Recursive partitioning of linear growth curve models

Marjolein Fokkema[1] and Achim Zeileis[2]
[1]Leiden University
[2]University of Innbruck

Abstract

Growth curve models are popular tools for studying development over time. Often, researchers have a specific interest in uncovering subgroups that show different patterns of growth over time. Recursive partitioning methods (RPMs) allow for detecting such subgroups. In this paper, we focus on the use of generalized linear mixed effects model trees (GLMM trees) for partitioning LGCMs. GLMM trees were originally proposed for subgroup detection in clustered cross-sectional data. Here, we introduce recent adaptations of the algorithm that that are of particular relevance for partitioning longitudinal models, and growth curve models in particulat. Using simulated and existing data, we compare the performance of GLMM trees and other methods for partitioning LGCMs: SEM trees, LongCART and longRPart. We find that GLMM trees provide similar accuracy compared to SEM trees, and better accuracy compared to LongCART and longRPart. In addition, GLMM trees allow for the analysis of trajectories with both discrete and continuous time, are less sensitive to (mis-)specification of the random-effects structure and require substantially lower computation times.

Introduction

In longitudinal studies, (generalized) linear growth curve modeling is often used to model change over time, and inter individual differences in change. For example, researchers may want to model student's reading or math development in an educational study, or symptom reduction over the course of a clinical trail. Inter individual heterogeneity is likely in such studies, where (groups of) participants show higher or lower rates of growth over time. Through the use of mixed-effects or latent-variable models, such heterogeneity can

be captured, and explained by covariates of a-priori known relevance (McNeish & Matta, 2018).

Often, the covariates that predict or affect the trajectories may not be known a-priori, or researchers may have a specific interest in finding subgroups of participants with similar trajectories. Recursive partitioning methods (RPMs), also known as decision-tree methods, are particularly suited in such cases. Several existing RPMs allow for partitioning growth trajectories: For example, GUIDE (Loh, 2002), longRPart (Abdolell, LeBlanc, Stephens, & Harrison, 2002), GEE-based decision trees (Lee, 2005), longitudinal interaction trees (IT; Su, Meneses, McNees, & Johnson, 2011), Structural Equation Model (SEM) trees (Arnold, Voelkle, & Brandmaier, 2021; Brandmaier, von Oertzen, McArdle, & Lindenberger, 2013), mixed-effects longitudinal trees (MELT; Eo & Cho, 2014) and LongCART Kundu and Harezlak (2019) allow for partitioning linear growth curve models (LGCMs). Partitioning based on non-linear growth curve models can be performed with the longRPart2 (Stegmann, Jacobucci, Serang, & Grimm, 2018) and IT-LT (Wei, Liu, Su, Zhao, & Jiang, 2020) methods[1].

In this paper, we focus on generalized linear mixed-effects model trees (GLMM trees; Fokkema, Smits, Zeileis, Hothorn, & Kelderman, 2018), which allow for subgroup detection in a wide range of mixed-effects models. Although originally not specifically developed for partitioning LGCMs, application of GLMM trees for partitioning LGCMs is straightforward. We propose two adaptations to the algorithm, which we expect to be of particular relevance for partitioning LGCMs.

The main difference between GLMM trees and the aforementioned RPMs, is that the latter all fit the full parametric model in each of the nodes. In contrast, GLMM trees uses the observations in each node to estimate the fixed-effects parameters, while using all observations to estimate the random-effects parameters. This may not only reduce computation time; it also reduces the total number of parameters to be estimated, which may improve stability and reduce overfitting. This local/global estimation approach was originally proposed by Hajjem, Bellavance, and Larocque (2011) and Sela and Simonoff (2012) for trees with constant fits in the terminal nodes; Fokkema et al. (2018) generalized it to model-based trees.

The longRPart, longRPart2, IT-LT and original SEM tree algorithms employ an exhaustive split detection procedure: For every possible cut point in the current node, the full parametric model is re-estimated in the two resulting nodes. To choose the optimal

[1]Both IT methods specifically focus at detecting so-called *treatment-subgroup interactions*: subgroups that respond more favorably to one treatment than another.

cutpoint, a likelihood ratio test is computed for every possible cut point. Drawbacks of this exhaustive search are a heavy computational burden, as well as a selection bias towards covariates with a larger number of possible cut points (Shih, 2004; Shih & Tsai, 2004).

LongCART, MELT, GEE-based decision trees and score-based SEM trees fit full parametric models in each of the nodes as well, but do not refit models for cutpoint selection; they employ the predictions or residuals from the fitted model in the current node for selecting the best split. This reduces computational burden. Furthermore, these approaches allow for employing a two-step approach for split detection, which separates variable and cutpoint selection, thus preventing variable selection bias. The GLMM tree algorithm is more similar to the latter group of RPMs, in that it employs a two-step approach to split detection.

The remainder of this paper is structured as follows: In the next section (Estimation) we explain how GLMM trees are estimated, and how estimation can be adjusted to (better) account for dependence structures encountered in LGCMs. Next, in Study I, we perform a simulation study to assess performance of GLMM trees and the proposed adaptations in partitioning LGCMs. In Study II, we use the same simulation design to compare the performance of GLMM trees with that of two other mixed-effects RPMs: SEM trees and LongCART. In Study III, we compare the performance of GLMM trees and longRPart in partitioning children's trajectories of reading, math and science abilities. In the Discussion, we summarize our findings and discuss implications.

Estimation of GLMM trees

In the GLMM tree model (Fokkema et al., 2018), expectation μ_i of outcome vector y_i is modeled through a linear predictor and suitable link function:

$$E[y_i|X_i] = \mu_i, \quad (1)$$

$$g(\mu_i) = X_i\beta_j + Z_i b_i \quad (2)$$

In the current paper, g is the identity function and we assume a continuous response with normally distributed errors. Further, X_i is the $N_i \times (p+1)$ fixed-effects design matrix for subject i ($i = 1, \ldots, N$). We assume time is the predictor variable of interest, thus $p = 1$ and X_i comprises a column of 1s for the intercept and a column for the observed timepoints of subject i. The set of observed timepoints may differ between subjects, and thus the number of rows of X_i may differ between subjects. β_j is a column vector of node-specific fixed-effects parameters, their values depending on terminal node j of which observation i is part. Subscript j is what distinguishes the GLMM tree model from a traditional GLMM.

As in a traditional GLMM, Z_i is the random-effects design matrix for subject i and contains (a subset of) the columns of X_i. Further, b_i is a vector of random effects of subject i. We assume that b follows a (possibly multivariate) normal distribution with mean zero and (co)variance Σ_b. We assume Gaussian errors, with constant variance across subjects.

The parameters of a traditional GLMM can be estimated by, for example, (restricted) maximum likelihood. The GLMM tree model in Eq. 2 is estimated by alternating between estimating the partition (i.e., subgroups or terminal nodes j), and estimating the random- and fixed-effects parameters, as per the following algorithm:

0. Initialize by setting r and all values $\hat{b}_{(r)}$ to 0.
1. Set $r = r + 1$. Estimate the partition using $Z_i\hat{b}_{i(r-1)}$ as an offset.
2. Fit the mixed-effects model $g(\mu_i) = X_i\beta_j + Z_ib_i$ with terminal node $j(r)$ from Step 1. Extract posterior predictions $\hat{b}_{(r)}$ from the estimated model.
3. Repeat Steps 1 and 2 until convergence.

The algorithm initializes by setting b to 0, since the random effects are initially unknown. In every iteration, the partition is (re-)estimated using a GLM-based recursive partition in Step 1, while the fixed- and random-effects parameters are (re-)estimated in Step 2. Note that the random effects are not partitioned, but estimated *globally*, using all observations in the dataset. The fixed effects are estimated *locally*, using the observations in the current node. Convergence of the algorithm is monitored by computing the log-likelihood criterion of the mixed-effects model fitted in Step 3. Typically, this converges if the tree does not change from one iteration to the next.

Initialization

Studies to date indicate that initializing estimation with zero random effects yields an accurate final model (Fokkema et al., 2018; Fu & Simonoff, 2015; Hajjem et al., 2011; Hajjem, Bellavance, & Larocque, 2014; Hajjem, Larocque, & Bellavance, 2017; Sela & Simonoff, 2012). Sela and Simonoff (2012) assessed the impact of different initial values and found only minor impact on the final model, which became smaller with increasing sample size. In Fokkema et al. (2018), we found initializing estimation of GLMM trees with zero random effects performed well in cross-sectional datasets. With longitudinal data (e.g., LGCMs), observations from the same unit tend to be more strongly correlated than in cross-sectional data; initialization with zero random effects may provide an unrealistic starting point which may be difficult to overcome in subsequent iterations. Initializing GLMM-tree estimation

by estimating the random effects, assuming no grouping structure, may provide a better starting point. We thus hypothesize that for partitioning LGCMs, initializing GLMM-tree estimation with the random effects can improve subgroup recovery and predictive accuracy.

