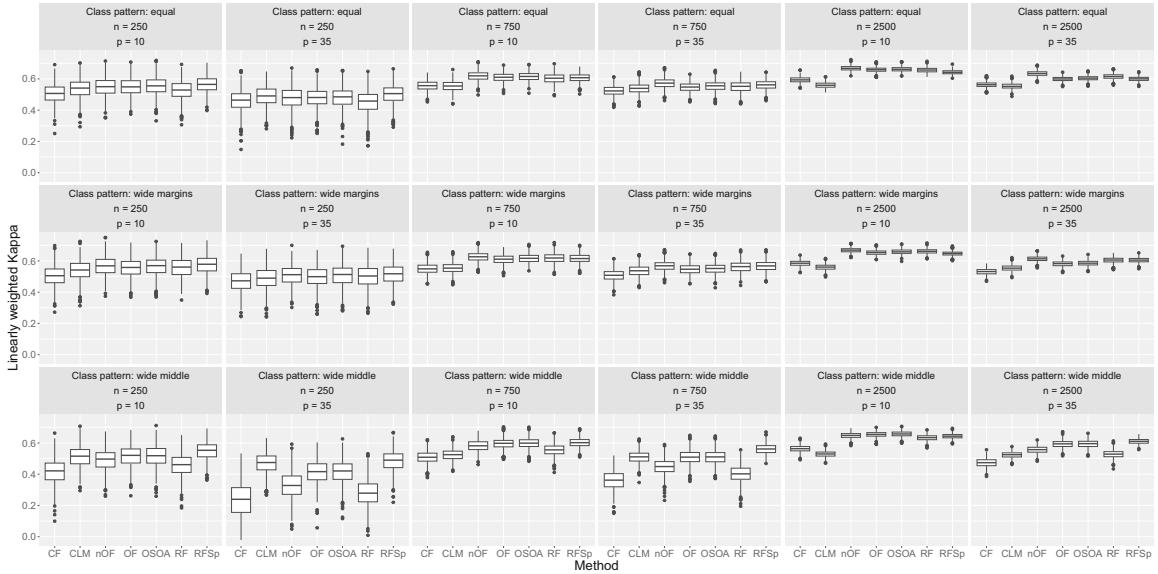

Fig. 3 Linearly weighted Kappa values achieved by prediction methods on data generated according to DGP 3 with $k = 5$ categories, and varying sample sizes, dimensions, and class patterns

behind all RF-based learners for 750 observations and even more for 2500 observations, especially when the number of covariates was low. However, it was still ahead of CF in most scenarios. As before, the results for equally distributed classes and classes with wide margins were similar, while the scenarios with a wide middle revealed more differences between the learners. For the latter class pattern, RFSp, OF, and OSOA were frequently among the best performing methods. As for the two other data generating processes, OF and OSOA could improve upon naive OF for class distributions with a wide middle, while naive OF performed similarly or (slightly) better for classes that were populated equally or more strongly in the margin categories. For equally distributed classes, naive OF was one of the best performing learners. However, the differences between the RF-based learners (apart from CF) were again mostly subtle.

Regarding the Kendall rank correlation scores, Fig. 8 (Appendix B) generally mirrors the results obtained for the weighted Kappa. The CLM was competitive when the number of observations was low, but fell with increasing sample sizes. Similarly, CF was commonly outperformed by the RF-based methods and was on par with CLM, or in some scenarios even behind. For the RF-based methods, the differences were once more subtle. As for the previous data generating processes, OF and OSOA could not notably improve upon naive OF.

3.3 Robustness of Data Generation

To study whether the results were sensitive to the generative model, we additionally reran parts of the simulation for a generative model that created data according to a linear regression model and binned the outcome into ordinal response categories. As such, this generative model mimicked the approach of transforming numeric outcomes to ordinal outcomes commonly used in practice. To this end, we generated the linear predictor term according to DGP 2 and added standard normal noise. The resulting numeric outcomes were binned such that the targeted relative class frequencies in Table 1 were obtained. This was achieved by approximating the empirical distribution function of the numeric outcome using 100 000 simulated observations. As binning values, the respective quantiles leading to the targeted class distribution pattern were selected. For instance, for the wide middle example mentioned above with $k = 5$ categories and targeted relative frequencies of 0.11, 0.22, 0.33, 0.22, and 0.11, respectively, we have selected the 11.11%, 33.33%, 66.66%, and 88.88% quantiles as binning values. This approach was inspired by the simulation study in Hornung (2019). Using the alternative generative model for DGP 2, we have obtained results (see Online Supplement) that were consistent with the results described above which indicates that our findings are robust w.r.t. the generative model used for simulation.

3.4 Runtime Comparison

Apart from the predictive performance, the required computation time should also be taken into account when comparing the different methods. Figure 4 shows the time needed for training and predicting using a given method relative to the time needed for the CLM. This implies that if absolute runtimes increase overall between conditions, but by the same factor for any given method and the CLM, relative runtimes will be the same in both conditions. Relative runtimes have the advantage of being less machine-specific than the absolute runtimes. The CLM was chosen as the reference method as it represents the classical approach for ordinal prediction as well as the computationally least expensive method. Note that the values have

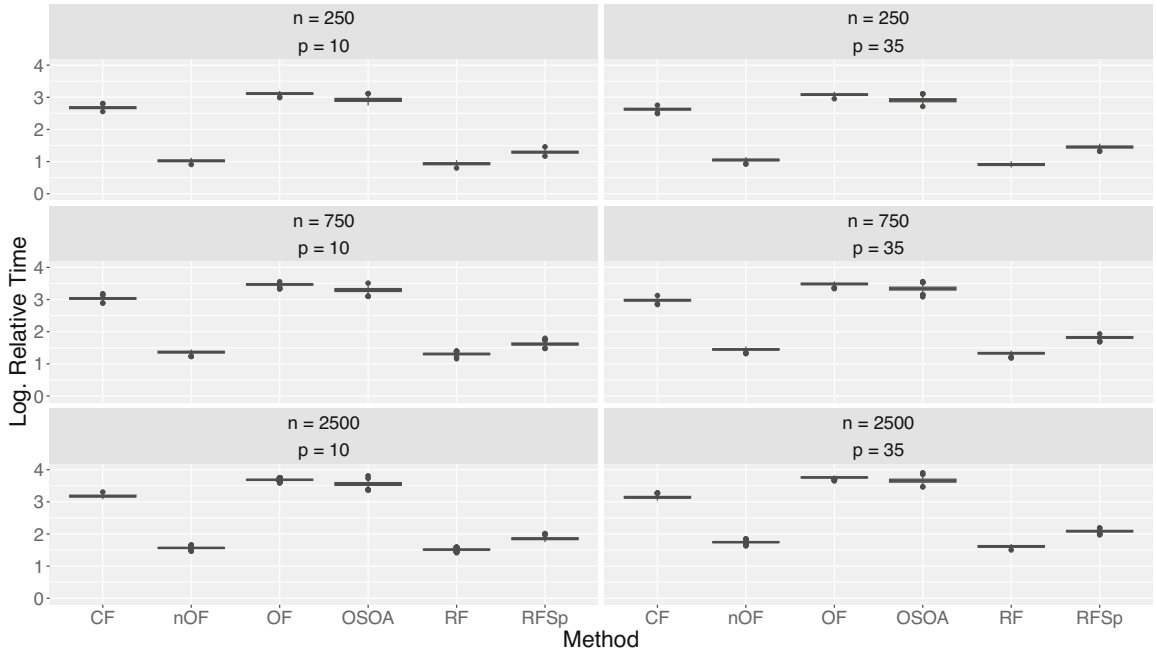

Fig. 4 Computation time of TE prediction methods relative to CLM runtime. Values have been logarithmized using base 10

been logarithmized using base 10. As such, a value of 0 means that a given method needed the same time as the CLM, while a value of 2 means that the computation time of a given method was larger by a factor of 100. Since the choices of DGP, number of categories (except for RFSp in which the number of RF fits within RFSp scales linearly with the number of categories), and class pattern had little impact on the relative runtime results, we are only showing the runtime comparison for data generated according to DGP 2 with five categories that were distributed equally. Regarding RFSp, we have decided for five categories as it posed the middle ground of the different numbers of categories considered in our simulation. Since all computations were performed on a compute cluster, it cannot be guaranteed that the individual computations were performed on identical CPUs. Furthermore, we have restricted all methods to only use a single core for computation which may have negatively impacted the runtime of some methods due to not being able to employ parallelization for faster computations. Due to these limitations, our results should be interpreted as broad estimates rather than exact benchmarks. When comparing the relative runtimes, one can see that for all methods, the required runtime was higher by at least a factor of 10 compared to the CLM. From the TE methods, the lowest runtimes were achieved by RF and naive OF which both fit a single RF. As RFSp needed to fit four RF models, it required more runtime than RF and naive OF as expected. In comparison, to the RF-based methods, CF needed relatively higher runtimes for fitting a single model. As expected, the runtimes of OF and OSOA were quite high due to the optimization step. Their computation time was between a 1000 and 10,000 times higher than for the CLM. However, one should keep in mind that the runtime of OF and OSOA is directly linked to the number of optimization iterations, i.e., the number of different score sets evaluated in OF and the maximum number of evaluations in OSOA.

Regarding the impact of n and p, one can see that the relative computation times of the TE methods increase for larger sample sizes. This means that the CLM's runtimes were less impacted by increased sample sizes than the runtimes of the TE methods. Increasing the number of covariates did not result in a similar effect, indicating that the CLM scaled similarly to the TE methods regarding the computation time.

For practical implications, however, one should also consider the actual runtimes instead of solely relying on the relative runtimes. For example, in the computationally most demanding case of $n = 2500$ observations and $p = 35$ covariates, the CLM needed a median time of 0.08 s, RF 3.40 s, naive OF 4.66 s, RFSp 10.26 s, CF 114.93 s, OSOA 394.76 s, and OF 479.79 s. Depending on the dataset and computational power at hand, the discrepancy between, say, a CLM and naive OF or RFSp may be negligible for practical purposes as long as one does not employ a hyperparameter tuning for the RF-based methods.

4 Real Data Examples

In addition to the simulation study, the seven methods were also evaluated on eight real datasets. For our selection of datasets, we have strived for incorporating a variety of different application domains and dataset characteristics. Therefore, we have included datasets from psychology, (bio-)medicine, and the social sciences, as these are common application fields for ordinal prediction. The datasets also vary regarding their size and target variable properties (i.e., number of categories and their distribution). Table 2 provides an overview of the datasets. Out of these eight datasets, five (Birthweight, Boston, Hearth, Medical Care, and Wine Quality) were already analyzed in Tutz (2021), while the Mammography data were also used in Janitza et al. (2016) and Hornung (2019). The Birthweight dataset is concerned with predicting the birthweight of newborns. It was obtained from the `MASS` package. The original numeric target variable was categorized according to Tutz (2021). For the Boston dataset, it is of interest to predict the median value of owner-occupied homes in Boston. It was obtained from the `mlbench` package (Leisch & Dimitriadou, 2021). The numeric target variable was binned according to Tutz (2021). For the Hearth dataset, the goal is to predict the severity of coronary artery disease. It was taken from the `ordinalForest` package (Hornung, 2022). The Mammography dataset contains information about mammography experiences and was taken from the `TH.data` package (Hothorn, 2023).

The Medical Care dataset originates from the US National Medical Expenditure Survey from 1987. It was obtained from the `AER` package (Kleiber & Zeileis, 2008). We have chosen the same subset of observations and covariates to predict the number of physician office visits as well as the same target variable binning as Tutz (2021). The Student dataset contains information about the final grade of students from a Portuguese language course. The data was taken from the UCI Machine Learning Repository (Cortez, 2014). We have binned the target variable that was originally on a 20-point scale into five categories (see Table 2). As covariates, we have selected gender, age, region (rural vs. urban), parents' cohabitation status, mother's education, father's education, weekly study time, presence of educational support from the school, presence of educational support from the family, partaking in paid extra classes, interest in taking higher education as well as access to the internet at home. The Wage dataset was obtained from the `ISLR` package (James et al., 2021). The goal is to predict the wage of workers in the Mid-Atlantic region. The target variable was binned into five categories for our analysis (see Table 2). Lastly, the task for the Wine Quality dataset is predicting the quality score of wine. It was taken from Cortez et al. (2009). The original categories were coarsened according to Tutz (2021). None of the obtained datasets contained any missing values. To evaluate the seven learners, we performed a five-fold cross-validation with 50 replications. For the learners, we used the same settings as in the simulation study before.

Table 2 Description of real datasets used for evaluation

Name	Obs	Cov	Description and categories
Birthweight	189	8	Birth weight in grams
			1: < 2500 ($n = 59$), 2: 2500-3000 ($n = 38$)
			3: 3000–3500 ($n = 45$), 4: > 3500 ($n = 47$)
Boston	506	13	Median value of owner-occupied homes in $1000
			1: < 15 ($n = 97$), 2: 15-19 ($n = 78$), 3: 19-22 ($n = 109$)
			4: 22–25 ($n = 98$), 5: 25-32 ($n = 57$), 6: > 32 ($n = 67$)
Hearth	294	10	Severity of coronary artery disease
			1: no disease ($n = 188$), 2: degree 1 ($n = 37$)
			3: degree 2 ($n = 26$), 4: degree 3 ($n = 28$), 5: degree 4 ($n = 15$)
Mammography	412	5	Last mammography visit
			1: Never ($n = 234$), 2: Within a year ($n = 104$)
			3: Over a year ($n = 74$)
Medical Care	1778	10	Number of physician office visits
			1: 0 ($n = 329$), 2: 1 ($n = 183$), 3: 2-3 ($n = 362$), 4: 4-6 ($n = 398$)
			5: 7–8 ($n = 149$), 6: 9-11 ($n = 149$), 7: > 11 ($n = 208$)
Student	649	12	Final grade in Portuguese language course
			1: 0–10 ($n = 100$), 2: 10-11 ($n = 201$), 3: 12-13 ($n = 154$)
			4: 14–15 ($n = 112$), 5: 15-20 ($n = 82$)
Wage	3000	8	Wage of workers in Mid-Atlantic region in $100k
			1: < 75 ($n = 430$), 2: 75-100 ($n = 913$), 3: 100-125 ($n = 789$)
			4: 125–150 ($n = 525$), 5: > 150 ($n = 343$)
Wine Quality	4898	6	Wine quality rating
			1: < 5 ($n = 183$), 2: 5 ($n = 1457$), 3: 6 ($n = 2198$)
			4: 7 ($n = 880$), 5: > 7 ($n = 180$)

Figure 5 shows the values for the weighted Kappa with linear weights achieved by the learners on the eight datasets. It can be seen that the CLM was notably outperformed on the Boston, Mammography, and Wine Quality datasets. For the Medical Care data, the CLM, however, achieved the best performance of all learners. For the remaining datasets, it was competitive with the TE learners. When comparing the RF-based learners and CF, CF and the classification RF could never outperform the other learners and were either on par or lagged (slightly) behind. OF could improve upon naive OF for the Birthweight, Mammography, Medical Care, Student, Wage, and Wine Quality datasets. The predictive performance of OSOA was mostly aligned with the performance achieved by OF except for the Medical Care data. RFSp was among the most consistently performing methods. It outperformed all other learners on the Wage dataset and was competitive for the remaining datasets. Overall, however, the performance differences between the RF-based learners were often on a small scale apart from situational advantages or disadvantages for some methods, respectively. The fluctuating hierarchies regarding the performance could not reveal a general advantage for any method. When using Cohen's Kappa with quadratic weights, a similar picture emerged (see Online Supplement).

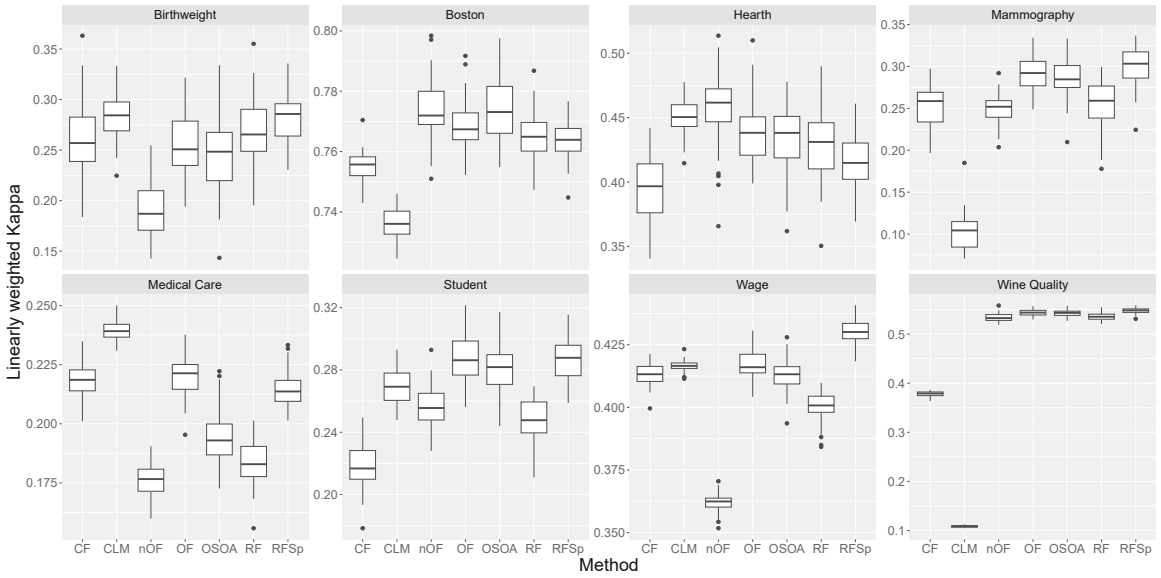

Fig. 5 Linearly weighted Kappa values achieved by prediction methods on real datasets

Figure 9 (Appendix B) shows the Kendall rank correlation scores achieved by the seven learners. Overall, the result patterns resembled the findings for the weighted Kappa. The CLM was outperformed on the Wine Quality and Mammography datasets as well as slightly on the Boston dataset. For the Medical Care dataset, the CLM performed best and was competitive for the remaining datasets. In contrast to the weighted Kappa with linear weights, CF was lagging behind less compared to the RF-based learners for Kendall's rank correlation scores. OF could not notably outperform its naive counterpart. As before, OSOA's performance mostly fell between OF and naive OF. Generally, the differences between the RF-based learners were mostly subtle. When taking all learners into account, the largest differences were mostly caused by the CLM when it either performed particularly well as for the Medical Care data or when it was not suitable as for the Wine Quality and Mammography datasets. These findings were echoed when looking at the Spearman rank correlation scores. For the latter results, we refer to the Online Supplement. While the predictive performance of the individual methods was based on relative comparisons so far, for practical purposes, it is also relevant to consider the absolute predictive performance of the methods. Figures 5 and 9 show that overall, the highest predictive performance was achieved on the Boston dataset with Kendall rank correlations between 0.78 and 0.84, followed by the Wine Quality (only for TE learners), Hearth, and Wage datasets. For the Birthweight, Student, Mammography, and Medical Care data, the predictive performance values achieved were generally lower (e.g., for the latter two datasets, rank correlations lower than 0.34 were achieved for all methods), indicating that these prediction tasks were more difficult.

5 Discussion

In this work, we provided an extensive comparison of the CLM and TE methods for ordinal classification such as RF (Breiman, 2001), CF (Hothorn et al., 2006), (naive) OF (Hornung, 2019), and RFSp (Tutz, 2021). We further contributed a new method through our proposed OSOA. OSOA employs a non-linear optimization algorithm for determining optimal numeric scores to be assigned to the ordinal response categories within a regression RF framework.

We studied all methods in a wide range of varying scenarios including three different DGPs that were characterized by an increasing amount of non-linear effects. Inspired by Hornung (2019), we further varied the class distributions using three different distribution patterns. Creating such diverse data settings helped us investigating under which circumstances traditional parametric models such as the CLM are competitive with modern, computationally more demanding TE methods, and at which point the latter offer a noticeable improvement in predictive performance. Furthermore, by including different TE approaches such as classification RFs, binarized RFs, regression RFs with unoptimized scores as well as regression RFs with optimized scores, we could study differences among the TE methods, particularly regarding the question whether the computational cost of score optimization yields relevant benefits. Our extensive comparison has revealed several important insights which we discuss in the following.

Finding 1: CLM Remains Competitive for Small Sample Sizes and Limited Non-linear Effects

Similar to the findings in Tutz (2021), our results were also mainly characterized by cases in which the CLM either outperformed the other methods, was on par or notably lagged behind. For the first DGP that included linear effects only, the CLM notably outperformed all ML approaches. This was to be expected as the data generated from a proportional odds model with the all-linear effect structure represented the optimal use case for the CLM. With an increasing amount of non-linear effects, the CLM's performance suffered in comparison to the TE methods. However, it required quite strong non-linear effects for the CLM to be outperformed. Especially for small sample sizes, the CLM was regularly competitive even in the presence of non-linear effects. For larger sample sizes, the TE learners usually closed the performance gap (when they lagged behind for smaller samples) or widened it (when they were slightly ahead or on par for smaller samples), respectively. Our analysis of real datasets revealed a similar pattern where the most discrepancies between the different methods were caused by the performance of the CLM in relation to the TE methods.

Finding 2: TE Methods Reveal Only Small Differences Among Themselves

When comparing the TE methods, CF fell behind the RF-based learners once the DGPs increasingly included non-linear effects. This was more prevalent when using the weighted Kappa to assess the predictive performance and less when using rank correlation scores. The differences between the RF-based methods themselves were mostly subtle. The latter finding is in line with Tutz (2021). The performance of our newly proposed OSOA mostly matched the performance of OF. When evaluating all methods on real datasets, we similarly found small differences between the TE methods. The largest performance differences were mostly caused by the CLM that either performed particularly well on a dataset or was outperformed notably by the set of RF-based methods.

Regarding the number of covariates, we observed the RF-based learners to incur a more notable performance loss than the CLM. However, this effect could be attributed to a lacking hyperparameter tuning. Even though RFs are relatively robust regarding their parameter choices, the condition with a higher number of covariates added only noise to the model. Consequently, only 7 of the 35 covariates affected the outcome. By using the default setting (i.e., the square root of the number of covariates) for the hyperparameter `mtry` that regulates how many covariates are randomly sampled for consideration in a given split, RFs potentially

had to rely often on noise variables only for splitting which ultimately harmed the predictive performance.

Finding 3: Limited Benefit of Score Optimization in OF and OSOA

Regarding the question of whether optimizing scores in score-based methods such as OF and OSOA nets a benefit over naive OF, our results were mixed. In our simulation, we found that OF improved upon naive OF in cases where the distribution of classes was characterized by dominant middle categories. As such, this supports the findings in Hornung (2019). For the other distribution patterns where classes were distributed equally or a dominant margin categories were present, however, OF could not outperform naive OF. As OSOA's performance mostly aligned with OF, the findings above also hold when comparing OSOA to naive OF. For the real datasets, OF and OSOA could improve upon naive OF for six out of eight datasets, but the differences were often on a small scale. As such, the benefits of score optimization were rather situational. While it can improve the predictive performance, it is not guaranteed for any given dataset and the high runtimes demonstrated in the runtime comparison must be kept in mind.

Limitations of Simulation Study

While we aimed to make our DGPs as diverse as possible by varying the amount of non-linear effects and the distribution of the response categories, all datasets were generated from models with identical effects across the categories. Future work could study DGPs that include category-specific effects. To this end, it would also be sensible to analyze more parametric models such as partial proportional odds models (see e.g., Brant, 1990; Peterson & Harrell, 1990) or the more recently proposed location-shift model by Tutz and Berger (2022). Additionally, all covariates were simulated as uncorrelated. While it is likely that in individual simulation runs some correlations between covariates randomly occurred, systematically varying the simulated correlation between covariates in future work may help illuminate further differences between the ordinal prediction methods studied here. Furthermore, our work only focused on datasets with relatively few covariates. For high-dimensional data, classical parametric models may run into problems (see, e.g., Zahid & Tutz, 2013). As such, it would be of interest to study how the findings from our comparison translate to high-dimensional settings. Another limitation of our work is posed by the lack of hyperparameter tuning for TE methods. Although this was in line with other works from the field (Janitza et al., 2016; Hornung, 2019; Tutz, 2021) and despite RF's relative robustness regarding the parameter choice (Probst et al., 2019), a hyperparameter tuning could have amended the performance loss suffered by the TE methods in the simulation scenarios with many noise variables. Further, we did not optimize the parameters of OSOA as its inclusion in the comparison study served more as a showcase. More fine tuning of OSOA's parameters may yield higher predictive performance or limit the computational burden of the optimization procedure while potentially sacrificing only little predictive performance.

Combining Multiple Prediction Methods as Ensembles

While this work focused on comparing individual prediction methods to assess their respective viability in different data scenarios, Tutz (2021) proposed combining multiple prediction

models (e.g., CLM, OF) into a joint ensemble prediction model. To this end, the individual models are trained separately on the training data. For predicting new observations, the predicted response category probabilities of the individual models are aggregated via a weighted mean. The weights are determined in an advance step where the prediction methods are evaluated separately on a subset of the training data and higher predictive performance (relative to the other methods) results in a higher weight. For more details on the joint ensemble approach, we refer to Tutz (2021). When running an ensemble consisting of a CLM, an OF and a RFSp model on the real data examples, Fig. 10 (Appendix C) shows that the joint ensemble achieved the highest linearly weighted Kappa values for two datasets, indicating that this approach can yield benefits. For some datasets, the joint ensemble was partly held back when one of its prediction models performed poorly on the given dataset (e.g., CLM for the Wine Quality data). As such, the joint ensemble approach can be worthwhile to consider, but requires a few design choices whose optimality is likely to vary between datasets, e.g., how many and which prediction methods should be included, what the optimal weighting strategy is, etc. As the joint ensemble fits each prediction method multiple times (due to the weight computation), this approach also leads to extended computational runtimes. Furthermore, if interpretability is of interest, then the ensemble approach loses the interpretability its individual models potentially provide. However, if only the predictive performance is relevant for the application at hand, then joint ensembles can be a sensible approach to employ.

Avenues for Further Methodological Research

A possible explanation for the lack of consistent improvement upon naive OF could be that OF and OSOA are internally both optimizing a class balanced version of Youden's J. As Youden's J is a measure reflecting sensitivity and specificity, it is not a performance measure specifically tailored towards ordinal classification. As such, optimizing Youden's J may not necessarily result in a more optimal classification result from an ordinal perspective. One could investigate whether optimizing other performance measures such as the weighted Kappa would lead to more consistent improvement upon naive OF.

Specifically for OSOA, the proposed optimization procedure could further be enhanced by including restarts which could help the optimizer in escaping local minima to possibly find better solutions. Additionally, one could try to use multiple sets of starting score values for initializing the optimization instead of only using scores derived from a partition of the $[0, 1]$ interval where all sub-intervals are of equal width. Furthermore, one could explore alternative optimization approaches, e.g., evolutionary algorithms (EAs) such as the covariance matrix adaptation evolution strategy (Hansen & Ostermeier, 1996). EAs keep a population of solution candidates which is continually optimized through recombining and mutating current population members, or sampling new candidates from distributions influenced by current candidates (for an introduction to EAs see e.g., Pétrowski & Ben-Hamida, 2017).

Another possible explanation for the lack of improvement achieved through score optimization may be found in the prediction procedure of OF and OSOA. OF and naive OF determine predictions for new observations by first obtaining the predicted numeric scores from all trees. For each of the tree models, the predicted score is first transformed into a class label by using the associated class borders. The final class prediction is then computed by aggregating over all class labels via majority voting. As OSOA was modeled after OF in this regard, it follows the same prediction procedure. However, since the individual trees in a forest are usually grown without restricting their complexity too much, the resulting terminal nodes may often be pure, i.e., contain (almost) only observations from a single category.

