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.,

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

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.

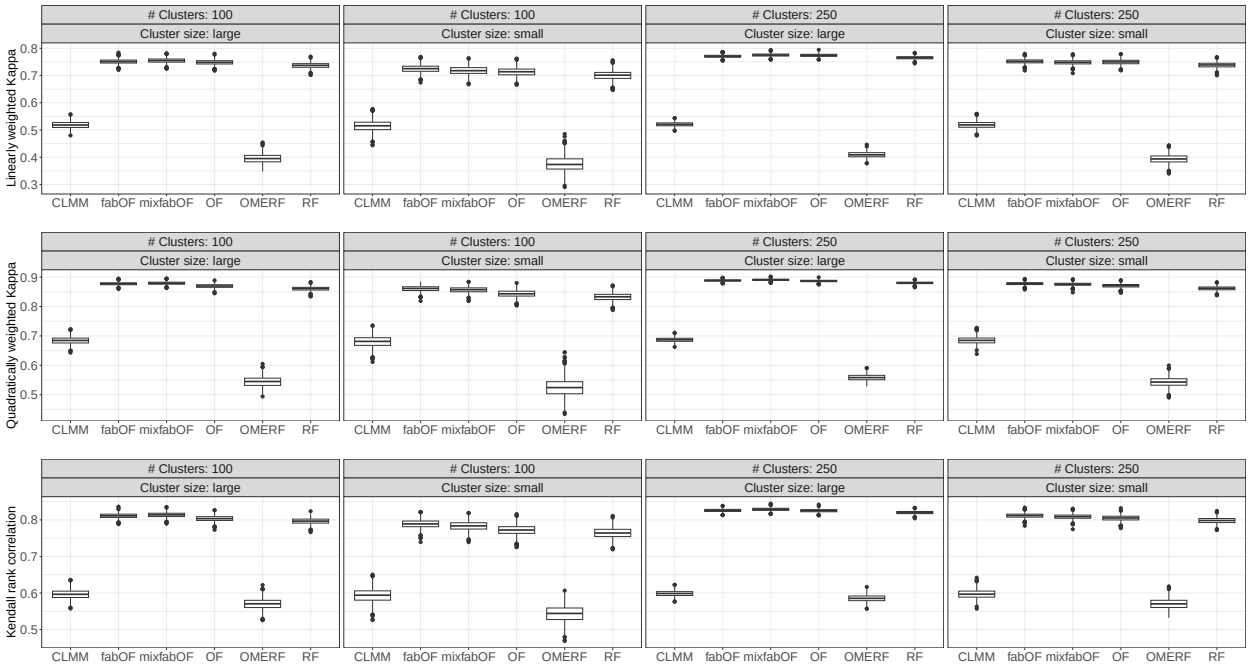

Figure 1

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.