Partitioning

The partition (or tree) in Step 1 is estimated using model-based recursive partitioning (MOB; Zeileis, Hothorn, & Hornik, 2008). The MOB algorithm cycles iteratively through the following steps:

a) Fit the parametric model to all observations in the dataset.

b) Statistically test for parameter instability with respect to each of the partitioning variables.

c) If there is some overall parameter instability, split the dataset with respect to the partitioning variable associated with the highest instability.

d) Repeat Steps (a) through (c) in each of the resulting subgroups.

Parameter stability is tested in Step (b), using the the *scores* (gradient contributions) from the model fitted in Step (a). Under mild regularity conditions, the scores have an expected value of 0. The parameter stability tests evaluate whether the scores fluctuate randomly around their mean of 0, or if they exhibit systematic deviations when they are ordered by the values of one or more covariates. For continuous covariates (or ordered covariates with a large enough number of unique values), this involves computing the cumulative score process $W_k(t)$ with respect to each potential partitioning variable U_k (Zeileis et al., 2008):

$$W_k(t) = \hat{J}^{-1/2} n^{-1/2} \sum_{i=1}^{[nt]} \hat{\psi}_{\sigma(U_{ik})} \tag{3}$$

where $\hat{J}$ is a suitable estimate of the covariance matrix of the parametric model fitted in the current node, and n gives the number of observations in the current node. Further, $\hat{\psi}_{\sigma(U_{ik})}$ denotes estimated scores (denoted $\hat{\psi}$), with the subscript $\sigma(U_{ik})$ denoting their ordering by the values of partitioning variable U_k. Note that $0 \leq t \leq 1$, thus $nt = 1$ for an observation with a unique minimum on the partitioning variable, and $nt = n$ for an observation with a unique maximum.

From the cumulative score process $W_k(t)$, a range of test statistics can be derived which capture non-random fluctuations. For numerical partitioning variables, a maximum Lagrange multiplier test statistic can be computed, which takes the maximum of the squared Euclidean norm of $W_k(t)$, weighted by it's variance (Zeileis & Hornik, 2007). This statistic is referred to as the *supLM* statistic, and is asymptotically equivalent to the maximum of likelihood-ratio statistics. Approximate asymptotic p-values for the *supLM* statistic can be computed with the method of Hansen (1997). Categorical covariates do not provide an implicit ordering. Scores are therefore binned at each level of the covariate and a test statistic is computed that does not depend on the ordering of the levels (Merkle, Fan, & Zeileis, 2014).

When partitioning LGCMs, covariates will often not be measured at the lowest level, but at the subject level. The covariance matrix employed in Eq. 3 can account for this clustering of observations within subjects (e.g., Zeileis, Köll, & Graham, 2020) by employing a clustered covariance matrix $\hat{J}$ in computation of the score process. @AZ: Thus, only computation of $\hat{J}$ is affected by cluster argument? Sscores are summed before computing $\hat{J}$, but computation of $\sum_{i=1}^{[nt]} \hat{\psi}_{\sigma(U_{ik})}$ is unaffected? This resembles a GEE-type approach, where dependence between observations is accounted for in the variance structure. We thus hypothesize that for partitioning LGCMs, use of clustered covariances yields more accurate subgroup recovery and improves predictive accuracy.

Study I: Performance of LM(M) trees

To test whether initializing GLMM-tree estimation with the random effects and use of clustered covariances in computation of the parameter stability tests yields more accurate subgroup recovery when partitioning LGCMs, we performed a simulation study.

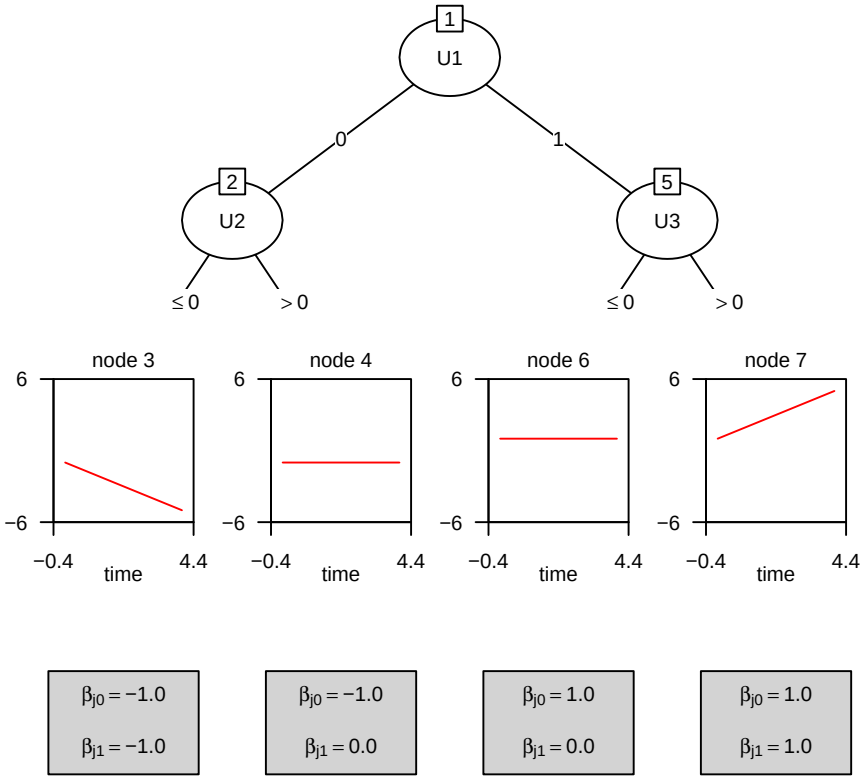
Method

Data-generating design

We simulated datasets according to the subgroup structure depicted in Figure 1. Specifically, four non-overlapping subgroups corresponding to the terminal nodes of the tree in Figure 1 were created. Observations in terminal nodes 3 and 4, and in terminal nodes 6 and 7 have the same fixed intercept (β_{j0}). To facilitate comparison with the SEM tree algorithm, the same timepoints were generated for all subjects (i.e., $t = 0, 1, 2, 3, 4$). The response was computed as:

Figure 1

Design of the subgroup-specific fixed effects in the simulation.



$$y_i = X_i\beta_j + Z_i b_i + \epsilon_i,$$

where β_j corresponds to the fixed effects in terminal node j of which subject i is part. The fixed- and random-effects design matrices X_i and Z_i are identical, each comprising two columns: a vector of 1s for the intercept, and a vector of timepoints. The value for b_i was generated from a multivariate normal distribution with mean zero and a 2×2 diagonal covariance matrix Σ, the diagonal entries determined by the level of the data-generating design described next. Values in ϵ_i were independently generated from $N(\mu = 0, \sigma = \sqrt{5})$.

We varied the following data-generating characteristics:

- Number of subjects: small ($N = 100$) or large ($N = 250$).

- Variance of the random intercept: small ($\sigma^2_{b_0} = 1$) or large ($\sigma^2_{b_0} = 2$).
- Variance of the random slope: small ($\sigma^2_{b_1} = .1$) or large ($\sigma^2_{b_1} = .4$).
- Number of noise partitioning variables: small ($p = 5$) or large ($p = 25$).
- Intercorrelation between partitioning variables: absent ($\rho = 0$) or present ($\rho = .3$).

We employed a full factorial design, yielding $2^5 = 32$ cells.We performed 100 repetitions for each cell of the design. For generating and analysing data, we used R (R Core Team, 2022, version 4.1.2).

Partitioning methods

We fitted two different LM trees to each dataset: One employing default settings (i.e., parameter stability tests performed at level I) and one where the parameter stability tests employed clustered covariance matrices (i.e., parameter stability tests performed at level 2). To fit LM trees, we used package **partykit** (version 1.2-15 Hothorn & Zeileis, 2015).

