Detecting Treatment-Subgroup Interactions in Clustered Data with Generalized Linear Mixed-Effects Model Trees

M. Fokkema[1], N. Smits[2], A. Zeileis[3], T. Hothorn[4], H. Kelderman[5]
[1]Universiteit Leiden, [2]Universiteit van Amsterdam, [3]Universität Innsbruck, [4]Universität Zürich, [5]Universiteit Leiden and Vrije Universiteit, Amsterdam

Abstract

Identification of subgroups of patients for which treatment A is more effective than treatment B, and vice versa, is of key importance to the development of personalized medicine. Tree-based algorithms are helpful tools for the detection of such interactions, but none of the available tree-based algorithms allow for taking into account a clustered or nested structure. Therefore, we propose the generalized linear mixed-effects model trees (GLMM tree) algorithm, which allows for the detection of treatment-subgroup interactions, while accounting for the clustered structure of a dataset. The algorithm uses model-based recursive partitioning (MOB) to detect treatment-subgroup interactions, and a GLMM for the estimation of random-effects parameters. In a simulation study, we evaluate the performance of GLMM trees and compare it with that of trees without random effects, and GLMMs with pre-specified interactions. GLMM tree was found to accurately recover treatment-subgroup interactions in 90% of the simulated datasets, whereas trees without random effects only did so in 61% of datasets. GLMM tree also outperformed trees without random effects in terms of predictive accuracy (.94 and .88 on average, respectively) and Type-I error rates (4% and 33%, respectively). Furthermore, compared to GLMMs with pre-specified interaction effects, GLMM tree showed equal predictive accuracy (.60 on average), but better accuracy in detecting interactions. We illustrate the application of GLMM trees on an individual patient-level data meta-analysis on treatments for depression. We conclude that GLMM tree is a promising algorithm for

The authors would like to thank Prof. Pim Cuijpers, Prof. Jeanne Miranda, Dr. Boadie Dunlop, Prof. Rob DeRubeis, Prof. Zindel Segal, Dr. Sona Dimidjian, Prof. Steve Hollon and Erica Weitz for granting access to the dataset for the application. The work for this paper was partially done while MF, AZ and TH were visiting the Institute for Mathematical Sciences, National University of Singapore in 2014. The visit was supported by the Institute.

the detection of treatment-subgroup interactions in clustered datasets.

Keywords: model-based recursive partitioning, treatment-subgroup interactions, random effects, generalized linear mixed-effects model, classification and regression trees

Introduction

In research on the efficacy of treatments for somatic and psychological disorders, the one-size-fits-all paradigm is slowly losing ground, and stratified (or personalized) medicine is becoming increasingly important. Stratified medicine presents the challenge of finding which patients respond best to which treatments. This can be referred to as the detection of treatment-subgroup interactions (e.g., Doove, Dusseldorp, Van Deun, & Van Mechelen, 2014). Often, treatment-subgroup interactions are studied using linear models, such as factorial analysis of variance techniques, in which potential moderators have to be specified a-priori, have to be checked one at a time, and continuous moderator variables have to be discretized. This may hamper identification of which treatment works best for whom, especially when there are no a-priori hypotheses about treatment-subgroup interactions. As noted by Kraemer, Frank, and Kupfer (2006), there is a need for methods that generate instead of test such hypotheses.

Tree-based methods are such hypothesis-generating methods, as they can automatically detect subgroups which differ in the expected outcomes for one or more treatments. Due to their flexibility, tree-based methods are particularly useful for exploratory purposes, as they can handle many potential predictor variables at once and can automatically detect (higher order) interactions between predictor variables (Strobl, Malley, & Tutz, 2009). As such, tree-based methods are preeminently suited to the detection of treatment-subgroup interactions. Several tree-based algorithms for the detection of treatment-subgroup interactions have already been developed (Dusseldorp, Doove, & Van Mechelen, 2015; Dusseldorp & Meulman, 2004; Su, Tsai, Wang, Nickerson, & Li, 2009; Foster, Taylor, & Ruberg, 2011; Lipkovich, Dmitrienko, Denne, & Enas, 2011; Zeileis, Hothorn, & Hornik, 2008; Seibold, Zeileis, & Hothorn, 2016; see Doove et al., 2014 for an overview). Also, Zhang, Tsiatis, Laber, and Davidian (2012); Zhang, Tsiatis, Davidian, Zhang, and Laber (2012) have developed a flexible classification-based approach, allowing users to select from a range of statistical methods, including trees.

In many instances, researchers may want to detect treatment-subgroup interactions in a generalized linear mixed-effects (GLMM) type model. For example, in individual-level patient data meta-analysis, where datasets of multiple clinical trials on the same treatments are pooled (e.g., Koopman, Heijden, Glasziou, Grobbee, & Rovers, 2007). In such analyses, the nested or clustered structure of the dataset should be taken into account by including study-specific random effects in the model, prompting the need for a mixed-effects model (e.g., Cooper & Patall, 2009; Higgins, Whitehead, Turner, Omar, & Thompson, 2001). In linear models, ignoring the clustered structure may lead, for example, to biased inference due to underestimated standard errors in linear models (e.g., Bryk & Raudenbush, 1992; Noortgate, Opdenakker, & Onghena, 2005). For tree-based methods, ignoring the clustered structure has been found to result in the detection of spurious subgroups and inaccurate predictor variable selection (e.g., Sela & Simonoff, 2012; Martin, 2015). However, none of the purely tree-based methods for treatment-subgroup interaction detection allow for taking into account the clustered structure of a dataset. Therefore, in the current paper, we present a tree-based algorithm which can be used for the detection of (treatment-subgroup) interactions and non-linearities in GLMM type models: generalized linear mixed-effects model trees, or GLMM trees.

The GLMM tree algorithm builds on model-based recursive partitioning (MOB, Zeileis et al., 2008), which offers a flexible framework for subgroup detection. For example, GLM-based MOB has been applied to detect treatment-subgroup interactions for the treatment of depression (Driessen et al., 2016) and amyotrophic lateral sclerosis (Seibold et al., 2016). In contrast to other purely tree-based methods (e.g., Zeileis et al., 2008; Su et al., 2009; Dusseldorp & Van Mechelen, 2014)), GLMM tree allows for taking into account the clustered structure of datasets. In contrast to previously suggested regression trees with random effects, GLMM tree allows for treatment effect estimation, with continuous as well as non-continuous response variables (e.g., Hajjem, Bellavance, & Larocque, 2011; Sela & Simonoff, 2012).

In what follows, we first introduce an artificial motivating dataset, which will be used to illustrate and explain treatment-subgroup interaction detection with GLM-based MOB and GLMM tree. In the Simulation-based Evaluation, we evaluate the performance of GLMM tree in simulated datasets, and compare it with that of GLM-based MOB and GLMMs with pre-specified interaction effects. The comparison with GLM-based MOB allows for assessing the extent to which including random-effects estimation improves the performance of MOB trees. The comparison with GLMMs with pre-specified interactions will allow for assessing the extent to which using a tree-based method will improve detection of treatment-subgroup interactions. In the Application, we apply the algorithm to an existing dataset of a patient-level meta-analysis on the effects of psycho- and pharmacotherapy for depression. Finally, in the Discussion we summarize the results and point out some directions for future research.

Artificial motivating dataset

We created a dataset representing observations on 150 participants in a randomized clinical trial. Every participant was randomly assigned to Treatment 1 or Treatment 2, and

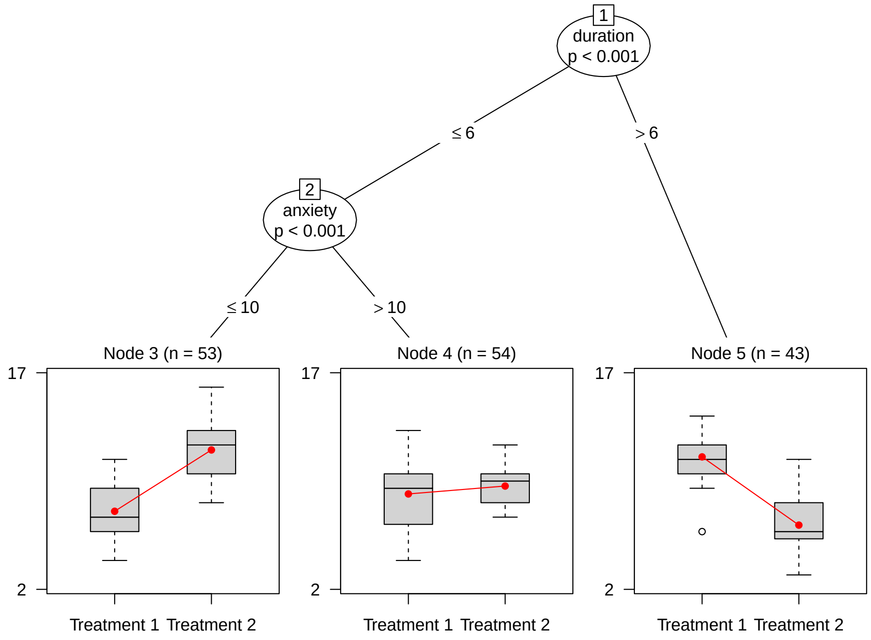

Figure 1. GLMM tree of the motivating example dataset. Three covariates (anxiety questionnaire score, duration of depressive symptoms at baseline in months and age) were used as potential splitting variables, and the clustering structure was taken into account by estimating random intercepts.

has a value for the response variable, with which the effect of treatment is assessed: the posttreatment total score on a depression inventory. For all participants, three covariate values are available: duration of depressive symptoms prior to treatment in months (duration, range 0–15); age in years (age, range 18–75); anxiety inventory total score (anxiety, range 3–18).

