Detection of Treatment-Subgroup Interactions in Clustered Datasets: Combining Model-Based Recursive Partitioning and Random-Effects Estimation

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. Several tree-based algorithms have been developed for the detection of such treatment-subgroup interactions. In many instances, however, datasets may have a clustered structure, where observations are clustered within, for example, research centers, studies or persons. In the current paper we propose a new algorithm, GLMMtree, that allows for detection of treatment-subgroup interactions, as well as estimation of cluster-specific random effects. The algorithm uses model-based recursive partitioning (MOB) to detect treatment-subgroup interactions, and a linear mixed-effects model for estimation of random-effects parameters. In a simulation study, we evaluate the performance of GLMMtree and compare it with that of MOB trees without random-effects estimation. In datasets without treatment-subgroup interactions, GLMMtree was found to have a much lower Type I error rate than MOB trees without random effects (4 and 33%, respectively). Furthermore, in datasets with treatment-subgroup interactions, GLMMtree recovered the true treatment subgroups much more often than MOB without random effects (90% and 61% of the datasets, respectively). Also, GLMMtree predicted treatment outcome differences more accurate than MOB trees without random effects (average accuracy of .94 and .88, respectively). We illustrate the application of GLMMtree on a patient-level dataset of a meta-analysis on the effects of psycho- and pharmacotherapy for depression. We conclude that GLMMtree is a promising algorithm for the detection of treatment-subgroup interactions in clustered datasets, and discuss directions for future research.

Introduction

In medicine-efficacy research, the one-size-fits-all paradigm is slowly losing ground, and personalized medicine is becoming increasingly important. Personalized 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). In most cases, 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 a-priori. This may hamper identification of which treatments work 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, hypotheses and that are specifically directed at the detection of treatment interactions.

Tree-based methods are such hypothesis-generating methods, as they can automatically detect subgroups which differ on the expected outcomes for one or more treatments. Due to their flexibility, tree-based methods are preeminently suited to the detection of treatment-subgroup interactions: they can handle many potential predictor variables at once, and can automatically detect (higher order) interactions between predictor variables. Several promising tree-based algorithms and software packages have been developed to assist in the detection of treatment-subgroup interactions (e.g., Dusseldorp & Van Mechelen, 2014; Dusseldorp & Meulman, 2004; Su, Tsai, Wang, Nickerson, & Li, 2009; Foster, Taylor, & Ruberg, 2011; Lipkovich, Dmitrienko, Denne, & Enas, 2011; Zeileis, Hothorn, & Hornik, 2008; see Doove et al., 2014 for an overview). Of these tree-based methods, model-based recursive partitioning (MOB; Zeileis et al., 2008) may be the most flexible in detecting treatment-subgroup interactions, as it offers a very generic data-analytic framework for detecting partitions in a dataset, with different model parameter estimates. The recursive partitioning in MOB can be based on a broad class of parametric models that can be fitted using M-type estimators (Zeileis et al., 2008), the most well-known example being the generalized linear model (GLM). Earlier, MOB has been successfully applied by Driessen et al. (2014) in the detection of subgroups with differential treatment outcomes for two different psychotherapies.

However, none of the aforementioned tree-based algorithms allow for taking into account the clustered structure of datasets. In many cases, researchers may want to detect treatment-subgroup interactions in datasets with a clustered structure (e.g., Koopman, Van der Heijden, Glasziou, Grobbee, & Rovers, 2007). For example, in individual-level patient data meta-analyses, where datasets of multiple trials evaluating the effects of the same treatments are pooled. In such analyses, the clustered structure of the dataset should be taken into account by including study-specific effects in the model, prompting the need for modeling random effects (e.g., Friedenreich, 1993; DerSimonian & Laird, 1986; Higgins, Whitehead, Turner, Omar, & Thompson, 2001). Likewise, longitudinal datasets, and datasets from multi-center trials also require modeling of random effects. Ignoring the clustered structure of datasets may lead to biased inference, due to underestimated standard errors (e.g., Bryk & Raudenbush, 1992; Hox, 1998; Van den Noortgate, Opdenakker, & Onghena, 2005). When the interest is in subgroup detection, ignoring random effects on the outcome variable may result in the detection of spurious subgroups (e.g., Sela & Simonoff, 2012).