We fitted eight different LMM trees to each dataset: We estimated an LMM tree using default settings (i.e., parameter stability tests performed using observation-level covariances, estimation initialized with the tree structure); an LMM tree with parameter stability tests performed using cluster-level covariances; an LMM tree estimated by initializing estimation with the random effects; an LMM tree estimated by combining the latter two approaches. Each of the four approaches were applied with a specification comprising random intercepts only, as well as with a specification comprising both random intercepts and slopes, yielding eight different LMM trees. The different specifications allow for assessing the effect of mis-specification of the random-effects structure. To fit LMMs, we used package **lme4** (version 1.1-29 Bates, Mächler, Bolker, & Walker, 2015) and to fit LMM trees we used package **glmertree** (version 0.2-0 Fokkema et al., 2018).

Evaluation of performance

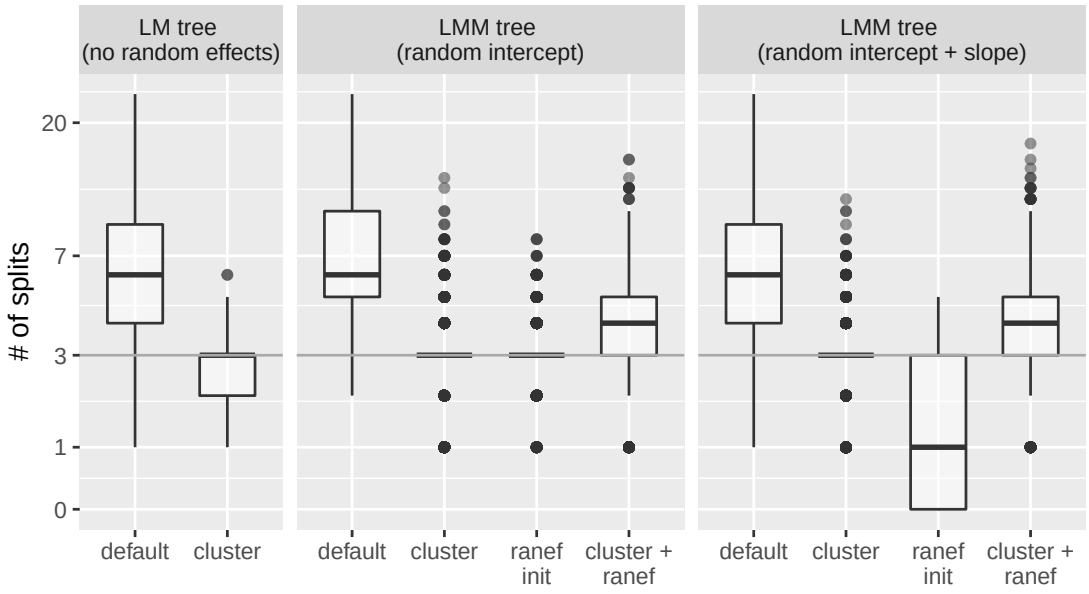
We evaluated tree accuracy by counting the number of splits in each tree. Trees with > 3 splits are indicative of Type-I error, while trees with > 3 splits are indicative of Type-II errors (i.e., too low power for tests of the true partitioning variables). We also assessed which variable was selected for the first split in every tree.

Results

Figure 2 depicts the number of splits implemented by each partitioning approach. The default fitting approach overfits, and implemented > 3 splits in most datasets. This

Figure 2

Tree size distributions for LM(M) trees.



Note. The y-axis is on the log scale and represents tree size. Grey lines indicate true number of splits (3 nodes). cluster = cluster-level covariances employed in parameter stability tests; ranef init = estimation initialized with the random effects. Distances on y-axis are on the log scale.

overfitting is successfully mitigated by the use of cluster-level covariances in the parameter stability tests (mean number of splits 2.87), which appears most effective for LMM trees (mean number of splits 3.02 for trees estimating only random intercepts; 2.96 for trees estimating random intercepts and slopes).

Initializing estimation with the random effects also effectively reduces tree size, but too strongly when random slopes are estimated (mean number of splits without random slopes 3.15; with random slopes 1.15). Heterogeneity related to partitioning variables measured at level 2 can either be captured and explained by the tree, or captured by the random effects. By initializing estimation with the random effects effects, this heterogeneity can no longer be captured by the tree; this likely worsens with more complex random-effects specification. Combining the use of random-effects initialization and cluster-level covariances does not appear beneficial, yielding a similar and too high number of splits, irrespective of the random-effects specification (mean number of splits 4.1).

Distributions of the number of splits, separated according to the levels of σ_{b_0}, N,

p and σ_{b_1} are depicted in Figure A1 (Appendix A). These plots show a pattern similar to Figure 2, with use of cluster-level covariances providing the most accurate split recovery. In addition, the plots show that increasing values of σ_{b_0}, N, p and σ_{b_1} yield higher numbers of splits, while the effect of ρ is very minor. I moved plots to Appendix because of their size and because they are not of main interest. Not sure if following results should be discussed here or in the Appendix: The main deviation from this pattern of main effects is that increased variance of the random effects (increases σ_{b_0} and/or σ_{b_1}) yields lower numbers of splits if random slopes are estimated and estimation is initialized with the random effects.

Table 1 shows the variables selected for the first split. Only rarely is the wrong variable selected for the first split, and this occurs only with random-effects initialization and observation-level covariances in the parameter stability tests are employed. In the large majority of those cases, U_2 and U_3 were selected for the first split.

Table 1

Variables selected for the first split by each LM(M) tree approach.

Random effects	Fitting approach	U_1	U_2	U_3	U_4 - U_{25}	no split
$\sigma_{b_0} = \sigma_{b_1} = 0$		1.000	0.000	0.000	0.000	0.000
$\sigma_{b_0} = \sigma_{b_1} = 0$	clustered cov.	1.000	0.000	0.000	0.000	0.000
$\sigma_{b_0} > 0$, $\sigma_{b_1} = 0$		1.000	0.000	0.000	0.000	0.000
$\sigma_{b_0} > 0$, $\sigma_{b_1} = 0$	clustered cov.	1.000	0.000	0.000	0.000	0.000
$\sigma_{b_0} > 0$, $\sigma_{b_1} = 0$	ran.eff. init.	0.996	0.002	0.002	0.001	0.000
$\sigma_{b_0} > 0$, $\sigma_{b_1} = 0$	both	1.000	0.000	0.000	0.000	0.000
$\sigma_{b_0} > 0$, $\sigma_{b_1} > 0$		1.000	0.000	0.000	0.000	0.000
$\sigma_{b_0} > 0$, $\sigma_{b_1} > 0$	clustered cov.	1.000	0.000	0.000	0.000	0.000
$\sigma_{b_0} > 0$, $\sigma_{b_1} > 0$	ran.eff. init.	0.499	0.007	0.006	0.003	0.485
$\sigma_{b_0} > 0$, $\sigma_{b_1} > 0$	both	1.000	0.000	0.000	0.000	0.000

Note. U_1 is the true first splitting variable and is binary; all other partitioning variables are continuous, with U_2 and U_3 being true splitting variables (nodes 2 and 3). σ_{b_0} and σ_{b_1} are the standard deviations of the random intercept and slope, respectively.

Study II: Comparison of LM(M) trees with SEM trees and LongCART

In this second study, we compare performance of LM(M) trees with that of SEM trees and LongCART. This allows us to evaluate the possible (dis)advantages of global versus local estimation of random-effects parameters, as well as the performance of differ-

ent splitting criteria. The same data-generating design as in Study I was employed. We compared performance of LM(M) trees employing clustered covariances only, which showed good performance in Study I.

Method

We fitted SEM trees using two different splitting criteria: First, we used the default "naive" splitting approach (Brandmaier et al., 2013), in which likelihood ratio tests (LRTs) are used as the splitting criterion. The likelihood of the SEM fitted to the observations in the current node is compared against the likelihood of a range of two-group SEMs, in which the two groups are defined by each possible candidate split. An LRT can thus be computed for each candidate split, which quantifies the improvement in fit that may be obtained by implementing the split. In each step, the candidate yielding the highest LRT is selected for splitting, and splitting is continued as long as a candidate split yields a p-value of the LRT above a pre-specified α level (0.05, by default).

Second, we fitted score-based SEM trees, as proposed by Arnold et al. (2021). This approach uses the MOB algorithm described in the Introduction, where the parametric model fitted in step (a) is a SEM. While for GLMM trees, parameter stability tests are computed for the fixed-effects parameters only, score-based SEM trees compute parameter stability tests based on both fixed- and random-effects parameters.