Thus, the predicted value for observations falling into the terminal node would simply be the numeric score assigned to the respective category, or a value very close to it in case the node is not entirely pure. Since the predicted numeric score would be immediately transformed into a class label and not processed for further aggregation in its numeric state, the actual numeric score assigned to the respective category may have little impact on the overall prediction behavior. In future work, one could investigate whether score optimization yields greater benefits when first aggregating the numeric scores from the individual trees by averaging and assigning a class to the aggregated prediction score.

Practical Recommendations

Practical machine learning applications often follow a workflow in which the data problem at hand is not approached by only fitting a single prediction method, but instead comparing several prediction methods on a subset of the data to determine which of the many prediction methods available achieves the best performance for the given data. While it would be ideal to compare as many methods as possible, this is often not feasible in practice due to limited time and computational resources. Therefore, one typically resorts to benchmarking only a selected set of prediction methods. Acknowledging this common compromise, we therefore offer the following recommendations that may help arriving at a feasible set of methods to compare for a given data situation. Our work demonstrated that while the RF-based methods can outperform the CLM for non-linear effects and larger sample sizes, it required relatively strong non-linear effects for causing a performance gap. Even in the presence of non-linear effects, the CLM was competitive for small sample sizes. In line with Tutz (2021), we therefore recommend always considering a parametric model as a benchmark model against which other methods should be gauged. For small sample sizes and weak non-linear effects, the parametric model may already suffice and even lead to better performance as the TE methods often require a certain baseline of observations before they can achieve satisfying performance. Having small sample sizes is not uncommon in fields like psychology and social sciences, where ordinal responses are often encountered. For example, in their review of 1568 samples used in psychological publications between 1995 and 2008, Shen et al. (2011) found a median sample size of 172 and a mean sample size of 690 observations. As the TE methods often performed similarly and most differences in our comparison arose from performance gaps between the CLM and the relatively homogeneous set of TE methods, we recommend comparing the parametric benchmark model with a naive OF or RFSp. Classical RF was often lagging slightly behind the other TE methods. The score optimization procedures of OF and OSOA have been shown to be quite computationally demanding and their benefit was mostly situational. Therefore, comparing a parametric model with a computationally less demanding naive OF or RFSp model can give a first indication whether the application at hand is more suited towards parametric models or TE methods. Should naive OF or RFSp outperform the parametric model and should the distribution of the response categories be characterized by dominant middle categories, employing a score optimizing TE method may be worthwhile (if the computational resources permit). Regarding the TE methods, however, one should further keep in mind that despite the RF's robustness regarding its hyperparameter settings (Probst et al., 2019), scenarios in which one expects a high rate of noisy covariates and a low rate of influential covariates may warrant a hyperparameter tuning. This, in turn, would increase the runtime even further and in combination with a score optimization approach as in OF and OSOA may be less feasible for practical purposes.

Instead of arriving at a prediction model based on the recommendations above, one can also follow the joint ensemble approach from Tutz (2021) with the implications discussed earlier. Since the joint ensemble performs a model selection process internally, it offers a viable alternative to the benchmarking approach. In this case, our recommendations could be used to guide which prediction models to include in the joint ensemble.

Appendix A: Threshold Values for Simulation Study

Table 3 Threshold values for combinations of DGP, class distribution pattern, and k

DGP	Class pattern	k	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7
DGP 1	Equal	3	−3	0.75	∞	–	–	–	–
		5	−4.75	−2	0.25	3	∞	–	–
		7	−5.75	−3.5	−1.75	−0.25	1.5	3.75	∞
	Wide margins	3	−2	0.25	∞	–	–	–	–
		5	−3.75	−1.75	−0.5	1.75	∞	–	–
		7	−4.75	−2.75	−1.5	−0.75	0.5	2.5	∞
	Wide middle	3	−4	2	∞	–	–	–	–
		5	−6.25	−2.75	1	4.5	∞	–	–
		7	−8	−5	−2.5	0.25	2.75	5.5	∞
DGP 2	Equal	3	−5.25	−1.25	∞	–	–	–	–
		5	−7	−4.25	−2	0.75	∞	–	–
		7	−8	−5.75	−4	−2.25	−0.5	1.75	∞
	Wide margins	3	−4.25	−2	∞	–	–	–	–
		5	−6	−3.75	−2.5	−0.25	∞	–	–
		7	−7	−5	−3.75	−3	−1.75	0.5	∞
	Wide middle	3	−6.25	0	∞	–	–	–	–
		5	−8.75	−5.25	−1.25	2.5	∞	–	–
		7	−10	−7.25	−4.75	−1.75	1	4	∞
DGP 3	Equal	3	−6	−2	∞	–	–	–	–
		5	−8	−5.25	−3	−0.25	∞	–	–
		7	−9	−6.75	−5	−3.25	−1.5	0.75	∞
	Wide margins	3	−5.25	−3	∞	–	–	–	–
		5	−7	−4.75	−3.5	−1.25	∞	–	–
		7	−8	−5.75	−4.5	−3.75	−2.5	−0.25	∞
	Wide middle	3	−7.25	−1	∞	–	–	–	–
		5	−10	−6	−2	1.5	∞	–	–
		7	−11.25	−8.25	−5.5	−2.5	0	3	∞

Appendix B: Kendall Rank Correlation for Simulation and Real Data Comparisons

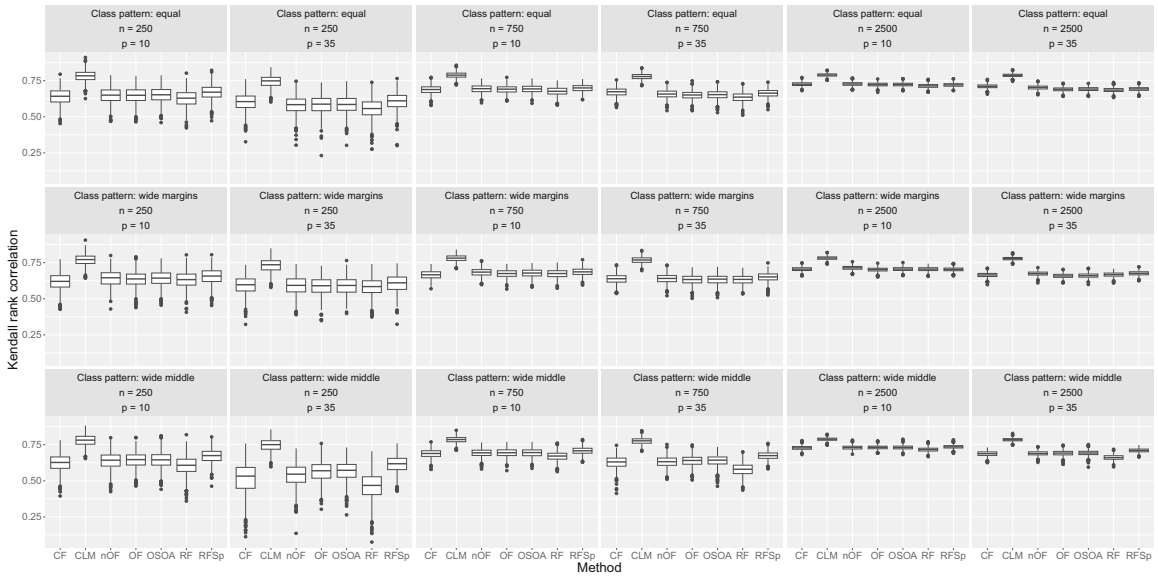

Fig. 6 Kendall rank correlation scores achieved by prediction methods on data generated according to DGP 1 with $k = 5$ categories, and varying sample sizes, dimensions, and class patterns

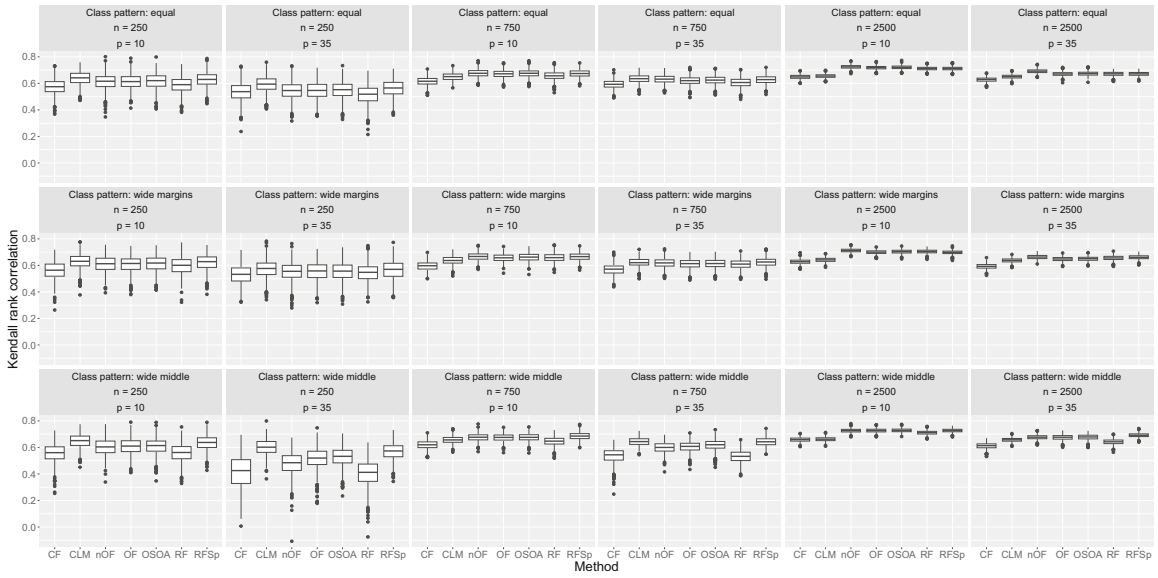

Fig. 7 Kendall rank correlation scores achieved by prediction methods on data generated according to DGP 2 with $k = 5$ categories, and varying sample sizes, dimensions, and class patterns

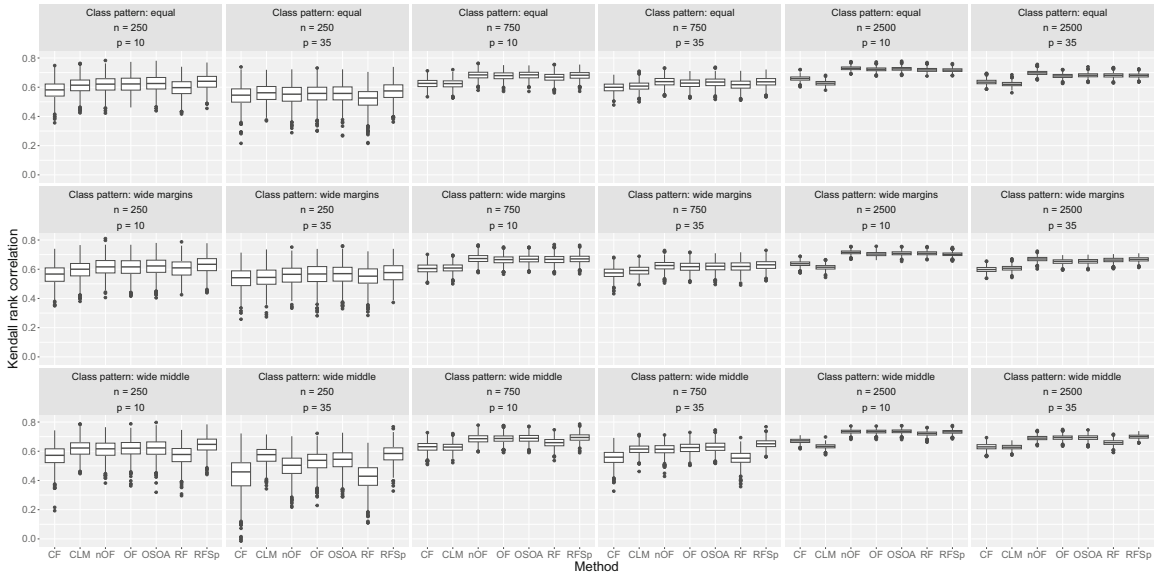

Fig. 8 Kendall rank correlation scores achieved by prediction methods on data generated according to DGP 3 with $k = 5$ categories, and varying sample sizes, dimensions, and class patterns

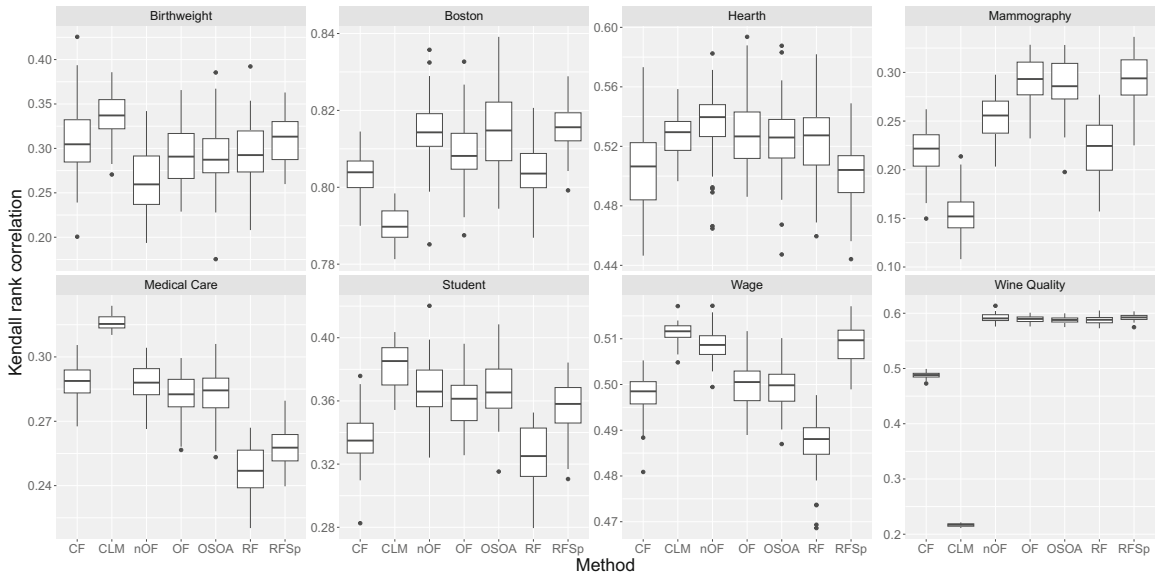

Fig. 9 Kendall rank correlation scores achieved by prediction methods on real datasets

Appendix C: Real Data Example Results for Joint Ensemble Learner

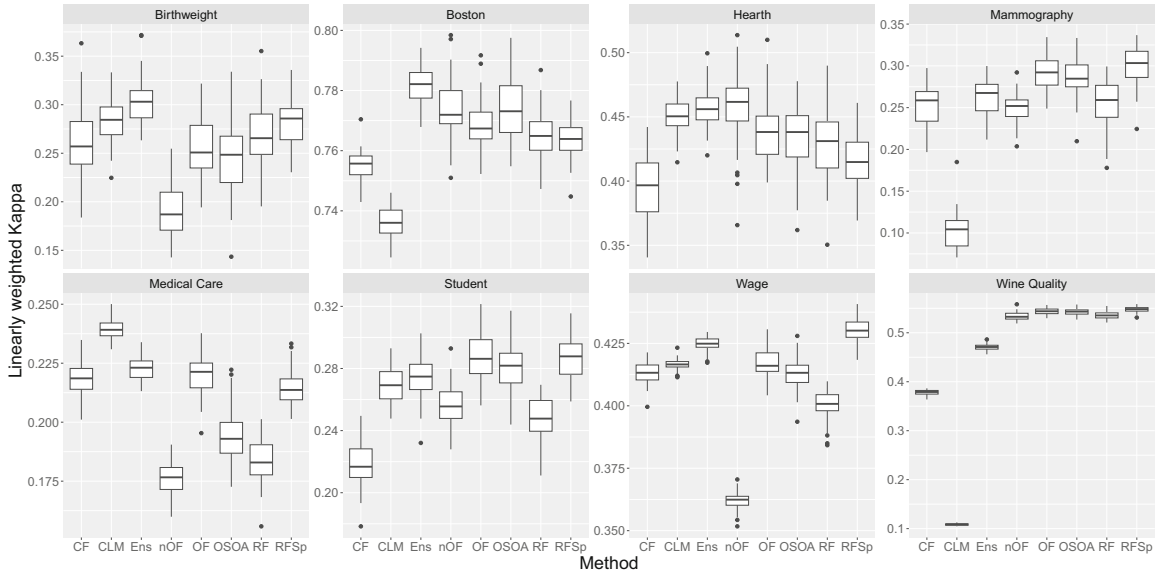

Fig. 10 Linearly weighted Kappa values on real datasets when additionally including an ensemble (Ens) of a CLM, OF, and RFSp

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00357-024-09497-9.

Acknowledgements The authors would like to thank Dr. Marie Beisemann for providing helpful discussion and valuable feedback. This work has been partly supported by the Research Center Trustworthy Data Science and Security (https://rc-trust.ai), one of the Research Alliance centers within the UA Ruhr (https://uaruhr.de). Additionally, the authors gratefully acknowledge the computing time provided on the Linux HPC cluster at TU Dortmund University Dortmund (LiDO3), partially funded in the course of the LargeScale Equipment Initiative by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as project 271512359.

Funding Open Access funding enabled and organized by Projekt DEAL.

Code Availability The R code for our work can be obtained from our corresponding OSF repository https://osf.io/v64d9/.

Declarations

Conflict of Interest The authors declare no competing interests.

References

Archer, K. J. (2010). rpartOrdinal: An R package for deriving a classification tree for predicting an ordinal response. *Journal of Statistical Software, 34*(7), 1–17. https://doi.org/10.18637/jss.v034.i07

Ben-David, A. (2008). Comparison of classification accuracy using Cohen's weighted kappa. *Expert Systems with Applications, 34*(2), 825–832. https://doi.org/10.1016/j.eswa.2006.10.022

Brant, R. (1990). Assessing proportionality in the proportional odds model for ordinal logistic regression. *Biometrics, 46*(4), 1171. https://doi.org/10.2307/2532457

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 123–140. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. New York, NY: Chapman and Hall/CRC. https://doi.org/10.1201/9781315139470

Buri, M., & Hothorn, T. (2020). Model-based random forests for ordinal regression. *The International Journal of Biostatistics, 16*(2), 1–17. https://doi.org/10.1515/ijb-2019-0063

Christensen, R. H. B. (2022). ordinal: Regression models for ordinal data [Computer software manual]. (R package version 2022.11-16). https://CRAN.R-project.org/package=ordinal

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*(4), 213–220. https://doi.org/10.1037/h0026256

Cortez, P. (2014). *Student performance*. UCI Machine Learning Repository. https://doi.org/10.24432/C5TG7T

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). *Wine quality*. UCI Machine Learning Repository. https://doi.org/10.24432/C56S3T

Frank, E., & Hall, M. (2001). A simple approach to ordinal classification. In L. De Raedt, & P. Flach (Eds.), *Lecture Notes in Computer Science: Vol. 2167. Machine learning: ECML 2001* (pp. 145–156). https://doi.org/10.1007/3-540-44795-4_13

Friedrich, S., & Friede, T. (2024). On the role of benchmarking data sets and simulations in method comparison studies. *Biometrical Journal, 66*(1), 2200212. https://doi.org/10.1002/bimj.202200212

Galimberti, G., Soffritti, G., & Maso, M. D. (2012). Classification trees for ordinal responses in R: The rpartScore package. *Journal of Statistical Software, 47*(10), 1–25. https://doi.org/10.18637/jss.v047.i10

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*.

Hansen, N., & Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In Proceedings of *IEEE International Conference on Evolutionary Computation* (pp. 312–317). https://doi.org/10.1109/ICEC.1996.542381

Hornung, R. (2022). ordinalForest: Ordinal forests: Prediction and variable ranking with ordinal target variables [Computer software manual]. (R package version 2.4-3). https://CRAN.R-project.org/package=ordinalForest

Hornung, R. (2019). Ordinal forests. *Journal of Classification, 37*(1), 4–17. https://doi.org/10.1007/s00357-018-9302-x

Hothorn, T. (2023). TH.data: TH's data archive [Computer software manual]. (R package version 1.1-2). https://CRAN.R-project.org/package=TH.data

Hothorn, T., & Zeileis, A. (2015). partykit: A modular toolkit for recursive partitioning in R. *Journal of Machine Learning Research, 16*, 3905–3909. https://jmlr.org/papers/v16/hothorn15a.html

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics, 15*(3), 651–674. https://doi.org/10.1198/106186006X133933

Immekus, J. C., Jeong, T.-s., & Yoo, J. E. (2022). Machine learning procedures for predictor variable selection for schoolwork-related anxiety: Evidence from PISA 2015 mathematics, reading, and science assessments. *Large-scale Assessments in Education, 10*(1). https://doi.org/10.1186/s40536-022-00150-8

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). ISLR: Data for an introduction to statistical learning with applications in R [Computer software manual]. (R package version 1.4). https://CRAN.R-project.org/package=ISLR

Janitza, S., Tutz, G., & Boulesteix, A.-L. (2016). Random forest for ordinal responses: Prediction and variable selection. *Computational Statistics & Data Analysis, 96*, 57–73. https://doi.org/10.1016/j.csda.2015.10.005

Johnson, S. G. (2007). The NLopt nonlinear-optimization package [Computer software manual]. https://github.com/stevengj/nlopt

Kendall, M. G. (1945). The treatment of ties in ranking problems. *Biometrika, 33*(3), 239–251. https://doi.org/10.1093/biomet/33.3.239

Kendall, M. G. (1948). *Rank correlation methods*. London, UK: Griffin.

Kleiber, C., & Zeileis, A. (2008). AER: Applied econometrics with R [Computer software manual]. https://CRAN.Rproject.org/package=AER

Kramer, S., Widmer, G., Pfahringer, B., & de Groeve, M. (2000). Prediction of Ordinal Classes Using Regression Trees. In Z. W. Raś, & S. Ohsuga (Eds.), *Lecture Notes in Computer Science: Vol. 1932. Foundations of Intelligent Systems. ISMIS 2000* (pp. 426–434). https://doi.org/10.1007/3-540-39963-1_45

Leisch, F., & Dimitriadou, E. (2021). mlbench: Machine learning benchmark problems [Computer software manual]. (R package version 2.1-3.1)

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News, 2*(3), 18–22. https://CRAN.R-project.org/doc/Rnews/

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society: Series B (Methodological), 42*(2), 109–127. https://doi.org/10.1111/j.2517-6161.1980.tb01109.x

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal, 7*, 308–313. https://doi.org/10.1093/comjnl/7.4.308

Peterson, B., & Harrell, F. E. (1990). Partial proportional odds models for ordinal response variables. *Applied Statistics, 39*(2), 205. https://doi.org/10.2307/2347760

Pétrowski, A., & Ben-Hamida, S. (2017). *Evolutionary algorithms*. Hoboken, NJ: Wiley.

Piccarreta, R. (2007). Classification trees for ordinal variables. *Computational Statistics, 23*(3), 407–427. https://doi.org/10.1007/s00180-007-0077-5

Probst, P., Wright, M. N., & Boulesteix, A. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery, 9*(3), 1–15. https://doi.org/10.1002/widm.1301

R Core Team (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

Rowan, T. H. (1990). *Functional stability analysis of numerical algorithms* (Doctoral dissertation). Retrieved from ProQuest Dissertations and Theses database. (UMI No. 9031702)

Shen, W., Kiger, T. B., Davies, S. E., Rasch, R. L., Simon, K. M., & Ones, D. S. (2011). Samples in applied psychology: Over a decade of research in review. *Journal of Applied Psychology, 96*(5), 1055–1064. https://doi.org/10.1037/a0023322

Tutz, G. (2021). Ordinal trees and random forests: Score-free recursive partitioning and improved ensembles. *Journal of Classification, 39*(2), 241–263. https://doi.org/10.1007/s00357-021-09406-4

Tutz, G. (2022). Ordinal regression: A review and a taxonomy of models. *WIREs Computational Statistics, 14*(2), e1545. https://doi.org/10.1002/wics.1545

Tutz, G., & Berger, M. (2022). Sparser ordinal regression models based on parametric and additive location-shift approaches. *International Statistical Review, 90*(2), 306–327. https://doi.org/10.1111/insr.12484

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. https://doi.org/10.18637/jss.v077.i01

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer, 3*(1), 32–35. https://doi.org/10.1002/1097-0142(1950)3:1<32::aid-cncr2820030106>3.0.co;2-3

Zahid, F. M., & Tutz, G. (2013). Proportional odds models with high-dimensional data structure. *International Statistical Review, 81*(3), 388–406. https://doi.org/10.1111/insr.12032

Article 2

Buczak, P. (2025). Frequency-adjusted borders ordinal forest: A novel tree ensemble method for ordinal prediction. *British Journal of Mathematical and Statistical Psychology*, 78(2), 594–616. https://doi.org/10.1111/bmsp.12375

Received: 22 July 2024 | Accepted: 9 November 2024

DOI: 10.1111/bmsp.12375

the british psychological society
promoting excellence in psychology

ARTICLE

Frequency-adjusted borders ordinal forest: A novel tree ensemble method for ordinal prediction

Philip Buczak[1,2]

[1]Department of Statistics, TU Dortmund University, Dortmund, Germany

[2]Research Center Trustworthy Data Science and Security, UA Ruhr, Dortmund, Germany

Correspondence
Philip Buczak, Department of Statistics, TU Dortmund University, Dortmund, Germany.
Email: buczak@statistik.tu-dortmund.de

Abstract

Ordinal responses commonly occur in psychology, e.g., through school grades or rating scales. Where traditionally parametric statistical models like the proportional odds model have been used, machine learning (ML) methods such as random forest (RF) are increasingly employed for ordinal prediction. With new developments in assessment and new data sources yielding increasing quantities of data in the psychological sciences, such ML approaches promise high predictive performance. As RF does not inherently account for ordinality, several extensions have been proposed. A promising approach lies in assigning optimized numeric scores to the ordinal response categories and using regression RF. However, these optimization procedures are computationally expensive and have been shown to yield only situational benefit. In this work, I propose Frequency-Adjusted Borders Ordinal Forest (fabOF), a novel tree ensemble method for ordinal prediction forgoing extensive optimization while offering improved predictive performance in simulation and an illustrative example of student performance. To aid interpretation, I additionally introduce a permutation variable importance measure for fabOF tailored towards ordinal prediction. When applied to the illustrative example, an interest in higher education, mother's education, and study time are identified as important predictors of student performance. The presented methodology is made available through an accompanying R package.