In the current paper, we present a tree-based algorithm for treatment-subgroup interaction detection, which takes the clustered nature of datasets into account. The algorithm combines MOB with the estimation of random effects, thus allowing for the detection of treatment-subgroup interactions, as well as accounting for variation between clusters (e.g., trials). In what follows, we first discuss the existing frameworks for estimating treatment effects: the GLM, model-based recursive partitioning of the GLM, and the generalized linear mixed-effects model (GLMM). Then, we present a new algorithm, which combines model-based recursive partitioning and random-effects estimation: GLMMtree. In a simulation study, we evaluate the comparative accuracy of the new algorithm. Finally, we apply GLMMtree to an existing dataset on the effects of treatments for depression, to illustrate the application of the algorithm.

General modeling framework

GLM

In a clinical trial, where the outcomes of two or more treatments are compared, an overall GLM may be used to estimate treatment effects. The goal is to estimate a model for predicting the value of treatment outcome y for observation $i = 1, \ldots, N$ (an overview of notation used is provided in the Appendix):

$$g(\mu_i) = g(E(y_i)) = x_i\beta \tag{1}$$

Where y_i is the value of the linear predictor of the response variable for observation i, and g is the link function, characterizing the relationship between the linear predictor and the mean of the response distribution function. In case of a continuous response variable, g is the identity function. Further, x_i is a vector of fixed-effects predictor variable values for observation i, of which the first element takes a value of 1 for the intercept, and the second element takes the value of a dummy indicator for treatment type. β is a vector of fixed-effects regression coefficients, containing the intercept, which is the mean value of the linear predictor in the first treatment group, and the slope, which is the difference in mean values of the linear predictor between the first and second treatment groups.

To keep notation and examples simple, we assume x_i and β to have length 2, That is, the effects of only two treatment conditions are estimated and no additional covariates are included in the linear model. However, additional treatment conditions and covariates can easily be included.

An example of such a GLM for a continuous outcome variable is graphically represented in Figure 1. The example is based on simulated data. The boxplots in Figure 1 show the distribution of the outcome variable (posttreatment depression score) among 150 participants, who were randomly assigned to treatments 1 and 2. Little overall difference between the outcomes of both treatments is suggested, as the slope of the regression line is nearly zero. We shall see that this does not necessarily mean that posttreatment depression score and treatment type are unrelated, as the effect of treatment may be moderated by other variables. Conditional on other variables (i.e., subgroup indicators), the relationship between x and y may vary in strength and/or direction (c.f., Simpson's paradox; Simpson, 1951).

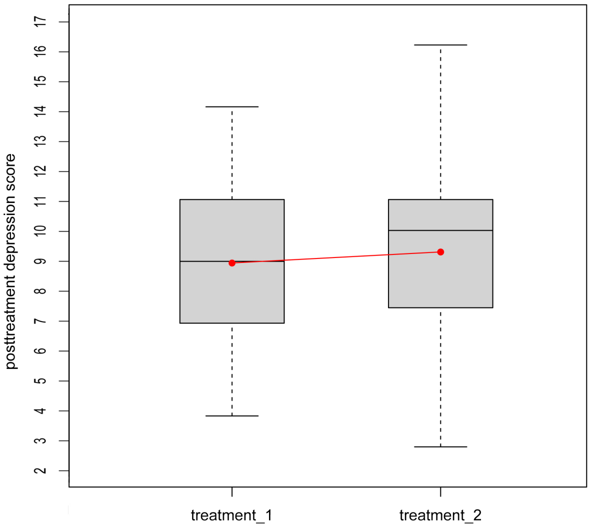

Figure 1. Example of a linear fixed-effects model for treatment outcomes (N=150). The dot for treatment 1 represents the first, and the slope of the regression line represents the second element of β.

Model-based recursive partitioning

The rationale behind MOB is that a global model for all observations, like that in Equation 1, 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. Whereas CART trees have constant fits in the terminal nodes, MOB trees have parametric models with one or more predictor variables in their terminal nodes.

To find partitions and better-fitting local linear models, the MOB algorithm tests for parameter instability. 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 additional covariates. To find 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 subsamples.