To specify the node-specific models for SEM trees, we employed an LGCM specification with the response at each timepoint regressed on a latent intercept and slope. Intercept loadings were fixed to 1; slope loadings were fixed to the value of t at each timepoint. With both LRT- and score-based splitting criteria, we employed three different model specifications: a) variances of latent slope and intercept fixed to 0; b) variance of latent intercept freely estimated, variance of latent slope fixed to 0; c) variances of latent intercept and slope freely estimated. Errors were assumed uncorrelated between timepoints, error variances freely were estimated for each timepoint. To fit SEMs, we used package **lavaan** (version 0.6-11 Rosseel, 2012) and to fit SEM trees, we used package **semtree** (version 0.9.17 Brandmaier et al., 2013).

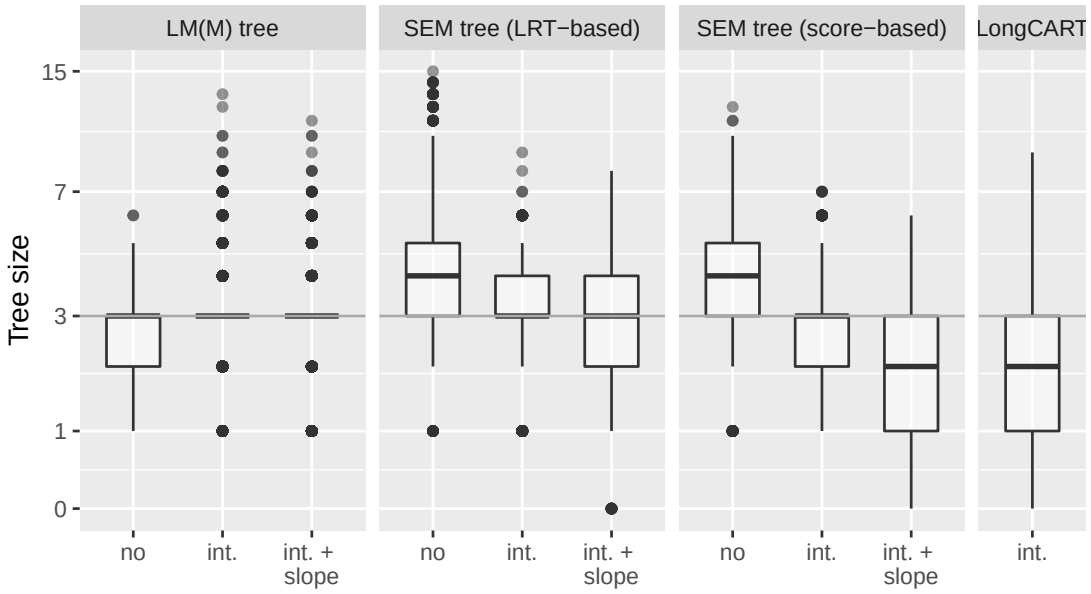
We fitted a LongCART tree to each dataset, employing default settings. The LongCART function estimates node-specific models comprising a random intercept term; this default cannot be overridden. A fixed-effects model was specified with the response regressed on time and a subject-specific random intercept. We used package **semtree** to fit SEM trees (Brandmaier et al., 2013, version 0.9.17). To fit LMMs, we used package **nlme** (version 3.1-153 Pinheiro & Bates, 2000) and to fit LongCART trees we used package **LongCART**

(version 3.1 Kundu, 2021).

Results

Figure 3

Tree size distributions for LM(M) trees with clustered covariances, SEM trees and LongCART.



Note. Values on x-axis indicate whether random intercepts and/or slopes were estimated: no = random intercepts and slopes fixed to 0; int. = variance of random intercept freely estimated, variance of random slope fixed to 0; int. + slope = variance of random intercept and slope (and correlation between the two) freely estimated. Distances on y-axis are on the log scale. Grey lines indicate true number of splits (3 splits).

Figure 3 depicts tree size distributions for the different algorithms. LRT-based SEM trees appear somewhat overpowered with a mean number of splits of 3.61 (SD =1.71), while score-based SEM trees comprised the correct number of splits on average (M =3.09; SD =1.61). SEM trees appear more systematically affected by mis-specification of the random-effects structure than LMM trees and under-specification yields a higher number of splits. LongCART trees appear underpowered, with a mean number of splits of 2.01 (SD =1.45). This could be the result of under-specification of the random-effects structure: LongCART trees incorporate a random-intercept term only and this default cannot be

overridden.

Figure A2 (Appendix A) presents tree size distributions, separated according to the levels of the data-generating parameters. Strongest effects were observed for N, followed by σ_{b_0}, p_{noise}, ρ and σ_{b_1}. I moved plots to Appendix because of their size and because they are not of main interest. Not sure if following results should even be discussed here or in the Appendix: Results for N were as expected for all methods: with increasing sample size, more splits are implemented. Both LRT- and score-based SEM trees seem only affected by levels of σ_{b_0}, p_{noise} and ρ under misspecification of the random effects. Especially when both random intercept and slope variances are fixed to 0, higher levels of σ_{b_0} and p_{noise} yield more splits with both SEM tree approached. LRT-based SEM trees seem unaffected by levels of ρ, while score-based SEM trees seemed to implement a larger number of splits with increasing values of ρ. This pattern seemed reversed for increased magnitude of σ_{b_1}, which yields a lower number of splits for both SEM tree approaches, but only when the random effects were correctly specified. LongCART implemented more splits with higher levels of p_{noise} and ρ, but was unaffected by levels of σ_{b_0} and σ_{b_1}.

Table 2

Variable selected for the first split by each of the partitioning methods.

Algorithm	Random effects	U_1	U_2	U_3	U_4 - U_{25}	no split
LM(M) tree	$\sigma_{b_0} = \sigma_{b_1} = 0$	1.000	0.000	0.000	0.000	0.000
(clustered	$\sigma_{b_0} > 0$, $\sigma_{b_1} = 0$	1.000	0.000	0.000	0.000	0.000
covariances)	$\sigma_{b_0} > 0$, $\sigma_{b_1} > 0$	1.000	0.000	0.000	0.000	0.000
SEM tree	$\sigma_{b_0} = \sigma_{b_1} = 0$	1.000	0.000	0.000	0.000	0.000
(LRT-based)	$\sigma_{b_0} > 0$, $\sigma_{b_1} = 0$	0.998	0.000	0.002	0.000	0.000
	$\sigma_{b_0} > 0$, $\sigma_{b_1} > 0$	0.992	0.001	0.001	0.002	0.004
SEM tree	$\sigma_{b_0} = \sigma_{b_1} = 0$	0.761	0.176	0.052	0.012	0.000
(score-based)	$\sigma_{b_0} > 0$, $\sigma_{b_1} = 0$	0.879	0.077	0.044	0.000	0.000
	$\sigma_{b_0} > 0$, $\sigma_{b_1} > 0$	0.999	0.000	0.000	0.000	0.001
LongCART	$\sigma_{b_0} > 0$, $\sigma_{b_1} = 0$		0.419	0.297	0.092	0.192

Note. U_1 is the true first splitting variable and is binary; all other partitioning variables are continuous, with U_2 and U_3 being true splitting variables (nodes 2 and 3). σ_{b_0} and σ_{b_1} are the standard deviations of the random intercept and slope, respectively.

Table 2 presents variable selection frequencies for the first split in the fitted trees. For SEM trees, the LRT criterion yields almost perfect accuracy for the first split. The

score-based approach selected the wrong splitting variable for the first split in about 12% of datasets, but only when the random-effects structure was mis-specified (i.e., σ_{b_0} and/or σ_{b_1} fixed to 0). When random effects were correctly specified, score-based SEM trees also provided perfect recovery of the first split. Closer inspection of parameter stability tests suggested that the tests are (much) more sensitive to instability in the fixed slope than in the fixed intercept.

LongCART trees exhibit strikingly low accuracy for recovering the first split, selecting the wrong variable in all datase. LongCART showed a more dramatic tendency than score-based SEM trees to use U_2 and U_3 for implementing the first split. Of note, in the LongCART trees where no split was implemented (19% of datasets), 99% U_1 was the strongest splitting candidate, but the parameter stability test simply did not exceed the criterion of significance in those datasets. This suggests that either the parameter stability tests proposed by Kundu and Harezlak (2019) are less sensitive to instabilities with respect to the fixed intercept, and/or with respect to categorical covariates.

Figure 4

Computation time distributions for the different partitioning methods.