KEYWORDS

machine learning, ordinal forest, ordinal prediction, random forest

1 | INTRODUCTION

Ordinal data are an everyday occurrence in psychology. School grades are strictly speaking ordinal, rating scales used in questionnaires produce ordinal responses, and raters award ordinal scores to performance in tasks or assessments. In statistical analyses, ordinal responses have long been modelled using parametric models. A well-known parametric model class (containing, e.g., the proportional odds model) are cumulative models, which assume that the ordinal response results from an underlying latent variable that is only observable through thresholds (McCullagh, 1980). For a more general overview of parametric models for ordinal response data, the reader is referred to Tutz (2022). Similarly, ordinal regression can be approached from a Bayesian perspective. To this end, Johnson and Albert (1999) and Agresti and Hitchcock (2005) provide an overview of Bayesian ordinal regression models. For a tutorial on fitting ordinal regression models in the Bayesian framework, I refer the reader to Bürkner and Vuorre (2019). In the more traditional statistical modelling literature for psychology, efforts have been made to popularize statistical models for ordinal data (e.g., Bürkner & Vuorre, 2019; Sönning et al., 2024). At the same time, machine learning (ML) methods have been on the rise in these fields (e.g., Fife & D'Onofrio, 2022; Hilbert et al., 2021; Ulitzsch et al., 2022). These prediction-oriented methods can easily handle large quantities of data, as are increasingly becoming available in the psychological sciences (Hilbert et al., 2021; Ulitzsch et al., 2022). Prediction is often of crucial interest for these fields, e.g., in aptitude-fit assessment for trying to predict which students might benefit from what support. Further, there is a growing interest in predicting ordinal data, such as school grades (e.g., Costa-Mendes et al., 2020) or creativity ratings (e.g., Buczak et al., 2022). These studies often rely on ML approaches that do not provide inherent support for ordinal responses such as random forest (RF; Breiman, 2001). To work around this limitation, different strategies have been proposed in the statistical and ML literature that enable the use of RF while accounting for the ordinal nature of the response. A common approach is assigning numeric scores and category borders to the ordinal response categories and using the scores as the outcome variable for training a regression RF and the category borders for predicting new observations (Buczak et al., 2024; Hornung, 2019). While in principle one could assign default scores of $1, 2, \ldots, k$ to the k ordinal response categories, it is unclear if such a choice is optimal and represents the assumed underlying latent variable well. Thus, ordinal forest (OF; Hornung, 2019) and the ordinal score optimization algorithm (OSOA; Buczak et al., 2024) both first aim to optimize the scores that are assigned to the ordinal categories. Split-based ordinal forest (RFSp) proposed by Tutz (2021), on the other hand, does not rely on using numeric scores for the ordinal response categories and instead transforms the ordinal prediction task into a series of binary prediction tasks for which regular binary classification RFs are trained respectively. Using the individual RF models, cumulative probabilities for the ordinal response categories can be computed.

Apart from RF, other ML methods have been extended to ordinal prediction as well. Examples include boosting algorithms (Riccardi et al., 2014; Tutz & Hechenbichler, 2005), support vector machine (Chu & Keerthi, 2007; Herbrich, 1999), and neural networks (e.g., Cao et al., 2020; Cheng et al., 2008; Shi et al., 2023). While the ML literature generally offers a plethora of different methods, tree-based methods have proven particularly effective for tabular data (Grinsztajn et al., 2022). As tabular data are common in psychology, and RF is currently among the most actively researched tree-based methods for ordinal prediction (e.g., Buczak et al., 2024; Hornung, 2019; Janitza et al., 2016; Tutz, 2021), this work will focus mainly on the use of RF for ordinal prediction. A further practical benefit of RF is its relative robustness of hyperparameter choices, whereas other ML methods may require a computationally expensive hyperparameter tuning beforehand (see Probst et al., 2019).

In their comparison studies, Tutz (2021) and Buczak et al. (2024) only found slight differences regarding the predictive performance of the different RF-based approaches. Further, Buczak et al. (2024) found the benefit of optimizing the scores assigned to the ordinal response categories as in OF and OSOA rather situational. The authors discuss possible reasons for the lack of consistent improvement such as the prediction procedure used in both methods. In this work, I introduce a novel prediction procedure for score-based RFs as sketched in Buczak et al. (2024). Further, I point out that the direct link

between scores and category borders (one is always determined by the other) in OF and OSOA can be another factor hindering improvement regarding predictive performance. To remedy this, I propose a novel score-based RF method called *Frequency-Adjusted Borders Ordinal Forest* (fabOF) which builds on the newly introduced prediction procedure and decouples the scores assigned to the ordinal response categories and the borders used for transforming the numeric predictions into ordinal categories. Instead of a computationally expensive optimization procedure as in OF and OSOA, fabOF chooses its borders using a heuristic based on the response category frequencies. Through simulation and an illustrative data example, I will demonstrate that fabOF improves upon the predictive performance of existing methods while requiring significantly less runtime than OF and OSOA, for example. Apart from predictive performance, explaining and interpreting results remains of crucial interest to psychological researchers, even in primarily predictive settings (Henninger et al., 2023). Particularly in the context of RF models, variable importance measures (VIMs) are a popular tool for quantifying the importance of individual covariates on model predictions (Molnar, 2022). While Janitza et al. (2016) proposed a VIM for ordinal prediction based on the ranked probability score (Epstein, 1969), their VIM is not applicable to the newly proposed fabOF, thereby necessitating an alternative solution. Therefore, I additionally introduce a custom VIM for fabOF that helps interpret the impact of individual covariates on fabOF model predictions.

The remainder of this paper is structured as follows. In the next section, I will present prior work on score-based RF approaches to ordinal prediction in more detail. Subsequently, I will contribute to the literature by introducing the alternative prediction procedure as well as the newly proposed fabOF method and its custom VIM. Further, I will evaluate fabOF as well as its permutation VIM through simulation and showcase both using an illustrative data example. The work closes with a discussion and potential avenues for further research.

2 | PRIOR WORK ON SCORE-BASED ORDINAL PREDICTION WITH RANDOM FOREST

Assigning numeric scores to ordinal response categories is a common approach to extending existing ML algorithms to the ordinal case. Kramer et al. (2000) used numeric scores for predicting ordinal responses with regression trees. Piccarreta (2007), Archer (2010), and Galimberti et al. (2012) extended split criteria in classification trees based on numeric scores. The conditional inference tree framework by Hothorn et al. (2006) uses permutation tests to assess the association between the outcome variable and covariates. In their framework, which supports nominal, ordinal, and numeric responses, ordinal response categories are transformed into numeric scores that can be prespecified by the user. Janitza et al. (2016) have studied in detail the use of conditional inference forests for ordinal classification. In this work, a particular focus is placed on the score-based RF methods OF (Hornung, 2019) and OSOA (Buczak et al., 2024).

Similar to the cumulative model, both methods assume that the ordinal response variable results from an underlying latent numeric variable. However, instead of the original numeric values, one can only observe a coarser version of the latent variable where individual observations take one of k ordinal categories. Both methods aim to approximate the latent variable by partitioning the $[0, 1]$ interval into k category-specific subintervals characterized by the category borders $0 \leq b_1 < b_2 < \cdots < b_{k+1} \leq 1$. These category intervals are represented by a numeric score where for the rth category, OF and OSOA use the midpoint of the corresponding category interval $[b_r, b_{r+1})$ as its representative score s_r, i.e., $s_r = \frac{b_r + b_{r+1}}{2}, r = 1, \ldots, k$ (Buczak et al., 2024; Hornung, 2019). Finally, the numeric scores (and category borders) are transformed using the quantile function of the standard normal distribution (probit function) and used to fit a regression RF model. As a toy example, consider the following partition of the $[0, 1]$ interval into five category subintervals: $[0, 0.2)$, $[0.2, 0.4)$, $[0.4, 0.6)$, $[0.6, 0.8)$ and $[0.8, 1]$. Using the midpoints of these intervals, the five categories would be represented by the numeric scores .1, .3, .5, .7,

and .9 respectively. Transforming these scores (and category borders) using the probit function leads to numeric values in the interval $(-\infty, \infty)$, which are in turn used to fit a regression RF that can output numeric predictions for new observations. Numeric predictions are translated back into ordinal categories using the category borders. Ignoring the probit transformation for the sake of the toy example, a numeric prediction of .15 for a given observation would fall into the first subinterval $[0, 0.2)$ and, thus, translate to the first ordinal category.

However, this toy example only showcases one possible partition of the $[0, 1]$ interval. In practice, it is not known which partition best represents the unknown underlying variable. Therefore, OF and OSOA employ an optimization procedure that aims to find the optimal partition regarding the predictive performance achieved with it. However, the two methods differ in their optimization strategy. OF starts by generating random sets of partitions. For each partition, a regression RF is trained using the numeric scores resulting from the respective partition as sketched in the toy example. The RF fits are evaluated using their out-of-bag (OOB) performance (i.e., for each observation only the trees which did not include the given observation in the training data are used for prediction and performance evaluation) as measured by Youden's index J (Youden, 1950). The partitions leading to the best performance are combined into a final partition that is used for fitting the final RF in a second step (Hornung, 2019). OSOA similarly evaluates partitions by their predictive performance using Youden's J. However, in contrast to OF, OSOA uses a non-linear optimization algorithm that iteratively searches for an optimal partition. As such, the optimization procedure in OSOA can explore promising regions in the solution space instead of relying on a pregenerated set of possible partitions (Buczak et al., 2024).

For the non-linear optimization, OSOA relies on the Sbplx optimizer from the `NLopt` library (Johnson, 2008), which is a variant of the Nelder–Mead method (Nelder & Mead, 1965). Relating back to the toy example, OSOA would select the partition $[0, 0.2)$, $[0.2, 0.4)$, $[0.4, 0.6)$, $[0.6, 0.8)$, and $[0.8, 1]$ as its starting point and would aim to iteratively pivot towards an optimal partition. Similar to OF, partitions are evaluated based on the predictive performance they achieve when training a regression RF and assessing its OOB performance. The internal optimizer traverses the space of possible partitions and aims to find the partition leading to the optimal value of Youden's index J. As such, OF and OSOA both aim to optimize the same performance measure, but with different means. Apart from their different optimization approaches, OF and OSOA are equivalent. In their comparison study, Buczak et al. (2024) found both methods to perform mostly similarly. Furthermore, the authors found the benefit of the score optimization to be situational as OF and OSOA could not consistently outperform a naive OF variant, which instead simply used the default scores $1, 2, \ldots, k$ and category borders $0.5, 1.5, \ldots, k + 0.5$.

Discussing potential improvements for OF and OSOA, Buczak et al. (2024) suggest using an alternative prediction procedure than the one currently employed. To predict new observations, both methods use the approach described in pseudocode in Algorithm 1. For all n_{pred} observations to be predicted, each of the B trees is used to generate a numeric score prediction $\hat{y}_{ij}^{\text{num}}$, with $i = 1, \ldots, n_{\text{pred}}$ and

Algorithm 1 **Transform-first-aggregate-after (TFAA) prediction**

Obtain numeric prediction $\hat{y}_{ij}^{\text{num}}$ for observation $i = 1, \ldots, n_{\text{pred}}$ from tree $j = 1, \ldots, B$.

Transform $\hat{y}_{ij}^{\text{num}}$ into category prediction $\hat{y}_{ij}^{\text{cat}}$ using category borders $b_1, \ldots, b_{k+1}$.

Determine aggregated category prediction via majority voting over all tree-level category predictions, i.e., $\hat{y}_i^{\text{cat}} = \text{mode}(\hat{y}_{i1}^{\text{cat}}, \ldots, \hat{y}_{iB}^{\text{cat}})$.

$j = 1, \ldots, B$. Using the category borders $b_1, \ldots, b_{k+1}$ of the k response categories, the numeric score prediction $\hat{y}_{ij}^{\text{num}}$ can be transformed into a category label prediction $\hat{y}_{ij}^{\text{cat}}$. For each observation, the category label predictions from the B trees are aggregated via majority voting, resulting in the category label prediction for observation i, $\hat{y}_i^{\text{cat}} = \text{mode}(\hat{y}_{i1}^{\text{cat}}, \ldots, \hat{y}_{iB}^{\text{cat}})$. As this procedure first transforms the numeric score predictions into a category label and then aggregates all predicted category labels into a single category label prediction, it will be referred to as *transform-first-aggregate-after* (TFAA) prediction in what follows.

Buczak et al. (2024) argue that transforming numeric scores into category labels already at the tree level may limit the impact of the actual choices of the numeric scores. Instead, the authors suggest exploring an alternative prediction approach where the numeric predictions from the individual trees are first aggregated by averaging and then transforming the aggregated numeric prediction to a category label. I follow their suggestion and introduce *aggregate-first-transform-after* (AFTA) prediction in the next section. Additionally, I address another limitation of OF and OSOA. For both methods, scores and category borders are directly linked since the scores are always determined as the midpoints of the category intervals. While always representing the categories by their midpoint may seem intuitive, it is not necessarily a decision based on predictive performance for the given data context. Therefore, separating category scores and borders from one another allows for additional flexibility. To this end, I propose fabOF, a novel score-based RF method for ordinal prediction that builds on AFTA prediction and determines its category borders separately from its category scores in an adaptive way.

3 | NEW CONTRIBUTIONS TO SCORE-BASED ORDINAL PREDICTION WITH RANDOM FOREST

3.1 | Aggregate-first-transform-after prediction

AFTA prediction is described in pseudocode in Algorithm 2. In contrast to TFAA prediction, the tree-level numeric score predictions $\hat{y}_{ij}^{\text{num}}, i = 1, \ldots, n_{\text{pred}}, j = 1, \ldots, B$ are first averaged to obtain an aggregated numeric score prediction, i.e.,

$$\hat{y}_i^{\text{num}} = \frac{1}{B} \sum_{j=1}^{B} \hat{y}_{ij}^{\text{num}}.$$

By using the category borders, the aggregated score prediction can in turn be transformed into a category label prediction $\hat{y}_i^{\text{cat}}$. In this work, I investigate the use of AFTA prediction for existing methods such as OF and OSOA as well as for the newly proposed fabOF introduced in what follows.

Algorithm 2 **Aggregate-first-transform-after (AFTA) prediction**

Obtain numeric prediction $\hat{y}_{ij}^{\text{num}}$ for observation $i = 1, \ldots, n$ from tree $j = 1, \ldots, B$.

Compute aggregated numeric prediction $\hat{y}_i^{\text{num}} = \frac{1}{B} \sum_{j=1}^{B} \hat{y}_{ij}^{\text{num}}$.

Transform $\hat{y}_i^{\text{num}}$ into category prediction $\hat{y}_i^{\text{cat}}$ using category borders $b_1, \ldots, b_{k+1}$.

3.2 | Frequency-adjusted borders ordinal forest

The new fabOF method follows (naive) OF and OSOA in assigning scores to ordinal categories and using these to train a regression RF, which in turn outputs numeric score predictions for new observations that are transformed back into ordinal categories via category borders. However, in contrast to OF and OSOA, fabOF relies on AFTA prediction and separates the choice of scores and category borders. While in principle one could extend the optimization procedure of OF and OSOA to simultaneously optimize scores and category borders, such an approach would greatly increase the complexity of the optimization problem. Instead, fabOF employs a heuristic for deriving its category borders based on the distribution of the ordinal response categories. This design decision is motivated by the results obtained by Buczak et al. (2024) and Hornung (2019), both of whom indicate the response category distributions affect the predictive performance of ordinal prediction methods. I will demonstrate through simulation that combining AFTA prediction with the heuristic already leads to consistent improvements over existing methods in many cases while using the default scores $1, 2, \ldots, k$. As such, fabOF does not make use of a computationally expensive optimization step.

Algorithm 3 describes fabOF in detail. After assigning the default scores $1, 2, \ldots, k$ to the ordinal response categories, a regression RF is trained using the respective covariates as predictors and the numeric score variable Y^{num} as the target variable. From the RF model, numeric OOB predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$ for all observations are obtained. These numeric OOB predictions already constitute the aggregation step from the AFTA prediction approach since OOB predictions are generated at the tree level and then averaged into a combined OOB prediction for each observation. A key idea of fabOF is to employ a heuristic that determines the category borders in a way that matches the distribution of the predicted categories with the category distribution one would expect in the general population. To this end, fabOF computes the cumulative relative frequencies $\pi_1, \pi_2, \ldots, \pi_{k-1}$ of the ordinal response categories in the training data up to (but not including) category k. The inner set of category borders, i.e., $b_2, \ldots, b_k$, are then determined by the quantiles $q_{\pi_1}, q_{\pi_2}, \ldots, q_{\pi_{k-1}}$ of the OOB predictions for probabilities $\pi_1, \pi_2, \ldots, \pi_{k-1}$. The bounding borders, i.e., b_1 and b_{k+1}, are set to 1 and k respectively, as they

Algorithm 3 **Frequency-Adjusted Borders Ordinal Forest (fabOF)**

procedure FABOF

Assign scores $(s_1, s_2, \ldots, s_k) \leftarrow (1, 2, \ldots, k)$.

Fit regression RF using numeric score variable $Y^{\text{num}} \in \{s_1, s_2, \ldots, s_k\}$ as target and respective covariates as predictors.

Obtain numeric OOB predictions $\hat{y}_i^{\text{num}}, i = 1, \ldots, n$, from RF model.

Compute cumulative relative frequencies $\pi_1, \pi_2, \ldots, \pi_{k-1}$ of ordinal response categories up to (and not including) category k.

Compute quantiles $q_{\pi_1}, q_{\pi_2}, \ldots, q_{\pi_{k-1}}$ of OOB predictions for probabilities $\pi_1, \pi_2, \ldots, \pi_{k-1}$.

Assign category borders $(b_1, b_2, \ldots, b_k, b_{k+1}) \leftarrow \left(s_1, q_{\pi_1}, \ldots, q_{\pi_{k-1}}, s_k\right)$.

return RF model and borders

end procedure

represent the minimum and maximum prediction values possible. By assigning the respective quantiles of the OOB predictions, the category borders are chosen such that the distribution of the predicted categories approximates the category distribution in the training data. This implicitly assumes that the training dataset is a suitable representation of the general population. The benefit of using the OOB predictions for determining the borders, instead of, for example, predictions obtained using the entire forest for each observation in the training set is that it reduces the risk of overfitting. As such, one can mimic the process of adjusting the borders to perform well on unseen data without having to set aside a separate set of data points solely to determine the appropriate borders. In the final step, fabOF returns the trained RF model as well as the category borders for predicting new observations. The proposed method is implemented in the `R` package `fabOF`, currently available via GitHub (https://github.com/phibuc/fabOF).

Variable importance measure

While it can offer high predictive performance, a drawback of RF is its lack of interpretability (Henninger et al., 2023). However, when used, for example, in the context of predicting student attainment, it may not necessarily only be of interest to achieve high predictive model performance, but also to learn which factors, i.e., covariates, play a particularly influential role in the process. To aid interpretation of RF models in this regard, VIMs are often computed. VIMs aim to assess the importance of covariates by quantifying their impact on the predictive performance of a RF model (Molnar, 2022). A common class of VIMs are permutation VIMs (Breiman, 2001). To this end, the values of a given covariate are permuted and the resulting loss in predictive performance is measured. The logic behind permutation VIMs is that for important covariates, permuting the values leads to a larger loss in predictive performance as compared to less important covariates, because permutation voids the original information contained in the covariate (Molnar, 2022). As fabOF directly builds on RF, variable importance can be used to enhance the interpretability of fabOF as well. For ordinal prediction, Janitza et al. (2016) proposed a permutation VIM based on the ranked probability score (RPS; Epstein, 1969). Similarly, OF as implemented in the `ordinalForest` package (Hornung, 2022) offers the possibility of computing variable importance based on the RPS or on classification accuracy (i.e., ignoring ordinality). However, the RPS operates on the predicted ordinal response category probabilities, which are not available for fabOF. As such, a VIM relying on the RPS cannot be used for fabOF. Instead, I propose a custom permutation VIM for fabOF based on weighted Kappa as presented in pseudocode in Algorithm 4. It follows the classic flow of permutation VIMs as described in Fisher et al. (2019), where for each of the p covariates, the covariate values are permuted and the variable importance is computed as the difference in performance when using the original and the permuted data respectively. To account for the ordinality of the responses, weighted Kappa (Cohen, 1968) with linear weights (denoted by κ^{lin}) is used to measure predictive performance.

Typically, VIMs for RFs are computed at the tree level such that for each tree variable importance values for all covariates are obtained, which in turn are averaged, resulting in a single forest-level importance value for each covariate. Since fabOF does not transform its numerical predictions into ordinal categories until the former are aggregated at the forest level, it is not feasible to compute variable importance based on an ordinal loss function at the tree level with fabOF. Computing variable importance at the forest level only, however, would mean that each covariate is permuted only once (as opposed to once per tree as in the typical approach), which could lead to unstable results (cf. Molnar, 2022). Therefore, fabOF's permutation VIM replicates the permutation process for each variable `reps` times where `reps` is a prespecified parameter modulating the trade-off between stability of the VIM results and computation time.

Algorithm 4 **Permutation VIM for fabOF**

Compute original OOB model performance $\kappa^{\text{lin}}_{\text{orig}}$.

for q in $1, \ldots, p$ **do**

  for v in $1, \ldots,$ `reps` **do**

    Permute values of variable q in original data.

    Compute OOB performance $\kappa_{q,v}$ based on predictions for permuted data.

    Compute $VI_{q,v} = \kappa^{\text{lin}}_{\text{orig}} - \kappa^{\text{lin}}_{q,v}$.

  end for

  Compute variable importance for variable q as $VI_q = \frac{1}{\texttt{reps}} \sum_{v=1}^{\texttt{reps}} VI_{q,v}$.

end for

4 | SIMULATION STUDY 1: PREDICTIVE PERFORMANCE

4.1 | Simulation setup

To evaluate fabOF, I performed a simulation study that compared fabOF with existing ordinal prediction methods such as (naive) OF, OSOA, RFSp, and a proportional odds model (referred to as CLM and specified with all linear main effects) as well as modified versions of (naive) OF and OSOA which employ AFTA prediction. For further computational details including software implementations and parameter choices, I refer the reader to Appendix A. The simulation setup was inspired by the simulation studies in Janitza et al. (2016) and Buczak et al. (2024). I simulated datasets with $n \in \{750, 1500\}$ observations and $p = 15$ standard normally distributed and uncorrelated covariates. Six of the 15 covariates were influential, while the remaining were noise variables. To create diverse data scenarios, I created four different data-generating processes (DGPs). To this end, I used two linear predictor functions $g_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$ with

$$\begin{aligned} g_1(\boldsymbol{x}) &= \begin{cases} 1, & x_1 \in (-1,1] \\ -1, & x_1 \notin (-1,1] \end{cases} + 1_{x_2>0} + 0.75x_3 + 0.25x_3^2 + 0.75x_4 + 0.25 \cdot 1_{x_2>0.5 \wedge x_4 \leq 0.5} + 0.5x_5 + 0.5x_6, \\ g_2(\boldsymbol{x}) &= x_1 + x_2 - x_3 - x_4 + x_6. \end{aligned}$$

DGP 1 was simulated from a proportional odds model using $g_1(\boldsymbol{x})$ as the linear predictor. As such, the probability of the ordinal response Y taking at most category $r = 1, \ldots, k$ was simulated as

$$P(Y \leq r|\boldsymbol{x}) = \frac{\exp(\gamma_r + g_1(\boldsymbol{x}))}{1 + \exp(\gamma_r + g_1(\boldsymbol{x}))},$$

where γ_r is the respective threshold value with $-\infty < \gamma_1 < \cdots < \gamma_k = \infty$. For DGP 2, I added standard normal noise to $g_1(\boldsymbol{x})$, resulting in a linear regression model. The simulated numeric outcomes from the linear regression model were transformed into an ordinal response by binning. As binning is commonly used in practice, DGP 2 was designed to correspond to this frequent use-case of ordinal prediction methods. DGP 3 was simulated from a proportional odds model where the linear predictor $g(\boldsymbol{x})$ was obtained as a linear combination of $g_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$, with

$$g(\boldsymbol{x}) = 0.6g_1(\boldsymbol{x}) + 0.4g_2(\boldsymbol{x}).$$

For DGP 4, I replaced the normally distributed covariates X_5 and X_{15} in DGP 1 by binary covariates. As DGPs 1–3 only included normally distributed covariates, the aim of DGP 4 was to check whether the presence of two binary covariates (one being influential, the other being noise) affects the performance of the prediction methods. Both binary covariates were simulated from a Bernoulli distribution with probability .5. To better reflect the original effect magnitude in DGP 1, I increased the effect of X_5 from .5 to 1 for DGP 4. The covariate X_{15} remained a noise predictor for DGP 4. The effect sizes in $g_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$ as well as the mixture component approach and mixture parameter choice were inspired by Janitza et al. (2016). Combining non-linear and linear effects as in $g_1(\boldsymbol{x})$ was inspired by Buczak et al. (2024). I introduced further variety into the data generation using two different distribution patterns for the ordinal response categories, similar to Hornung (2019) and Buczak et al. (2024). By choosing specific threshold and binning values for the DGPs, I created a pattern of equally distributed categories, and a pattern with prominent middle categories (referred to as wide middle pattern) emerged. For more details on creating the patterns, see Appendix B.

The different prediction methods were evaluated using weighted Kappa with linear and quadratic weights (see Appendix C for more details) as well as Kendall's rank correlation (Kendall, 1945). Both have been used on ordinal prediction (e.g., Ben-David, 2008; Buczak et al., 2024; Hornung, 2019). To this end, $\frac{2n}{3}$ observations were used to fit the prediction model while the remaining observations were used for evaluating the model on unseen data. Since fabOF determines its category borders based on the frequencies of the ordinal response categories in the training data, it works on the assumption that the training data are a suitable representation of the general population data. To study how fabOF reacts to a violation of this assumption, I considered two different sampling procedures in the simulation. After generating an initial sample of 10,000 observations (based on the DGP and response distribution pattern as specified by the respective simulation condition), I either drew a category-stratified subsample or a random subsample without stratification of size $\frac{2n}{3}$ for the training set. The test set of $\frac{n}{3}$ observations was sampled with category stratification in both cases. This created two scenarios, where for the first scenario both training and test sets were a suitable representation of the general population, while for the second scenario the category distribution in the training set was (potentially) a misrepresentation of the category distribution in the population sample. All conditions were simulated with 1000 replications. As the development of the `fabOF` `R` package began at a later stage, a prototype implementation for fabOF was used in the simulation. This prototype implementation uses the exact same routine as the internal functions of the `fabOF` `R` package and differs only in its user interface and input handling. The prototype implementation is available from the corresponding OSF repository of this work at https://osf.io/fn8bg/.

4.2 | Simulation results

In the following section, I will present the results from the simulation for each DGP. From the simulation parameters, the number of categories and the sampling procedure did not affect the results, while the impact of an increased number of observations was mostly limited to reducing the variability of the results as well as increasing the disparity between the CLM and the RF-based methods. Therefore, I will only show results for simulation scenarios with $n = 750$ observations, $k = 5$ categories, and stratified sampling of the training sets in the following due to space reasons. For the remaining results, I refer to the Supporting Information.