More specifically, in step (2), to test for parameter instability, the so-called *scores* are computed, using the score function. The expected value of the scores over all observations in a dataset is zero, by definition. Under the null hypothesis of parameter stability, the scores do not systematically deviate from the expected value of zero, when the observations are ordered by the values of a potential partitioning variable U_k. To statistically test whether there are systematic deviations of the scores 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 has 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). In step (3), a binary 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 residual sum of squares 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.

Due to the binary recursive nature of MOB, the resulting partitions can be represented as a tree. If the partitioning is based on the GLM, the result is a GLMtree, which has a local fixed-effects regression model in every jth (where $j = 1, ..., J$) terminal node of the tree. As a result, in the GLMtree model, the value for β depends on terminal node j in which observation i 'falls':

$$g(\mu_i) = g(E(y_i)) = x_i \beta_j \tag{2}$$

Figure 2 provides an example of the GLMtree model in Equation 2, based on the same data as was used for Figure 1. By using four additional covariates (anxiety questionnaire score, duration of depressive symptoms at baseline, age), MOB partitioned the observations into four subgroups, each with a different estimate for β_j. The leftmost subgroup in Figure 2 represents observations with low duration and low anxiety, for which treatment 1 is much more beneficial then treatment 2 (i.e., lower posttreatment depression scores). The rightmost subgroup represents observations with longer duration of depressive symptoms, for whom treatment 2 is much more beneficial than treatment 1. The two terminal nodes in the middle indicate that for patients with low duration, treatment 1 is only beneficial to those with moderate levels of anxiety (i.e., a values > 10 and ≤ 12). Age did not have an effect on treatment outcome, and therefore does not appear as a splitting variable in the tree.

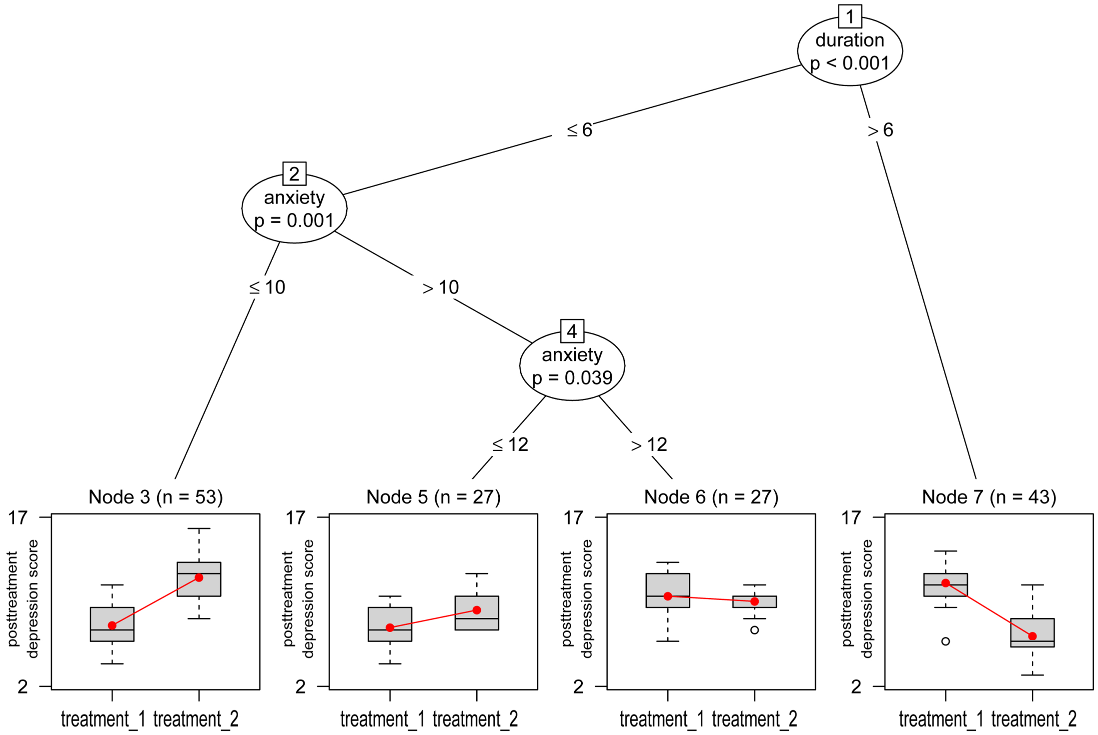

Figure 2. Example of tree representation of model-based recursive partitions, based on the same data as Figure 1. Three additional covariates (anxiety questionnaire score, duration of depressive symptoms at baseline in months and age) were used as potential splitting variables.