The simulated dataset has 3 subgroups with differential treatment effectiveness. The first subgroup consists of observations with duration ≤ 6 and anxiety ≤ 10. In this subgroup, Treatment 1 is more effective than Treatment 2: the mean of the response variable is 7 for Treatment 1, and 11 for Treatment 2. The second subgroup consists of observations with duration ≤ 6 and anxiety > 10. In this subgroup, both therapies are equally effective: the mean value of the response variable is 9 for Treatment 1, and 9 for Treatment 2. The third subgroup consists of observations with duration > 6. In this subgroup, Treatment 2 is more effective than Treatment 1: the mean value of the response variable is 12 for Treatment 1, and 7 for Treatment 2.

Participants were part of one of ten clusters, each with a different value for the random intercept. Data were generated such that covariates and cluster-specific intercepts were uncorrelated. 43% of variance in posttreatment depression scores was due to treatment-subgroup interactions, while 8% of variance was due to cluster-specific variation.

See Figure 1 for the GLMM tree fitted to the simulated data set which recovers the true underlying structure.

Model-based recursive partitioning

The rationale behind MOB is that a global parametric model may not describe the data well, and when additional covariates are available it may be possible to partition the dataset with respect to these covariates, and find a better model in each cell of the partition (Zeileis et al., 2008). This is reminiscent of the classification and regression tree (CART) algorithm of Breiman, Friedman, Olshen, and Stone (1984), which splits the dataset into subsets, for which the distributions of the outcome variable are most different. However, CART trees detect differences in the mean (or median) value across terminal nodes, whereas MOB trees detect differences in parameters of more complex models across terminal nodes.

For example, let us take a global GLM to estimate the overall treatment effect in the motivating dataset. The expectation μ_i of outcome y_i given the treatment regressors x_i is modeled through a linear predictor and suitable link function[1]:

$$E[y_i|x_i] = \mu_i, \tag{1}$$

$$g(\mu_i) = x_i^\top \beta, \tag{2}$$

where $x_i^\top \beta$ is the linear predictor for observation i and g is the link function. β is a vector of fixed-effects regression coefficients, the first element representing the intercept, corresponding to the mean value of the linear predictor in the first treatment group, and the second element representing the slope, which is the mean difference in the linear predictor between the first and second treatment groups. Thus, for simplicity we assume x_i and β to have length 2 in the current paper. Also, for the continuous response variable in the motivating data set, we employ a Gaussian distribution with identity link and denote the error by $\epsilon_i = y_i - \mu_i$ with variance σ^2_ϵ. However, the model can easily accomodate additional treatment conditions and covariates, and binary or count outcome variables.

The result of fitting a global GLM to the motivating dataset is depicted in Figure 2; the boxplots show the distribution of the posttreatment depression scores in both treatment groups. The global model does not describe the data well: there is substantial residual variance and the slope of the regression line is nearly zero. This does not necessarily mean that posttreatment depression score and treatment type are unrelated, as the effect of treatment may be moderated by variables not yet included in the model.

The MOB algorithm can be used to detect such moderation, by testing for parameter

[1] An overview of notation used is provided in the appendix.

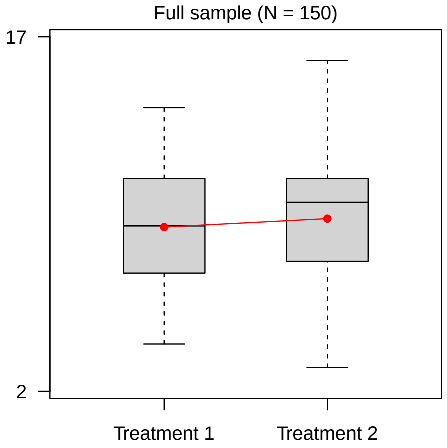

Figure 2. Example of a global GLM for treatment outcomes, based on the artificial motivating dataset ($N = 150$). The dot for Treatment 1 represents the first, and the slope of the regression line represents the second element of β.

stability over a set of auxiliary covariates, also known as *partitioning variables*. When the partitioning is based on a GLM, instabilities are differences in $\hat{\beta}$ across partitions of the dataset, which are defined by one or more auxiliary covariates not included in the linear predictor. To find these partitions, the MOB algorithm cycles iteratively through the following steps (Zeileis et al., 2008): (1) fit the parametric model to the dataset, (2) test for parameter instability over a set of partitioning variables, (3) if there is some overall parameter instability, split the dataset with respect to the variable associated with the highest instability, (4) repeat the procedure in each of the resulting subgroups.

More specifically, in step (2), to test for parameter instability, the so-called *scores* are computed, using the score function. When the model is correctly specified, the expected value of the score for each observation equals zero. Therefore, under the null hypothesis of parameter stability, the scores do not systematically deviate from the expected value of zero, when observations are ordered by the values of a potential partitioning variable U_k (c.f., Merkle & Zeileis, 2013). To statistically test whether the scores systematically deviate from zero with respect to variable U_k, the class of generalized M-fluctuation tests is used (Zeileis, 2005; Zeileis & Hornik, 2007).

If the null hypothesis of parameter stability in step (2) can be rejected, that is, if at least one of the partitioning variables U_k yields a p-value for the M-fluctuation test below the pre-specified significance level α, the dataset is partitioned into two subsets in step (3).

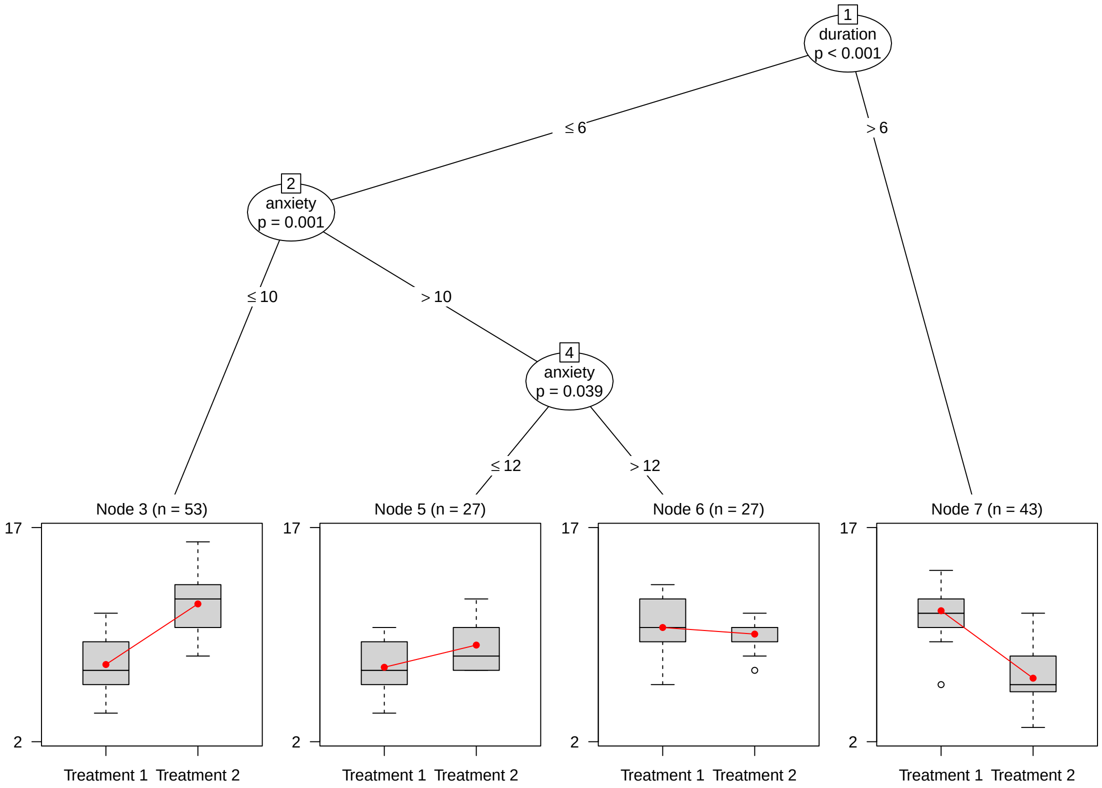

Figure 3. Example of a tree representation of model-based recursive partition, based on the artificial motivating dataset. Three additional covariates (anxiety questionnaire score, duration of depressive symptoms at baseline in months and age) were used as potential splitting variables.

This partition is created using U_{k^*}, the variable with the minimal p-value in step (2). The split point for U_{k^*} is selected, by taking the value that minimizes the sum of the values of the objective function in both partitions (Zeileis et al., 2008). In step (4), steps (1) through (3) are repeated in each partition, until the null hypothesis of parameter stability can no longer be rejected (or the subsets become too small).

Due to the binary recursive nature of MOB, the resulting partition can be represented as a binary tree. If the partitioning is based on a GLM, the result is a GLM tree, with a local fixed-effects regression model in every j-th ($j = 1, \dots, J$) terminal node or subgroup:

$$g(\mu_{ij}) = x_i^\top \beta_j \tag{3}$$

Note that, if the recursive subgroup structure (i.e., the partition) were known, the tree could be estimated as a single GLM where all x-variables interact with the subgroup indicator. Somewhat more formally, the model could then be written: $g(\mu_i) = x_i^{*\top} \beta^*$, where x_i^* are the values of the $2J$ interactions between the subgroups from the tree, and the elements of x_i. The corresponding β^* would have length $2J$ as well, containing subgroup-specific fixed-effects coefficients.

Figure 3 shows the GLM tree grown on the motivating dataset. By using the three auxiliary covariates (anxiety, duration and age), MOB partitioned the observations into four subgroups, each with a different estimate for β_j. Age was correctly not detected as a partitioning variable, and the left- and rightmost subgroups are in accordance with the true treatment-subgroup interactions described above. However, the two subgroups in the middle do not represent true subgroups, which is due to the clustered structure of the dataset not being taken into account.