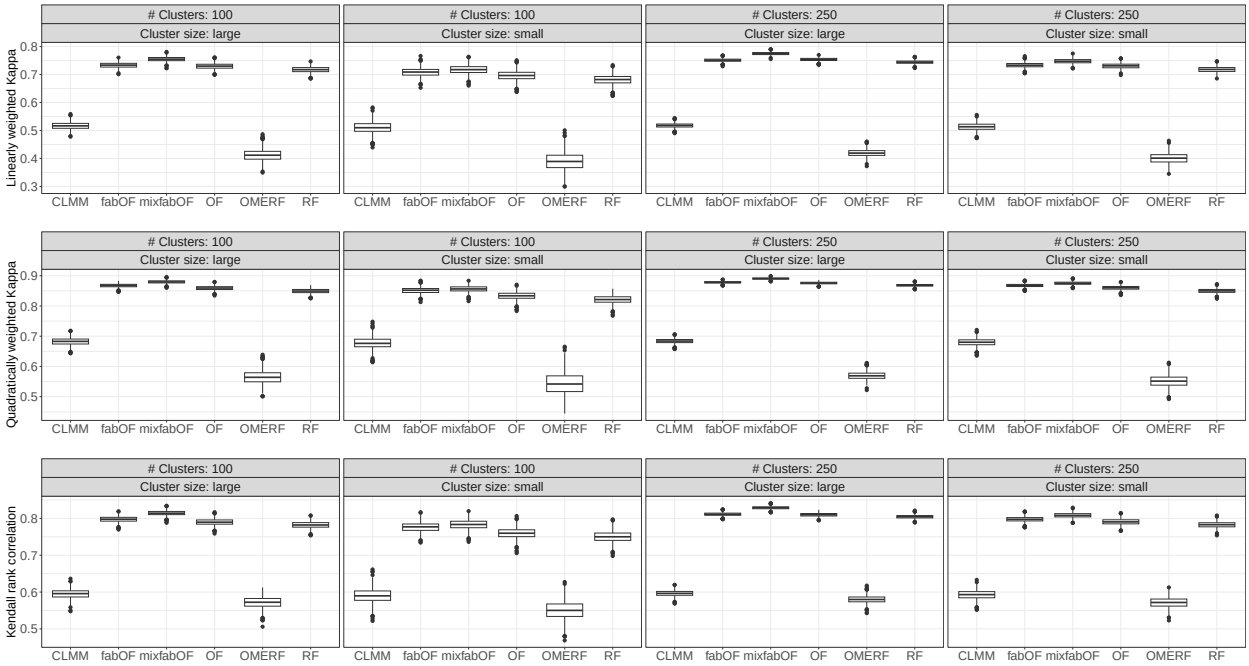

Figure 2

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

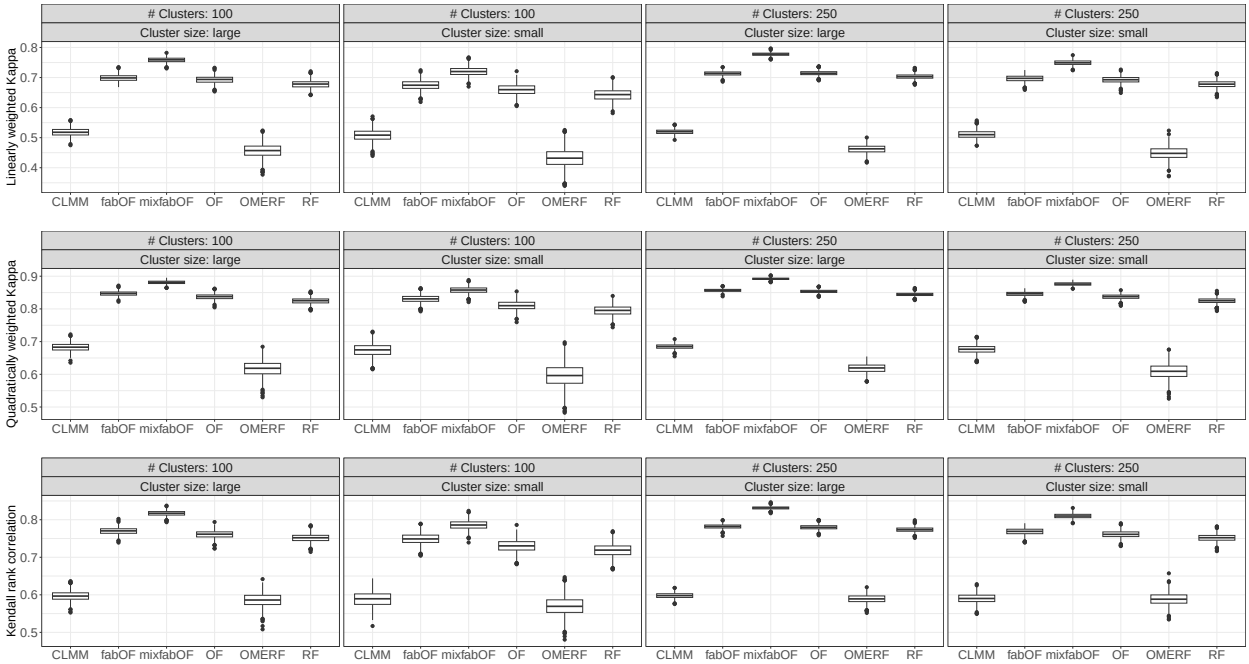

Figure 3

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.