GLMM

When a dataset contains observations from multiple clusters (i.e., trials, research centers, or individuals in longitudinal datasets), the GLM in Equation 1 can be extended to include cluster-specific, or random effects, and the model becomes a GLMM:

$$g(\mu_i) = g(E(y_i)) = x_i\beta + z_i b_m \tag{3}$$

Where z_i is a vector of random-effects predictor variables values for observation i, and b_m is the vector of random-effects regression coefficients in cluster m, of which observation i is part. Within the linear mixed-effects model, it is assumed that values of b_m are normally distributed, with mean zero and variance $\sigma^2_{b_m}$. For simplicity, we assume z_i and b_m to have length 1 in the current paper; that is, only cluster-specific intercepts are included in the models. However, random-effects covariates can easily be included. The parameters of the GLMM can be estimated with, for example, maximum likelihood (ML) and restricted ML (REML), as described in for example (Bryk & Raudenbush, 1992). Note that for prediction of the treatment outcome for a new observation from a new cluster, only the fixed-effects part of Equation 3 would be used.

Combining model-based recursive partitioning and random-effects estimation

As noted earlier, ordinary GLM(M)s are not well suited for the detection of treatment-subgroup interactions, whereas the MOB algorithm is, but does not allow for estimation of random effects. Therefore, we propose the GLMMtree, which combines the GLMM in Equation 3 with the GLMtree in Equation 2:

$$g(\mu_i) = g(E(y_i)) = x_i\beta_j + z_i b_m \tag{4}$$

To estimate the partitions, and values of β_j and b_m for this model, we take an iterative approach, alternating between (1) assuming the random-effects coefficients b_m known, allowing for partitioning the datasets and estimating the corresponding β_j values; and (2) assuming the partition and corresponding β_j values known, allowing for estimation of the random-effects coefficients b_m.

In Figure 3, a schematic representation of the GLMMtree algorithm is presented. The algorithm initializes by setting all values b_m to 0, since the random-effects (and also the fixed-effects) parts are initially unknown. In every iteration, the GLMtree (i.e., the partition and corresponding fixed-effects coefficients β_j) and random-effects coefficients b_m are re-estimated. The GLMtree is estimated, given the estimated b_m values from the last iteration, and the b_m values are estimated, given the estimated GLMtree 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 3. A similar approach has been taken by Hajjem, Bellavance, and Larocque (2011) and Sela and Simonoff (2012), who added random-effects estimation to CART trees with constant fits, instead of linear models, in the terminal nodes.

```
Algorithm GLMMtree
    Step 0: Initialize by setting all values $b_m$ to 0.
    Step 1: Given the current $b_m$ values, partition dataset and estimate corresponding values of $\beta_j$.
    Step 2: Given the current partition and corresponding $\beta_j$ values, estimate $b_m$ values.
    Step 3: Repeat steps 1 and 2 until convergence.
```

Figure 3. Description of the GLMMtree algorithm

In what follows, we present a simulation study in which we assess the performance of GLMMtree in recovering treatment-subgroup interactions, and predicting differences between the outcomes of two treatments, for continuous outcomes. In addition, we will compare the performance of GLMMtree with that of GLMtree. In the simulation study, the main interest will be in the effects of sample size, and the presence and magnitude of treatment-subgroup interactions and random effects, but other parameters will be varied, as well.

For GLMMtree, we expect the accuracy of recovered trees and predictions to improve with increasing sample size, and magnitude of the differences in treatment outcomes. For GLMtree, we have the same expectation, when random effects are absent; that is, when the variance of the random coefficients is zero, we expect GLMtree and GLMMtree to perform equally well. When random effects are present, we expect GLMMtree to perform better than GLMtree, and more so when the variance of random-effects coefficients increases.

Simulation: Method

Software

`R` (R Core Team, 2014) was used for generation and analysis of all datasets. Two additional R packages were used: `partykit` (Hothorn & Zeileis, 2014; Zeileis et al., 2008) for the estimation of GLMtrees, and `lme4` (Bates, Maechler, & Bolker, 2012) for the estimation of random-effects coefficients. For all functions, default settings were used, with exception of maximum tree depth. Maximum tree depth was set to four (i.e., maximum of eight terminal nodes) for all trees, as this yields a model and graphical representation which is easy to interpret.