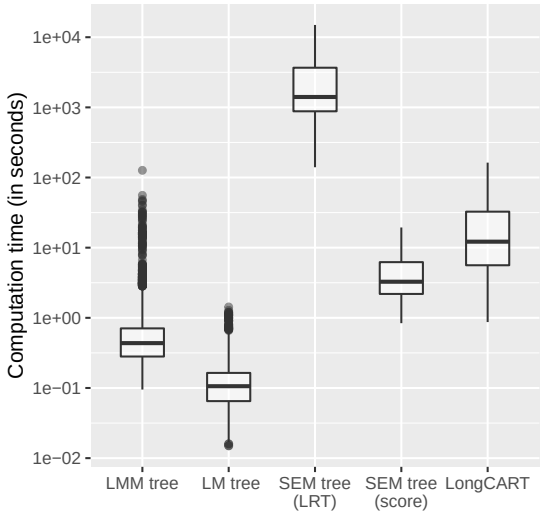


Figure 4 presents computation time distributions for the partitioning algorithms. A clear computational advantage is observed for LM trees, with an average computation time of 0.141 seconds. Including random-effects estimation increased computation time: for LMM trees, average computation time was 0.65 seconds. LongCART and SEM trees yielded much longer computation times, with LongCART requiring 23.36 seconds, score-based SEM

trees 4.4 seconds, and LRT-based SEM trees 2685.37 seconds computation time, on average.

Study III: Application

Method

Dataset

We analyzed longitudinal assessments of children's reading, math and science abilities from the Early Childhood Longitudinal Study-Kindergarten class of 1998-1999 (ECLS-K; "Early Childhood Longitudinal Program: Kindergarten class of 1998-1999 (ECLS-K)", 2010). Data were collected from kindergarten 1998 through eighth grade 2007. In the current analyses, we focus on assessments during kindergarten, 1st, 3rd, 5th and 8th grade. In the ECLS-K study, data from 21,304 children from 1,018 schools across the USA were collected.

Response variables are reading, math and science ability. These were assessed using multi-item cognitive tests, from which latent ability estimates were computed with a mean of zero and variance of one. Reading and math abilities were assessed in all five rounds of data collection, while science knowledge was assessed in the 3rd, 5th and 8th grade only. We analyzed data from children who completed all assessments ($N = 6,277$ for reading; $N = 6,512$ for math; $N = 6,625$ for science).

Our analyses focus on predicting ability trajectories, based on baseline characteristics. Time was measured as the number of months since the baseline assessment. In order to obtain approximately linear trajectories, used $\sqrt{\text{months}}$ as the timing metric for reading and math trajectories, and $\text{months}^{\frac{2}{3}}$ for science trajectories.

We used the following time-invariant covariates as potential partitioning variables:

- Gender (51.1% male)
- Age in months at baseline (range 53 to 96; M = 6.14 years at baseline).
- Race (8 categories)
- First time in kindergarten (yes/no)
- Socio-economic status (range -5 to 3)
- Fine motor skills (e.g., drawing figures; range 0 to 9)
- Gross motor skills (e.g., ability to hop, skip and jump; range 0 to 8)
- Interpersonal skills (range 1 to 4)

- Self-control (range 1 to 4)
- Internalizing problem behavior (range 1 to 4)
- Externalizing problem behavior (range 1 to 4)

Partitioning methods

We fitted four LM(M) trees, employing the same approaches applied in Studies I and II, except for the default approaches, and the random-effects initialization approach when both random intercepts and slopes were estimated; approaches which showed far from optimal performance in Study I.

For comparison, we fitted longRPart (Abdolell et al., 2002) trees. We opted for longRPart, because neither SEM tree, LongCART, longRPart or longRPart2 implementations provide functionality to obtain predictions for new observations. We wrote custom functionality to allow for prediction with longRPart trees, but could not produce this for the other methods. We expect, however, that performance of longRPart will be representative of the other methods, in the sense that all fit the full mixed-effects model locally. Furthermore, SEM tree could not be employed because assessments at each wave did not take place on identical timepoints, which is a requirement for SEM-based growth curve modeling.

Using package **longRPart** (version 1.0 Stewart & Abdolell, 2012), we fitted two longRPart trees with different random effects specifications: One tree comprising a random intercept only, and one tree comprising both a random intercept and slope, with respect to student. In order to implement a split, longRPart uses a criterion based on the minimum decrease in deviance, which is set to 0.01, by default. This yielded a very large number of splits in the current analyses and poor predictive accuracy. We therefore restricted maximum tree depth to 5, yielding a maximum of $2^5 = 32$ terminal nodes, or 31 splits. Significance of the LRT, or a cross-validation approach to determine whether to implement a split or not may yield better results, but neither approach is implemented in longRPart (nor longRPart2).

Evaluation of performance

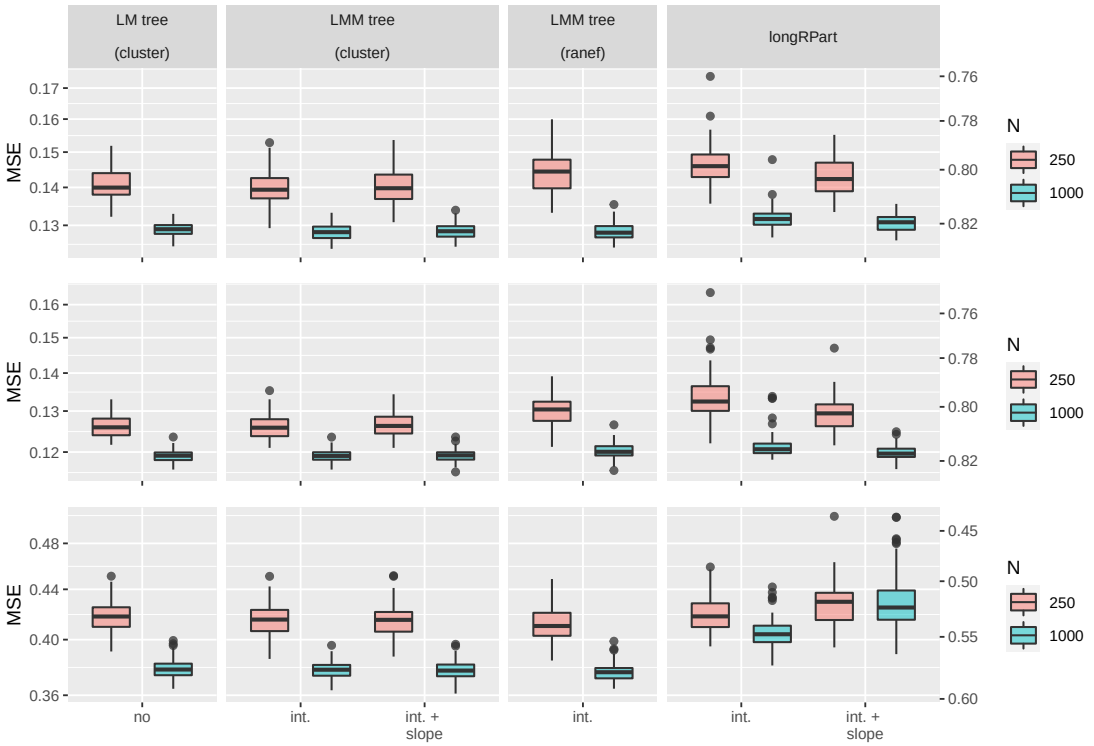
The ECLS-K datasets have exceptionally large sample sizes, which may not be representative of real-world applications. We therefore employed random sampling to obtain training samples of $N = 250$ and $N = 1,000$ children, and evaluated predictive accuracy using the responses of children not included in the training sample. For each combination of response variable and sample size, we performed 100 repetitions. Because we used disjunct

samples of children for training and testing, only the fixed-effects parts of each fitted model were used for prediction; random effects were fixed to 0. Accuracy was assessed by computing the test MSE, by taking the mean squared difference between predicted and observed ability estimates of test observations. We measured tree size by counting the number of splits in each tree, and measured computation time in seconds.

Results

Figure 5

Mean squared errors for trees fitted to math (top), reading (middle) and science (bottom) ability trajectories.



Note. The y-axis on the right gives an estimate of the proportion of variance explained, computed as $1 - \frac{MSE}{var(y)}$. Distances on y-axis are on the log scale.

Figure 5 and Table 3 present MSE distributions for each of the partitioning approaches, separated by sample size[2]. Differences in performance between the partitioning

[2]LongRPart trees comprising random intercepts and slopes ran into estimation errors in some datasets: In the Math ability data, 2 out of 200 datasets did not yield a tree. All tree estimations converged successfully

Table 3

Cross-validated mean squared errors for each of the response variables.