Generalized linear mixed-effects model trees

For datasets containing observations from multiple clusters (e.g., trials or research centers), application of a GLMM would be more appropriate. The model in Equation 2 is then extended to include cluster-specific, or random effects:

$$g(\mu_i) = x_i^\top \beta + z_i^\top b \tag{4}$$

For a simple random intercept z_i is a unit vector of length M, of which the m-th element takes a value of 1, and all other elements take a value of 0; m $(m = 1, \ldots, M)$ denotes the cluster which observation i is part of. Further, b is a random vector of length M, each m-th element representing the random intercept for cluster m. For simplicity, we just employ such a cluster-specific intercept here but further random effects can be easily included in z_i. Furthermore, within the GLMM, it is assumed that b is normally distributed, with mean zero and variance σ_b^2. The parameters of the GLMM can be estimated with, for example, maximum likelihood (ML) and restricted ML (REML).

Note that, if the random-effects coefficients were known, the model could be estimated by a simple GLM as in Equation 2 where $z_i^\top b$ would only be added as an offset (i.e., a variable with a fixed coefficient of 1) to the linear predictor.

Although the random-effects part of the GLMM in Equation 4 accounts for the nested structure of the dataset, the global fixed-effects part may not describe the data well. Therefore, we propose the GLMM tree model, in which the fixed-effects part may be partitioned as in Equation 3 while still adjusting for random effects:

$$g(\mu_i) = x_i^\top \beta_j + z_i^\top b \tag{5}$$

To estimate the parameters of this model, we take an approach similar to that of the mixed-effects regression tree (MERT) approach of Hajjem et al. (2011) and Sela and Simonoff (2012). In the MERT approach, the fixed-effects part of a GLMM is replaced by a CART tree with constant fits in the nodes, and the random-effects part is estimated as usual. To estimate a MERT, an iterative approach is taken, alternating between (1) assuming random effects known, allowing for estimation of the CART tree, and (2) assuming the CART tree known, allowing for estimation of the random effects.

For estimating GLMM trees, we take this approach two steps further: (1) Instead of a CART tree with constant fits to estimate the fixed-effects part of the GLMM, we use a GLM tree. This allows not only for detection of differences in intercepts/levels across terminal nodes, but also for detection of differences in slopes/regression coefficients such as treatment effects. (2) By using generalized linear (mixed) models, the response may also be a binary or count variable instead of a continuous variable. The GLMM tree algorithm takes the following steps to estimate the model in Equation 5:

Step 0: Initialize by setting r and all values $\hat{b}_{(r)}$ to 0.

Step 1: Set $r = r + 1$. Estimate GLM tree $(x_i^\top \hat{\beta}_{j(r)})$, with $z_i^\top \hat{b}_{(r-1)}$ as an offset.

Step 2: Estimate random effects in the mixed-effects model $x_i^\top \hat{\beta}_{j(r)} + z_i^\top \hat{b}_{(r)}$ with subgroups $j(r)$ from the GLM tree.

Step 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 GLM tree and coefficients $\beta_{j(r)}$ and $b_{(r)}$ are re-estimated. The GLM tree is estimated, given the estimated random effects from the last iteration, and the random effects are estimated, given the estimated GLM tree from the current iteration. Iterations are continued until convergence, which is monitored by computing the log-likelihood criterion of the mixed-effects model in Equation 5. Typically, this converges if the tree does not change from one iteration to the next.

In Figure 1, the result of applying the GLMM tree algorithm to the motivating dataset is presented. As already mentioned above, by taking into account the clustering of observations, the true treatment subgroups have been recovered, and the spurious split involving the anxiety variable no longer appears in the tree.

Simulation-based evaluation: Method

To assess the performance of GLMM tree, we carried out three simulation studies. In Study I, we compared the accuracy of GLMM tree with that of GLM tree in datasets with treatment-subgroup interactions. In Study II, we compared the accuracy of GLMM tree with that of GLM tree in datasets without treatment-subgroup interactions, allowing for assessment of the Type-I error rate. In Study III, we compared the accuracy of GLMM tree with that of a more classical approach of interaction detection: a GLMM with pre-specified interactions.

Software

R (R Core Team, 2016) was used for the generation and analysis of all datasets. The `partykit` package (version 1.0-2; Hothorn & Zeileis, 2015) was employed for estimating GLM trees, using the `lmtree` function for normal linear regressions and `glmtree` is available for other response distributions. For estimation of GLMMs the `lmer` function (or `glmer` function, respectively) from the `lme4` package (version 1.1-7; Bates, Mächler, Bolker, & Walker, 2015) was employed, using restricted maximum likelihood (REML) estimation. For estimation of GLMM trees the former two packages were combined in a new package `glmertree` (version 0.1-0; Fokkema & Zeileis, 2016; available from R-Forge). This package provides functions `lmertree` and `glmertree` that iterate between estimation of the `lmtree`/`glmtree` model and the `lmer`/`glmer` model.

In all simulations, the significance level α for the parameter instability tests in the trees was set to .05, with a Bonferroni correction applied for multiple testing. The minimum number of observations per node in the tree was set to 20 and the maximum tree depth was set to four, thus limiting the number of potential subgroups to eight.

Simulation design

(Z: My impression is that the description of the data-generating process is too far ahead of the simulation itself. I would recommend to first explain what is the same in all three simulation studies. And then what is specific could maybe be included at the beginning of each of the individual studies so that readers have it in mind when looking at the results. So in this section here I would just briefly explain the idea and defer all details until later.) README

Treatment-subgroup interactions. For generating datasets with treatment-subgroup interactions, we used one of the treatment-subgroup interaction designs from Dusseldorp and Van Mechelen (2014), which is depicted in Figure 4. Figure 4 shows two subgroups with mean differences in treatment outcomes, and two subgroups without mean differences in treatment outcomes. The four subgroups are characterized by values on the (true) partitioning variables U_2, and U_1 or U_5. The other potential partitioning variables (U_3, U_4, U_6 through U_{15}) are noise variables.

Continuous interactions. The subgroups depicted in Figure 4 are piecewise functions of the partitioning variables, therefore these interactions can be referred to as piecewise interactions. However, interactions may also consist of continuous functions of the partitioning variables. For comparing the accuracy of GLMM tree with that of GLMMs with pre-specified interaction terms, we generated datasets in which interactions were varied to be purely piecewise, purely continuous, or a combination of both. To generate purely continuous interactions, the outcome variable y was generated as the sum of:

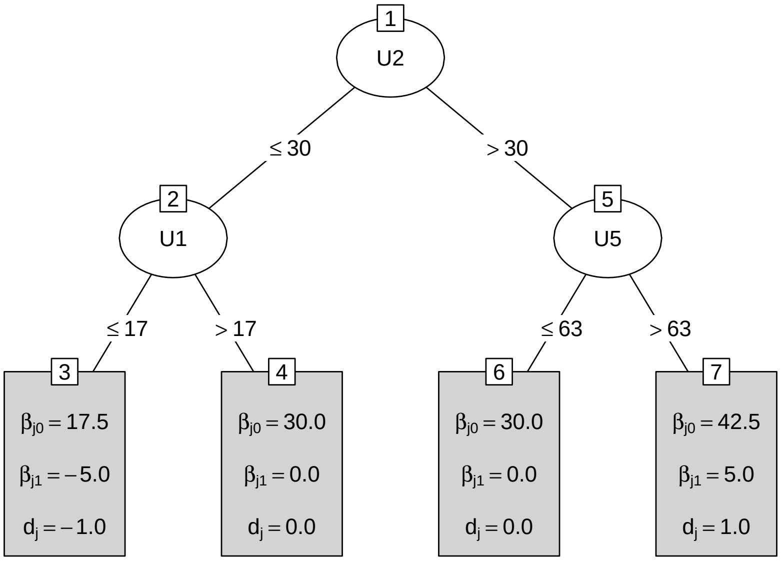

Figure 4. Data-generating model for treatment-subgroup interactions. Parameter d denotes the standardized mean difference between the outcomes of Treatment 1 and 2 (i.e., $\beta_{j1}/\sigma_\epsilon$).

1. a main effect of U_2 (coefficient = 1.5)

2. an interaction between U_1 and U_2 with a negative effect (coefficient = $-.25$), and an interaction between U_2 and U_5 with a positive effect (coefficient = .25)

3. an interaction between U_1, U_2 and treatment with a negative effect (coefficient = $-.25$), and an interaction between U_2, U_5 and the treatment indicator with a positive effect (coefficient = .25)

(Z: Also include the formula for the linear predictor in this setup so that this can be easily matched with the GLMM estimated.) README

Note that the direction (positive or negative) and type (main or interaction) of these effects correspond to those in Figure 4. All interactions were created by using centered U_k variables, calculated by subtracting the variable means.

Data-generating parameters. In generating datasets, we varied seven parameters of the data-generating process:

1. Three levels for sample size: $N = 200$, $N = 500$, $N = 1000$.

2. Two levels for the number of potential partitioning covariates U_1 through U_K: $K = 5$, $K = 15$ (where only U_1, U_2 and U_5 are true partitioning variables).

3. Two levels of intercorrelations between the covariates U_1 through U_K: $\rho_{U_k,U_{k'}} = 0.0$, $\rho_{U_k,U_{k'}} = 0.3$.

4. Three levels for the number of clusters: $M = 5$, $M = 10$, $M = 25$.

5. Three levels for the population standard deviation of the normal distribution from which the cluster specific intercepts were drawn: $\sigma_b = 0$, $\sigma_b = 5$, $\sigma_b = 10$.

6. Three levels for the intercorrelations between b and one of the U_k variables: b and U_k uncorrelated, b correlated with a true partitioning variable (i.e., U_2, U_1, or U_5, introducing a correlation of ≈ 0.42), b correlated with a non-partitioning covariate (i.e., U_3 or U_4, introducing a correlation of ≈ 0.42)[2].

7. Two levels for β_{j1}, the unstandardized mean difference in treatment outcomes. In datasets with treatment-subgroup interactions, the treatment effect difference varied only in nodes 4 and 7, with levels $|\beta_{j1}| = 2.5$ (corresponding to a medium effect size, Cohen's $d = 0.5$; Cohen, 1992) and $|\beta_{j1}| = 5.0$ (corresponding to a large effect size; Cohen's $d = 1.0$).