Results for DGP 1

Figure 1 shows the results for DGP 1 for the two response category distribution patterns. For weighted Kappa with linear and quadratic weights, fabOF generally reached the best performance alongside OF

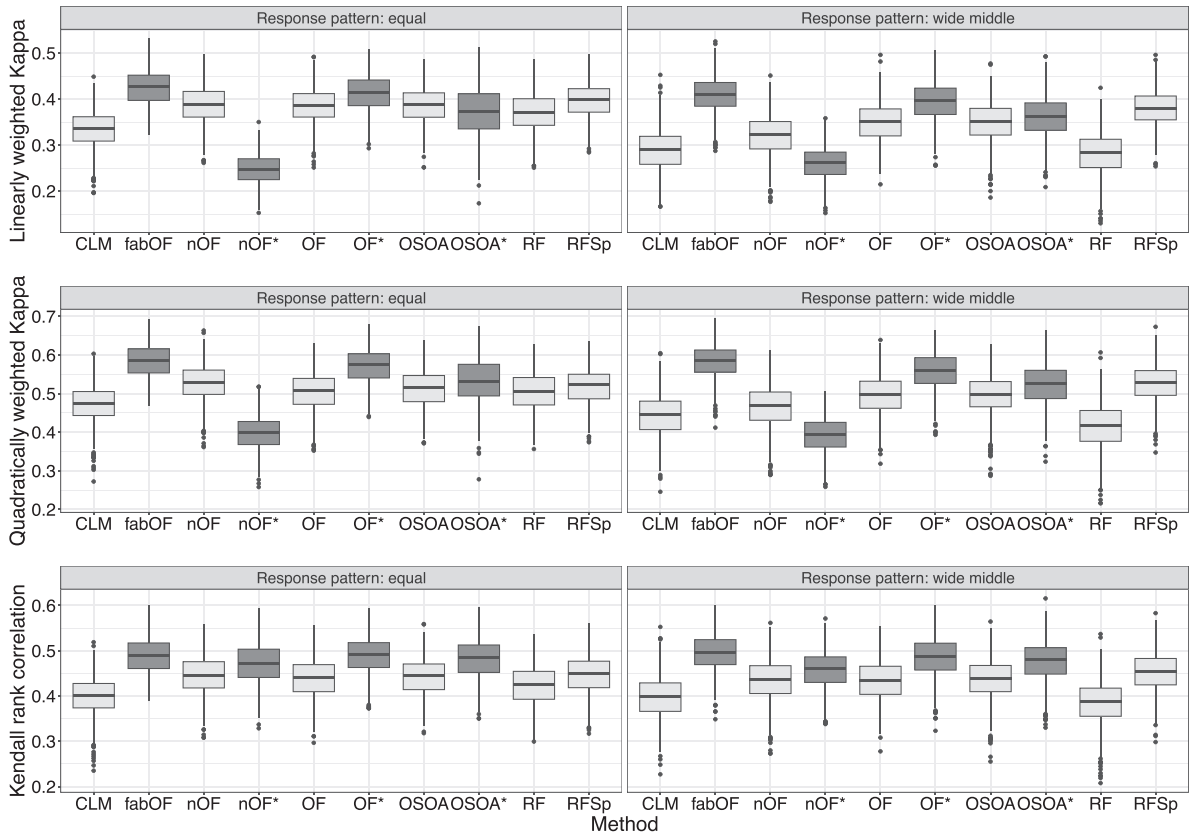

FIGURE 1 Predictive performance of all methods and modifications for data simulated from DGP 1 with $n = 750$, $k = 5$, and stratified sampling for both response category distribution patterns. Approaches using aggregate-first-transform-after prediction indicated through dark grey coloured boxplots with additional asterisk indicating modification of existing method. CLM, cumulative link model (proportional odds); fabOF, Frequency-Adjusted Borders Ordinal Forest; nOF, naive ordinal forest; OF, ordinal forest; OSOA, ordinal score optimization algorithm; RF, random forest; RFSp, split-based ordinal forest.

with AFTA prediction. Overall, the existing ordinal prediction methods performed similarly in the case of equally distributed response categories, while finer differences emerged for the wide middle pattern. For the latter case, all existing methods except RFSp suffered from the imbalanced data scenario. fabOF, on the other hand, held up quite well and gained more ground on most of the other methods. When Kendall's rank correlation was used as the performance measure, the differences between the individual methods (and modifications) were more subtle. For both response category distribution patterns, fabOF as well as AFTA-modified OF and OSOA tended to achieve slightly higher correlation scores than the remaining methods. Overall, the effect of AFTA prediction was rather mixed. While it increased the performance of OF and tended to do so as well for OSOA, the effect direction for naive OF was dependent on the performance measure.

Results for DGP 2

Figure 2 displays the results for DGP 2 for which, in contrast to DGP 1, the ordinal response was generated by binning numeric responses simulated from a linear regression model. Overall, these results mirror the results obtained for DGP 1. This is not surprising as the effect structure from DGP 1 carried over to DGP 2, and the only difference between the two DGPs was the model used for simulating the outcome. fabOF generally led to the best performance for weighted Kappa with linear and quadratic weights as well as Kendall's rank correlation. The differences were more apparent for weighted Kappa and the wide middle response category distribution pattern, while they were smaller for Kendall rank correlation scores. The AFTA prediction modified versions of OF and OSOA trailed only slightly behind fabOF and were also on par in some scenarios. While AFTA

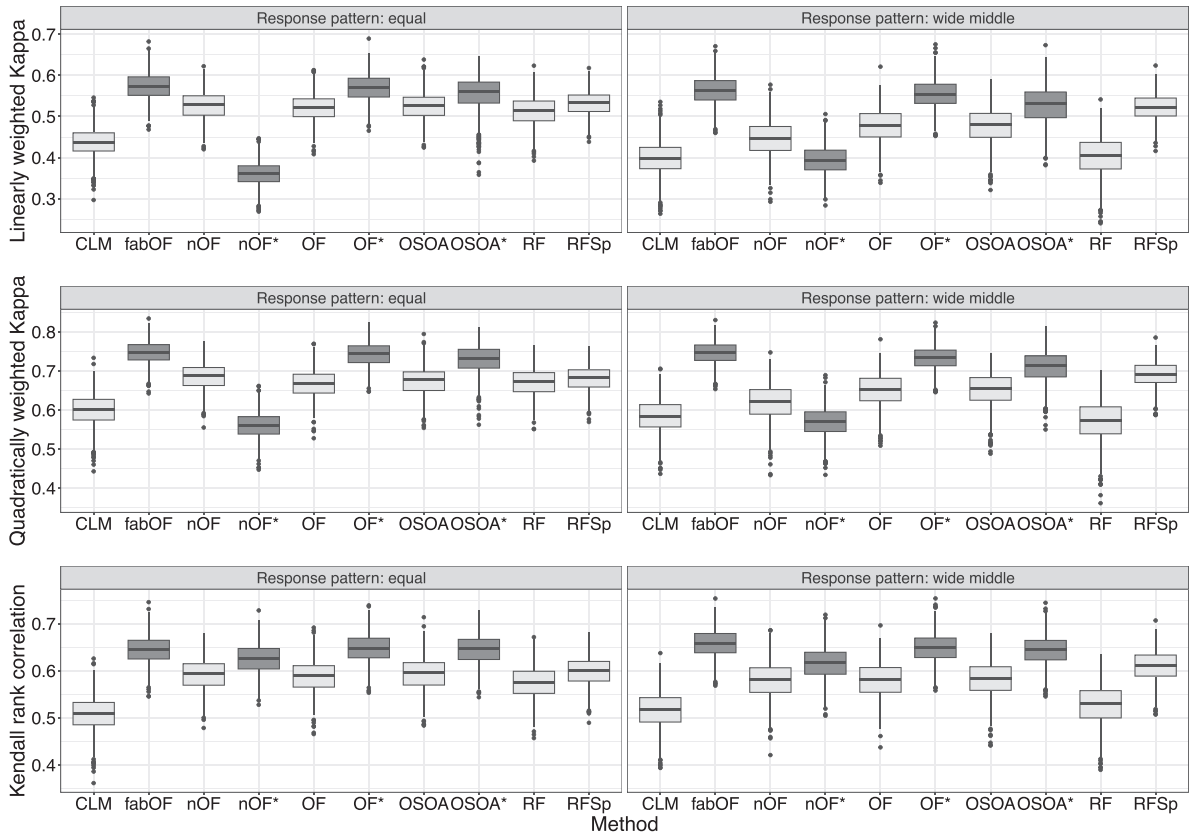

FIGURE 2 Predictive performance of all methods and modifications for data simulated from DGP 2 with $n = 750$, $k = 5$, and stratified sampling for both response category distribution patterns. Approaches using aggregate-first-transform-after prediction indicated through dark grey coloured boxplots, with additional asterisk indicating modification of existing method. CLM, cumulative link model (proportional odds); fabOF, Frequency-Adjusted Borders Ordinal Forest; nOF, naive ordinal forest; OF, ordinal forest; OSOA, ordinal score optimization algorithm; RF, random forest; RFSp, split-based ordinal forest.

prediction benefited the predictive performance of OF and OSOA for all measures, it notably decreased the performance of naive OF for weighted Kappa and slightly increased it for Kendall's rank correlation.

Results for DGP 3

Figure 3 shows the results for DGP 3 in which the outcome was simulated through a mixture distribution. Compared to DGP 1 and DGP 2, the results were more homogeneous between methods while still adhering to the result patterns observed for DGPs 1 and 2. For weighted Kappa, the best performance was achieved by fabOF and AFTA-modified OF. For Kendall's rank correlation, all methods were much closer in performance while still displaying a slight advantage for methods based on or modified with AFTA prediction. As before, only OF benefited consistently from the AFTA modification, while the effects were mixed and mostly subtle for OSOA. For naive OF, AFTA prediction greatly decreased predictive performance for weighted Kappa and slightly increased it for Kendall's rank correlation.

Results for DGP 4

Figure 4 shows the results for DGP 4 in which the effect structure of DGP 1 was modified to include two binary covariates, of which one was influential and one was noise. In comparison to the results for DGP 1, the inclusion of binary covariates did not affect the results to a notable degree. For

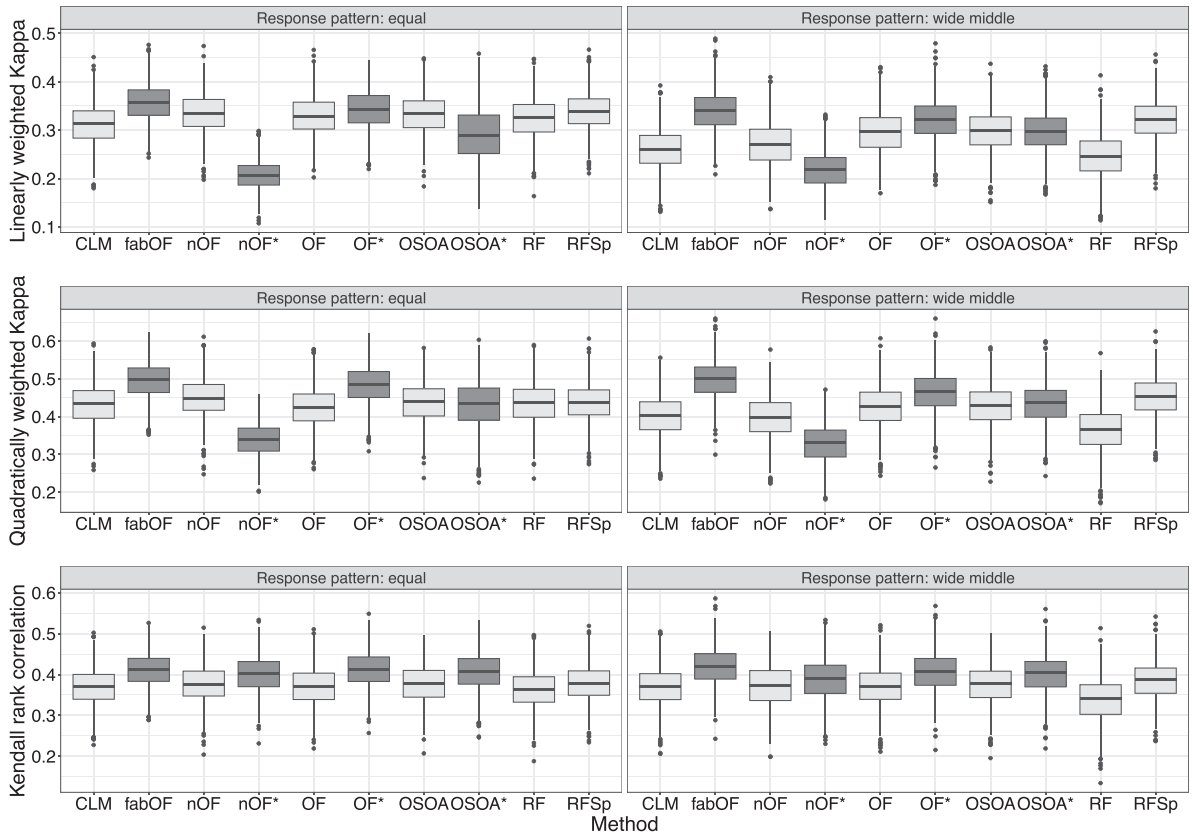

FIGURE 3 Predictive performance of all methods and modifications for data simulated from DGP 3 with $n = 750$, $k = 5$, and stratified sampling for both response category distribution patterns. Approaches using aggregate-first-transform-after prediction indicated through dark grey coloured boxplots, with additional asterisk indicating modification of existing method. CLM, cumulative link model (proportional odds); fabOF, Frequency-Adjusted Borders Ordinal Forest; nOF, naive ordinal forest; OF, ordinal forest; OSOA, ordinal score optimization algorithm; RF, random forest; RFSp, split-based ordinal forest.

weighted Kappa, fabOF and AFTA-modified OF achieved the highest predictive performance, while for Kendall's rank correlation, AFTA-modified OSOA was on par with the former two methods. For the wide middle response pattern, fabOF's heuristic led to slight performance advantages when compared to the pattern of equally distributed responses. Similar to the previous DGPs, AFTA prediction increased OF's predictive performance for all measures while the results were more mixed for OSOA and for naive OF.

4.3 | Runtime comparison

Buczak et al. (2024) have shown that apart from the predictive performance, the runtime of ordinal prediction methods is important to consider as well, especially given the costly optimization procedures in OF and OSOA. To this end, I also compared the computation time required by the methods during the simulation in a similar fashion to that in Buczak et al. (2024). Note that since the computation of the simulation was carried out using a compute cluster, it cannot be guaranteed that the individual computations ran in perfectly comparable conditions (e.g., regarding the node selected by the cluster's workload manager or regarding the overall workload of the cluster at a given time). Further, all methods were restricted to only use one CPU for computation. This may have put methods relying on parallelization at a disadvantage. However, all RF-based methods except for RFSp (which uses the randomForest package; Liaw & Wiener, 2002) used the RF implementation from the same package (i.e., ranger; Wright & Ziegler, 2017). In any case, the runtime results presented here should not be seen as an exact comparison, but rather as a ballpark estimate. As the choice of DGP, category distribution pattern or sampling strategy did not impact the runtime, I only show results for DGP 1 with equally distributed

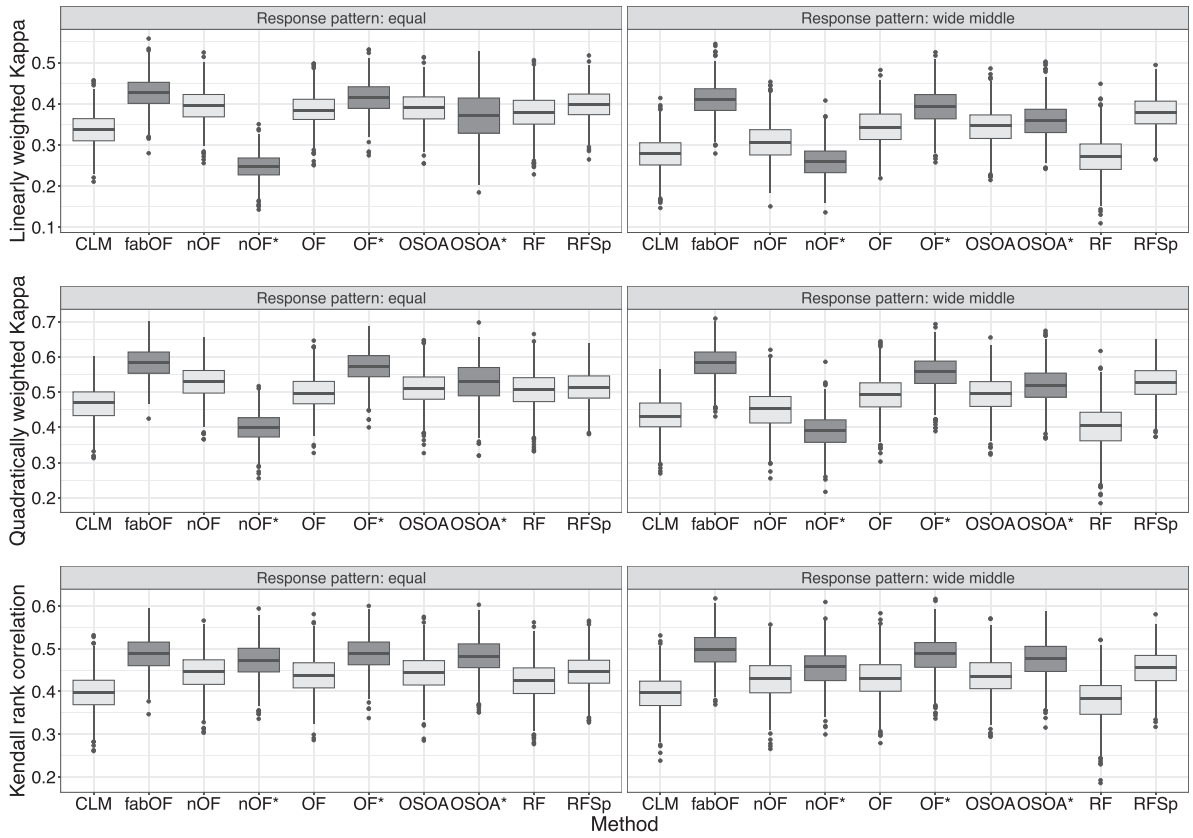

FIGURE 4 Predictive performance of all methods and modifications for data simulated from DGP 4 with $n = 750$, $k = 5$, and stratified sampling for both response category distribution patterns. Approaches using aggregate-first-transform-after prediction indicated through dark grey coloured boxplots with additional asterisk indicating modification of existing method. CLM, cumulative link model (proportional odds); fabOF, Frequency-Adjusted Borders Ordinal Forest; nOF, naive ordinal forest; OF, ordinal forest; OSOA, ordinal score optimization algorithm; RF, random forest; RFSp, split-based ordinal forest.

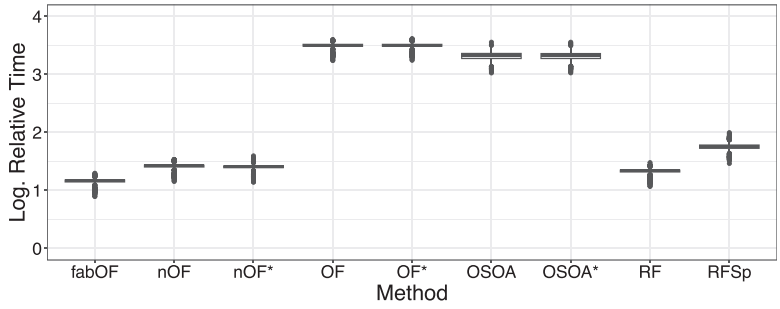

FIGURE 5 Logarithmized runtime (using base 10) relative to CLM for RF-based methods. Modification of (naive) OF and OSOA with aggregate-first-transform-after prediction indicated through asterisk. CLM, cumulative link model (proportional odds); fabOF, Frequency-Adjusted Borders Ordinal Forest; nOF, naive ordinal forest; OF, ordinal forest; OSOA, ordinal score optimization algorithm; RF, random forest; RFSp, split-based ordinal forest.

response categories and stratified sampling. Further, I limit the number of categories to five. While the other methods are not impacted by the choice of k, RFSp fits k-1 RF models. As such, the number of categories is directly linked with the runtime of RFSp. For consistency with the presentation of the simulation results above, I selected $k = 5$ for the runtime comparison here. For lower values of k, lesser runtimes are to be expected, and for higher values of k, longer runtimes of RFSp are to be expected.

Figure 5 shows the relative runtimes of the RF-based methods when using the CLM as the reference method (similar to Buczak et al., 2024). Relative runtimes have the advantage of being less dependent on the machine used for computation. As the CLM is the computationally least expensive method from the methods considered here, it serves as a sensible reference. For better visibility, the relative runtimes were

logarithmized with a base of 10. As such, values of 0 indicate that a given method required the same runtime as the CLM in a given run, while a value of 1 indicates that a given method's runtime was larger than the CLM runtime by a factor of 10. It can be seen that from the set of RF-based methods, fabOF achieved the lowest runtimes together with RF and naive OF. As RFSp had to fit four RF models in this case, it slightly trailed the three leading methods, as was to be expected. As already seen in Buczak et al. (2024), the optimization procedures make OF and OSOA computationally quite expensive. However, their runtime is also directly affected by the resources allotted to the optimization process (i.e., number of score sets for OF or maximum number of evaluations for OSOA). For OF and OSOA, changing the prediction procedure did not notably impact the runtime. This is not surprising as the bulk of the runtime is spent during the optimization procedure and the prediction method is only used once at the very end with the final model.

While fabOF achieved even lower runtimes than RF and naive OF, the differences between these three methods should not be overstated since the prototype fabOF implementation used for the simulation did not use any form of input checking which might add more overhead. Still, the results show a significant disparity in (relative) runtime between fabOF and OF which both were the best performing methods (using AFTA prediction for OF) when considering predictive performance. Since fabOF only needed to fit a single RF model and achieved results similar to those of OF with its extensive optimization procedure, this represents a meaningful advantage of fabOF.

5 | SIMULATION STUDY 2: VARIABLE IMPORTANCE

Apart from evaluating the predictive performance of fabOF, I also evaluated the custom permutation VIM of fabOF using simulation data. To investigate whether the VIM consistently discovered the influential covariates, I computed the variable importance for data generated according to DGP 1 (including only normally distributed covariates) and DGP 4 (including 13 normally distributed and two binary covariates) from the first simulation study. For each DGP, I simulated 100 datasets of size 1000 and computed the variable importance using 100 permutation replications for a fabOF model consisting of 500 trees.

To visualize the results, I followed the `iml` package (Molnar et al., 2018), i.e., for each covariate the range of importance values between the 5% and 95% quantiles are displayed with a dot indicating the median variable importance. For DGP 1, Figure 6 shows that fabOF's VIM was able to recover the influential covariates quite well. All six influential covariates $X_1, \ldots, X_6$ achieved notably higher

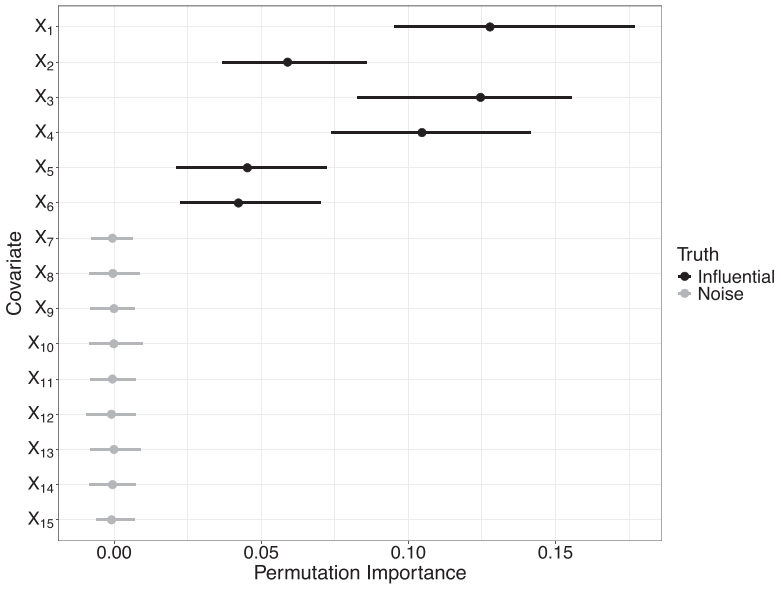

FIGURE 6 Permutation variable importance values for data generated using DGP 1. Colour coding indicates whether covariates were simulated as influential or as noise.

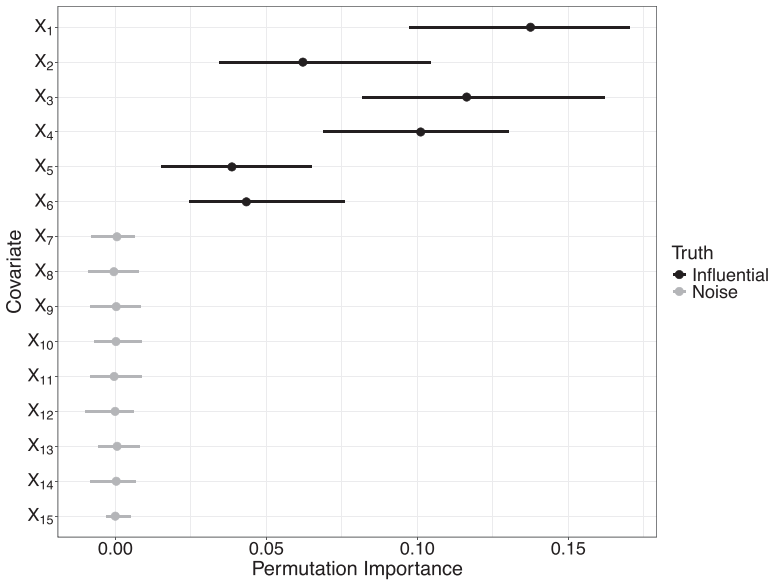

FIGURE 7 Permutation variable importance values for data generated using DGP 4. Colour coding indicates whether covariates were simulated as influential or as noise.

importance values than the noise variables $X_7, \ldots, X_{15}$. For the latter, importance values around 0 were mostly observed.

For DGP 4 in which X_5 and X_{15} were replaced by binary covariates, a similar picture emerged in Figure 7. The permutation VIM was able to distinguish between influential and noise covariates fairly well. Again, the noise covariates achieved importance values around 0 while all influential covariates reached higher importance values. When comparing the importance of X_5 for DGP 1 (where it was a normally distributed covariate) and DGP 4 (where it was a binary covariate), it can be seen that the importance values were slightly lower for the binary case. A potential explanation could be that the effect magnitude was not translated equivalently when changing from to DGP 1 to DGP 4. Another explanation may be that RFs are known to be biased towards covariates with many different split points, which in turn can also affect VIM results (Strobl et al., 2007).

6 | ILLUSTRATIVE DATA EXAMPLE

Apart from simulation data, I also evaluated fabOF on the basis of an illustrative data example on student performance in a language course from two Portuguese high schools. The original data were first introduced in Cortez and Silva (2008). I used the dataset provided in Cortez (2014) for this work, which consisted of 649 observations and 30 covariates with no missing values present. Further, I considered the same subset of the data already analysed in Buczak et al. (2024), which used 12 of the 30 original covariates. These included age, gender, residence (rural or urban), parental education status, parental cohabitation status, educational support from the family and the school, taking of private tutoring, internet access at home, and interest in pursuing higher education. The target variable is the final grade in a Portuguese language course. The original grades ranging from 0 to 20 were binned using the same binning values as in Cortez and Silva (2008), resulting in five categories: 0–9 ($n = 100$), 10–11 ($n = 201$), 12–13 ($n = 154$), 14–15 ($n = 112$), 16–20 ($n = 82$). For comparison, I used the same prediction methods as in the first simulation study. Predictive performance was assessed using Cohen's weighted Kappa (Cohen, 1968) with linear and quadratic weights as well as Kendall's rank correlation (Kendall, 1945). To avoid overconfident performance values, I used a five-fold cross-validation (CV) with 50 replications, where the ordinal prediction models were trained on the respective training set of the CV partition and evaluated on the test set.

Figure 8 shows that fabOF achieved the highest weighted Kappa values for both linear and quadratic weights, outperforming existing and AFTA-modified methods. For Kendall's rank correlation, fabOF was slightly ahead as well, although the methods were closer in performance for this performance measure. For OF, AFTA prediction improved the predictive performance for all three measures. In the case of OSOA, performance increased slightly for Kendall's rank correlation but remained mostly unaffected for weighted Kappa. For naive OF, AFTA prediction slightly increased performance for Kendall's rank correlation but notably decreased it for Cohen's weighted Kappa.

Overall, the increased predictive performance of fabOF over existing methods is a promising finding as data-driven prediction of student performance can play a key role in informing policymaking and the establishment of student support systems (Costa-Mendes et al., 2020; van der Scheer & Visscher, 2017).