	Math		Reading		Science	
	N = 250	N = 1,000	N = 250	N = 1,000	N = 250	N = 1,000
LM tree (c)	0.1407 (0.004)	0.1290 (0.002)	0.1262 (0.003)	0.1191 (0.001)	0.4178 (0.012)	0.3787 (0.007)
LMM tree (i,c)	0.1398 (0.004)	0.1283 (0.002)	0.1261 (0.003)	0.1191 (0.001)	0.4151 (0.013)	0.3777 (0.006)
LMM tree (i,s,c)	0.1403 (0.005)	0.1287 (0.002)	0.1264 (0.003)	0.1192 (0.001)	0.4143 (0.012)	0.3777 (0.006)
LMM tree (i,r)	0.1441 (0.006)	0.1285 (0.002)	0.1302 (0.004)	0.1202 (0.002)	0.4124 (0.013)	0.3762 (0.006)
longRPart (i)	0.1463 (0.006)	0.1318 (0.003)	0.1335 (0.006)	0.1213 (0.003)	0.4197 (0.015)	0.4050 (0.011)
longRPart (i,s)	0.1430 (0.005)	0.1306 (0.002)	0.1291 (0.004)	0.1199 (0.002)	0.4289 (0.020)	0.4296 (0.022)

Note. Values represent means over 100 repetitions, parenthesized values represent SDs and can be interpreted as standard errors. c = cluster-level covariances; r = estimation initialized with random effects; i = random-intercept variance freely estimated; s = random-slope variance freely estimated.

approaches are of smaller magnitude than differences due to sample size. All methods show higher accuracy with larger sample sizes (with one exception: longRPart trees comprising random intercepts and slopes fitted on science ability data).

LM(M) trees tend to show better predictive accuracy than longRPart trees. Pairwise t-tests indicated significantly better performance for each of the LM(M) tree approaches employing clustered covariances, compared to both longRPart approaches (Bonferroni corrected p-values all $< .01$). However, it should be noted that differences in performance are small; for example, the standard errors presented in Table 3 indicate differences > 1 *SE* only for Reading with $N = 250$, and Science (both $N = 250$ and $N = 1,000$)[3].

Table 3 indicates that LMM trees comprising a random intercept term and employing cluster-level covariances ("LMM tree (i,c)") performed best for predicting Math and Reading abilities. For predicting science abilities, LMM trees initializing estimation with the random effects and estimating only a random intercept performed best, while this was the worst-performing approach for math and reading. For longRPart, a somewhat similar shift in performance occurs: While trees comprising both a random intercept and slope performed best for predicting Math and Reading, trees with a random intercept only performed best for predicting Science. This shift may be due to the lower number of Science ability assessments,

in the Reading ability data. In the Science ability data, 40 out of 200 datasets did not yield a tree. Results are presented for iterations did yield a tree.

[3]The significance of pairwise t-tests may seem in contrast with the but small differences in terms of standard errors presented in Table 3, but the standard errors in the pairwise t-tests are sensitive to the number of replications, while the standard errors in Table 3 are not, and thus provide a more useful measure of the strength of the effect.

compared to Math and Reading.

Interestingly, the performance of LM(M) trees employing cluster-level covariances seems hardly affected by the specification of the random-effects: Figure 5 shows very similar performance for LM(M) trees with(out) random intercepts or slopes. Pairwise t-tests did not indicate significant differences between the three approaches for any of the response variables.

Figure 6 present tree size distributions for each of the partitioning approaches. Interestingly, whereas LM(M) trees implement a larger number of splits with increasing sample size, this pattern appears reversed for longRPart trees. For math and reading, the best-performing LM(M) tree approach in terms of predictive accuracy yields a smaller number of splits than the best-performing longRPart approach when sample size is small ($N = 250$). This pattern reverses with large sample size ($N = 1,000$), then the best-performing LM(M)

Figure 6

Tree sizes for trees fitted to math (top), reading (middle) and science (bottom) ability trajectories.

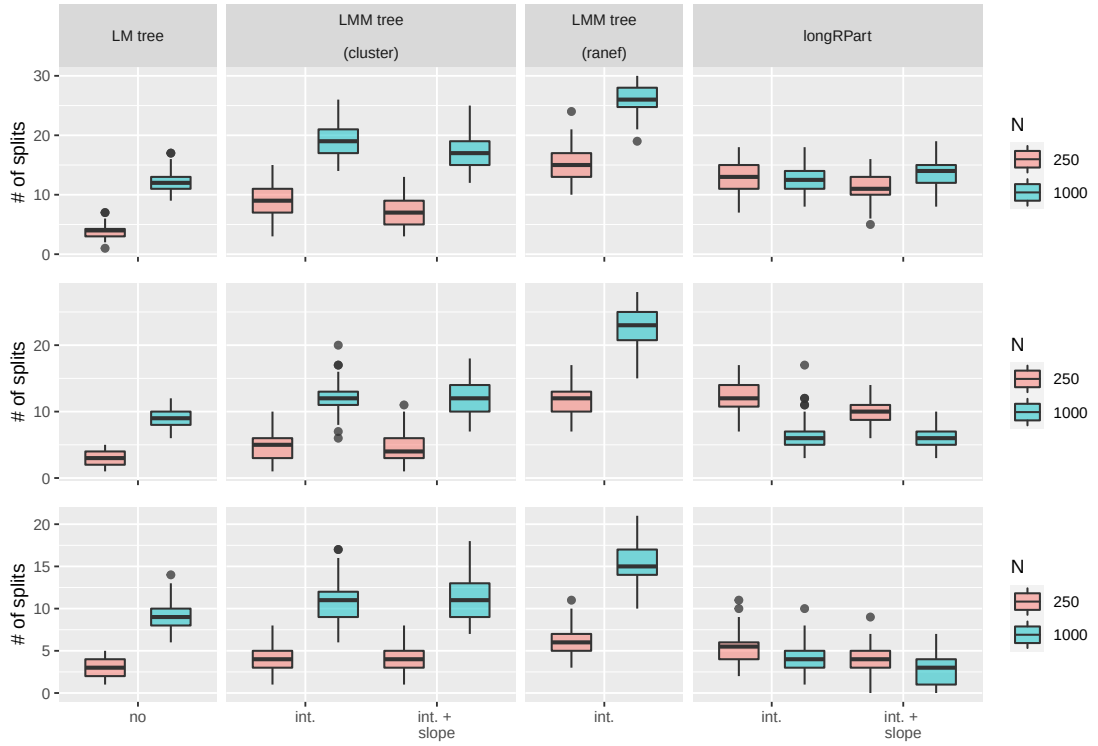
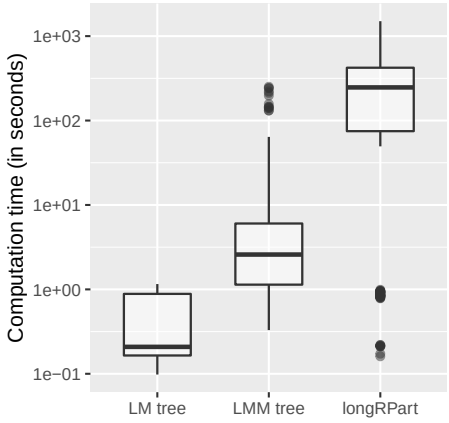


Figure 7

Computation times for partitioning math, reading and science ability trajectories.



tree approach yields a larger number of splits than the best-performing longRPart approach. For the math data, the best-performing longRPart approach yields a smaller number of splits than the best-performing LM(M) tree approach.

Given the very minor differences in performance for LM(M) tree employing clustered covariances with different specifications of the random effects (Figure 5), Figure 6 suggests that LM trees with clustered covariances may provide very good complexity-accuracy trade-off.

Figure 7 depicts the computation time distributions of the different partitioning algorithms. A clear computational advantage is observed for LM trees, with an average computation time of 0.47 seconds. Including estimation of random effects increases computation time, which was 5.62 seconds on average for LMM trees. LongRPart trees required longest computation times, with an average of 364.06 seconds.

Discussion

In Study I, we found the default estimation approach for GLMM trees proposed in Fokkema et al. (2018) yields high Type-I error rates when partitioning LGCMs. As hypothesized, performing parameter-stability tests using clustered covariances strongly improved performance. Also as hypothesized, we found initializing estimation with the random effects to be beneficial, but only when the random-effects specification comprised a random intercept only. Combining cluster-level covariances and random-effects initialization was not beneficial. The performance of of LM(M) trees using clustered covariances appeared

largely unaffected by (mis-)specification of the random effects.