Variable distributions. Following the approach of Dusseldorp and Van Mechelen (2014), all covariates U_1 through U_K were drawn from a multivariate normal distribution with means $\mu_{U1} = 10$, $\mu_{U2} = 30$, $\mu_{U4} = -40$, and $\mu_{U5} = 70$. The means for all other covariates (i.e., μ_{U3}, and μ_{U6} through μ_{U15}) were drawn from a discrete uniform distribution on the interval $[-70, 70]$. All covariates U_1 through U_{15} have the same standard deviation: $\sigma_{Uk} = 10$. Correlations between U variables vary according to the third facet of the data-generating design described above.

To generate the cluster-specific intercepts b_m, we partitioned the sample into equally-sized clusters, conditional on one of the variables U_1 through U_5, producing the correlations in the sixth facet of the simulation design. For each cluster, a single value b_m was drawn from a normal distribution with mean 0 and the value of σ_b given by the fifth facet of the simulation design. When b was correlated with one of the potential partitioning variables, this variable was randomly selected.

To generate node-specific fixed effects, we partitioned the sample according to the terminal nodes of the tree in Figure 4.3. In combination with the seventh facet of the simulation design, this determines the values of β_j. For every observation, we generated a binomial variable (with probability .5) as an indicator for treatment type. Random errors ϵ were drawn from a normal distribution with $\mu_\epsilon = 0$ and $\sigma_\epsilon = 5$.

Finally, the response variable was calculated as the sum of the (node-specific) fixed effects, the random intercept and the error term: $y_i = x_i^\top \beta_j + z_i^\top b_m + \epsilon_i$.

Assessment of performance. Firstly, the total number of nodes in estimated GLM and GLMM tree were calculated to assess the extent to which the algorithms may detect spurious subgroups. Also, the number of nodes allow for assessing Type-I (the extent to

[2] Note that, when $\sigma_b = 0$, the correlation between b and one of the U_k variables is 0, by definition. However, datasets were created for this condition, to allow for a full factorial design of the simulation study; in reality, b and U are uncorrelated in these instances.

which datasets are erroneously partitioned) and Type-II (the extent to which datasets are erroneously not partitioned) error.

Secondly, in datasets with treatment-subgroup interactions, we assessed the accuracy of estimated models. For GLM and GLMM trees, an accurately recovered tree was defined as a tree with (1) the true tree size (i.e., tree size $= 7$ in datasets with treatment-subgroup interactions), (2) the first split in the tree involving variable U_2 and a value of 30 ± 5, (3) the next split on the left involving variable U_1 and a value of 17 ± 5, and (4) the next split on the right involving variable U_5 and a value of 63 ± 5. Note that the allowance of ± 5 equals an allowance of plus or minus half the population standard deviation of the partitioning variable (σ_{U_k}).

Predictive accuracy of each method was assessed by calculating the correlation between true and predicted treatment-effect differences (β_{j1} in Figure 4) in each dataset. Consequently, predictive accuracy was only assessed in datasets with treatment-subgroup interactions, as the true treatment differences have a constant value in datasets without treatment-subgroup interactions. To prevent overly optimistic estimates of predictive accuracy (Hastie, Tibshirani, & Friedman, 2009), predictive accuracy was assessed using test datasets. Test datasets were generated from the same population as training datasets, but test observations were not drawn from the same clusters as the training observations, but from 'new' clusters.

Simulation-based evaluation: Results

Study I: Detection of Treatment-Subgroup Interactions

For each cell of the data-generating design described above, 50 datasets were generated, resulting in $50 \times 3 \times 2 \times 2 \times 3 \times 3 \times 3 \times 2 =$ 32,400 training datasets with treatment-subgroup interactions. In these datasets, the true tree size was 7 (4 terminal nodes and 3 inner nodes; Figure 4).

Tree size. GLMM tree yielded trees with an average size of 7.15 nodes (SD = 0.61), whereas GLM tree yielded an average tree size of 8.15 nodes (SD = 2.05), indicating that GLM trees involved more spurious splits than GLMM trees, on average. The estimated probability that a dataset was erroneously not partitioned (the Type-II error) was 0 for both algorithms.

To find the most important predictors of tree size, we performed an ANOVA with algorithm type and the parameters of the data-generating process as independent variables. First-order interactions between algorithm type and each of the data-generating parameters were also included. The most important predictors of tree size were algorithm type, sample size, variance of the random intercept, and the correlation between partitioning variables. These effects were further assessed by means of a graphical display (Figure 5).

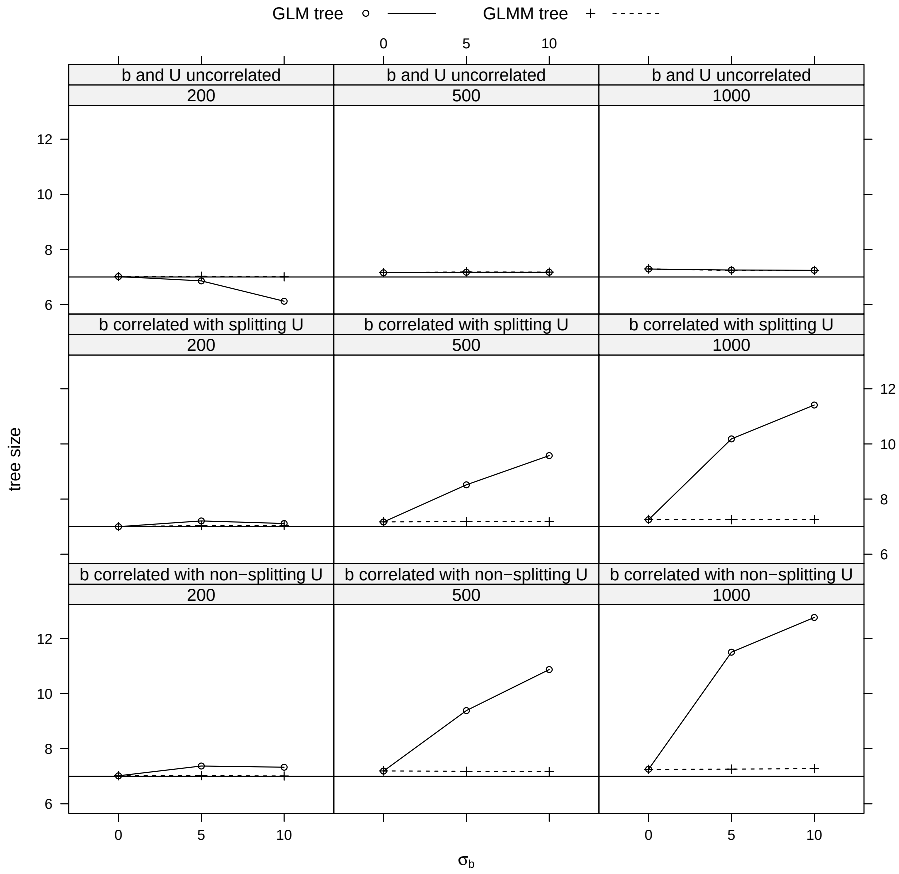

Figure 5. Average tree size of GLM and GLMM trees for datasets with treatment-subgroup interactions. Rows represent levels of dependence between random effects (b) and one of the partitioning variables U_k; columns represent levels of sample size.

Figure 5 indicates that GLMM tree grows trees of about the true tree size in all conditions. On average, tree size increases slightly with sample size for GLMM tree, and dramatically for GLM tree. In the absence of random effects (i.e., $\sigma_b = 0$), GLM and GLMM trees grow trees of about the same size. However, clear differences in tree size were observed when $\sigma_b > 0$. When sample size is small and random intercepts are not correlated with a partitioning covariate, GLM tree lacks power and grows trees that are too small, on average. When sample size is larger and random intercepts are not correlated with a partitioning covariate, GLM and GLMM trees are about equally sized. However, when random intercepts are correlated with a partitioning covariate, GLM starts to create spurious splits, especially with larger sample sizes (i.e., 500 or 1000) and larger σ_b values (i.e., 10). This effect was somewhat stronger when the random intercept was correlated

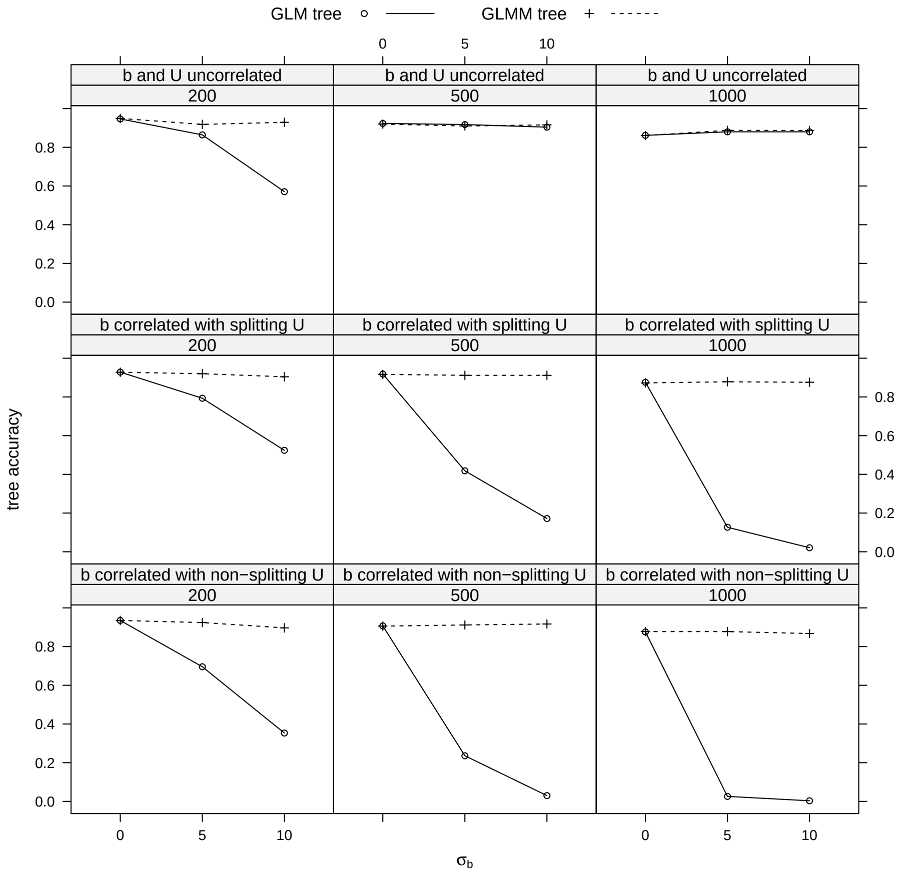

Figure 6. Tree accuracy of GLM and GLMM tree. Tree accuracy is defined as the proportion of datasets in which the true tree was accurately recovered. Rows represent levels of dependence between random effects (b) and one of the partitioning variables U_k, columns represent levels of sample size.

with a covariate that did not appear in the true tree.