For more detailed insights about the impact of individual covariates on the prediction of student performance, I computed variable importance values using fabOF's custom permutation VIM (with 100 replications). Figure 9 shows that the most important variables are an interest in higher education, mother's education, and study time. These findings are largely consistent with the educational research literature where interest in (higher) education has been found to be an important motivational variable associated with student performance (Hidi & Harackiewicz, 2000). Parental education has also been found to be a significant predictor of student achievement (e.g., Wößmann, 2003), with mothers particularly impacting a child's educational attainment (Cabus & Ariës, 2016). The low importance of father's education could potentially be partly explained by the relatively high correlation between mother's and father's education (Kendall's $\tau_B = 0.57$). Including highly correlated variables can lead to importance being split between them (Molnar, 2022). When removing mother's education from the model, the importance of father's education increases comparatively (see Supporting Information). Research about

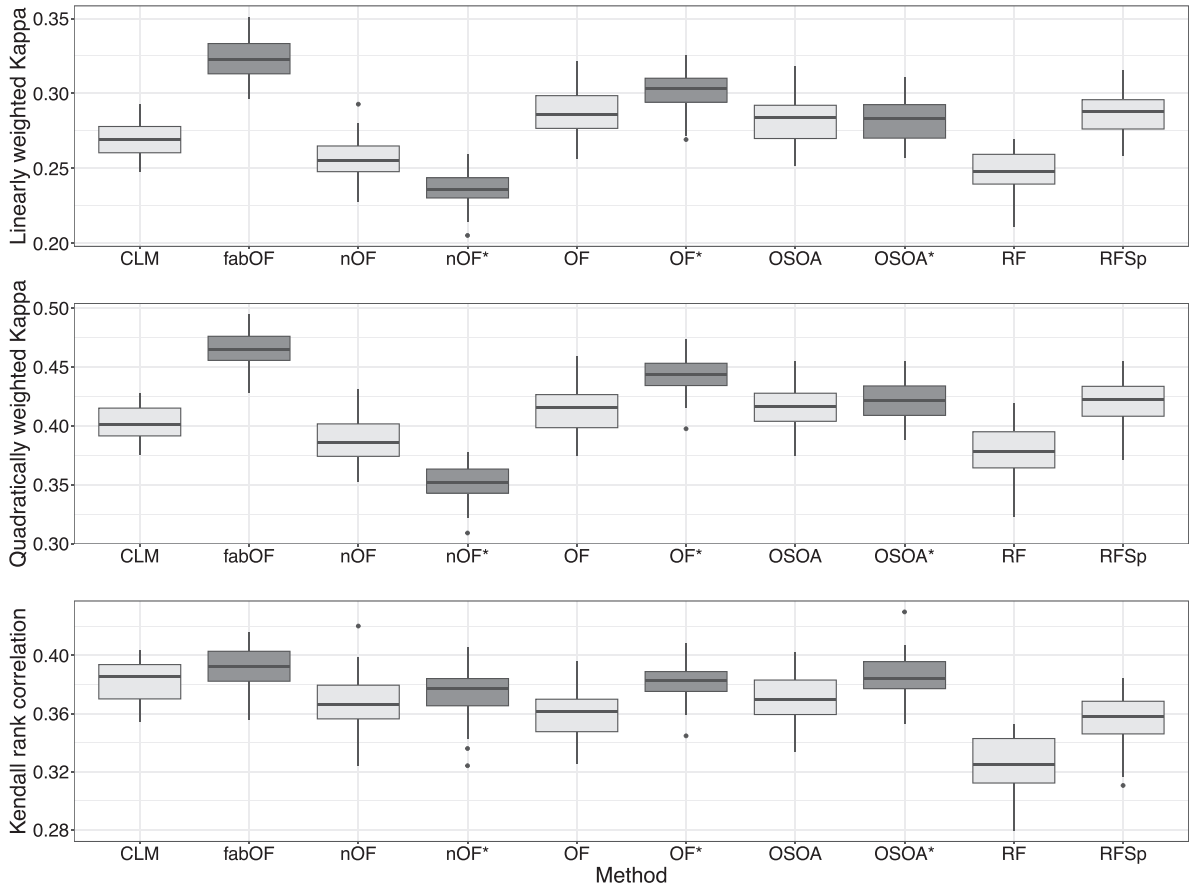

FIGURE 8 Predictive performance of existing methods and modifications for student performance data. Approaches using aggregate-first-transform-after prediction indicated through dark grey coloured boxplots with additional asterisk indicating modification of existing method. CLM, cumulative link model (proportional odds); fabOF, Frequency-Adjusted Borders Ordinal Forest; nOF, naive ordinal forest; OF, ordinal forest; OSOA, ordinal score optimization algorithm; RF, random forest; RFSp, split-based ordinal forest.

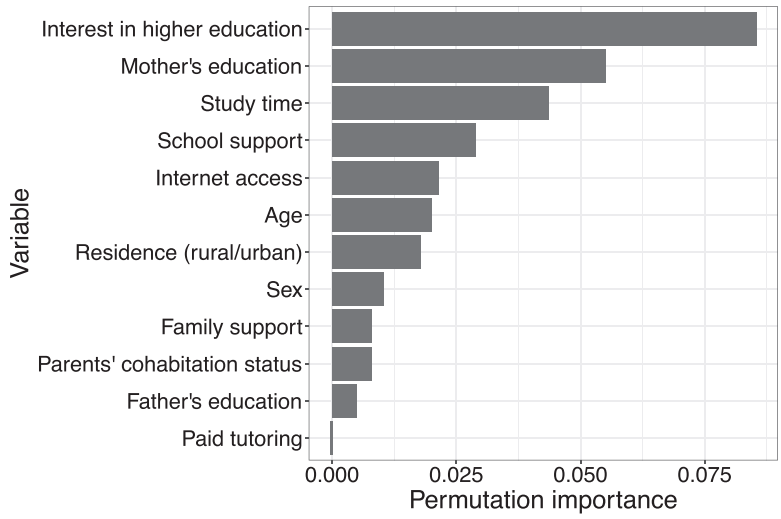

FIGURE 9 Permutation variable importance for student performance data.

the effect of study time on student achievement has been rather inconclusive, with some studies finding positive effects but others studies only finding an effect when the relation was mediated by motivational factors (see e.g., Keith, 1982; Masui et al., 2012; Rosário et al., 2012).

Overall, however, the presence of high correlations among some covariates indicates that the results should be interpreted with caution. VIMs are known to be affected by high correlations between covariates (Strobl et al., 2008). As a remedy, Strobl et al. (2008) proposed conditional permutation VIMs, which aim to preserve the original correlation structure in the data by restricting the way permutations can be performed. Therefore, comparing the foregoing results to variable importance results from a conditional VIM would be helpful to assess how reliable the interpretations are.

7 | DISCUSSION

In this work, I proposed fabOF, a novel method for ordinal prediction that adds to the methodological stream of ordinal prediction methods based on RF (Breiman, 2001), such as ordinal forest (OF; Hornung, 2019), split-based ordinal random forest (RFSp; Tutz, 2021), and the ordinal score optimization algorithm (OSOA; Buczak et al., 2024). Through simulation and an illustrative data example of student performance in a Portuguese language course (Cortez & Silva, 2008), I demonstrated that fabOF shows promising predictive performance and can improve upon existing methods in many data scenarios. Similar to OF and OSOA, fabOF assigns numeric scores to ordinal response categories and fits a regression RF using numeric scores as the target variable. For unseen observations, the predicted numeric scores from the RF fit are transformed into ordinal response categories using category borders that reflect a partition of the assumed latent variable's domain. Whereas in OF and OSOA numeric scores and category borders are directly linked (numeric scores are always chosen as the midpoint of the respective category interval as defined by the category borders), fabOF separates the choice of scores and category borders. Using default scores (i.e., $1, 2, \ldots, k$ for k categories), fabOF employs a heuristic for deriving adaptive category borders based on the cumulative relative frequencies of the response categories in the data. A simulation study showed that this approach was particularly effective in scenarios where response categories are not equally occupied. Furthermore, in comparison to OF and OSOA, the heuristic eliminates the need for an extensive optimization procedure in fabOF.

Buczak et al. (2024) found that in real data applications, differences in comparative predictive performance emerged, stressing the importance of evaluating ordinal prediction methods on various datasets. Apart from the student performance data covered in detail in this work, I have also benchmarked fabOF on seven other datasets used in Buczak et al. (2024). These additional results are included in the

Supporting Information. For a detailed description of the datasets, see Buczak et al. (2024). Compared to the RF-based ordinal prediction methods, fabOF achieved the highest predictive performance or places among the best performing methods for the majority of datasets. fabOF only fell behind notably for one dataset. Further investigation would be needed to determine whether one could infer more detailed reasons as to why fabOF did not perform well for this dataset or whether the data rather represented an outlier. Overall, however, these additional benchmarks underline the promising findings obtained from the simulation study and the student performance data presented here. Future work could also explore fabOF's performance in other data scenarios such as data generated from partial proportional odds models (see e.g., Brant, 1990; Peterson & Harrell, 1990), where not all covariate effects are global but some may instead differ across ordinal response categories.

Additionally, my simulation results indicate that the newly introduced prediction scheme of aggregating numeric score predictions at the tree level first and then transforming them to an ordinal response category via category borders (referred to as AFTA) as opposed to the reverse order (referred to as TFAA), which is currently employed in OF and OSOA, can yield benefits for OF and OSOA as well. Particularly for OF, AFTA prediction could improve predictive performance across all measures. For the simulation data, AFTA-modified OF often reached similar performance values compared to fabOF. As both share the underlying regression RF framework, the same prediction scheme (AFTA) and both aim to find a high-performing choice of category borders (through optimization or the heuristic), similar performance of these two approaches is likely commonly encountered. However, the simulation data and the real data example have also shown that the additional flexibility of fabOF (category borders and scores are decoupled from one another) and its frequency-based heuristic can lead to performance advantages depending on the data at hand. However, this does not imply that fabOF will always perform at least as well as AFTA-modified OF as such statements are generally not tenable in machine learning.

To enhance the interpretability of fabOF, I additionally introduced a custom permutation VIM based on Cohen's weighted Kappa (Cohen, 1968). Using simulated data, the permutation VIM was able to recover the influential covariates quite well. The importance of noise covariates was mostly around 0, while the importance of influential covariates was notably higher. When applied to the illustrative data example, findings that were largely consistent with the educational research literature emerged. However, it is not clear how fabOF's permutation VIM reacts to highly correlated covariates as the simulation data only included covariates which were simulated as uncorrelated. Unconditional VIMs (i.e., VIMs which do not restrict the permutation process in any way) can be affected by highly correlated covariates (Strobl et al., 2008). For the illustrative data example, high correlations between covariates were present. This should be taken into account when interpreting results. Future work could further evaluate fabOF's VIM in the presence of high correlations and consider the development of a conditional permutation VIM, as proposed by Strobl et al. (2008). To preserve the original correlation structure of the data, conditional permutation VIMs only permute covariates within certain regions of the covariate space. Future work could also study how the number of replications affects the stability of fabOF's VIM such that sensible compromises between stability and computation time can be derived.

Regarding methodological improvement of fabOF, future research could investigate further approaches to selecting appropriate category borders. For example, one could try to optimize category borders in a way similar to the optimization procedure in OSOA (while keeping the default numeric scores). In principle, one could also optimize numeric scores and category borders at the same time. However, this may pose a complex optimization problem. A possible remedy for this could be to employ a procedure in the spirit of the EM algorithm (Dempster et al., 1977), where the optimization procedure iterates between optimizing one while keeping the other fixed.

A further avenue for future research may be motivated by the student performance data used in this work. Typically, data from an educational context possess a hierarchical structure where individual observations are nested within groups, e.g., school classes. Another classic example of hierarchical data are longitudinal studies where multiple assessments of each person are performed, so individual assessments are nested within the respective individuals. Hierarchical data structures can introduce group-specific effects that need to be accounted for in the modelling process, e.g., through the inclusion of additional group-specific

random effects as in the (generalized) linear mixed model literature (see e.g., Hedeker & Gibbons, 1994; Tutz & Hennevogl, 1996, for extensions to ordinal regression). To extend fabOF to hierarchical data, one could adapt an approach similar to that of Hajjem et al. (2011) or Sela and Simonoff (2012), as was used in multiple extensions of RF to hierarchical data for different response types (for an overview, see Hu & Szymczak, 2023). Since ordinal data from psychological fields are often characterized by a hierarchical structure, such an extension could be a promising endeavour for future work.

AUTHOR CONTRIBUTIONS

Philip Buczak: conceptualization; methodology; software; writing – original draft; visualization; formal analysis; investigation; writing – review and editing; project administration.

ACKNOWLEDGEMENTS

The author would like to thank Dr. Marie Beisemann for providing helpful discussion and valuable feedback. This work has been supported by the Research Center Trustworthy Data Science and Security (https://rc-trust.ai), one of the Research Alliance centers within the UA Ruhr (https://uaruhr.de). Additionally, the author gratefully acknowledges the computing time provided on the Linux HPC cluster at TU Dortmund University Dortmund (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as project 271512359.

DATA AVAILABILITY STATEMENT

The `R` code for this work can be obtained from the corresponding OSF repository https://osf.io/fn8bg/. The accompanying `R` package `fabOF` can be obtained from https://github.com/phibuc/fabOF. The illustrative data example used were obtained from https://archive.ics.uci.edu/dataset/320/student+performance.

ORCID


Philip Buczak https://orcid.org/0000-0001-6980-8110

REFERENCES

Agresti, A., & Hitchcock, D. B. (2005). Bayesian inference for categorical data analysis. *Statistical Methods and Applications*, *14*(3), 297–330. https://doi.org/10.1007/s10260-005-0121-y

Archer, K. J. (2010). rpartOrdinal: An R package for deriving a classification tree for predicting an ordinal response. *Journal of Statistical Software*, *34*(7), 1–17. https://doi.org/10.18637/jss.v034.i07

Ben-David, A. (2008). Comparison of classification accuracy using Cohen's weighted kappa. *Expert Systems with Applications*, *34*(2), 825–832. https://doi.org/10.1016/j.eswa.2006.10.022

Brant, R. (1990). Assessing proportionality in the proportional odds model for ordinal logistic regression. *Biometrics*, *46*(4), 1171–1178. https://doi.org/10.2307/2532457

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 123–140. https://doi.org/10.1023/A:1010933404324

Buczak, P., Horn, D., & Pauly, M. (2024). *Old but gold or new and shiny? Comparing tree ensembles for ordinal prediction with a classic parametric approach*. Pre-print version 1.1 https://doi.org/10.31219/osf.io/v7bcf

Buczak, P., Huang, H., Forthmann, B., & Doebler, P. (2022). The machines take over: A comparison of various supervised learning approaches for automated scoring of divergent thinking tasks. *The Journal of Creative Behavior*, *57*(1), 17–36. https://doi.org/10.1002/jocb.559

Bürkner, P.-C., & Vuorre, M. (2019). Ordinal regression models in psychology: A tutorial. *Advances in Methods and Practices in Psychological Science*, *2*(1), 77–101. https://doi.org/10.1177/2515245918823199

Cabus, S. J., & Ariës, R. J. (2016). What do parents teach their children?—The effects of parental involvement on student performance in Dutch compulsory education. *Educational Review*, *69*(3), 285–302. https://doi.org/10.1080/00131911.2016.1208148

Cao, W., Mirjalili, V., & Raschka, S. (2020). Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, *140*, 325–331. https://doi.org/10.1016/j.patrec.2020.11.008

Cheng, J., Wang, Z., & Pollastri, G. (2008). A neural network approach to ordinal regression. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE. https://doi.org/10.1109/ijcnn.2008.4633963

Christensen, R. H. B. (2022). *Ordinal—Regression models for ordinal data. R package version 2022.11-16*. https://CRAN.R-project.org/package=ordinal

Chu, W., & Keerthi, S. S. (2007). Support vector ordinal regression. *Neural Computation*, *19*(3), 792–815. https://doi.org/10.1162/neco.2007.19.3.792

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, *70*(4), 213–220. https://doi.org/10.1037/h0026256

Cortez, P. (2014). *Student performance data. UCI Machine Learning Repository*. Retrieved from UCI Machine Learning Repository https://archive.ics.uci.edu/dataset/320/student+performance

Cortez, P., & Silva, A. M. G. (2008). *Using data mining to predict secondary school student performance*. https://api.semanticscholar.org/CorpusID:16621299

Costa-Mendes, R., Oliveira, T., Castelli, M., & Cruz-Jesus, F. (2020). A machine learning approximation of the 2015 Portuguese high school student grades: A hybrid approach. *Education and Information Technologies*, *26*(2), 1527–1547. https://doi.org/10.1007/s10639-020-10316-y

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, *39*(1), 1–38.

Epstein, E. S. (1969). A scoring system for probability forecasts of ranked categories. *Journal of Applied Meteorology*, *8*(6), 985–987.

Fife, D. A., & D'Onofrio, J. (2022). Common, uncommon, and novel applications of random forest in psychological research. *Behavior Research Methods*, *55*(5), 2447–2466. https://doi.org/10.3758/s13428-022-01901-9

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, *20*(177), 1–81.

Galimberti, G., Soffritti, G., & Maso, M. D. (2012). Classification trees for ordinal responses in R: The rpartScore package. *Journal of Statistical Software*, *47*(10), 1–25. https://doi.org/10.18637/jss.v047.i10

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022) - Track on Datasets and Benchmarks*. NeurIPS.

Hajjem, A., Bellavance, F., & Larocque, D. (2011). Mixed effects regression trees for clustered data. *Statistics & Probability Letters*, *81*(4), 451–459.

Hedeker, D., & Gibbons, R. D. (1994). A random-effects ordinal regression model for multilevel analysis. *Biometrics*, *50*(4), 933. https://doi.org/10.2307/2533433

Henninger, M., Debelak, R., Rothacher, Y., & Strobl, C. (2023). Interpretable machine learning for psychological research: Opportunities and pitfalls. *Psychological Methods*. https://doi.org/10.1037/met0000560

Herbrich, R. (1999). Support vector learning for ordinal regression. In *9th International Conference on Artificial Neural Networks: ICANN '99*, volume 1999 (pp. 97–102). IET. https://doi.org/10.1049/cp:19991091

Hidi, S., & Harackiewicz, J. M. (2000). Motivating the academically unmotivated: A critical issue for the 21st century. *Review of Educational Research*, *70*(2), 151–179. https://doi.org/10.3102/00346543070002151

Hilbert, S., Coors, S., Kraus, E., Bischl, B., Lindl, A., Frei, M., Wild, J., Krauss, S., Goretzko, D., & Stachl, C. (2021). Machine learning for the educational sciences. *Review of Education*, *9*(3), e3310. https://doi.org/10.1002/rev3.3310

Hornung, R. (2019). Ordinal forests. *Journal of Classification*, *37*(1), 4–17. https://doi.org/10.1007/s00357-018-9302-x

Hornung, R. (2022). *ordinalForest: Ordinal forests: Prediction and variable ranking with ordinal target variables. R package version 2.4-3*. https://CRAN.R-project.org/package=ordinalForest

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, *15*(3), 651–674. https://doi.org/10.1198/106186006X133933

Hu, J., & Szymczak, S. (2023). A review on longitudinal data analysis with random forest. *Briefings in Bioinformatics*, *24*(2), bbad002. https://doi.org/10.1093/bib/bbad002

Janitza, S., Tutz, G., & Boulesteix, A.-L. (2016). Random forest for ordinal responses: Prediction and variable selection. *Computational Statistics & Data Analysis*, *96*, 57–73. https://doi.org/10.1016/j.csda.2015.10.005

Johnson, S. G. (2008). *The NLopt nonlinear-optimization package*. https://github.com/stevengj/nlopt

Johnson, V. E., & Albert, J. H. (1999). *Ordinal data modeling*. Springer. https://doi.org/10.1007/b98832

Keith, T. Z. (1982). Time spent on homework and high school grades: A large-sample path analysis. *Journal of Educational Psychology*, *74*(2), 248–253. https://doi.org/10.1037/0022-0663.74.2.248

Kendall, M. G. (1945). The treatment of ties in ranking problems. *Biometrika*, *33*(3), 239–251. https://doi.org/10.1093/biomet/33.3.239

Kramer, S., Widmer, G., Pfahringer, B., & de Groeve, M. (2000). Prediction of ordinal classes using regression trees. In *Lecture notes in computer science* (pp. 426–434). Springer.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R News*, *2*(3), 18–22. https://CRAN.R-project.org/doc/Rnews/

Masui, C., Broeckmans, J., Doumen, S., Groenen, A., & Molenberghs, G. (2012). Do diligent students perform better? Complex relations between student and course characteristics, study time, and academic performance in higher education. *Studies in Higher Education*, *39*(4), 621–643. https://doi.org/10.1080/03075079.2012.721350

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society: Series B: Methodological*, *42*(2), 109–127. https://doi.org/10.1111/j.2517-6161.1980.tb01109.x

Molnar, C. (2022). *Interpretable machine learning* (2nd ed.) https://christophm.github.io/interpretable-ml-book

Molnar, C., Bischl, B., & Casalicchio, G. (2018). iml: An r package for interpretable machine learning. *Journal of Open Source Software*, *3*(26), 786. 10.21105/joss.00786

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, *7*, 308–313. https://doi.org/10.1093/comjnl/7.4.308

Peterson, B., & Harrell, F. E. (1990). Partial proportional odds models for ordinal response variables. *Applied Statistics*, *39*(2), 205–217. https://doi.org/10.2307/2347760

Piccarreta, R. (2007). Classification trees for ordinal variables. *Computational Statistics*, *23*(3), 407–427. https://doi.org/10.1007/s00180-007-0077-5

Probst, P., Wright, M. N., & Boulesteix, A. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, *9*(3), 1–15. https://doi.org/10.1002/widm.1301

R Core Team. (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/

Riccardi, A., Fernandez-Navarro, F., & Carloni, S. (2014). Cost-sensitive adaboost algorithm for ordinal regression based on extreme learning machine. *IEEE Transactions on Cybernetics*, *44*(10), 1898–1909. https://doi.org/10.1109/tcyb.2014.2299291

Rosário, P., Núñez, J. C., Valle, A., González-Pienda, J., & Lourenço, A. (2012). Grade level, study time, and grade retention and their effects on motivation, self-regulated learning strategies, and mathematics achievement: A structural equation model. *European Journal of Psychology of Education*, *28*(4), 1311–1331. https://doi.org/10.1007/s10212-012-0167-9

Sela, R. J., & Simonoff, J. S. (2012). RE-EM trees: A data mining approach for longitudinal and clustered data. *Machine Learning*, *86*, 169–207.

Shi, X., Cao, W., & Raschka, S. (2023). Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, *26*(3), 941–955. https://doi.org/10.1007/s10044-023-01181-9

Sönning, L., Krug, M., Vetter, F., Schmid, T., Leucht, A., & Messer, P. (2024). Latent-variable modelling of ordinal outcomes in language data analysis. *Journal of Quantitative Linguistics*, *31*, 77–106. https://doi.org/10.1080/09296174.2024.2329448

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, *9*(1), 307. https://doi.org/10.1186/1471-2105-9-307

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, *8*(1), 25. https://doi.org/10.1186/1471-2105-8-25

Tutz, G. (2021). Ordinal trees and random forests: Score-free recursive partitioning and improved ensembles. *Journal of Classification*, *39*(2), 241–263. https://doi.org/10.1007/s00357-021-09406-4

Tutz, G. (2022). Ordinal regression: A review and a taxonomy of models. *WIREs Computational Statistics*, *14*(2), e1545. https://doi.org/10.1002/wics.1545

Tutz, G., & Hechenbichler, K. (2005). Aggregating classifiers with ordinal response structure. *Journal of Statistical Computation and Simulation*, *75*(5), 391–408. https://doi.org/10.1080/00949650410001729481

Tutz, G., & Hennevogl, W. (1996). Random effects in ordinal regression models. *Computational Statistics & Data Analysis*, *22*(5), 537–557. https://doi.org/10.1016/0167-9473(96)00004-7

Ulitzsch, E., Ulitzsch, V., He, Q., & Lüdtke, O. (2022). A machine learning-based procedure for leveraging clickstream data to investigate early predictability of failure on interactive tasks. *Behavior Research Methods*, *55*(3), 1392–1412. https://doi.org/10.3758/s13428-022-01844-1

van der Scheer, E. A., & Visscher, A. J. (2017). Effects of a data-based decision-making intervention for teachers on students' mathematical achievement. *Journal of Teacher Education*, *69*(3), 307–320. https://doi.org/10.1177/0022487117704170

Wößmann, L. (2003). Schooling resources, educational institutions and student performance: the international evidence. *Oxford Bulletin of Economics and Statistics*, *65*(2), 117–170. https://doi.org/10.1111/1468-0084.00045

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. 10.18637/jss.v077.i01

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer*, *3*(1), 32–35. https://doi.org/10.1002/1097-0142(1950)3:1¡32::aid-cncr2820030106¿3.0.co;2-3

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Buczak, P. (2025). Frequency-adjusted borders ordinal forest: A novel tree ensemble method for ordinal prediction. *British Journal of Mathematical and Statistical Psychology*, *78*, 594–616. https://doi.org/10.1111/bmsp.12375

APPENDIX A

COMPUTATIONAL DETAILS

All computations were performed using `R` (R Core Team, 2022) version 4.2.1 with the packages `ordinalForest` (Hornung, 2022) for fitting (naive) OFs, `ranger` (Wright & Ziegler, 2017) for fitting multilabel classification RFs and `ordinal` (Christensen, 2022) for fitting proportional odds models with the `clm()` function. For OSOA I used the implementation provided by the authors in https://osf.io/v64d9/, while for RFSP I used the implementation provided by the author in https://github.com/GerhardTutz/ScoreFreeTrees. For all RF-based methods, I used 500 trees, which is the default value for the RF implementations in the `ranger` (Wright & Ziegler, 2017) and `randomForest` (Liaw & Wiener, 2002) packages. For other hyperparameters I used the default values as provided in the respective implementations. In particular, I refrained from performing a hyperparameter tuning as I mainly compared RF-based methods among themselves, and RFs have been shown to be relatively robust regarding their choice of hyperparameters (Probst et al., 2019). This design choice is in line with previous research from the field (e.g., Buczak et al., 2024; Hornung, 2019; Janitza et al., 2016; Tutz, 2021).

APPENDIX B

RESPONSE CATEGORY DISTRIBUTION PATTERNS AND THRESHOLD CHOICES

The two distribution patterns for DGPs 1, 3 and 4 from the simulation study were enforced through the choice of threshold values using the same approach as in Buczak et al. (2024). To this end, I determined threshold values for each DGP that for a simulated dataset of 100,000 observations approximately resulted in the targeted relative frequencies per category as displayed in Table B1.

For DGP 2 in which data were simulated from a linear regression model and binned into ordinal categories, the distribution patterns were obtained using specific binning values. To determine suitable binning values, I followed the approach in Buczak et al. (2024) of approximating the empirical distribution function of the numeric outcome through 100,000 simulated observations. The quantiles matching the intended relative frequencies per category in Table B1 were selected as binning values. Table B2 displays the values chosen as thresholds for DGPs 1, 3, and 4 as well as the binning values for DGP 2.

APPENDIX C

COHEN'S WEIGHTED KAPPA

Cohen's weighted Kappa (Cohen, 1968) κ_w is given by

$$\kappa_w = \frac{\sum_{r=1}^{k}\sum_{s=1}^{k} w_{rs} p^o_{rs} - \sum_{r=1}^{k}\sum_{s=1}^{k} w_{rs} p^e_{rs}}{1 - \sum_{r=1}^{k}\sum_{s=1}^{k} w_{rs} p^e_{rs}},$$

where p^o_{rs} denotes the observed proportion of instances with true category r and predicted category s, and p^e_{rs} denotes the proportion that is expected by chance (Cohen, 1968). Furthermore, w_{rs} denotes the weight

TABLE B1 Targeted relative frequencies $\pi_r, r = 1, \ldots, k$ based on number of categories k and response category distribution pattern.

		Targeted relative frequencies						
Categories	**Pattern**	π_1	π_2	π_3	π_4	π_5	π_6	π_7
$k = 5$	Equal	0.20	0.20	0.20	0.20	0.20	–	–
	Wide middle	0.11	0.22	0.33	0.22	0.11	–	–
$k = 7$	Equal	0.14	0.14	0.14	0.14	0.14	0.14	0.14
	Wide middle	0.06	0.13	0.19	0.25	0.19	0.13	0.06

TABLE B2 Threshold/binning values for combinations of DGP, number of categories k, and response category distribution pattern.