Estimation of GLM- and GLMMtrees. For estimating GLMtrees, the `glmtree` function from the `partykit` package was used. The `glmtree` function builds a generalized linear model tree: a model-based recursive partition based on a GLM (Equation 2).

For estimating GLMMtrees, we implemented the algorithm as described in Figure 3 in a function that iterates between (1) estimation of a linear model tree using the `glmtree` function, and (2) estimation of the random-effects coefficients b_m using the `glmer` function. Convergence of `GLMMtree` is monitored using the log-likelihood value of the generalized linear mixed-effects model estimated in step (2). When the difference in the log-likelihoods of two consecutive iterations is less than a prespecified value (.001 by default), `GLMMtree` has converged.

Simulation design

Datasets with treatment-subgroup interactions. For generating datasets with treatment-subgroup interactions, we used a treatment-subgroup interaction design from Dusseldorp and Van Mechelen (2014), which is also 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 their values on the partitioning variables U_2, and U_1 or U_5. In other words, U_1, U_2 and U_5 are true partitioning variables, whereas the other potential partitioning variables (U_3, U_4, U_6 through U_{15}) are noise variables.

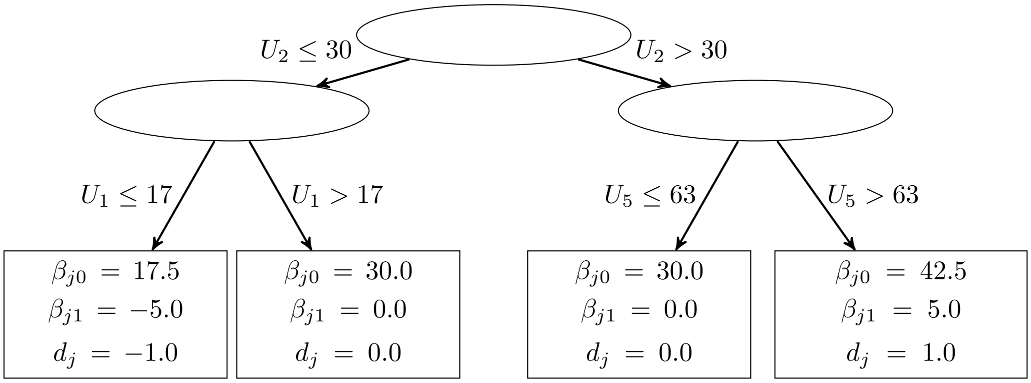

Figure 4. data-generating model for treatment-subgroup interactions. d denotes the standardized mean difference between the outcomes of treatment 1 and 2 (i.e., $\beta_j 1/\sigma_\epsilon$).

Datasets without treatment-subgroup interactions. For generating datasets without treatment-subgroup interactions, we used a design in which there is only a main effect of treatment in the population. In these datasets, the mean of the outcome variable was 30, and d and the first and second elements of β have the same value for all observations (i.e., when $d = 1.0$, the first and second elements of β are 37.5 and 42.5, respectively).

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

1. Three levels for the total number of observations: $N = 200$, $N = 500$, $N = 1000$.

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 are drawn: $\sigma_{b_m} = 0$, $\sigma_{b_m} = 5$, $\sigma_{b_m} = 10$.

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

7. Two different levels for β_1, the unstandardized mean difference in treatment outcomes, in subgroups with differential effects for treatment 1 ($X_1 = 0$) and treatment 2 ($X_1 = 1$). The levels for mean differences in subgroups with differential treatment effect were $|\beta_1| = 2.5$ (corresponding to a medium effect size, Cohen's $d = 0.5$; Cohen, 1992) and $|\beta_1| = 5.0$ (corresponding to a large effect size; Cohen's $d = 1.0$).

For each cell, 50 datasets with treatment-subgroup interactions were generated, resulting in 50 x 3 x 2 x 2 x 3 x 3 x 3 x 2 = 32,400 training datasets. For the datasets with main treatment effect only, the 6th parameter of the data-generating process had only two levels (b_m correlated with one of the U_k variables, and b_m not correlated with any of the U_k variables). Therefore, 50 x 3 x 2 x 2 x 3 x 3 x 2 x 2 = 21,600 datasets without treatment-subgroup interactions were generated.