Accuracy of recovered trees. To assess the accuracy of trees recovered by both algorithms, we inspected variables and values that were selected for partitioning. For the first split, GLMM tree selected the true partitioning variable (U_2) in all datasets, and GLM tree in all but one datasets. The splitting value for U_2 selected for the first split was 29.94 for both GLM and GLMM tree, which is very close to the true splitting value of 30 (Figure 4).

However, further splits were more accurately recovered by GLMM tree, which accurately recovered the partition in 90.40% of datasets, and GLM tree in only 61.44% of datasets. To find the most important predictors of tree accuracy, we fitted a GLM with algorithm type and the parameters of the data-generating process as independent variables.

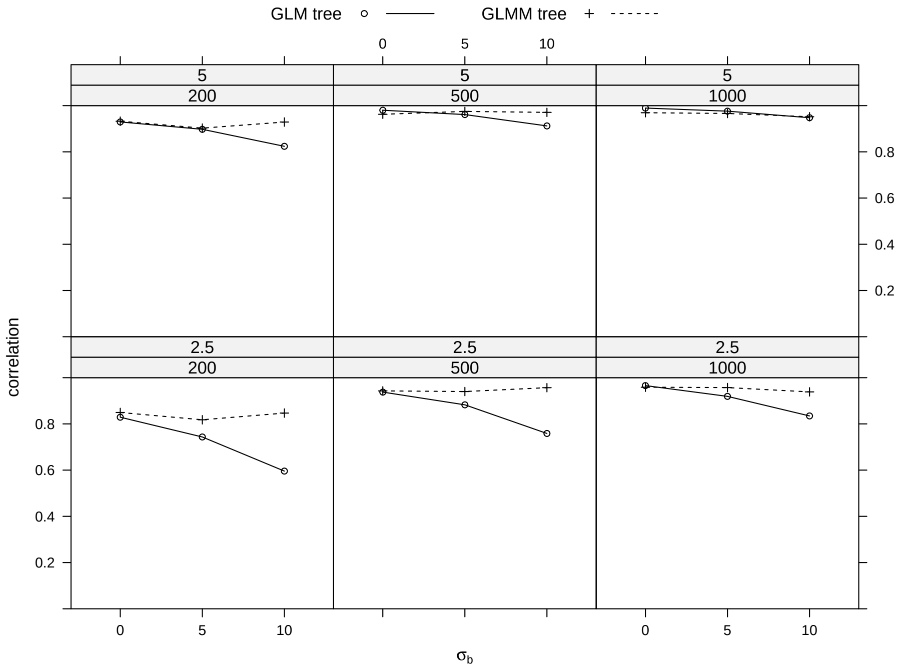

Figure 7. Average predictive accuracy of GLM and GLMM trees. Predictive accuracy of trees is defined as the correlation between the true and predicted differences between Treatment 1 and 2. Rows represent the levels of the absolute value of the unstandardized treatment-effect difference in subgroups with treatment-effect differences, columns represent levels of sample size.

First-order interactions between algorithm type and each of the data-generating parameters were also included. The most important predictors of tree accuracy were algorithm type, sample size, variance of the random intercept, and the level of dependence between the partitioning variables and the random intercept. These effects were further assessed by means of a graphical display (Figure 6).

Figure 6 indicates that in the absence of random effects, the trees recovered by GLM and GLMM tree were about equally accurate. In the presence of random effects, GLM trees were much less accurate than GLMM trees. This was found for all sample sizes, when random effects were correlated with a partitioning covariate. This effect was somewhat stronger when the correlated U_k was not a true partitioning variable. However, when random intercepts were not correlated with one of the U_k variables, GLMM tree outperformed GLM tree only when sample size was small (i.e., $N = 200$).

Predictive accuracy of trees. The predicted treatment-effect differences of GLMM tree were closer to the true differences than those of GLM tree, with a mean correlation of .93 (SD = 0.12) for GLMM tree and .88 (SD = 0.19) for GLM tree. An ANOVA indicated that the most important predictors of predictive accuracy were algorithm type, sample size, variance of the random intercept, and the treatment-difference effect size. These effects were further assessed by means of a graphical display (Figure 7).

Figure 7 shows clear main effects of sample size and treatment-effect differences for both algorithms. In the absence of random effects (i.e., $\sigma_b = 0$), predictions of GLM and GLMM tree were about equally accurate. In the presence of random effects, GLM tree predictions were clearly less accurate than those of GLMM tree when treatment-effect differences were smaller (i.e., Cohen's $d = .5$). This effect was also observed, but much less pronounced, when treatment-effect differences were larger (i.e., Cohen's $d = 1.0$). In other words, GLMM tree outperforms GLM tree in the presence of random effects, especially when sample size and treatment-effect differences are smaller.

Study II: Type-I error

(Z: Wouldn't it be more natural to just look at P(tree size greater than 1), i.e., the probability of splitting at all. Then it could be worked out more clearly that GLMM trees approximately keep the significance level while GLM trees generally have an inflated irror.)

README

Design. In addition to the accuracy of treatment-subgroup interaction detection, we assessed the Type-I error: the extent to which datasets without treatment-subgroup interactions may erroneously be partitioned. For assessing Type-I error of GLM and GLMM tree, we generated datasets without treatment-subgroup interactions, in which there was only a main effect of treatment in the population. Put differently, there was only a single global value of $\beta_j = \beta$ in every dataset. For these datasets, we varied the same parameters of the data-generating process as in Study I. However, the sixth facet of the data-generating process had only two levels for these datasets, as there were no 'true' partitioning variables in these datasets. Therefore, $50 \times 3 \times 2 \times 2 \times 3 \times 3 \times 2 \times 2 = 21{,}600$ datasets without treatment-subgroup interactions were generated.

Tree size and Type-I error. In datasets without treatment-subgroup interactions, average tree size was 1.09 (SD = 0.44) for GLMM tree, and 2.02 (SD = 1.68) for GLM tree. The overall Type-I error rate was only .04 for GLMM tree, and .33 for GLM tree. Again, main predictors of tree size were found to be sample size, variance of the random intercepts, and dependence between the random intercept and one of the partitioning variables. These effects were further assessed by means of a graphical display (Figure 8). Figure 8 indicates that in the absence of random effects (i.e., $\sigma_b = 0$), both GLM and GLMM tree tend to create trees of size 1. In the presence of random effects, GLMM tree still tends to create trees of size 1, but GLM tree creates much larger trees, when the random effects are correlated with one of the partitioning covariates. This effect increases with sample size.

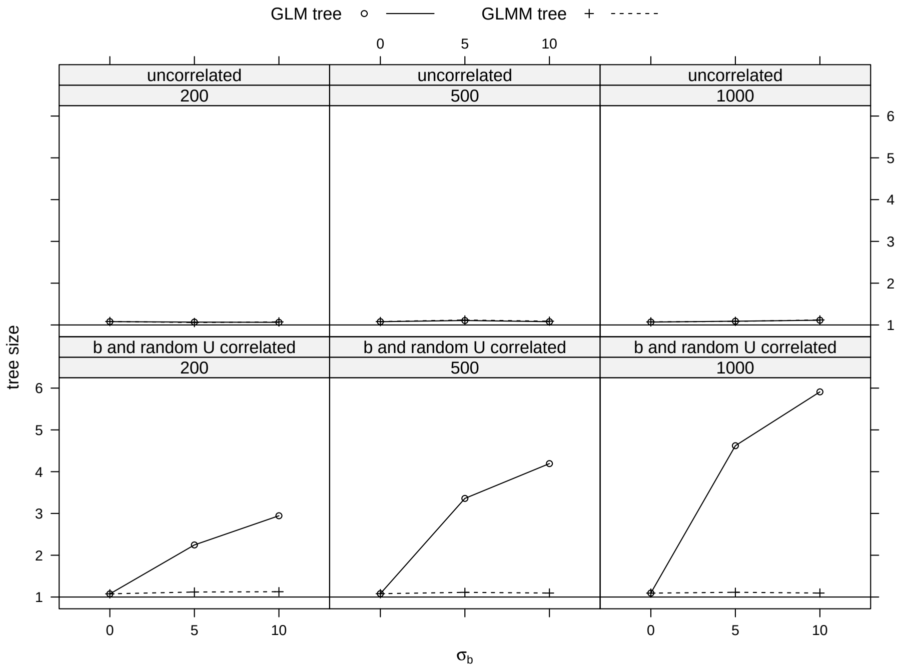

Figure 8. Average tree size of GLM and GLMM trees for datasets without treatment-subgroup interactions. Rows represent dependence between random effects (b) and one of the partitioning variables U_k; columns represent sample size.

Study III: Piecewise vs. Continuous Interactions

Design. To compare the performance of GLMM tree with that of GLMMs with prespecified interaction effects, we created datasets in which interactions were varied from purely continuous to purely piecewise. For these datasets, the seventh facet of the interaction design (effect size of treatment differences) was replaced by a factor with three levels, controlling the type of interactions: purely continuous interactions, purely piecewise interactions, and a combination of both. Purely continuous interactions were generated as described above, in the Method section. To allow for a smooth transition from purely continuous to purely continuous interactions, we aimed to keep the proportion of variance explained by each of the covariates equal across piecewise, continuous and combined interactions. Therefore, for piecewise interactions, we used the partition depicted in Figure 4, but not the coefficients. The coefficients used for data-generation were replaced by the subgroup-specific means, given purely continuous interactions. To generate the combined conitnuous and piecewise interactions, y values according to the purely piecewise and continuous interaction models were summed and divided by 2.