In Study II, we found good performance of SEM trees in partitioning LGCMs, but a stronger sensitivity to mis-specification of the random effects, where underspecification yielded higher Type-I errors. In accordance with results of Arnold et al. (2021), we found score-based SEM trees to have somewhat lower power than LRT-based SEM trees. LongCART tended to be underpowered, which could be due to the trees not incorporating random slopes, or the parameter stability test for categorical covariates proposed by Kundu and Harezlak (2019) may be underpowered.

In Study III, we found higher predictive accuracy for LM(M) trees than for longRPart, but differences in performance were small. Similar to Studies I and II, we found LM(M) trees to be comparably insensitive to (mis-)specification of the random effects, which may provide an important practical advantage.

All three studies showed substantial computational advantage of GLM(M) trees. This is likely due to the estimation approach employed by GLMM trees, where fixed-effects parameters are estimated locally within a node while random-effects parameters are estimated globally, using all observations. In contrast, SEM trees, LongCART and longRPart fit the full mixed-effects model in each node, which may substantially increase computational burden.

Possible points for further discussion

- Lower performance of longRPart possibly due to lack of robust criterion for determining statistical significance of a split. Cross-validation procedures might mitigate this problem, but feasibility of such approaches currently impeded by lack of functionality for generating predictions and high computational burden.

- MELT (Eo & Cho, 2014) employs random-effects predictions as a partitioning criterion, which is effective, as our results on random-effects initialization illustrate that random effects can capture effects of substantive interest that we want to use for explanation.

- Future work:

  - GLMM trees can be extended to allow for partitioning based on all mixed-effects model parameters simultaneously, by using the score-based tests for mixed-effects models proposed by Wang and Merkle (2018).

- GLMM trees can be extended to allow for partitioning non-linear growth curve models / GAMs.
- Performance of GLMM trees on non-gaussian responses (binomial or count responses) should be evaluated.

References

Abdolell, M., LeBlanc, M., Stephens, D., & Harrison, R. (2002). Binary partitioning for continuous longitudinal data: categorizing a prognostic variable. *Statistics in Medicine*, *21*(22), 3395–3409.

Arnold, M., Voelkle, M. C., & Brandmaier, A. M. (2021). Score-guided structural equation model trees. *Frontiers in Psychology*, *11*, 3913.

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

Brandmaier, A. M., von Oertzen, T., McArdle, J. J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, *18*(1), 71.

Early childhood longitudinal program: Kindergarten class of 1998-1999 (ecls-k) [Computer software manual]. (2010). Retrieved from `https://nces.ed.gov/ecls/kindergarten.asp`

Eo, S.-H., & Cho, H. (2014). Tree-structured mixed-effects regression modeling for longitudinal data. *Journal of Computational and Graphical Statistics*, *23*(3), 740–760.

Fokkema, M., Smits, N., Zeileis, A., Hothorn, T., & Kelderman, H. (2018). Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. *Behavior research methods*, *50*(5), 2016–2034. doi: doi:10.3758/s13428-017-0971-x

Fu, W., & Simonoff, J. S. (2015). Unbiased regression trees for longitudinal and clustered data. *Computational Statistics & Data Analysis*, *88*, 53–74.

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. (2014). Mixed-effects random forest for clustered data. *Journal of Statistical Computation and Simulation*, *84*(6), 1313–1328.

Hajjem, A., Larocque, D., & Bellavance, F. (2017). Generalized mixed effects regression trees. *Statistics & Probability Letters*, *126*, 114–118.

Hansen, B. E. (1997). Approximate asymptotic p values for structural-change tests. *Journal of Business & Economic Statistics*, *15*(1), 60–67.

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

Kundu, M. G. (2021). Longcart: Recursive partitioning for longitudinal data and right censored data using baseline covariates [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=LongCART` (R package)

Kundu, M. G., & Harezlak, J. (2019). Regression trees for longitudinal data with baseline covariates. *Biostatistics & epidemiology*, *3*(1), 1–22.

Lee, S. K. (2005). On generalized multivariate decision tree by using gee. *Computational Statistics & Data Analysis*, *49*(4), 1105–1119. doi: doi:10.1016/j.csda.2004.07.003

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

McNeish, D., & Matta, T. (2018). Differentiating between mixed-effects and latent-curve approaches to growth modeling. *Behavior research methods*, *50*(4), 1398–1414.

Merkle, E. C., Fan, J., & Zeileis, A. (2014). Testing for measurement invariance with respect to an ordinal variable. *Psychometrika*, *79*(4), 569–584.

Pinheiro, J. C., & Bates, D. M. (2000). *Mixed-effects models in s and s-plus*. New York: Springer. doi: doi:10.1007/b98882

R Core Team. (2022). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `https://www.R-project.org/`

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36. doi: doi:10.18637/jss.v048.i02

Sela, R. J., & Simonoff, J. S. (2012). Re-em trees: a data mining approach for longitudinal and clustered data. *Machine learning*, *86*(2), 169–207.

Shih, Y.-S. (2004). A note on split selection bias in classification trees. *Computational statistics & data analysis*, *45*(3), 457–466. doi: doi:10.1016/S0167-9473(03)00064-1

Shih, Y.-S., & Tsai, H.-W. (2004). Variable selection bias in regression trees with constant fits. *Computational statistics & data analysis*, *45*(3), 595–607. doi: doi:10.1016/S0167-9473(03)00036-7

Stegmann, G., Jacobucci, R., Serang, S., & Grimm, K. J. (2018). Recursive partitioning with nonlinear models of change. *Multivariate behavioral research*, 1–12.

Stewart, S., & Abdolell, M. (2012). longrpart: Recursive partitioning of longitudinal data using mixed-effects models [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=longRPart` (R package)

Su, X., Meneses, K., McNees, P., & Johnson, W. O. (2011). Interaction trees: exploring the differential effects of an intervention programme for breast cancer survivors. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *60*(3), 457–474. doi: doi:10.1111/j.1467-9876.2010.00754.x

Wang, T., & Merkle, E. C. (2018). merderiv: Derivative computations for linear mixed effects models with application to robust standard errors. *Journal of Statistical Software, Code Snippets*, *87*(1), 1–16. doi: doi:10.18637/jss.v087.c01

Wei, Y., Liu, L., Su, X., Zhao, L., & Jiang, H. (2020). Precision medicine: Subgroup identification in longitudinal trajectories. *Statistical Methods in Medical Research*. doi: doi:10.1177/0962280220904114

Zeileis, A., & Hornik, K. (2007). Generalized m-fluctuation tests for parameter instability. *Statistica*

Neerlandica, *61*(4), 488–508.

Zeileis, A., Hothorn, T., & Hornik, K. (2008). Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, *17*(2), 492–514. doi: doi:10.1198/106186008X319331

Zeileis, A., Köll, S., & Graham, N. (2020). Various versatile variances: An object-oriented implementation of clustered covariances in r. *Journal of Statistical Software*, *95*, 1–36.