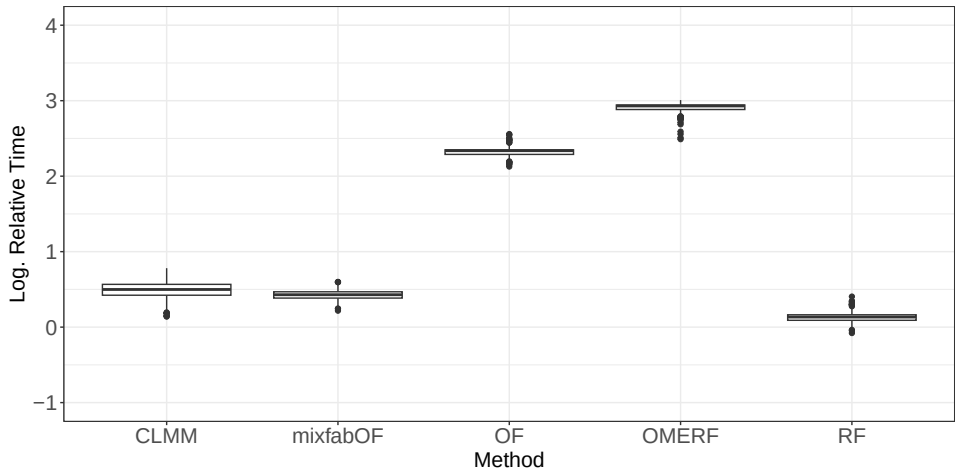

Figure 4

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

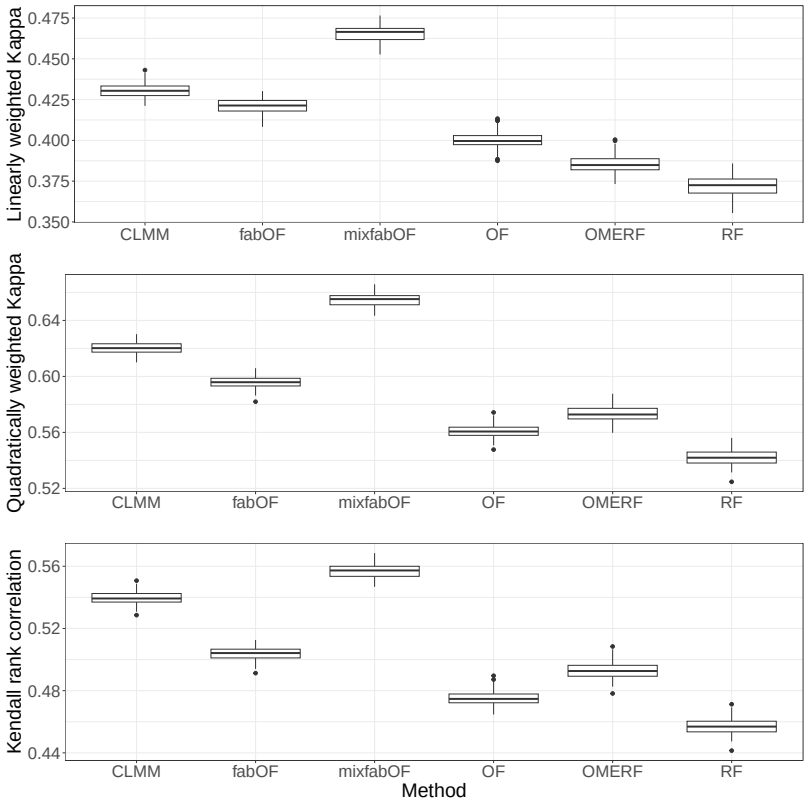

Figure 5

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).