Furthermore, in this simulation study, the number of cells in the simulation design was reduced by limiting the fourth facet of the data-generating design to a single level ($M = 25$ clusters), as the earlier simulations indicated no effects of the number of clusters. The fifth facet of the data-generating design was limited to two levels ($\sigma_b = 2.5$ and $\sigma_b = 7.5$). As in Study II, we limited the sixth facet to two levels (the random intercept was either correlated to one of the covariates, or not). To counter unidentifiability due to potentially large numbers of pre-specified interaction terms in the GLMMs, the second level of the second facet (number of potential partitioning covariates) was set to $k = 10$, instead of the value of $k = 15$ used in the earlier simulations. As a result, $50 \times 3 \times 2 \times 2 \times 1 \times 2 \times 2 \times 3$ = 7,200 training datasets were generated for this study.

GLMMs were estimated by specifying main effects for all covariates U_k and the treatment indicator, first-order (or two-way) interactions between all pairs of covariates U_k, and second-order (or three-way) interactions between all pairs of covariates U_k and treatment. Accurately recovered GLMMs were defined to be models in which significant (i.e., absolute value of the t-test statistic ≥ 1.96) effects were found for U_2 and (treatment) interactions of U_1 and U_2 and U_2 and U_5. Accurately recovered GLMM trees were defined to be trees in which the first split involved U_2, the second split to the right involved U_1 and the third split to the right involved U_5. Note that this criterion of accuracy is somewhat more lenient than in Study I.

Accuracy of recovered models. Overall, GLMM tree accurately recovered main and interaction effects in 53.58% of datasets, wheareas GLMMs accurately recovered main and interaction effects in 34.81% of datasets. To find the most important predictors of model accuracy, we performed an ANOVA, in which the algorithm type, sample size, the number of covariates and type of interaction (continuous, piecewise or both) were found to be most important. These effects were further assessed by means of the graphical display depicted in Figure 9.

Figure 9 shows a clear effect of interaction type: GLMM tree performs best in datasets with purely piecewise interactions, and poorest in datasets with purely continuous interactions. GLMMs perform best in datasets with purely continuous interactions, and poorest in datasets with purely piecewise interactions. Figure 9 also shows a clear effect of the number of covariates: with a large number of covariates, GLMMs never recover all true effects in the dataset, whereas they always recover the true effects when the number of covariates is small and interaction effects are at least partly continuous. At the same time, the accuracy of GLMM tree is unaffected by the number of covariates. Finally, the performance of both GLMM tree and GLMMs is affected by sample size: the effects are more often recovered in datasets with larger sample sizes.

Predictive accuracy. Both algorithms showed about equal average correlations between the true and predicted treatment-effect differences: average accuracy was .60 (SD =

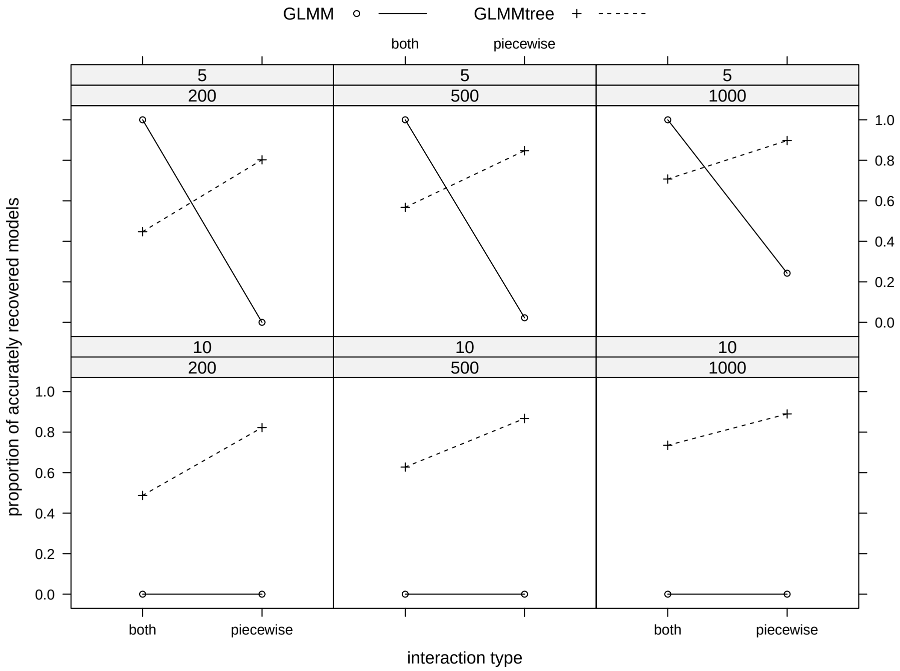

Figure 9. Model accuracy for GLMMs and GLMM trees. In an accurate model, the true main and interaction effects were accurately recovered. Columns represent sample size, rows represent the number of covariates in the model.

.36) for GLMMs and .60 (SD = .32) for GLMM trees. In terms of bias, GLMM treatment-effect difference predictions deviated more from the true values than the predictions of GLMM tree. The mean true treatment-effect difference averaged over all datasets was .04 (SD = 1.22), whereas the mean predicted treatment-effect differences averaged over all datasets for GLMMs were -.26 (SD = 100.33) and for GLMM trees .03 (SD = 3.88). Furthermore, it should be noted that the predictive models for GLMM trees were simpler than those of the GLMMs. Predictions for GLMMs in datasets with 10 covariates involved 11 (significant or non-significant) main effects, 100 first-order interactions and 100 second-order interactions. Predictions for GLMM tree involved a maximum of only 8 terminal nodes, as their maximum depth was restricted.

The most important predictors of predictive accuracy were algorithm type, sample size, the number of covariates, and type of interaction (continuous, piecewise or both). These effects were further assessed by means of the graphical display depicted in Figure 10. As Figure 10 indicates, GLMM trees show highest predictive accuracy in datasets with purely piecewise interactions, whereas GLMMs show highest predictive accuracy in datasets with purely continuous interactions. Furthermore, performance of GLMM tree is hardly affected

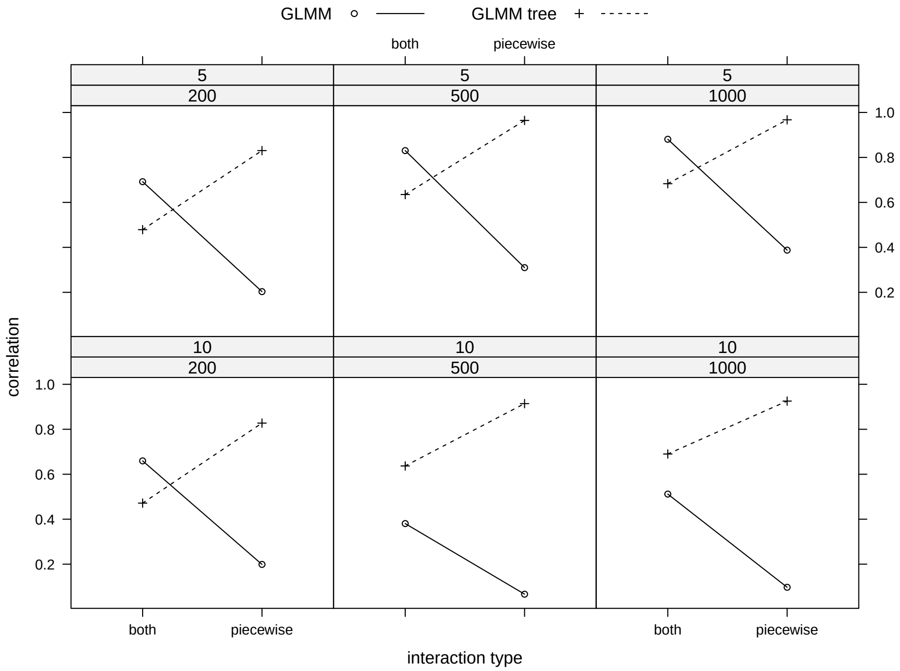

Figure 10. Average predictive accuracy of GLMMs and GLMM trees. Predictive accuracy of trees is defined as the correlation between the true and predicted differences between Treatment 1 and 2. Columns represent sample size, rows represent the number of covariates.

by the number of covariates, whereas the predictive accuracy of GLMMs deteriorates when the number of covariates increases. This indicates that GLMM tree may be especially useful for exploratory purposes.

Application: Individual patient-level meta-analysis dataset on treatments for depression

Method

To further illustrate the use of GLM and GLMM tree, we applied both algorithms to a dataset from an individual-patient data meta-analysis of Cuijpers et al. (2014). This meta-analysis was based on patient-level observations from 14 RCTs, comparing the effects of psychotherapy (cognitive behavioral therapy; CBT) and pharmacotherapy (PHA) in the treatment of depression. The study of Cuijpers et al. (2014) was aimed at establishing whether gender is a predictor or moderator of the outcomes of psychological and pharmacological treatments for depression. Treatment outcomes were assessed by means of the 17-item Hamilton Rating Scale for Depression (HAM-D; Hamilton, 1960). Cuijpers et al. (2014) found no indication that gender predicted or moderated treatment outcomes.

Further details on the dataset can be found in Cuijpers et al. (2014).

In our analyses, posttreatment HAM-D score was the outcome variable, and potential partitioning variables were age, gender, level of education, presence of a comorbid anxiety disorder at baseline, and pretreatment HAM-D score. The predictor variable in the linear model was treatment type (0 = CBT and 1 = PHA). An indicator for study was used as the cluster indicator.