			Threshold/binning values						
DGP	**Categories**	**Pattern**	γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7
DGP 1	$k = 5$	Equal	−3.25	−1.75	−0.5	1	∞	–	–
		Wide middle	−4	−2.25	−0.25	1.75	∞	–	–
	$k = 7$	Equal	−3.75	−2.5	−1.6	−0.75	0.2	1.4	∞
		Wide middle	−5	−3.25	−1.85	−0.35	1.1	2.65	∞
DGP 2	$k = 5$	Equal	−0.5	0.7	1.7	2.8	∞	–	–
		Wide middle	−1.25	0.3	2	3.5	∞	–	–
	$k = 7$	Equal	−0.95	0	0.8	1.5	2.2	3.1	∞
		Wide middle	−1.9	−0.5	0.6	1.8	3	4.4	∞
DGP 3	$k = 5$	Equal	−2.5	−1.25	−0.25	1	∞	–	–
		Wide middle	−3.25	−1.5	0.25	2	∞	–	–
	$k = 7$	Equal	−3	−2	−1.15	−0.4	0.4	1.5	∞
		Wide middle	−4	−2.5	−1.35	0	1.25	2.65	∞
DGP 4	$k = 5$	Equal	−3.7	−2.3	−1.1	0.3	∞	–	–
		Wide middle	−4.65	−2.7	−0.6	1.3	∞	–	–
	$k = 7$	Equal	−4.25	−3	−2.1	−1.25	−0.3	0.9	∞
		Wide middle	−5.4	−3.7	−2.35	−1	0.4	2	∞

assigned to instances with true category r and predicted category s. Weights must be specified in advance and act as a way of penalization regarding the distance between the true and predicted response category. In this work, I have used linear and quadratic weights, given by

$$w_{rs}^{\text{lin}} = 1 - \frac{|r-s|}{k-1},$$
$$w_{rs}^{\text{quad}} = 1 - \frac{|r-s|^2}{(k-1)^2}$$

as these are common choices for ordinal prediction (Ben-David, 2008; Buczak et al., 2024; Hornung, 2019). Compared to quadratic weights, linear weights place higher weight on instances for which the predicted categories are close or equal to the true categories. On the other hand, quadratic weights assign more weight to instances with predicted categories further away from the true categories than linear weights (Hornung, 2019).

Article 3

Buczak, P. (2024). Mixed-effects frequency-adjusted borders ordinal forest: A tree ensemble method for ordinal prediction with hierarchical data. *OSF pre-print, version 1.1*, 1–36. https://doi.org/10.31219/osf.io/ny6we (*Currently under review with Multivariate Behavioral Research*)

Mixed-Effects Frequency-Adjusted Borders Ordinal Forest: A Tree Ensemble Method for Ordinal Prediction with Hierarchical Data

Philip Buczak

Department of Statistics, TU Dortmund University, 44227 Dortmund, Germany

UA Ruhr, Research Center Trustworthy Data Science and Security, 44227 Dortmund, Germany

Correspondence should be addressed to: Philip Buczak, Department of Statistics, TU Dortmund University, Vogelpothsweg 87, 44227 Dortmund, Germany;

buczak@statistik.tu-dortmund.de

Pre-print version 1.1 (Oct 8th, 2024).

Author Note

The author would like to thank Dr. Marie Beisemann for providing helpful discussion and valuable feedback. This work has been partly supported by the Research Center Trustworthy Data Science and Security (https://rc-trust.ai), one of the Research Alliance centers within the UA Ruhr (https://uaruhr.de). Additionally, the author gratefully acknowledges the computing time provided on the Linux HPC cluster at TU Dortmund University Dortmund (LiDO3), partially funded in the course of the Large-Scale Equipment Initiative by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as project 271512359. The `R` code for this work can be obtained from the corresponding OSF repository https://osf.io/npem6/. A development version of the accompanying `R` package can be obtained from https://github.com/phibuc/fabOF.

Abstract

Predicting ordinal responses such as school grades or rating scale data is a common task in the social and life sciences. Currently, two major streams of methodology exist for ordinal prediction: parametric models such as the proportional odds model and machine learning (ML) methods such as random forest (RF) adapted to ordinal prediction. While methods from the latter stream have displayed high predictive performance, particularly for data characterized by non-linear effects, most of these methods do not support hierarchical data. As such data structures frequently occur in the social and life sciences, e.g., students nested in classes or individual measurements nested within the same person, accounting for hierarchical data is of importance for prediction in these fields. A recently proposed ML method for ordinal prediction displaying promising results for non-hierarchical data is Frequency-Adjusted Borders Ordinal Forest (fabOF). Building on an iterative expectation-maximization-type estimation procedure, I extend fabOF to hierarchical data settings in this work by proposing Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF). Through simulation and a real data example on math achievement, I demonstrate that mixfabOF can improve upon fabOF and other RF-based ordinal prediction methods for (non-)hierarchical data in the presence of random effects.

Keywords: Ordinal Prediction; Hierarchical Data; Random Forest; Machine Learning

Mixed-Effects Frequency-Adjusted Borders Ordinal Forest: A Tree Ensemble Method for Ordinal Prediction with Hierarchical Data

Introduction

Ordinal responses are commonly encountered in the social and life sciences. Students receive ordinal grades for their performance, participants in assessment studies voice their preferences or agreement towards given statements on ordinal rating scales, judges evaluate the performance, e.g., in creativity tasks, using ordinal scores. Historically, there are two major streams of methodology developed for modeling and predicting ordinal responses. First, the more traditional stream of parametric models, e.g., cumulative models which assume that the observed ordinal responses are generated by an underlying latent (numeric) variable that can only be observed through certain thresholds (McCullagh, 1980). A particularly popular special case of the cumulative model is the proportional odds model (McCullagh, 1980) which intuitively can be thought of as a series of logistic models holding simultaneously (Tutz, 2021). For a general overview of parametric models for ordinal responses, see Tutz (2022). The second methodological stream has developed more recently and involves using machine learning (ML) methods such as random forest (RF; Breiman, 2001) for ordinal prediction (Buczak, 2024; Buczak et al., 2024; Hornung, 2019; Janitza et al., 2016; Tutz, 2021). ML methods offer the prospect of high predictive performance for large datasets as are becoming increasingly available in the social and life sciences, e.g., through click-stream data (e.g., Ulitzsch et al., 2022), ecological momentary assessment data (e.g., Kathan et al., 2022) or other types of digital phenotyping and mobile sensing data (for an overview, see Montag & Baumeister, 2023). Another common source of large datasets in these fields are large-scale assessment studies such as PISA, PIRLS or TIMSS. A ML method that was recently proposed for ordinal prediction is Frequency-Adjusted Borders Ordinal Forest (fabOF; Buczak, 2024). Similar to Ordinal Forest (OF; Hornung, 2019) (and cumulative models), fabOF assumes the ordinal response to originate from an underlying latent numeric variable. To approximate the latent

variable, fabOF represents each ordinal response category as a numeric interval and assigns a representative numeric score to each category, respectively. Based on the numeric scores and category interval borders, fabOF trains a regression RF and transforms the resulting numeric predictions back into ordinal categories via the category borders. Whereas OF relies on a computationally extensive optimization procedure to arrive at suitable values for the scores and category borders, fabOF employs a heuristic based on the frequencies of the ordinal response categories. Apart from the notable advantage in computational runtime, Buczak (2024) has also demonstrated promising results regarding the predictive performance of fabOF. However, as indicated by the author, the lacking support for hierarchical data is a current limitation of fabOF. Hierarchical data structures occur when individual observations can be grouped into clusters, e.g., students nested within school classes or individual assessments nested within the same person in longitudinal study designs. Such structures can induce cluster-specific effects into the data which, e.g., in the case of (generalized) linear mixed models are accounted for by including cluster-specific random effects (Molenberghs & Verbeke, 2000). In the context of ordinal regression, extensions to hierarchical data have been proposed, e.g., in Hedeker and Gibbons (1994) and Tutz and Hennevogl (1996). While several extensions of ML algorithms to hierarchical data have been proposed for numeric outcomes (Capitaine et al., 2020; Hajjem et al., 2011, 2012; Pellagatti et al., 2021; Salditt et al., 2023; Sela & Simonoff, 2012), corresponding extensions for ordinal responses have long been lacking. Only recently, Bergonzoli et al. (2024) proposed Ordinal Mixed-Effect Random Forest (OMERF) building on the framework of the Generalized Mixed-Effect Random Forest (GMERF; Pellagatti et al., 2021). Developed independently in parallel and following a different approach, this work extends fabOF to hierarchical data by proposing Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF). The newly proposed mixfabOF method follows the logic of fabOF and combines it with the iterative estimation procedure of Mixed-Effects Random Forest (MERF; Hajjem et al., 2012). Through simulation and an illustrative data

example on math ability of fourth grade students, I will demonstrate that mixfabOF achieves higher predictive performance than fabOF and other non-hierarchical RF-approaches for ordinal prediction in the presence of moderate and large random effects. Furthermore, mixfabOF can improve upon OMERF in both, predictive performance and computational runtime. These promising findings underline the usefulness of the proposed mixfabOF method for ordinal prediction in hierarchical data scenarios as is common, e.g., in the context of educational achievement. To this end, improving predictive capabilities can help in better informing the development of educational policies and student support systems (Costa-Mendes et al., 2020; van der Scheer & Visscher, 2017).

The remainder of this work is structured as follows. In the next section, I will provide an overview of previous research including RF-based methods for ordinal prediction as well as extensions of classic ML methods to hierarchical data for various outcome types. Following this, I will introduce the newly proposed mixfabOF method and compare it with other ordinal prediction methods in a simulation study and an illustrative data example. This work will close with a discussion and potential avenues of further research.

Previous Research

Ordinal Prediction with RF

While enjoying popularity for classification and regression tasks, RF is lacking inherent support for ordinal response data. As a remedy, several workarounds and extensions to RF have been proposed. A commonly used approach is assigning numeric scores to the ordinal response categories. In the context of decision trees, Kramer et al. (2000) predicted ordinal responses using regression trees with numeric scores, while Piccarreta (2007), Archer (2010) and Galimberti et al. (2012) built on numeric scores to extend split criteria of classification trees to ordinal prediction tasks. Similarly, the Conditional Inference Tree framework (Hothorn et al., 2006) also relies on numeric scores for accommodating ordinal responses. The use of Conditional Inference Forests for ordinal prediction has been studied in Janitza et al. (2016). While these approaches all either

implicitly assume a concrete set of scores (e.g., $1, 2, \ldots, k$ for k categories) or otherwise expect a user-specified input of scores, Ordinal Forest (OF; Hornung, 2019) first employs an optimization procedure to determine an optimal set of numeric scores to be used within a regression RF context. An entirely score-free approach was proposed by Tutz (2021) who introduced Split-Based Random Forest (RFSp). Instead of relying on regression RF, RFSp transforms the ordinal prediction task into a series of binary prediction tasks for which classification RFs are trained. The individual RF models are then used to obtain combined predictions for the original ordinal prediction task in the spirit of cumulative models (Tutz, 2021). Tutz (2021) as well as Buczak et al. (2024) compared the different tree ensemble methods with parametric models. Both studies found that the tree ensemble methods performed mostly similarly, while the most pronounced differences occurred in relation to the parametric model(s) depending on the data generating processes (e.g., non-linearity of effects). As a compromise between parametric and ML models, Tutz (2021) proposed therefore combining both types in a joint prediction ensemble consisting of multiple individual prediction models. Regarding the optimization of the numeric scores assigned to the ordinal response categories, Buczak et al. (2024) found that the optimization procedures in OF and the authors' own *Ordinal Score Optimization Algorithm* (OSOA) yielded only situational benefits. Based on these findings, Buczak (2024) proposed Frequency-Adjusted Borders Ordinal Forest (fabOF). Following OF, fabOF assumes the ordinal response to be a coarser version of a latent numeric variable (similar to the cumulative model) and expresses the ordinal categories as numeric intervals that partition the assumed latent variable's domain. Each category interval is represented by a numeric score which is mapped to the ordinal response category and used to fit a regression RF. For new observations, the numeric predictions from the internal regression RF model are transformed into ordinal response categories through the category borders that define the category intervals. Where OF and fabOF differ is in their choice of category borders and scores. While OF uses an extensive optimization step to determine optimal settings, fabOF

avoids the optimization step and relies on a category frequency-based heuristic to derive its category borders using arbitrary category scores (Buczak, 2024). After assigning numeric scores (e.g., $1, 2, \ldots, k$) to the ordinal response categories, a regression RF is trained using the numeric scores as the target variable. From the RF model, numeric out-of-bag (OOB) predictions for the training data are obtained, i.e., for a given observation, only trees for which the observation was not used for training are used for prediction, respectively. To determine the the category borders, fabOF uses the OOB predictions for computing quantiles for probabilities matching the cumulative relative frequencies of the ordinal response categories up to (but not including) category k. Buczak (2024) reported promising findings regarding the predictive performance of fabOF and notably reduced computational runtime compared to OF. However, the author also identified a lacking support for hierarchical data structures as a current limitation of fabOF. This limitation is currently also shared with OF, RFSp and OSOA, as these all rely on RF internally. While RF as well as other classic ML methods were initially affected by this limitation, several extensions to hierarchical data have been proposed as a remedy which will be presented in the next section.

Extending Tree-based Methods to Hierarchical Data

Some of the earliest extensions of tree-based ML methods to hierarchical data were proposed by Segal (1992) and De'ath (2002). Both authors accommodated hierarchical data structures by extending univariate regression trees to multivariate regression trees where all (univariate) observations of a cluster were treated as a combined multivariate cluster observation vector. As such, only splits at the cluster-level could be performed which, e.g., in a longitudinal setting would imply that all covariates need to be fixed in time (Salditt et al., 2023). This limitation (also present in subsequent approaches, such as Loh & Zheng, 2013) was addressed by the Mixed Effects Regression Tree (MERT; Hajjem et al., 2011) and Random Effects Expectation Maximization (RE-EM) tree (Sela & Simonoff, 2012) which allow for splitting at the observation- and cluster-level alike. Both

approaches operate within the linear mixed model (LMM) framework where the n observations adhere to a hierarchical structure and are grouped into m clusters of sizes $n_1, \ldots, n_m$ (with $n_1 + \cdots + n_m = n$). It is assumed that the individual outcomes result from a linear combination of (global) fixed effects and cluster-specific random effects. The classic LMM (cf. Molenberghs & Verbeke, 2000) models the outcome vector $\boldsymbol{y}_j$ of cluster j as

$$\boldsymbol{y}_j = \boldsymbol{X}_j\boldsymbol{\beta} + \boldsymbol{Z}_j\boldsymbol{b}_j + \boldsymbol{\varepsilon}_j,\ j = 1, \ldots, m, \tag{1}$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ is the fixed effects vector and for cluster j, respectively, $\boldsymbol{X}_j \in \mathbb{R}^{n_j \times p}$ is the matrix of fixed effect covariate values, $\boldsymbol{Z}_j \in \mathbb{R}^{n_j \times q}$ is the matrix of random effect covariate values, $\boldsymbol{b}_j \in \mathbb{R}^q$ is the vector of random effects, and $\boldsymbol{\varepsilon}_j \in \mathbb{R}^{n_j}$ is the vector of error terms, $j = 1, \ldots, m$. It is assumed that $\boldsymbol{b}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{D})$ with $\boldsymbol{D} \in \mathbb{R}^{q \times q}$ as well as $\boldsymbol{\varepsilon}_j \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_j)$. For $\boldsymbol{R}_j$, it is often assumed that $\boldsymbol{R}_j = \sigma^2 \boldsymbol{I}_{n_j \times n_j}$ (Fahrmeir et al., 2021). It is further assumed that the random effects $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ and error terms $\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_m$ are independent (Molenberghs & Verbeke, 2000).

Hajjem et al. (2011) and Sela and Simonoff (2012) both approach their extension of regression trees to hierarchical data by modifying the model in Equation 1 and replacing the linear fixed effects structure through a (non-linear) function $f(\boldsymbol{X}_j)$. This results in the modified model

$$\boldsymbol{y}_j = f(\boldsymbol{X}_j) + \boldsymbol{Z}_j\boldsymbol{b}_j + \boldsymbol{\varepsilon}_j,\ j = 1, \ldots, m. \tag{2}$$

For estimation, both approaches use the Expectation Maximization (EM) Algorithm (Dempster et al., 1977) as can be used for the estimation of mixed models (see e.g., Laird & Ware, 1982). To this end, the estimation procedure iterates between estimating the fixed (i.e., $f(\boldsymbol{X}_j)$) and random effect components. However, MERT and RE-EM trees differ in their specification and estimation of the fixed effects component. In MERT, $f(\boldsymbol{X}_j)$ is estimated by fitting a regression tree to the modified outcome

$$\tilde{\boldsymbol{y}}_j = \boldsymbol{y}_j - \boldsymbol{Z}_j\boldsymbol{b}_j,\ j = 1, \ldots, m, \tag{3}$$

i.e., the outcome from which the random effect structure has been removed (Hajjem et al.,

2011). RE-EM trees, on the other hand, fit a regression tree to the modified outcome only to use the resulting partition to fit a LMM in which fixed effects are modeled locally (as determined by the partition specified by the regression tree model) and random effects globally (Sela & Simonoff, 2012). Both MERT and RE-EM trees have been extended for use with RF through Mixed-Effects Random Forest (MERF; Hajjem et al., 2012) and REEMforest (Capitaine et al., 2020). Capitaine et al. (2020) further proposed the inclusion of a stochastic model component, resulting in further extensions, namely SMERT, SMERF, SREEMtree and SREEMforest. For adapting MERT/MERF to response types from the exponential family, Generalized Mixed Effects Regression Trees (GMERT; Hajjem et al., 2017), Generalized Mixed-Effects Trees (GMET; Fontana et al., 2021) and Generalized Mixed-Effects Random Forest (GMERF; Pellagatti et al., 2021) have been proposed. Using a Bayesian approach for binary responses, Speiser et al. (2018) introduced Binary Mixed Model (BiMM) trees which were extended to BiMM forests (Speiser et al., 2019). Extensions of other ML methods to hierarchical data in the spirit of MERT and RE-EM trees have also been proposed for logistic regression (Lin & Luo, 2019) and gradient tree boosting (Salditt et al., 2023). For an overview of most of the above methods, see Hu and Szymczak (2023).

In the context of ordinal prediction for hierarchical data, Bergonzoli et al. (2024) have recently proposed Ordinal Mixed-Effects Random Forest (OMERF) which builds on the GMERF framework. OMERF initializes by fitting an OF model to the data, and then iterates between fitting a RF and a CLMM. The Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF) method proposed in this work was developed independently of OMERF, and instead combines the approaches of MERF and fabOF. I will present mixfabOF in detail in the next section.

Mixed-Effects Frequency-Adjusted Borders Ordinal Forest

The general idea of mixfabOF is assigning numeric scores to the ordinal response categories, performing the iterative estimation of fixed and random effects components

known from MERF and deriving suitable category borders using the heuristic from fabOF. The procedure is described in more detail with pseudocode in Algorithm 1. After assigning numeric scores (e.g., default scores $1, \ldots, k$ for k categories) to the ordinal response categories, the score-based numeric outcome values $\boldsymbol{y}_j^{\text{num}}, j = 1, \ldots, m$, are used to iterate between estimating the fixed and random effects components. To this end, mixfabOF follows the procedure proposed in MERF (cf. lines 7-14 of Algorithm 1 with pseudocode in Hajjem et al., 2011). For the current fixed effects component, a regression RF is trained on the current modified responses from which the random effects have been removed (cf. Equation 3). Having updated the fixed effect component, the estimates for the random effects, random effect variance and the residual variance are updated. The alternating estimation procedure continues until convergence or a maximum number of iterations is achieved. For assessing convergence, mixfabOF uses the generalized log-likelihood (GLL) criterion employed in MERF (cf. Hajjem et al., 2012), i.e.,

$$GLL(f, \boldsymbol{b_i}|\boldsymbol{y}^{\text{num}}) = \sum_{j=1}^{m} \Bigg\{ \left(\boldsymbol{y}_j^{\text{num}} - f\left(\boldsymbol{X}_j\right) - \boldsymbol{Z}_j \boldsymbol{b}_j\right)^T \boldsymbol{R}_j^{-1} \left(\boldsymbol{y}_j^{\text{num}} - f\left(\boldsymbol{X}_j\right) - \boldsymbol{Z}_j \boldsymbol{b}_j\right) + \boldsymbol{b}_j^T \boldsymbol{D}^{-1} \boldsymbol{b}_j + \log|\boldsymbol{D}| + \log|\boldsymbol{R}_i| \Bigg\}. \tag{4}$$

For a given iteration, the criterion is computed using the current estimates. When the relative change in the GLL compared to the previous iteration is smaller than a threshold value δ, the iterative procedure is stopped. Following Salditt et al. (2023), mixfabOF uses $\delta = 0.001$. After the iterative estimation procedure, the frequency-adjusted borders heuristic of fabOF is applied. To this end, numeric OOB-based predictions $\hat{\boldsymbol{y}}_j^{\text{num}}$ for the training data are computed using the final RF model's numeric OOB predictions $\hat{f}(\boldsymbol{X}_j)_{\text{OOB}}$ and the final random effect estimates, i.e., $\hat{\boldsymbol{y}}_j^{\text{num}} = \hat{f}(\boldsymbol{X}_j)_{\text{OOB}} + \boldsymbol{Z}_j \hat{\boldsymbol{b}}_j, j = 1, \ldots, m$. For readability, the subscript indicating the final iteration has been omitted. Based on the cumulative relative frequencies $\pi_1, \ldots, \pi_{k-1}$ of the ordinal response categories up to (but not including) category k, the respective quantiles $q_{\pi_1}(\hat{\boldsymbol{y}}^{\text{num}}), \ldots, q_{\pi_{k-1}}(\hat{\boldsymbol{y}}^{\text{num}})$ of the OOB-based predictions are determined. These quantiles are in turn assigned to the inner

set of category borders whereas the lower and upper bound are set to $-\infty$ and ∞, respectively. Note that fabOF's use of the lowest and highest numeric scores as bounds are not possible here anymore since due to the inclusion of the random effects, values smaller or larger than s_1 and s_k can occur. Lastly, the final RF fit, the final random effect estimates and the category borders are returned. New observations from known clusters are predicted by first obtaining numeric predictions based on the fixed effects component RF model and the random effect estimates. For observations from unknown clusters, only the fixed effects component is used while the random effects component is set to zero (similar, e.g., to the `lme4` package; Bates et al., 2015). In both cases, the numeric predictions are transformed into ordinal response category predictions using the category borders.

An implementation of mixfabOF is available in the `fabOF` package which can be obtained from GitHub (https://github.com/phibuc/fabOF). The implementation further includes the possibility of computing variable importance values for the covariates associated with the fixed effects. The custom permutation variable importance measure (VIM) is based on the VIM introduced in Buczak (2024) and was adapted for use with mixfabOF such that the hierarchical data context is accounted for. To this end, it additionally allows for permuting in a clusterwise fashion, i.e., permutations are only performed within the same cluster, respectively. Variable importance values can aid with interpreting RF-based models as RF inherently suffers from a lack of interpretability (Molnar, 2022). Permutation VIMs (Breiman, 2001) assess the impact of individual covariates on the model's predictive performance by randomly shuffling the values of a given covariate, thus, voiding the information it contains. The importance of the covariate is then determined by comparing the predictive performance achieved when using the original data and the permuted data. The underlying logic is that a comparatively large decrease in predictive performance indicates that the given covariate is important for the model's predictions (Molnar, 2022).

Algorithm 1 Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF)

procedure MIXFABOF

Unless specified otherwise, assign scores $(s_1, s_2, \ldots, s_k) \leftarrow (1, 2, \ldots, k)$.

Create $\boldsymbol{y}_j^{\text{num}}, j = 1, \ldots, m$, by assigning scores to ordinal response categories.

Set $r = 0, \hat{\boldsymbol{D}}_{(0)} = \boldsymbol{I}_{n_j \times n_j}, \hat{\boldsymbol{b}}_{j,(0)} = \boldsymbol{0}_{n_j}, j = 1, \ldots, m$.

while $r \leq$ max.iter and `not converged` **do**

$r = r + 1$

Update $\tilde{\boldsymbol{y}}_{j,(r)}^{num}, \hat{f}_{(r)}(\boldsymbol{X}_j)$ *and* $\hat{\boldsymbol{b}}_{j,(r)}$:

$\tilde{\boldsymbol{y}}_{j,(r)}^{\text{num}} = \boldsymbol{y}_j^{\text{num}} - \boldsymbol{Z}_j \hat{\boldsymbol{b}}_{j,(r-1)}, j = 1, \ldots, m$.

Obtain $\hat{f}_{(r)}(\boldsymbol{X}_j)$ by fitting a regression RF to response $\tilde{\boldsymbol{y}}_{(r)}^{\text{num}}$ and covariates $\boldsymbol{X}$.

$\hat{\boldsymbol{b}}_{j,(r)} = \hat{\boldsymbol{D}}_{(r-1)} \boldsymbol{Z}_j^T \hat{\boldsymbol{V}}_{j,(r-1)}^{-1} \left(\boldsymbol{y}_j^{\text{num}} - \hat{f}_{(r)}(\boldsymbol{X}_j) \right), j = 1, \ldots, m,$

where $\hat{\boldsymbol{V}}_{j,(r-1)}^{-1} = \boldsymbol{Z}_j \hat{\boldsymbol{D}}_{(r-1)} \boldsymbol{Z}_j^T + \hat{\sigma}_{(r-1)}^2 \boldsymbol{I}_{n_j \times n_j}$.

Update $\hat{\sigma}_{(r)}^2$ *and* $\hat{\boldsymbol{D}}_{(r)}$:

$$\hat{\sigma}_{(r)}^2 = \frac{1}{n} \sum_{j=1}^{m} \hat{\boldsymbol{\varepsilon}}_{j,(r)}^T \hat{\boldsymbol{\varepsilon}}_{j,(r)} + \hat{\sigma}_{(r-1)}^2 \left(n_j - \hat{\sigma}_{(r-1)}^2 \text{trace} \left(\hat{\boldsymbol{V}}_{j,(r-1)} \right) \right),$$

$$\hat{\boldsymbol{D}}_{(r)} = \frac{1}{m} \sum_{j=1}^{m} \left\{ \hat{\boldsymbol{b}}_{j,(r)} \hat{\boldsymbol{b}}_{j,(r)}^T + \left(\hat{\boldsymbol{D}}_{(r-1)} - \hat{\boldsymbol{D}}_{(r-1)} \boldsymbol{Z}_j^T \hat{\boldsymbol{V}}_{j,(r-1)}^{-1} \boldsymbol{Z}_j \hat{\boldsymbol{D}}_{(r-1)} \right) \right\},$$

where $\hat{\boldsymbol{\varepsilon}}_{j,(r)} = \boldsymbol{y}_j^{\text{num}} - \hat{f}_{(r)}(\boldsymbol{X}_j) - \boldsymbol{Z}_j \hat{\boldsymbol{b}}_{j,(r)}$.

Check convergence using GLL criterion.

end while

Compute numeric OOB predictions $\hat{f}(\boldsymbol{X}_j)_{\text{OOB}}$ with final RF model, $j = 1, \ldots, m$.

Compute OOB-based predictions $\hat{\boldsymbol{y}}_j^{\text{num}} = \hat{f}(\boldsymbol{X}_j)_{\text{OOB}} + \boldsymbol{Z}_j \hat{\boldsymbol{b}}_j, j = 1, \ldots, m$.

For categories up to category k, compute cumulative relative frequencies $\pi_1, \ldots, \pi_{k-1}$.

Obtain prediction quantiles $q_{\pi_1}(\hat{\boldsymbol{y}}^{\text{num}}), \ldots, q_{\pi_{k-1}}(\hat{\boldsymbol{y}}^{\text{num}})$ for probabilities $\pi_1, \ldots, \pi_{k-1}$.