Appendix

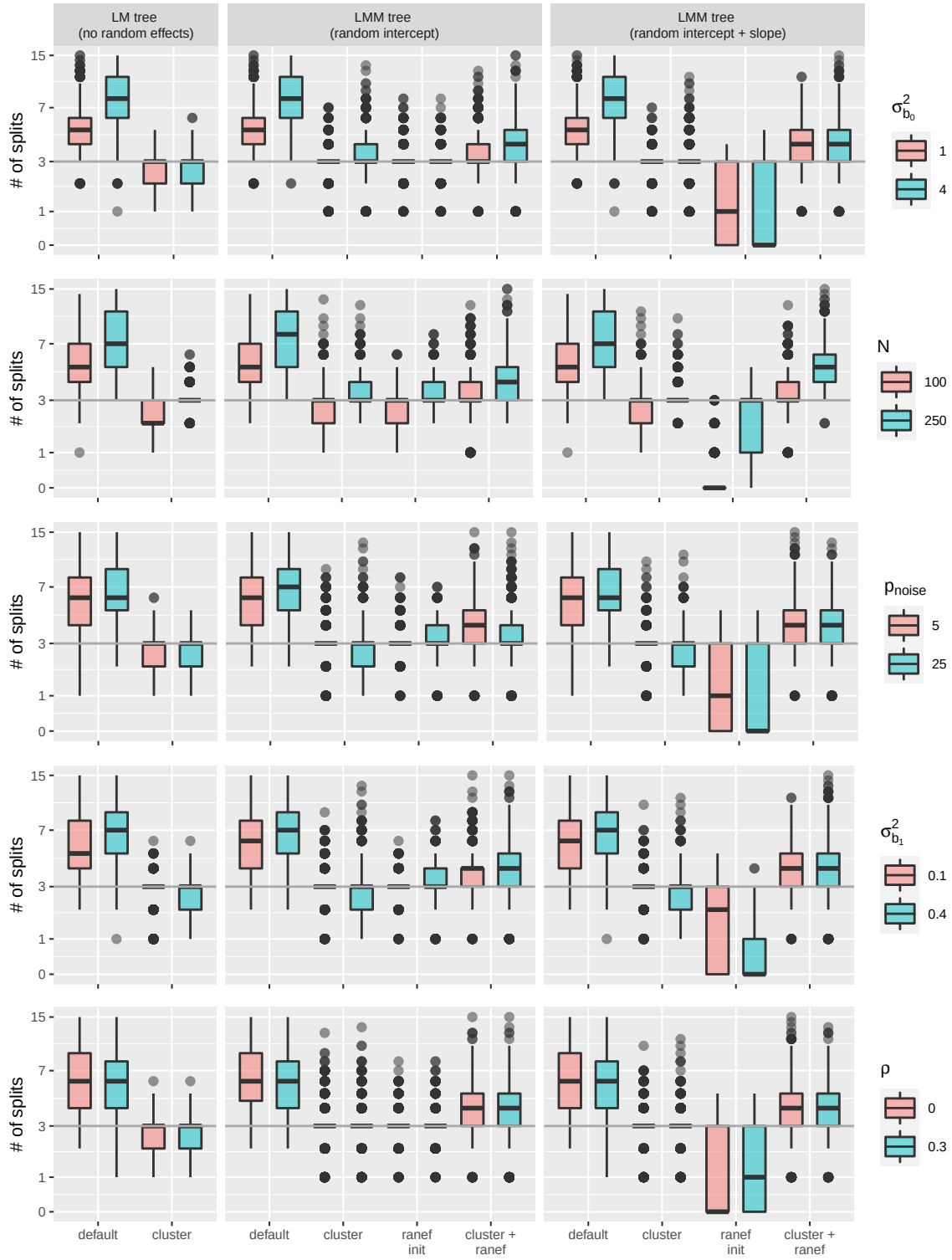
Effects of data-generating parameters on tree size

For LM(M) trees (Figure A1), use of cluster-level covariances provided the most robust improvement in split recovery. For the data-generating parameters, the effects on tree size were strongest for σ_{b_0}, followed by N, p, σ_{b_1} and ρ. Increasing values of σ_{b_0}, N, p and σ_{b_1} tend to yield higher numbers of splits, and the effect of ρ is minimal. The main exception to these main effects is that increased values of σ_{b_0} and σ_{b_1} yield lower numbers of splits when random slopes are estimated, and estimation is initialized with the random effects. In these cases, most variance may be captured by the random effects already in the first iteration of the algorithm, and will no longer be picked up by splits in the tree. Also, increased values of p and σ_{b_1} yield a slightly lower number of splits when cluster-level covariances are employed. Cluster-level covariances appear to be beneficial in accurately recovering the true subgroups, but with higher p, the Bonferroni correction will further reduce power to detect. Increased variance of σ_{b_1} may reduce the power to detect the second and third split in the tree, which are driven by subgroup differences in slopes.

For SEM trees and LongCART (Figure A2), strongest effects were observed for N, followed by σ_{b_0}, p_{noise}, ρ and σ_{b_1}. Results for N were as expected for all methods: with increasing sample size, more splits are implemented. Both LRT- and score-based SEM trees seem only affected by levels of σ_{b_0}, p_{noise} and ρ under misspecification of the random effects. Especially when both random intercept and slope variances are fixed to 0, higher levels of σ_{b_0} and p_{noise} yield more splits with both SEM tree approached. LRT-based SEM trees seem unaffected by levels of ρ, while score-based SEM trees seemed to implement a larger number of splits with increasing values of ρ. This pattern seemed reversed for increased magnitude of σ_{b_1}, which yields a lower number of splits for both SEM tree approaches, but only when the random effects were correctly specified. LongCART implemented more splits with higher levels of p_{noise} and ρ, but was unaffected by levels of σ_{b_0} and σ_{b_1}.

Figure A1

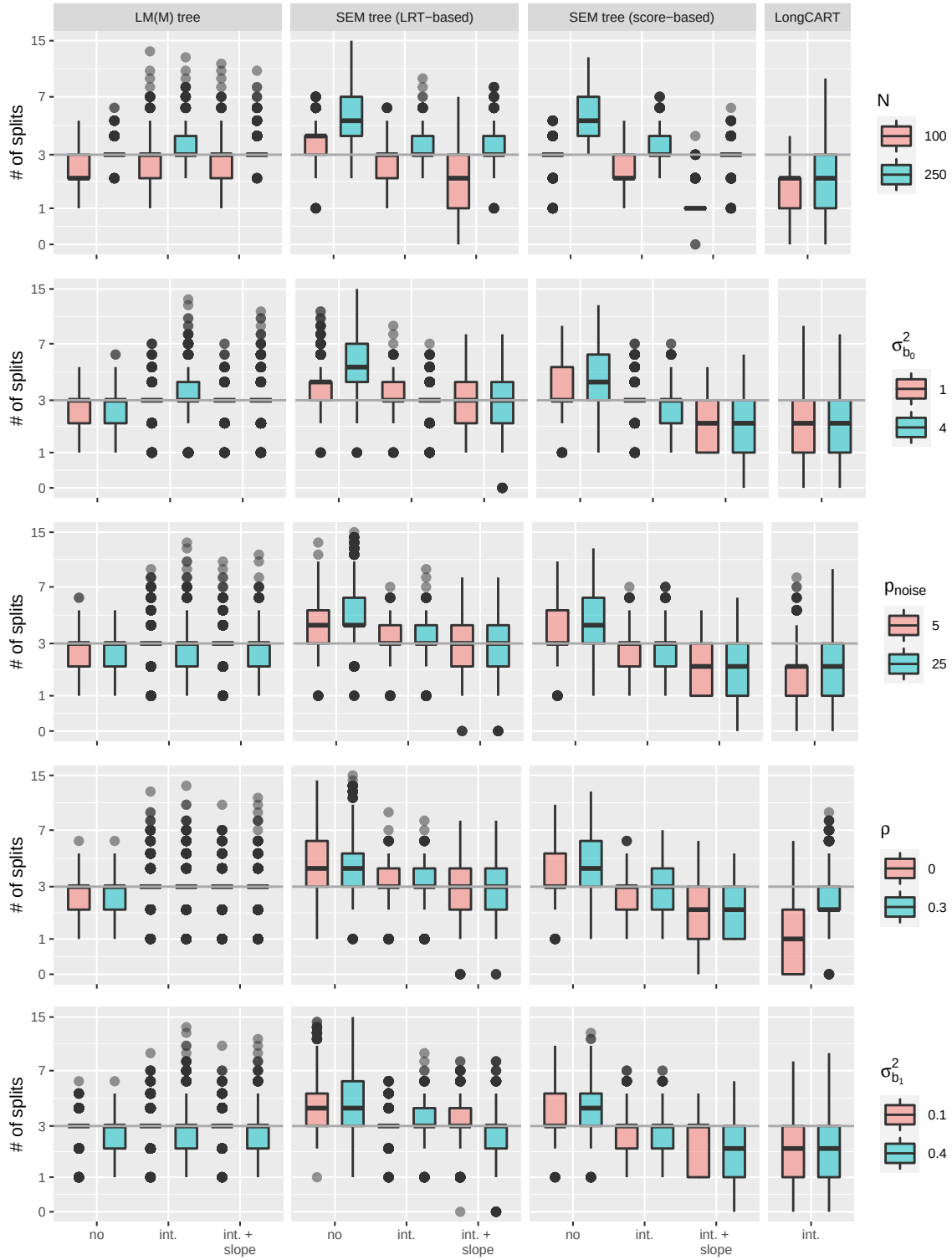
Effects of data-generating parameters on tree size for LM(M) trees.



Note. Grey lines indicate true number of splits; distances on y-axis are on log scale. $\sigma^2_{b_0}$ = variance of random intercept; $\sigma^2_{b_1}$ = variance of random slope; N = sample size at level 2; p_{noise} = number of noise variables; ρ = correlation between partitioning variables.

Figure A2

Effect of data-generating parameters on tree size for LM(M), SEM and LongCART trees.



Note. Grey lines indicate true number of splits; distances on y-axis are on log scale. N = sample size at level 2; $\sigma^2_{b_0}$ = variance of random intercept; $\sigma^2_{b_1}$ = variance of random slope; p_{noise} = number of noise variables; ρ = correlation between partitioning variables.