In RCTs, ANCOVAs are often employed, to linearly control posttreatment values on the outcome measure for pretreatment values. Therefore, we estimated the GLM and GLMM trees using posttreatment HAM-D scores, controlled for the linear effects of pretreatment HAM-D scores. The trees were grown using data of patients with complete observations; that is, observations with non-missing values for potential partitioning variables, and pre- and posttreatment HAM-D score. As a result, data from 694 patients from 7 studies were included in the analyses. Results of our analysis may therefore not be representative of the complete dataset of the meta-analysis by Cuijpers et al. (2014).

To provide a standardized estimate of the treatment effect differences in the final nodes of the trees, we calculated node-specific Cohen's d values. Cohen's d was calculated by dividing the node-specific predicted treatment outcome difference by the node-specific pooled standard deviation. Confidence intervals for Cohen's d could not be calculated, as these can not take into account the exploratory nature of our analyses (i.e., variable and split point selection). The predictive accuracy of GLM and GLMM trees was assessed by calculating average correlations between observed and predicted HAM-D posttreatment scores, based on 50-fold cross validation.

The results of recursive partitioning techniques are known to be potentially unstable, in the sense that small changes in the dataset may substantially alter the variables or values selected for partitioning. Therefore, we used the R package `stablelearner` (Philipp, Zeileis, & Strobl, 2016) to assess the stability of the selected splitting variables and values. Using the `stabletree` function, we calculated the number of times a variable and value were selected for partitioning, over 500 subsamples of size $.9 \times N$ of the dataset.

Results

The tree and effects sizes resulting from application of GLM tree are presented in Figure 11. Those resulting from application of GLMM tree are presented in Figure 12.

The GLM tree (Figure 11) selected level of education as the first partitioning variable, and presence of a comorbid anxiety disorder as a second partitioning variable, for observations with a higher level of education. By taking into account study-specific intercepts, the GLMM tree (Figure 12) indicates that the first split made by GLM tree may be a spurious one. The GLMM tree selected presence of a comorbid anxiety disorder as the only partitioning variable. The terminal nodes of Figure 12 show only a single treatment-subgroup

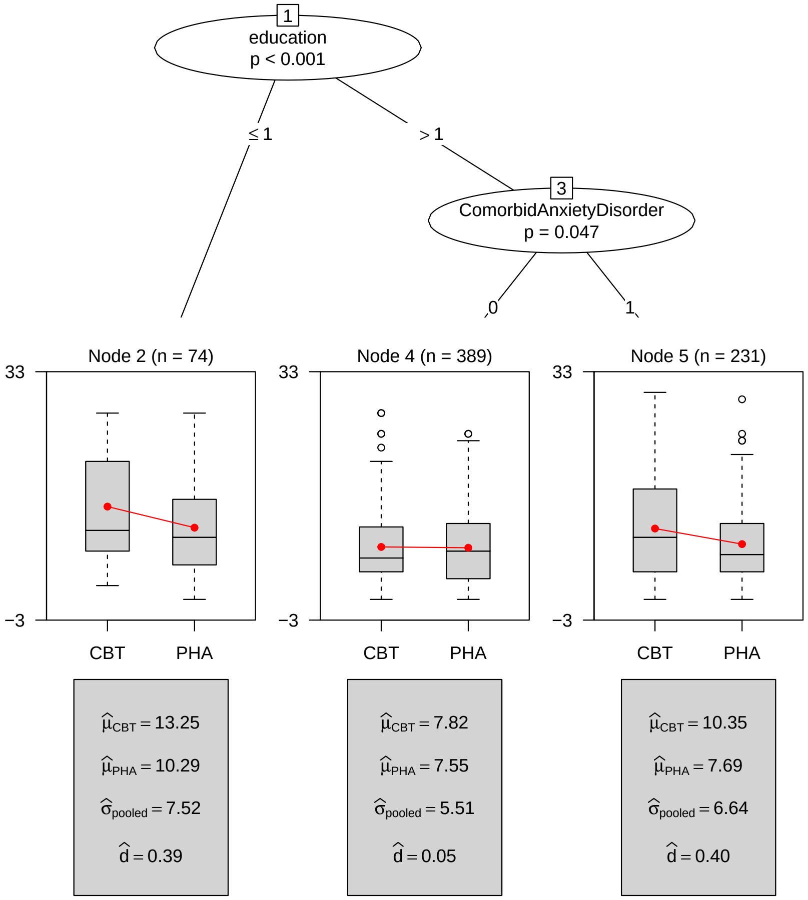

Figure 11. GLM tree for prediction of posttreatment total scores on the Hamilton Rating Scale for Depression (HAM-D). Upper panel: The y-axes of the boxplots represent posttreatment HAM-D scores, and the x-axes represent treatment levels (cognitive behavior therapy, CBT vs. pharmacotherapy, PHA). Lower panel: Subgroup-specific descriptive statistics.

interaction: for patients without a comorbid anxiety disorder, CBT and PHA provide more or less the same reduction in HAM-D scores (Cohen's $d = 0.05$; Figure 12). For patients with a comorbid anxiety disorder, PHA provides a greater reduction in HAM-D scores (Cohen's $d = 0.39$; Figure 12). The estimated intraclass correlation coefficient for the GLMM tree was .05.

Assessment of predictive accuracy by means of 50-fold cross validation indicated better predictive accuracy for GLMM tree than GLM tree. The correlation between true and predicted posttreatment HAM-D total scores, averaged over the 50 folds, was .28 ($var =$ .067) for GLMM tree, and .19 ($var =$ .084) for GLM tree. This indicates that GLMM

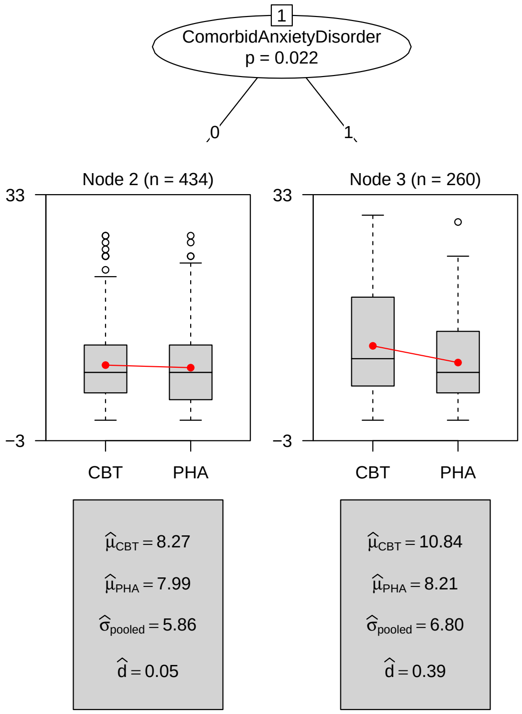

Figure 12. GLMM tree for prediction of posttreatment total scores on the Hamilton Rating Scale for Depression (HAM-D). Left panel: The y-axes of the boxplots represent posttreatment HAM-D scores, and the x-axes represent treatment levels (cognitive behavior therapy, CBT vs. pharmacotherapy, PHA). Right panel: Subgroup-specific descriptive statistics.

tree provided higher predictive accuracy, and also somewhat lower variability of predictive accuracy than GLM tree.

Table 1 presents statistics on the variables selected for partitioning in subsamples of the dataset. Note that the selection frequencies do not add up to 1, as trees may involve multiple, or no splits. Table 1 indicates that the presence of a comorbid anxiety disorder was selected for partitioning in the majority of GLMM trees grown on subsamples of the dataset, and all other variables were selected in less than 3% of the subsamples. As the comorbid anxiety disorder variable involved only a single splitting value, further assessment of the stability of splitting values was not necessary.

Table 1: Variable selection statistics

	Selection frequency	
Variable	GLM tree	GLMM tree
Education	.956	.014
ComorbidAnxietyDisorder	.398	.528
HRSDt0	.034	.002
Age	.000	.022
Gender	.002	.004

Note. Frequencies are calculated over 500 random subsamples of the complete dataset. Frequencies do not add up to 1, as trees may involve multiple or no splits.

Discussion

In the current paper, we presented the GLMM tree algorithm, which allows for estimation of a GLM-based recursive partition, as well as estimation of random-effects parameters. As such, we hypothesized GLMM tree to be well suited for the detection of treatment-subgroup interactions in clustered datasets, which was confirmed by our simulation studies. We performed simulation studies to assess the accuracy of GLMM tree in detecting treatment-subgroup interactions and illustrated the application of GLMM tree on an existing dataset.

GLMM tree performed very well in recovering treatment-subgroup interactions, accurately recovering interactions in 90% of datasets with treatment-subgroup interactions. In contrast, GLM tree accurately recovered interactions in only 61% of these datasets. In the absence of treatment-subgroup interactions, GLMM tree erroneously detected subgroups in 4% of the datasets, whereas GLM tree erroneously detected subgroups in 33% of those datasets. Put differently, the Type-I error rate of GLMM tree very closely approximated the α level used for testing parameter stability, whereas the Type-I error rate of GLM tree clearly exceeded this value.

The better performance of GLMM tree was mostly observed when random effects in the datasets were sizable, and random intercepts were correlated with potential partitioning variables. In these instances, random effects gave rise to spurious subgroup detection (spurious splits) by GLM tree, both in datasets with and without treatment-subgroup interactions.

GLMM tree also provided better predictive accuracy than GLM tree, with an average correlation between true and predicted treatment differences of .94 for GLMM tree, and .88 for GLM tree. GLMM tree clearly outperformed GLM tree when random effects in the datasets were sizable, differences in treatment effects across terminal nodes were relatively small (i.e., Cohen's $d = .5$), and/or sample size was small (i.e., $N < 1,000$). Such treatment-effect differences and sample sizes may be quite common, even in multi-center clinical trials, and GLMM tree may provide a helpful tool for subgroup detection in those instances.