Assign category borders $(b_1, b_2, \ldots, b_k, b_{k+1}) \leftarrow \left(-\infty, q_{\pi_1}(\hat{\boldsymbol{y}}^{\text{num}}), \ldots, q_{\pi_{k-1}}(\hat{\boldsymbol{y}}^{\text{num}}), \infty \right)$.

return RF model, random effect estimates and category borders

end procedure

Simulation Study

Simulation Setup

To evaluate mixfabOF, I performed a simulation study whose setup was largely inspired by the simulation studies in Hajjem et al. (2011) and Salditt et al. (2023). I used the same random intercept population model (cf. Salditt et al., 2023), i.e. the (numeric) outcome y_{ij} for observation i in cluster j was modeled was

$$y_{ij} = f(\boldsymbol{x}_{ij}) + b_j + \varepsilon_{ij},$$

where $f(\boldsymbol{x}_{ij})$ is the fixed effects linear predictor, b_j the random intercept effect of cluster j and ε_{ij} the respective error term. As covariates, I simulated nine standard normally distributed random variables $X_1, \dots X_9$ with all variables correlated to each other with a correlation of $\rho = 0.4$. As in Hajjem et al. (2011), the fixed effects linear predictor was simulated as

$$f(\boldsymbol{x}_{ij}) = 2x_{1ij} + x_{2ij}^2 + 4 \cdot \mathbb{1}_{x_{3ij}>0} + 2\log\left(|x_{1ij}|\right) x_{3ij}.$$

The random intercept effects were generated from a normal distribution with expected mean $\mu_b = 0$ and variance

$$\sigma_b^2 = \frac{ICC}{1 - ICC},$$

where ICC (intraclass correlation) was varied between $0.05, 0.25, 0.50$ as in Salditt et al. (2023) to cover different magnitudes of random effect variance. The error terms were simulated as standard normally distributed. To transform the numeric outcomes into ordinal response categories, I assigned five categories based on specifically selected threshold values. Using a similar approach as in Hornung (2019) and Buczak et al. (2024), the threshold values were chosen such that in a simulated population of size 100 000 a specific response category distribution pattern emerged. Analogously to Buczak (2024), I considered a response pattern with equally distributed categories as well as a pattern with prominent middle categories (denoted as wide middle pattern). For equally distributed response categories, relative category frequencies of $0.20, 0.20, 0.20, 0.20, 0.20$ were targeted,

while for the wide middle pattern relative category frequencies of $0.11, 0.22, 0.33, 0.22, 0.11$ were targeted, respectively. The threshold values derived from this are displayed in Table A1 (see Appendix A). The number of clusters was varied between 100 and 250. I further followed Salditt et al. (2023) regarding cluster sizes. For simulation conditions with 100 clusters, the number of observations from each cluster in the training data was randomly drawn from a discrete uniform distribution with bounds 10 and 15, while each cluster contained 10 observations in the test data. For simulation conditions with 250 clusters, the number of observations from each cluster was randomly drawn from a discrete uniform distribution with bounds 25 and 35, while each cluster contained 25 test observations, respectively.

I compared mixfabOF to the following (ordinal) prediction methods: fabOF (Buczak, 2024) as implemented in the `fabOF` package available from GitHub (https://github.com/phibuc/fabOF), OF (Hornung, 2019) using the `ordinalForest` package (Hornung, 2022), multi-label classification RF (Breiman, 2001) as implemented in the `ranger` package (Wright & Ziegler, 2017), OMERF (Bergonzoli et al., 2024) using the implementation provided by the authors on GitHub (https://github.com/giuliabergonzoli/OMERF) as well as a Cumulative Logit Mixed Model (CLMM; see e.g., Tutz & Hennevogl, 1996) as implemented in the `ordinal` package (Christensen, 2022). The CLMM was specified such that it included all linear main effects as well as a random intercept. Since fabOF, OF and RF do not support hierarchical data structures, I included the grouping variable as an additional covariate such that these methods can make use of the grouping information. All computations were run using `R` version 4.2.1 (R Core Team, 2023). For all RF-based methods, I used 500 trees as is a common default value, e.g., in the `ranger` package. As the maximum number of iterations for OMERF, I have selected 100 as is the suggested default setting by Bergonzoli et al. (2024). I used the same maximum number of iterations for mixfabOF. For the remaining parameters of the individual methods, I used the respective default values. I did not

perform a hyperparameter tuning as RFs have been shown to be relatively robust regarding their parameter settings (Probst et al., 2019). This design decision is in line with previous works from the field of RF-based ordinal prediction (Buczak et al., 2024; Hornung, 2019; Tutz, 2021). To assess the predictive performance of the different prediction methods, I have used Cohen's weighted Kappa (Cohen, 1968) with linear and quadratic weights as well as Kendall's rank correlation (Kendall, 1948) as performance measures. These measures are commonly used in the context of ordinal prediction (e.g., Ben-David, 2008; Buczak et al., 2024; Hornung, 2019). Similar to Cohen's Kappa (Cohen, 1960), weighted Kappa is a measure of agreement, in this case between the predicted and true response categories. Through the weights, the ordinal nature of the response is reflected as the "distance" between true and predicted categories is taken into account. Different weighting schemes allow for accentuating deviations from the true categories differently (Hornung, 2019). Linear and quadratic weights are among the most common choices for ordinal prediction (Ben-David, 2008; Hornung, 2019). All simulation conditions were run with 1 000 replications.

Simulation Results

In the following, the results from the simulation study will be presented. As the choice of response category distribution pattern only had little impact on the results, I will only be displaying results for the wide middle pattern here. For the remaining results, I refer to the Supplement. In all conditions, OMERF suffered from high rates of non-convergence. For small cluster sizes, OMERF converged in less than 1% of the runs, while for large cluster sizes OMERF only converged in about 11% of the runs. As such, this must be kept in mind when interpreting OMERF's results. In contrast to OMERF, mixfabOF converged in all simulation runs.

Figure 1 shows the results for data generated with $ICC = 0.05$, i.e., with small random effect variability. For all performance measures, similar result patterns emerged. It can be seen that the CLMM and OMERF fell notably behind the other methods. For the

CLMM, this can be explained by the highly non-linear effect structure. Whereas for mostly linear effects, parametric models tend to outperform RF-based approaches for ordinal prediction, RF-based methods tend to perform better under non-linear effects (Buczak et al., 2024). Regarding the remaining methods, RF slightly trailed mixfabOF, fabOF and OF which performed mostly similarly. For settings with 100 clusters, however, fabOF and mixfabOF tended to slightly outperform OF. As the random effect variability was low, the similar performance of fabOF and mixfabOF was to be expected. Generally, increasing the number of clusters and the size of the clusters led to reduced variability of the results for all methods and to improved predictive performance for mixfabOF, fabOF, OF and RF. For the CLMM and OMERF, predictive performance remained mostly unaffected by the number of clusters and cluster sizes.

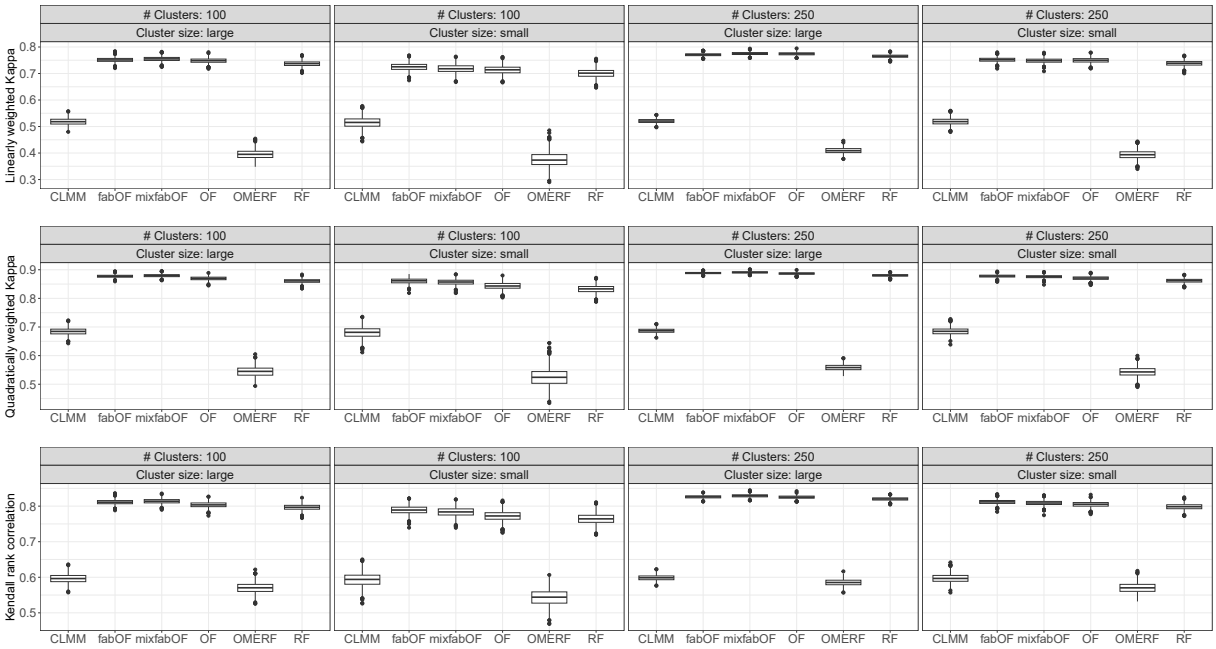

Figure 1

Predictive performance of prediction methods based on number of clusters and cluster sizes for $ICC = 0.05$.

Figure 2 shows the results for settings with moderate random effect variability

($ICC = 0.25$). Whereas for the low random effect variability conditions, mixfabOF, fabOF and OF performed similarly, the increased random effect variability resulted in mixfabOF pulling slightly ahead of fabOF and OF. This was most pronounced for settings with 250 clusters or large cluster sizes. Apart from this, the remaining findings from the low random effect variability settings mostly carried over. RF slightly trailed behind mixfabOF, fabOF and OF, while the CLMM and OMERF achieved notably lower predictive performance. As before, increasing the number of clusters and the cluster sizes, resulted in a reduction of variability and an improvement in predictive performance for mixfabOF, fabOF, OF and RF.

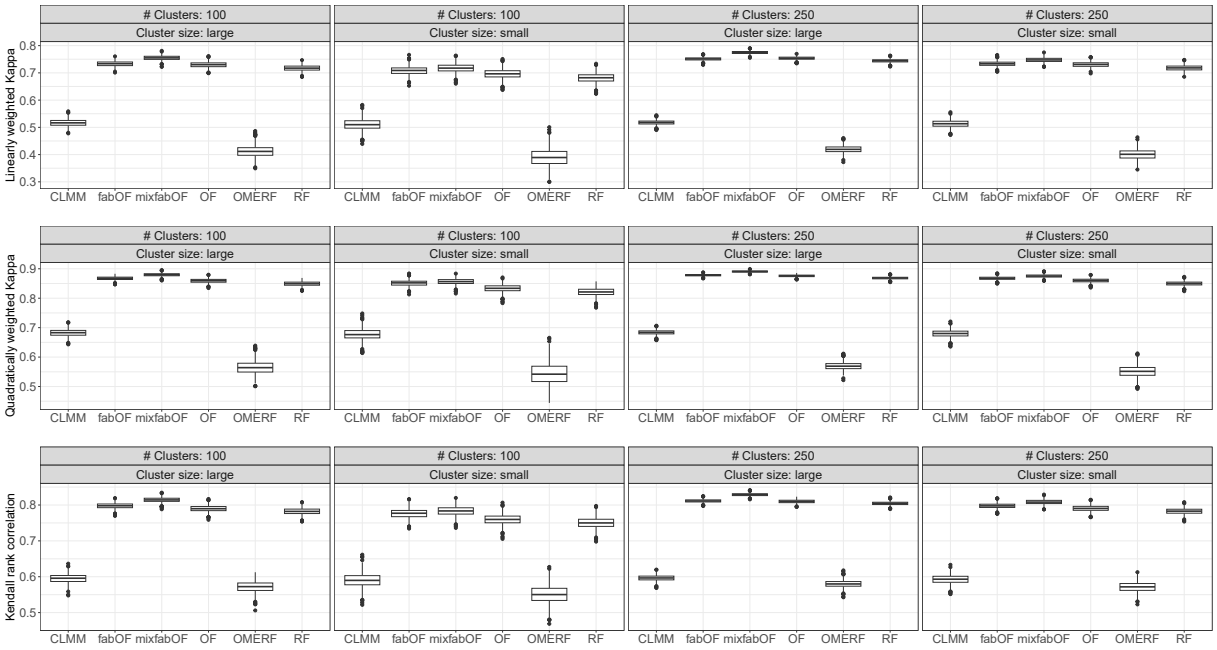

Figure 2

Predictive performance of prediction methods based on number of clusters and cluster sizes for $ICC = 0.25$.

Figure 3 displays the results for the simulation conditions with high random effect variability. It can be seen that mixfabOF achieved the highest predictive performance in all scenarios. The further increase in random effect variability has resulted in a wider

performance gap between mixfabOF and the two most competitive methods, fabOF and OF. As for the other two random effect variability settings, RF slightly lagged behind these three predictions methods, while the CLMM and OMERF fell further behind. Similarly, an increase in number of clusters and cluster sizes led to lower variability for all methods and higher predictive performance for mixfabOF, fabOF, OF and RF. Overall, the findings

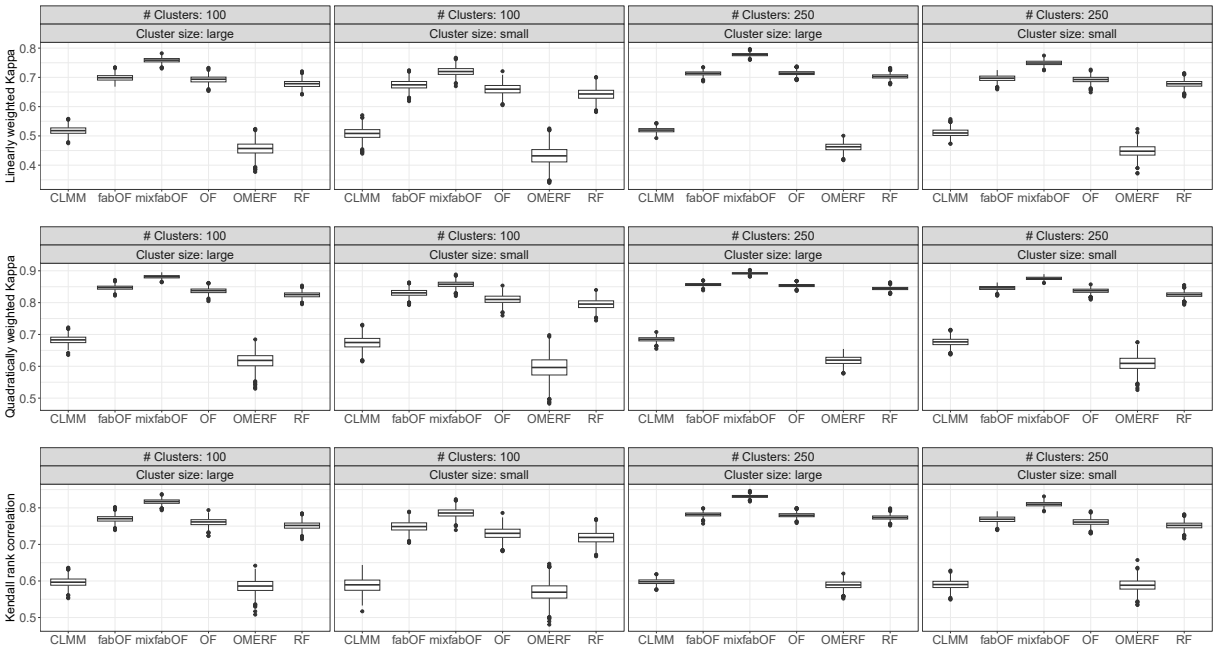

Figure 3

Predictive performance of prediction methods based on number of clusters and cluster sizes for $ICC = 0.50$.

from this simulation study are promising as mixfabOF displayed similar predictive performance as fabOF for low random effect variability and improved upon the latter for medium and high random effect variability for which it achieved the highest predictive performance of all methods.

Runtime Analysis

Apart from the predictive performance of the different ordinal prediction methods, their computational runtime is another factor warranting consideration. Buczak et al.

(2024) have demonstrated that the computational runtimes of ordinal prediction methods can vary notably. Therefore, I also performed a runtime analysis of the prediction methods compared in this work similar to the one in Buczak et al. (2024). Because all computations were performed on a compute cluster, the individual runs may not be perfectly comparable regarding the CPU nodes assigned by the cluster's workload manager or the current overall workload of the cluster at any given time. Additionally, all computations were restricted to using only a single CPU core which could have negatively impacted methods relying on parallelization. However, many prediction methods considered here are based on the same RF implementation from the `ranger` package (Wright & Ziegler, 2017), thus, benefiting comparability. Overall, the following results should not be interpreted as precise runtime comparisons, but rather as indications of the potential magnitudes of runtime differences between the prediction methods. For the runtime analysis, I have selected the simulation condition where data is generated using $ICC = 0.25$ and a wide middle response category distribution pattern for 250 clusters of large size (i.e, leading to the largest datasets).

Figure 4 shows the computational runtimes of the individual methods relative to the runtime of fabOF. Relative runtimes offer the benefit of being less dependent on the machine used for running the experiments. As fabOF was the fastest method overall, I have selected it as the reference method. For better visibility, I have logarithmized the relative runtimes using base 10. Consequently, a value of 0 indicates a runtime equal to fabOF while a value of 1 indicates a runtime larger than fabOF by a factor of 10. Since fabOF internally fits a single regression RF, it was to be expected that RF came closest to fabOF in runtime. For the data considered here, CLMM and mixfabOF required similar runtimes with mixfabOF's relative runtimes being slightly smaller on average and varying less. The relative runtimes of OF and OMERF were notably larger. As OMERF internally fits an OF model during its initialization, it can be seen that this step makes up a bulk of its runtime. It should be noted that OF's runtime is directly linked to the resources allotted to its optimization process. While the default values were used here, reducing the number

of score/category border sets generated during the optimization step, can reduce OF's runtime. Furthermore, as noted above, OMERF was affected by high non-convergence rates in this simulation. Increasing OMERF's maximum number of iterations may potentially remedy these issues, but would in turn increase OMERF's runtime even further.

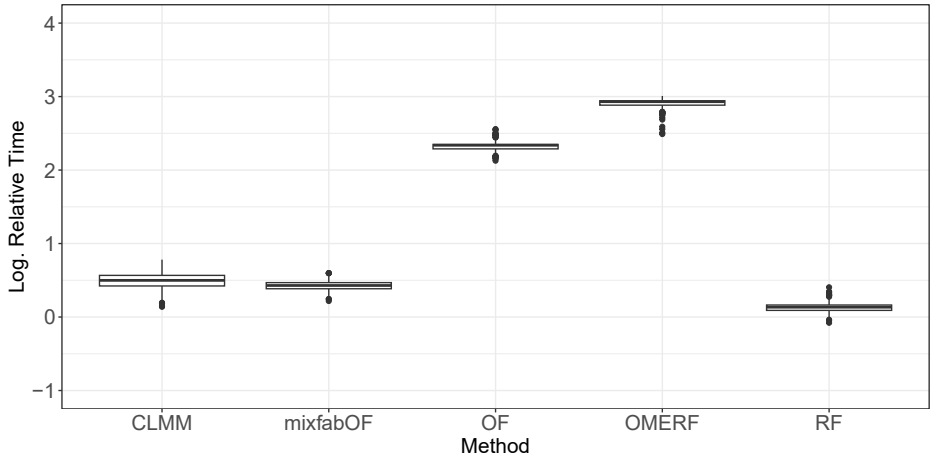

Figure 4

Computational runtime relative to fabOF for data with 250 clusters of large size. Values have been logarithmized using base 10.

Illustrative Data Example

In addition to the simulation study, I have also evaluated mixfabOF on an illustrative data example stemming from the Trends in International Mathematics and Science Study (TIMSS) 2019 data (Fishbein et al., 2021). TIMSS surveys the achievement of international fourth and eighth grade students in mathematics and science. For this analysis, I focused on a subset of the original data including only German students. As in Germany only data from fourth grade students is collected, the subset of the data accordingly only contained fourth graders. The goal of the prediction task constructed for this analysis was to predict the mathematical ability of students based on the students' sex, age, number of home study supports as well as their values on scales on disorderly behavior during math lessons, instructional clarity in math lessons, sense of school belonging,

bullying experiences, liking of learning math, confidence in math, and self-efficacy in computer use. Only complete observations from schools with at least five observations were considered, resulting in a sample size of 2773 students from 191 different schools. When fitting a random intercept LMM without any covariates to the numeric outcome, an estimated ICC of 0.23 resulted, indicating the presence of moderate random effect variability. For creating the ordinal response, I binned the original numeric mean ability score (ranging from 0-1000) into five ordinal categories: $[0, 450)$, $[450, 500)$, $[500, 550)$, $[550, 600)$ and $[600, 1000)$ with $n_1 = 354$, $n_2 = 598$, $n_3 = 778$, $n_4 = 681$ and $n_5 = 362$. I compared mixfabOF with the same methods as in the simulation study using the same settings. For the CLMM, all linear main effects and a random intercept were included. For RF, fabOF and OF, the grouping factor was included as an additional covariate. Predictive performance was assessed with a five-fold cross-validation (CV) using weighted Kappa with linear and quadratic weights as well as Kendall's rank correlation as performance measures. The sampling of the CV folds was performed at the cluster-level such that observations from each school were included in the training and the test data, respectively.

Figure 5 shows the predictive performance achieved by the different prediction methods in 100 replications. It can be seen that mixfabOF generally reached the best performance for all three performance measures. Comparing mixfabOF to the non-hierarchical prediction methods (particularly to its direct counterpart fabOF), the results demonstrate the usefulness of accounting for hierarchical structures for the present data. While performing better than the non-hierarchical OF and RF for weighted Kappa with quadratic weights and Kendall's rank correlation, OMERF falls behind mixfabOF, fabOF and the CLMM for all performance measures. Similar to the simulation study, OMERF was affected by convergence issues where for each run the maximum number of iterations was reached at least once during the CV loop. The differences between mixfabOF and the CLMM can likely be attributed to the nature of the underlying effects (linear vs. non-linear). It is to be expected that the relation between the predictive

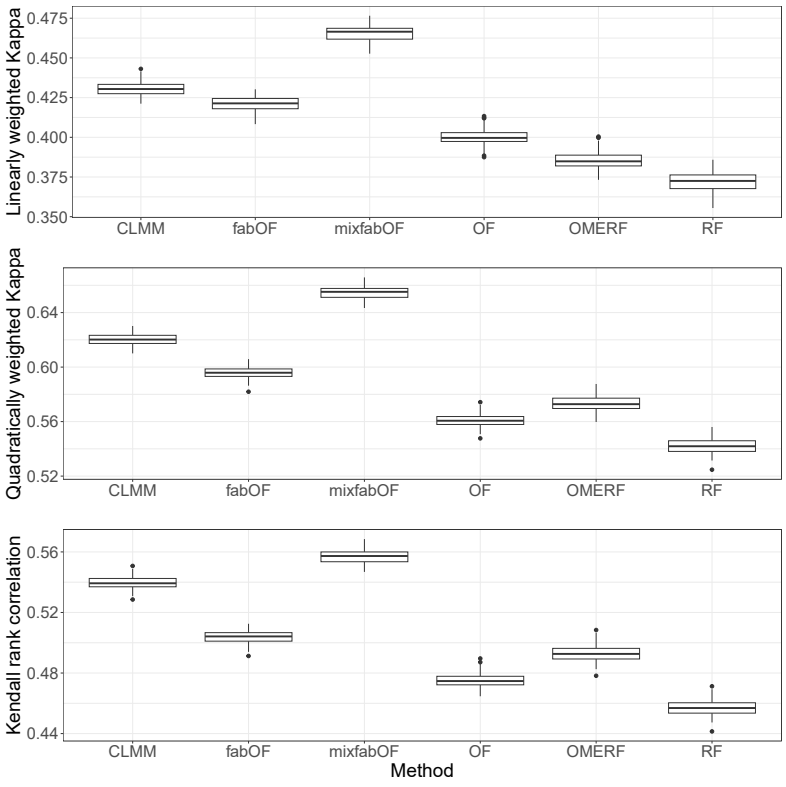

Figure 5

Predictive performance achieved by prediction methods on TIMSS data.

performance of mixfabOF and the CLMM is modulated by the effect nature (Buczak et al., 2024) and will likely vary across different datasets.

To examine the impact of the individual covariates on the predictive performance of mixfabOF, I computed the permutation variable importance values obtained when fitting a mixfabOF model to the entire dataset with clusterwise permutations. When allowing for permutations across all clusters, the results were affected only slightly. Figure 6 shows that the confidence in math scale is the most important covariate for the model's predictive performance. This is in line with results from the educational research literature which identified math confidence and the related concept of math self-efficacy as important predictors for math achievement (Jiang et al., 2013; Pitsia et al., 2017; Stankov et al., 2012). While the other covariates achieved notably lower importance values, some caution is advised when interpreting these results as some covariates displayed moderate to high

degrees of correlation. For example, the Pearson correlation between "confidence in math" and "like learning math" was 0.66. Unconditional VIMs as the one used here, are known to be affected by highly correlated covariates (see, e.g., Molnar, 2022; Nicodemus et al., 2010; Strobl et al., 2008). For assessing the reliability of the results, a comparison with results for a conditional VIM (e.g., in the vein of Strobl et al., 2008) would be desirable. In contrast to unconditional permutation VIMs, conditional permutation VIMs place restrictions on the permutation process such that the original correlation structure between covariates is better preserved (Strobl et al., 2008). Currently, there is no conditional VIM available for mixfabOF. Since conditional VIMs as proposed by Strobl et al. (2008) operate on the tree-level to determine permitted permutations and to compute the variable importance, an analogous implementation for mixfabOF would require further adjustments. This is due to the fact that mixfabOF does not transform its internal numeric scores used for representing the ordinal categories back into ordinal category predictions until they have been aggregated at the forest-level. As such, the variable importance cannot be evaluated at the tree-level (see also Buczak, 2024, for a more detailed discussion). As a consequence, implementing a conditional VIM for mixfabOF in future work would likely require a different approach than the one proposed by Strobl et al. (2008).

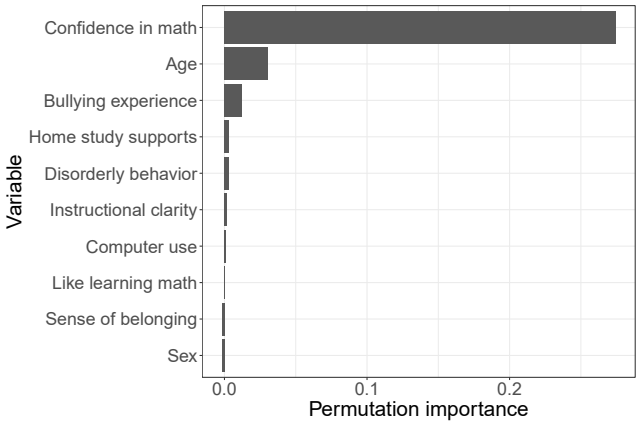

Figure 6

Permutation variable importance values for mixfabOF model on TIMSS data.