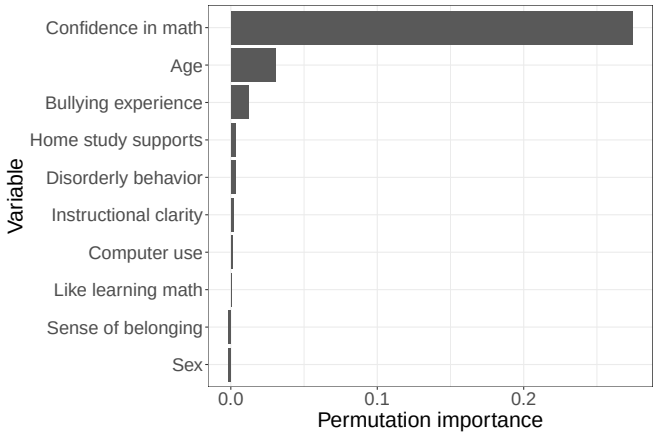

Figure 6

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)

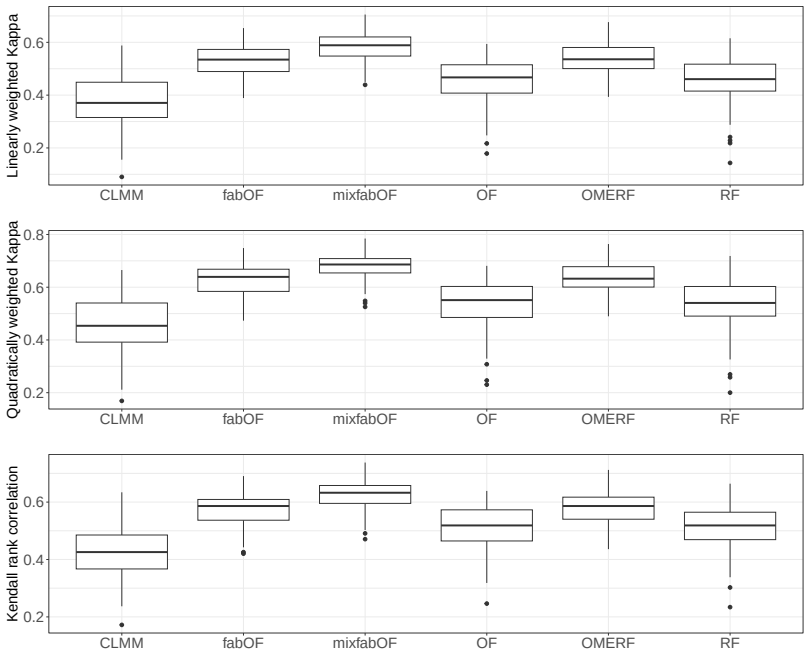

Figure B1

Predictive performance achieved by prediction methods on random intercept model simulation data from Bergonzoli et al. (2024).

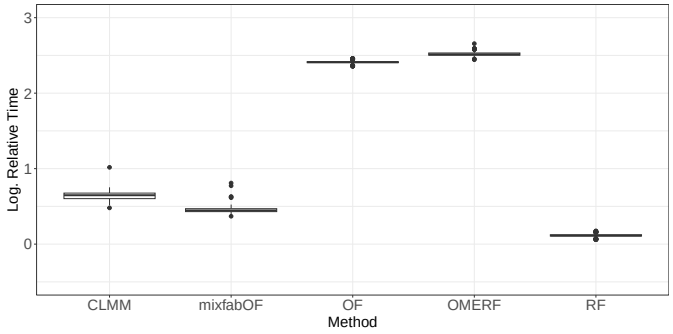

Figure B2

Computational runtime relative to fabOF for random intercept model simulation data from Bergonzoli et al. (2024). Values have been logarithmized using base 10.