As expected, when random effects were absent from the simulated datasets, GLM and GLMM tree yielded very similar predictive accuracy. This indicates that GLMM tree can be applied whenever cluster-specific random effects are expected: In the absence of random effects, GLM tree and GLMM tree are expected to perform about equally well and in the presence of random effects GLMM tree is expected to outperform GLM tree.

Compared to treatment-interaction detection by means of GLMMs with pre-specified interaction effects, GLMM trees provided similar predictive accuracy, on average. When interactions were purely continuous, GLMMs outperformed GLMM trees, and when interactions were purely piecewise, GLMM trees outperformed GLMMs. However, GLMM tree may have a clear advantage when there are a large number of potential moderator variables (i.e., > 5). With 10 potential moderator variables, t-tests of GLMMs were unable to identify the true moderator variables, whereas the number of potential moderator variables did not influence performance of GLMM tree. Therefore, our results indicate that GLMM trees are better suited than GLMMs for exploratory analyses, in which moderator variables need to be selected from a larger number of covariates. Furthermore, the number of terms in a GLMM increases quadratically with the number of potential moderator variables, yielding complex predictive models. The trees in our simulations were limited to a maximum number of 8 terminal nodes, and may therefore be much easier to use for prediction in practical decision-making contexts.

These findings are encouraging for the use of GLMM tree in the detection of treatment-subgroup interactions in datasets with clustered structures. However, it should be noted that the simulations show that GLMM tree performs very well, if the model is correctly specified. That is, if there are subgroups with respect to the variables specified as potential partitioning variables and these subgroups have different values for the parameters of the GLM, then GLMM tree will accurately recover those subgroups. However, misspecification of the model will negatively affect performance. The most important source of misspecification would be the omission of relevant variables, either in the GLM or as partitioning variables. When relevant variables are omitted, GLMM tree can only approximate the true subgroups using the specified variables. Another source of misspecification would be the inclusion of irrelevant variables. Our simulations indicate that the performance of GLMM tree was not negatively affected by the number of potential moderator variables specified, but the power to detect subgroups may still be reduced with larger numbers of potential partitioning variables. Including irrelevant variables in the GLM may also negatively affect performance, although we have not assessed this in our simulations.

In the Application, we found GLMM tree to provide results that were more easily interpretable, and also more accurate than a GLM tree without random effects. In addition, to judge clinical relevance of the findings, we calculated node-specific effect sizes, as would often be done in RCTs or meta-analysis. Although we have limited ourselves to calculating Cohen's d in the current paper, equivalent values of the success rate difference or the number needed to treat can be calculated, but this would involve additional distributional assumptions (Kraemer & Kupfer, 2006). Node-specific effect sizes can also be used to prune trees, when a researcher prefers to have a final tree which is based on statistical as well as clinical significance. A topic for further research would be the development of splitting procedures based on effect sizes, as this would allow for taking into account clinical significant in the tree-growing process.

As discussed in the Introduction, several tree-based methods for treatment-subgroup interaction detection are available. These methods have different objectives, and there is not yet an agreed-upon single best method. In a simulation study, Sies and Van Mechelen (2016) found the method of Zhang, Tsiatis, Davidian, et al. (2012) to perform best, followed by model-based recursive partitioning. However, the method of Zhang et al. performed worst under some conditions of the simulation study in terms of the Type I error rate. Further research comparing tree-based methods for treatment-subgroup interaction detection is needed, especially for clustered datasets, as our simulations were limited to GLM and GLMM-based MOB.

Furthermore, it should be stressed that tree-based methods are exploratory tools. They can be used to detect predictors, interactions and non-linear effects in a data-driven way, but users should take the exploratory nature of such analyses into account. The resulting trees are potentially unstable, and stability of the results should be assessed, preferably in a dataset not used for training, or by multi-fold cross-validation or resampling techniques. In the Application, we have shown how the stability of splitting variable selection can be assessed using resampling techniques.

In conclusion, GLMM tree provided highly accurate recovery of treatment-subgroup interactions and predictions of treatment effect differences, both in the presence and absence of cluster-specific random effects. Therefore, GLMM tree is a promising algorithm for the detection of treatment-subgroup interactions in datasets with a clustered structure, like for example in multi-center trials or individual-level patient data meta-analyses.

References

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). *Fitting linear mixed-effects models using lme4* (No. 1).

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees.* New

York: Wadsworth.

Bryk, A. S., & Raudenbush, S. W. (1992). *Hierarchical linear models: Applications and data analysis methods.* Newbury Park, CA: Sage.

Cohen, J. (1992). A power primer. *Psychological Bulletin*, *112*(1), 155–159.

Cooper, H., & Patall, E. A. (2009). The relative benefits of meta-analysis conducted with individual participant data versus aggregated data. *Psychological Methods*, *14*(2), 165.

Cuijpers, P., Weitz, E., Twisk, J., Kuehner, C., Cristea, I., David, D., et al. (2014). Gender as predictor and moderator of outcome in cognitive behavior therapy and pharmacotherapy for adult depression: An "individual-patients data" meta-analysis. *Depression and Anxiety*, *31*(11), 941–951.

Doove, L. L., Dusseldorp, E., Van Deun, K., & Van Mechelen, I. (2014). A comparison of five recursive partitioning methods to find person subgroups involved in meaningful treatment-subgroup interactions. *Advances in Data Analysis and Classification*, *8*, 403–425.

Driessen, E., Smits, N., Dekker, J., Peen, J., Don, F. J., Kool, S., et al. (2016). Differential efficacy of cognitive behavioral therapy and psychodynamic therapy for major depression: A study of prescriptive factors. *Psychological Medicine*, *46*(4), 731–744.

Dusseldorp, E., Doove, L., & Van Mechelen, I. (2015). Quint: An R package for the identification of subgroups of clients who differ in which treatment alternative is best for them. *Behavior Research Methods*, 1–14.

Dusseldorp, E., & Meulman, J. J. (2004). The regression trunk approach to discover treatment covariate interaction. *Psychometrika*, *69*(3), 355–374.

Dusseldorp, E., & Van Mechelen, I. (2014). Qualitative interaction trees: A tool to identify qualitative treatment-subgroup interactions. *Statistics in Medicine*, *33*(2), 219–237.

Fokkema, M., & Zeileis, A. (2016). *glmertree: Generalized linear mixed model trees.* (R package version 0.1-1)

Foster, J. C., Taylor, J. M. G., & Ruberg, S. J. (2011). Subgroup identification from randomized clinical trial data. *Statistics in Medicine*, *30*(24), 2867–2880.

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

Hamilton, M. (1960). A rating scale for depression. *Journal of Neurology, Neurosurgery and Psychiatry*, *23*(1), 56.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd ed.). New York: Springer.

Higgins, J., Whitehead, A., Turner, R. M., Omar, R. Z., & Thompson, S. G. (2001). Meta-analysis of continuous outcome data from individual patients. *Statistics in Medicine*, *20*(15), 2219–2241.

Hothorn, T., & Zeileis, A. (2015, December). partykit: A modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research*, *16*, 3905–3909.

Koopman, L., Heijden, G. J. M. G. Van der, Glasziou, P. P., Grobbee, D. E., & Rovers, M. M. (2007). A systematic review of analytical methods used to study subgroups in (individual patient data) meta-analyses. *Journal of Clinical Epidemiology*, *60*(10).

Kraemer, H. C., Frank, E., & Kupfer, D. J. (2006). Moderators of treatment outcomes: Clinical, research, and policy importance. *Journal of the American Medical Association*, *296*(10), 1286–1289.

Kraemer, H. C., & Kupfer, D. J. (2006). Size of treatment effects and their importance to clinical research and practice. *Biological Psychiatry*, *59*(11), 990–996.

Lipkovich, I., Dmitrienko, A., Denne, J., & Enas, G. (2011). Subgroup identification based on differential effect search – A recursive partitioning method for establishing response to treatment in patient subpopulations. *Statistics in Medicine*, *30*(21), 2601–2621.

Martin, D. (2015). *Efficiently exploring multilevel data with recursive partitioning.* Unpublished doctoral dissertation, University of Virginia.

Merkle, E. C., & Zeileis, A. (2013). Tests of measurement invariance without subgroups: A generalization of classical methods. *Psychometrika*, *78*(1), 59–82.

Noortgate, W. Van den, Opdenakker, M. C., & Onghena, P. (2005). The effects of ignoring a level in multilevel analysis. *School Effectiveness and School Improvement*, *16*(3), 281-203.

Philipp, M., Zeileis, A., & Strobl, C. (2016). A toolkit for stability assessment of tree-based learners. In A. Colubi, A. Blanco, & C. Gatu (Eds.), *Proceedings of COMPSTAT 2016 – 22nd international conference on computational statistics* (pp. 315–325). Oviedo: The International Statistical Institute/International Association for Statistical Computing.

R Core Team. (2016). *R: A language and environment for statistical computing.* Vienna, Austria.

Seibold, H., Zeileis, A., & Hothorn, T. (2016). Model-based recursive partitioning for subgroup analyses. *International Journal of Biostatistics*, *12*(1), 45–63.

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.

Sies, A., & Van Mechelen, I. (2016). *Comparing four methods for estimating tree-based treatment regimes.* (Submitted)

Strobl, C., Malley, J., & Tutz, G. (2009). An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychological Methods*, *14*(4), 323.

Su, X., Tsai, C.-L., Wang, H., Nickerson, D. M., & Li, B. (2009). Subgroup analysis via recursive partitioning. *The Journal of Machine Learning Research*, *10*, 141–158.

Zeileis, A. (2005). A unified approach to structural change tests based on ML scores, F statistics, and OLS residuals. *Econometric Reviews*, *24*(4), 445–466.

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.

Zhang, B., Tsiatis, A. A., Davidian, M., Zhang, M., & Laber, E. (2012). Estimating optimal treatment regimes from a classification perspective. *Stat*, *1*(1), 103–114.

Zhang, B., Tsiatis, A. A., Laber, E. B., & Davidian, M. (2012). A robust method for estimating optimal treatment regimes. *Biometrics*, *68*(4), 1010–1018.