Discussion

In this work, I proposed Mixed-Effects Frequency-Adjusted Borders Ordinal Forest (mixfabOF), an ordinal prediction method specifically tailored towards ordinal prediction tasks with hierarchical data structures. The proposed methods extends Frequency-Adjusted Borders Ordinal Forest (fabOF; Buczak, 2024) for use with hierarchical data by adapting the iterative fixed and random effects estimation procedure employed in Mixed-Effects Random Forest (MERF; Hajjem et al., 2012). To this end, mixfabOF assigns numeric scores to the ordinal response categories and uses these scores to iterate between fitting a regression random forest (RF; Breiman, 2001) to estimate the fixed effects component and fitting a linear mixed model (LMM; see e.g., Molenberghs & Verbeke, 2000) to estimate the random effects component. Having arrived at the final estimates for the fixed and random effect components, mixfabOF follows fabOF in determining the numeric category borders that are used for predicting new observations based on the cumulative relative frequencies of the ordinal response categories in the data. Through simulation and an illustrative example from the Trends in International Mathematics and Science Study (TIMSS) 2019 study (Fishbein et al., 2021), I demonstrated that mixfabOF can achieve higher predictive performance under medium and high random effect variability than existing (ordinal) prediction methods such as fabOF, Ordinal Forest (OF; Hornung, 2019) and multi-label classification RF.

Furthermore, mixfabOF achieved notably higher predictive performance for the simulated and real data considered in this work than Ordinal Mixed-Effect Random Forest (OMERF; Bergonzoli et al., 2024), which at the time of writing this work is (to my knowledge) the only method for ordinal prediction of hierarchical data proposed so far. Since OMERF relies on fitting an OF model internally, it is also affected by the computational runtime of OF's optimization procedure. As such, the runtime analysis performed in this work also revealed significant runtime advantages of mixfabOF over OMERF. However, some part of this disparity may be explained by the high rates of

non-convergence from which OMERF suffered in the simulation and real data experiments. This may have potentially affected OMERF's predictive performance as well. Experimenting with higher maximum numbers of iterations did not alleviate the convergence issues. As such, I was not able to obtain an explanation for OMERF's behavior. Since OMERF is a very recent method, available references and recommendations for OMERF are scarce. Therefore, an incorrect use of the implementation in this work cannot be ruled out with complete certainty. To obtain an additional comparison and to check for potential misuse of the method, I additionally performed a benchmark study on the random intercept model used for simulation in Bergonzoli et al. (2024). Figure B1 shows that mixfabOF achieved the highest predictive performance for all performance measures overall followed by OMERF and fabOF. For this data generating model, OMERF converged in all 100 replications. As Bergonzoli et al. (2024) only used data where the ordinal response consisted of three categories, perhaps the number of ordinal categories affects the convergence rates of OMERF. Figure B2 indicates that despite OMERF's improved convergence rates, mixfabOF still required notably less runtime than OMERF due to the computational runtime associated with fitting an OF model.

Apart from the RF-based approaches, I have also compared mixfabOF with a Cumulative Logit Mixed Model (CLMM; see e.g., Hedeker & Gibbons, 1994; Tutz & Hennevogl, 1996) in this work. While mixfabOF achieved higher predictive performance for the simulated and real data, it should be noted that this is likely caused by the effect structure of the data considered in this work. The data generating process in the simulation was characterized by mostly non-linear effects. In their comparison of RF-based ordinal prediction methods and a parametric model, Buczak et al. (2024) found that for predominantly linear effects, RF-based methods fell behind the parametric model, while for predominantly non-linear effects, the RF-based methods outperformed the parametric model. As such, it is plausible to expect that for data adhering to a mostly linear effect structure, the CLMM may outperform mixfabOF (and other RF-based prediction

methods). Therefore, the choice between a CLMM and mixfabOF should be guided either by prior knowledge or by benchmarking both methods on a subset of the data at hand.

While the simulation and illustrative data example only featured random intercept models, mixfabOF can in principle also account for random slopes or other random effect structures specifiable in an LMM. Future work could explore the use of mixfabOF for random effect structures beyond the random intercept model. In the context of ordinal regression models, e.g., the cumulative model (McCullagh, 1980), another type of random effects that can occur are random thresholds, i.e., cluster-specific category thresholds (Tutz & Hennevogl, 1996). As this type of random effect cannot be accounted for currently by mixfabOF, future work could study how such effects can be translated to the framework used by (mix)fabOF. One possibility could be to compute cluster-specific category borders instead of computing global category borders based on all observations.

Overall, this work has demonstrated the usefulness of accounting for hierarchical data structures in ordinal prediction tasks when using RF-based prediction methods. The newly proposed mixfabOF method extends fabOF in a meaningful way and could improve upon fabOF and other RF-based prediction methods for the data studied in this work. In light of the growing quantities of data in the social and life sciences sparking a rising interest in ML methods, these are promising findings that motivate further investigation and methodological refinement.

References

Archer, K. J. (2010). rpartOrdinal: An R package for deriving a classification tree for predicting an ordinal response. *Journal of Statistical Software*, *34*(7), 1–17. https://doi.org/10.18637/jss.v034.i07

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Ben-David, A. (2008). Comparison of classification accuracy using Cohen's weighted kappa. *Expert Systems with Applications*, *34*(2), 825–832. https://doi.org/10.1016/j.eswa.2006.10.022

Bergonzoli, G., Rossi, L., & Masci, C. (2024). Ordinal mixed-effects random forest [Pre-print version 1]. https://doi.org/10.48550/ARXIV.2406.03130

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 123–140. https://doi.org/10.1023/A:1010933404324

Buczak, P. (2024). fabOF: A novel tree ensemble method for ordinal prediction [Pre-print version 1.2]. https://doi.org/10.31219/osf.io/h8t4p

Buczak, P., Horn, D., & Pauly, M. (2024). Old but gold or new and shiny? Comparing tree ensembles for ordinal prediction with a classic parametric approach [Pre-print version 1.1]. https://doi.org/10.31219/osf.io/v7bcf

Capitaine, L., Genuer, R., & Thiébaut, R. (2020). Random forests for high-dimensional longitudinal data. *Statistical Methods in Medical Research*, *30*(1), 166–184. https://doi.org/10.1177/0962280220946080

Christensen, R. H. B. (2022). Ordinal—regression models for ordinal data [R package version 2022.11-16. https://CRAN.R-project.org/package=ordinal].

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*(1), 37–46. https://doi.org/10.1177/001316446002000104

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, *70*(4), 213–220. https://doi.org/10.1037/h0026256

Costa-Mendes, R., Oliveira, T., Castelli, M., & Cruz-Jesus, F. (2020). A machine learning approximation of the 2015 Portuguese high school student grades: A hybrid approach. *Education and Information Technologies*, *26*(2), 1527–1547. https://doi.org/10.1007/s10639-020-10316-y

De'ath, G. (2002). Multivariate regression trees: A new technique for modeling species–environment relationships. *Ecology*, *83*(4), 1105–1117. https://doi.org/10.1890/0012-9658(2002)083[1105:mrtant]2.0.co;2

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 1–38.

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. D. (2021). *Regression: Models, methods and applications* (2nd ed.). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-63882-8

Fishbein, B., Foy, P., & Yin, L. (2021). TIMSS 2019 user guide for the international database [Retrieved from Boston College, TIMSS & PIRLS International Study Center website: https://timssandpirls.bc.edu/timss2019/international-database/].

Fontana, L., Masci, C., Ieva, F., & Paganoni, A. M. (2021). Performing learning analytics via generalised mixed-effects trees. *Data*, *6*(7), 74. https://doi.org/10.3390/data6070074

Galimberti, G., Soffritti, G., & Maso, M. D. (2012). Classification trees for ordinal responses in R: The rpartScore package. *Journal of Statistical Software*, *47*(10), 1–25. https://doi.org/10.18637/jss.v047.i10

Hajjem, A., Bellavance, F., & Larocque, D. (2011). Mixed effects regression trees for clustered data. *Statistics & Probability Letters*, *81*(4), 451–459.

Hajjem, A., Bellavance, F., & Larocque, D. (2012). Mixed-effects random forest for clustered data. *Journal of Statistical Computation and Simulation*, *84*(6), 1313–1328. https://doi.org/10.1080/00949655.2012.741599

Hajjem, A., Larocque, D., & Bellavance, F. (2017). Generalized mixed effects regression trees. *Statistics amp; Probability Letters*, *126*, 114–118. https://doi.org/10.1016/j.spl.2017.02.033

Hedeker, D., & Gibbons, R. D. (1994). A random-effects ordinal regression model for multilevel analysis. *Biometrics*, *50*(4), 933. https://doi.org/10.2307/2533433

Hornung, R. (2019). Ordinal forests. *Journal of Classification*, *37*(1), 4–17. https://doi.org/10.1007/s00357-018-9302-x

Hornung, R. (2022). *ordinalForest: Ordinal forests: Prediction and variable ranking with ordinal target variables* [R package version 2.4-3]. https://CRAN.R-project.org/package=ordinalForest

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, *15*(3), 651–674. https://doi.org/10.1198/106186006X133933

Hu, J., & Szymczak, S. (2023). A review on longitudinal data analysis with random forest. *Briefings in Bioinformatics*, *24*(2). https://doi.org/10.1093/bib/bbad002

Janitza, S., Tutz, G., & Boulesteix, A.-L. (2016). Random forest for ordinal responses: Prediction and variable selection. *Computational Statistics & Data Analysis*, *96*, 57–73. https://doi.org/10.1016/j.csda.2015.10.005

Jiang, Y., Song, J., Lee, M., & Bong, M. (2013). Self-efficacy and achievement goals as motivational links between perceived contexts and achievement. *Educational Psychology*, *34*(1), 92–117. https://doi.org/10.1080/01443410.2013.863831

Kathan, A., Harrer, M., Küster, L., Triantafyllopoulos, A., He, X., Milling, M., Gerczuk, M., Yan, T., Rajamani, S. T., Heber, E., Grossmann, I., Ebert, D. D., & Schuller, B. W. (2022). Personalised depression forecasting using mobile sensor data

and ecological momentary assessment. *Frontiers in Digital Health*, *4*. https://doi.org/10.3389/fdgth.2022.964582

Kendall, M. G. (1948). *Rank correlation methods*. Griffin.

Kramer, S., Widmer, G., Pfahringer, B., & de Groeve, M. (2000). Prediction of ordinal classes using regression trees. In *Lecture notes in computer science* (pp. 426–434). Springer Berlin Heidelberg.

Laird, N. M., & Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics*, *38*(4), 963. https://doi.org/10.2307/2529876

Lin, S., & Luo, W. (2019). A new multilevel cart algorithm for multilevel data with binary outcomes. *Multivariate Behavioral Research*, *54*(4), 578–592. https://doi.org/10.1080/00273171.2018.1552555

Loh, W.-Y., & Zheng, W. (2013). Regression trees for longitudinal and multiresponse data. *The Annals of Applied Statistics*, *7*(1). https://doi.org/10.1214/12-aoas596

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society: Series B (Methodological)*, *42*(2), 109–127. https://doi.org/10.1111/j.2517-6161.1980.tb01109.x

Molenberghs, G., & Verbeke, G. (2000). *Linear mixed models for longitudinal data*. Springer New York. https://doi.org/10.1007/978-1-4419-0300-6

Molnar, C. (2022). *Interpretable machine learning: A guide for making black box models explainable* (2nd ed.). https://christophm.github.io/interpretable-ml-book

Montag, C., & Baumeister, H. (Eds.). (2023). *Digital phenotyping and mobile sensing: New developments in psychoinformatics*. Springer International Publishing. https://doi.org/10.1007/978-3-030-98546-2

Nicodemus, K. K., Malley, J. D., Strobl, C., & Ziegler, A. (2010). The behaviour of random forest permutation-based variable importance measures under predictor correlation. *BMC Bioinformatics*, *11*. https://doi.org/10.1186/1471-2105-11-110

Pellagatti, M., Masci, C., Ieva, F., & Paganoni, A. M. (2021). Generalized mixed-effects random forest: A flexible approach to predict university student dropout. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *14*(3), 241–257. https://doi.org/10.1002/sam.11505

Piccarreta, R. (2007). Classification trees for ordinal variables. *Computational Statistics*, *23*(3), 407–427. https://doi.org/10.1007/s00180-007-0077-5

Pitsia, V., Biggart, A., & Karakolidis, A. (2017). The role of students' self-beliefs, motivation and attitudes in predicting mathematics achievement: A multilevel analysis of the programme for international student assessment data. *Learning and Individual Differences*, *55*, 163–173. https://doi.org/10.1016/j.lindif.2017.03.014

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, *9*(3), 1–15. https://doi.org/10.1002/widm.1301

R Core Team. (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. https://www.R-project.org/

Salditt, M., Humberg, S., & Nestler, S. (2023). Gradient tree boosting for hierarchical data. *Multivariate Behavioral Research*, *58*(5), 911–937. https://doi.org/10.1080/00273171.2022.2146638

Segal, M. R. (1992). Tree-structured methods for longitudinal data. *Journal of the American Statistical Association*, *87*(418), 407–418. https://doi.org/10.1080/01621459.1992.10475220

Sela, R. J., & Simonoff, J. S. (2012). RE-EM trees: A data mining approach for longitudinal and clustered data. *Machine Learning*, *86*, 169–207.

Speiser, J. L., Wolf, B. J., Chung, D., Karvellas, C. J., Koch, D. G., & Durkalski, V. L. (2018). Bimm tree: A decision tree method for modeling clustered and longitudinal binary outcomes. *Communications in Statistics - Simulation and Computation*, (4), 1004–1023. https://doi.org/10.1080/03610918.2018.1490429

Speiser, J. L., Wolf, B. J., Chung, D., Karvellas, C. J., Koch, D. G., & Durkalski, V. L. (2019). Bimm forest: A random forest method for modeling clustered and longitudinal binary outcomes. *Chemometrics and Intelligent Laboratory Systems*, *185*, 122–134. https://doi.org/10.1016/j.chemolab.2019.01.002

Stankov, L., Lee, J., Luo, W., & Hogan, D. J. (2012). Confidence: A better predictor of academic achievement than self-efficacy, self-concept and anxiety? *Learning and Individual Differences*, *22*(6), 747–758. https://doi.org/10.1016/j.lindif.2012.05.013

Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, *9*. https://doi.org/10.1186/1471-2105-9-307

Tutz, G. (2021). Ordinal trees and random forests: Score-free recursive partitioning and improved ensembles. *Journal of Classification*, *39*(2), 241–263. https://doi.org/10.1007/s00357-021-09406-4

Tutz, G. (2022). Ordinal regression: A review and a taxonomy of models. *WIREs Computational Statistics*, *14*(2), e1545. https://doi.org/10.1002/wics.1545

Tutz, G., & Hennevogl, W. (1996). Random effects in ordinal regression models. *Computational Statistics & Data Analysis*, *22*(5), 537–557. https://doi.org/10.1016/0167-9473(96)00004-7

Ulitzsch, E., Ulitzsch, V., He, Q., & Lüdtke, O. (2022). A machine learning-based procedure for leveraging clickstream data to investigate early predictability of failure on interactive tasks. *Behavior Research Methods*, *55*(3), 1392–1412. https://doi.org/10.3758/s13428-022-01844-1

van der Scheer, E. A., & Visscher, A. J. (2017). Effects of a data-based decision-making intervention for teachers on students' mathematical achievement. *Journal of Teacher Education*, *69*(3), 307–320. https://doi.org/10.1177/0022487117704170

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. https://doi.org/10.18637/jss.v077.i01

Appendix A

Thresholds for Simulation Study

ICC	Response pattern	Threshold values				
		γ_1	γ_2	γ_3	γ_4	γ_5
0.05	equal	-0.08	2	3.61	6.05	∞
	wide middle	-1.71	1.41	4.2	7.62	∞
0.25	equal	-0.12	1.98	3.64	6.09	∞
	wide middle	-1.75	1.37	4.24	7.66	∞
0.50	equal	-0.22	1.95	3.71	6.16	∞
	wide middle	-1.85	1.3	4.31	7.76	∞

Table A1

Threshold values based on ICC and response category distribution pattern settings.

Appendix B

Comparison for Simulation Model from Bergonzoli et al. (2024)

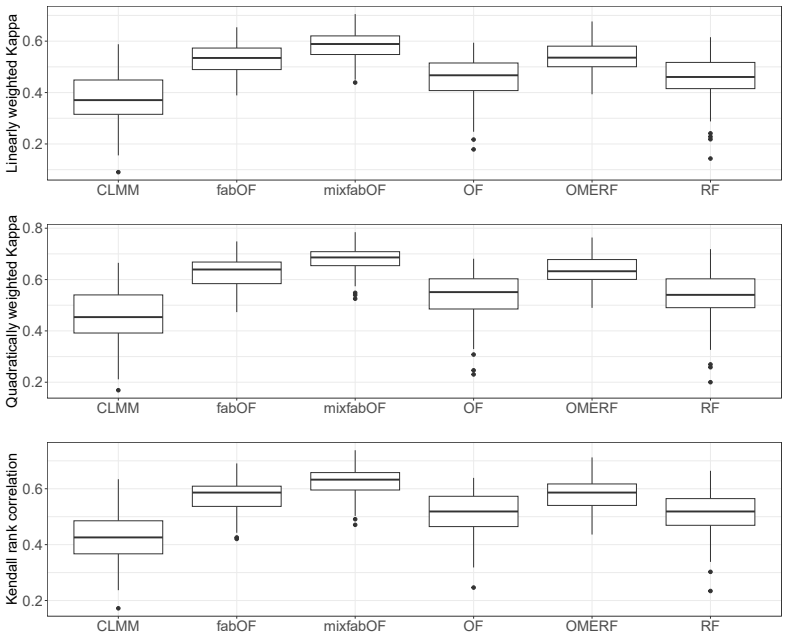

Figure B1

Predictive performance achieved by prediction methods on random intercept model simulation data from Bergonzoli et al. (2024).

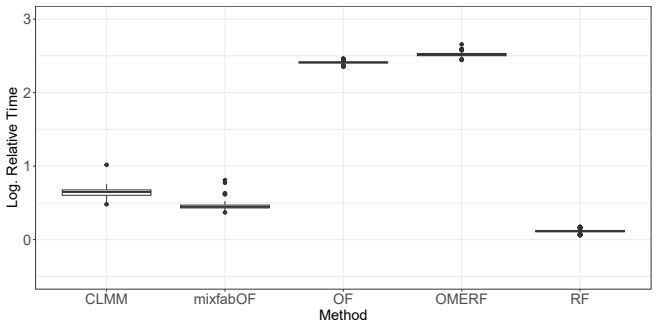

Figure B2

Computational runtime relative to fabOF for random intercept model simulation data from Bergonzoli et al. (2024). Values have been logarithmized using base 10.

Part III

Appendix

Appendix A - Hyperparameter Tuning

Table 6.1: Overview of hyperparameters and search spaces. *Note. For description of hyperparameters see* `ranger` *(Wright & Ziegler, 2017) and* `xgboost` *(Chen et al., 2024) packages.*

Method	Hyperparameter	Support	Search space
(mix)fabOF	`min.node.size`	$\{1, 2, \dots\}$	$\lfloor (n \cdot 0.2)^{\lambda} \rfloor$ with $\lambda \in [0, 1]$
	`mtry`	$\{1, \dots, p\}$	$\{1, \dots, p\}$
	`sample.fraction`	$[0, 1]$	$[0.2, 1]$
(mix)fabXGB	`alpha`	$[0, \infty)$	2^{λ} with $\lambda \in [-15, 15]$
	`eta`	$[0, 1]$	$[0, 1]$
	`max_depth`	$\{0, 1, \dots\}$	$\{1, \dots, 30\}$
	`min_child_weight`	$[0, \infty)$	2^{λ} with $\lambda \in [-15, 15]$
	`nrounds`	$\{1, 2, \dots\}$	$\{50, \dots, 300\}$

Appendix B - Pseudocode for mixfabXGB

Algorithm 11 Mixed-Effects Frequency-Adjusted Borders XGBoost

procedure MIXFABXGB

Unless specified otherwise, assign scores $(s_1, s_2, \ldots, s_k) \leftarrow (1, 2, \ldots, k)$.

Create $\boldsymbol{y}_j^{\text{num}}, j = 1, \ldots, m$, by assigning scores to ordinal response categories.

Set $\texttt{it} = 0, \hat{\boldsymbol{D}}_{(0)} = \boldsymbol{I}_{n_j \times n_j}, \hat{\boldsymbol{u}}_{j,(0)} = \boldsymbol{0}_{n_j}, j = 1, \ldots, m$.

while $\texttt{it} \leq \texttt{max.iter}$ and `not converged` **do**

$\texttt{it} = \texttt{it} + 1$

Update $\tilde{\boldsymbol{y}}_{j,(\texttt{it})}^{num}, \hat{f}_{(\texttt{it})}(\boldsymbol{X}_j)$ *and* $\hat{\boldsymbol{u}}_{j,(\texttt{it})}$:

$\tilde{\boldsymbol{y}}_{j,(\texttt{it})}^{\text{num}} = \boldsymbol{y}_j^{\text{num}} - \boldsymbol{Z}_j \hat{\boldsymbol{u}}_{j,(\texttt{it}-1)}, j = 1, \ldots, m$.

Obtain $\hat{f}_{(\texttt{it})}(\boldsymbol{X}_j)$ by fitting a regr. XGB model to $\tilde{\boldsymbol{y}}_{(\texttt{it})}^{\text{num}}$ with predictors $\boldsymbol{X}$.

$$\hat{\boldsymbol{u}}_{j,(\texttt{it})} = \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \boldsymbol{Z}_j^\top \hat{\boldsymbol{V}}_{j,(\texttt{it}-1)}^{-1} \left(\boldsymbol{y}_j^{\text{num}} - \hat{f}_{(\texttt{it})}(\boldsymbol{X}_j) \right), j = 1, \ldots, m,$$

where $\hat{\boldsymbol{V}}_{j,(\texttt{it}-1)}^{-1} = \boldsymbol{Z}_j \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \boldsymbol{Z}_j^\top + \hat{\sigma}_{(\texttt{it}-1)}^2 \boldsymbol{I}_{n_j \times n_j}$.

Update $\hat{\sigma}_{(it)}^2$ *and* $\hat{\boldsymbol{D}}_{(it)}$:

$$\hat{\sigma}_{(\texttt{it})}^2 = \frac{1}{n} \sum_{j=1}^{m} \hat{\boldsymbol{\varepsilon}}_{j,(\texttt{it})}^\top \hat{\boldsymbol{\varepsilon}}_{j,(\texttt{it})} + \hat{\sigma}_{(\texttt{it}-1)}^2 \left(n_j - \hat{\sigma}_{(\texttt{it}-1)}^2 \text{trace} \left(\hat{\boldsymbol{V}}_{j,(\texttt{it}-1)} \right) \right),$$

$$\hat{\boldsymbol{D}}_{(\texttt{it})} = \frac{1}{m} \sum_{j=1}^{m} \left\{ \hat{\boldsymbol{u}}_{j,(\texttt{it})} \hat{\boldsymbol{u}}_{j,(\texttt{it})}^\top + \left(\hat{\boldsymbol{D}}_{(\texttt{it}-1)} - \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \boldsymbol{Z}_j^\top \hat{\boldsymbol{V}}_{j,(\texttt{it}-1)}^{-1} \boldsymbol{Z}_j \hat{\boldsymbol{D}}_{(\texttt{it}-1)} \right) \right\},$$

where $\hat{\boldsymbol{\varepsilon}}_{j,(\texttt{it})} = \boldsymbol{y}_j^{\text{num}} - \hat{f}_{(\texttt{it})}(\boldsymbol{X}_j) - \boldsymbol{Z}_j \hat{\boldsymbol{u}}_{j,(\texttt{it})}$.

Check convergence using GLL criterion.

end while

Compute numeric fixed effects predictions $\hat{f}(\boldsymbol{X}_j)$ with final XGB model, $j = 1, \ldots, m$.

Compute numeric predictions $\hat{\boldsymbol{y}}_j^{\text{num}} = \hat{f}(\boldsymbol{X}_j) + \boldsymbol{Z}_j \hat{\boldsymbol{u}}_j, j = 1, \ldots, m$.

For categories up to category k, compute cumulative relative frequencies $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1}$.

Obtain prediction quantiles $q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}$ of numeric predictions $\hat{y}_{ij}^{\text{num}}$ for probabilities $\hat{\pi}_1, \ldots, \hat{\pi}_{k-1},\ i = 1, \ldots, n_j, j = 1, \ldots, m$.

Assign category borders $(b_1, b_2, \ldots, b_k, b_{k+1}) \leftarrow \left(-\infty, q_{\hat{\pi}_1}, \ldots, q_{\hat{\pi}_{k-1}}, \infty\right)$.

return XGB model, final random effect estimates and category borders

end procedure

Appendix C - Additional Results for mixfabXGB

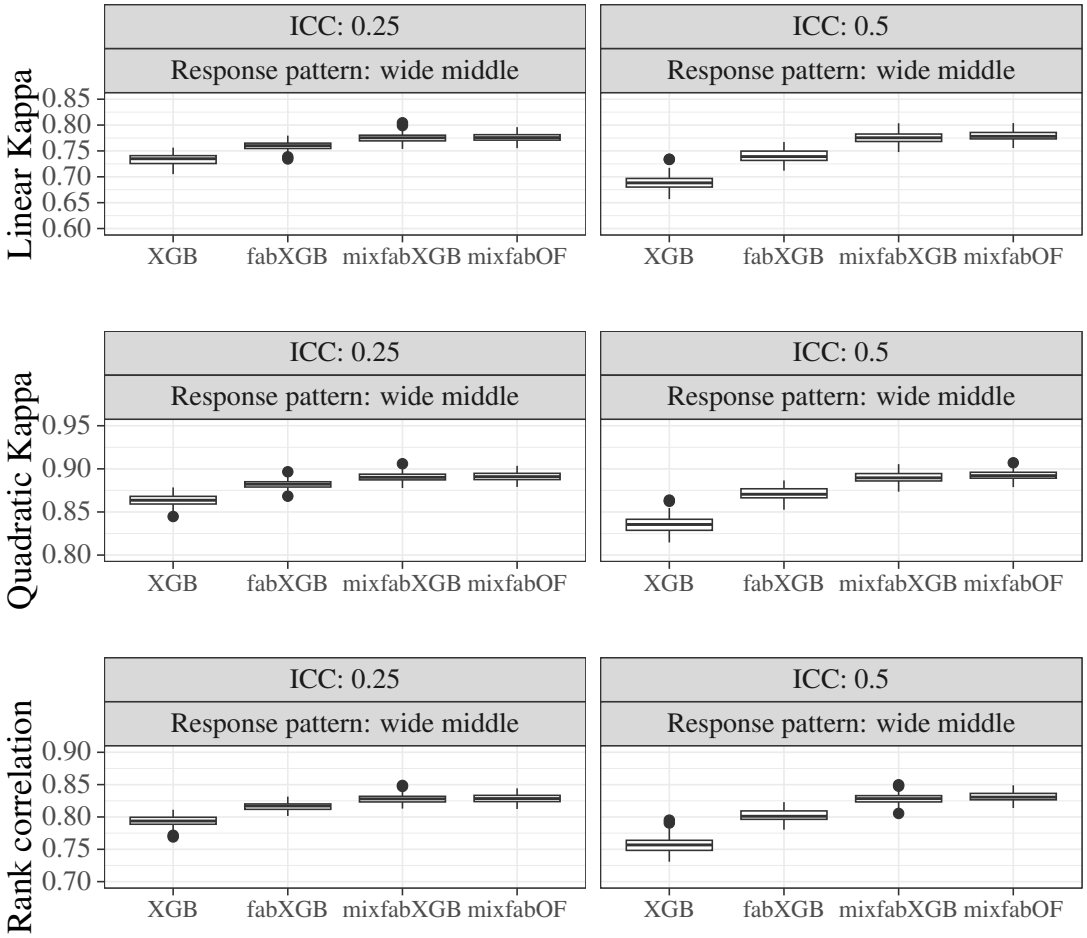

Figure 6.1: Predictive performance of mixfabXGB extension for wide middle response category distribution pattern.