Variable distributions. As in Dusseldorp and Van Mechelen (2014), all covariates U_1 through U_M were drawn from a multivariate normal distribution with μ_{U1}, μ_{U2}, μ_{U4}, and μ_{U5} fixed at 10, 30, -40 and 70, respectively. 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 the variables in U vary according to the third facet of the simulation design described above.

To generate the random error term ϵ, for every observation we drew a value from a normal distribution with $\mu_\epsilon = 0$ and $\sigma_\epsilon = 5$.

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 we drew a single b_m from a normal distribution with $\mu_{b_m} = 0$ and the value of σ_{b_m} given by the fifth facet of the simulation design. When b_m was correlated with one of the potential partitioning variables, the partitioning or non-partitioning covariate correlated with b_m was randomly selected.

To generate the 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 $p = .5$) as an indicator for treatment type.

Finally, outcome variable y was calculated according to the model in Equation 1.

Evaluation of performance

Tree size and accuracy. For every dataset, the accuracy and size of the GLM- and GLMMtrees were evaluated. We calculated the total number of nodes in every tree, and compared it with the true tree size. For datasets with a main treatment effect only, this allowed us to assess the accuracy in terms of Type I error: the probability that the dataset is erroneously partitioned. For datasets with treatment-subgroup interactions, this allowed us to assess the probability that the dataset is erroneously not partitioned, and the extent to which the algorithms may detect spurious subgroups.

For datasets with treatment-subgroup interactions, we assessed the accuracy of GLMtree and GLMMtree in recovering the true tree. An accurately recovered tree was defined as a tree with (1) the true tree size (i.e., total number of nodes is 7), (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 (σ_{U_k}) of the partitioning variable.

To detect predictors of the performance of both algorithms, GLMtrees were built using the `glmtree` function. Two trees were built for predicting the number of nodes in the trees generated by both algorithms: one for datasets without, and one for datasets with treatment-subgroup interactions. In these trees, the outcome variable is the number of nodes in a tree, the predictor variable for the linear model is algorithm type (GLMtree or GLMMtree), and the (potential) partitioning variables are the parameters of the data-generating process, described above. In addition, for datasets with treatment-subgroup interactions, a GLMtree was built for predicting the probability that the tree of treatment-subgroup interactions was accurately recovered. In this tree, the outcome variable is a binary indicator for whether the tree was accurately recovered (or not). Again, the predictor variable for the linear model is algorithm type (GLMtree or GLMMtree), and the (potential) partitioning variables are the parameters of the data-generating process.

Predictive accuracy. We evaluated predictive accuracy of GLM- and GLMMtrees by calculating correlations between the predicted and true treatment-effect differences (β_{j1}, Figure 4) for test observations. Note that this correlation was only assessed for datasets with treatment-subgroup interactions, as the true treatment differences have a constant value in datasets with a main treatment effect only.

Using the same data for training and evaluation of a model results in overly optimistic estimates of predictive accuracy (Hastie, Tibshirani, Friedman, Hastie, & Friedman, 2009). Therefore, GLM- and GLMMtrees were used for prediction of new observations in test datasets. These test datasets were generated from the same population as the training datasets. Because the cluster-specific intercepts b_m were randomly generated for training as well as test datasets, test observations were from 'new' clusters. As a result, a model without random effects was used for prediction with GLMMtree.

For every dataset, two correlation coefficients were calculated, representing the linear association between the true and predicted treatment differences: one for GLMtree, and one for GLMMtree. To detect predictors of predictive accuracy, a GLMtree was built. The outcome variable in this tree is the correlation between the true and predicted treatment differences. The predictor variable for the linear model is algorithm type (GLMtree or GLMMtree), and the (potential) partitioning variables are the parameters of the data-generating process, described above.

Simulation: Results

Tree size and accuracy in datasets with main treatment effect only

In Table 1, tree sizes for GLMtree and GLMMtree are presented for datasets with a main treatment effect only. Overall, smaller trees were created by GLMMtree: the average tree size was 1.09 (SD=0.44) for GLMMtree, and 2.02 (SD=1.68) for GLMtree. The estimated probability that a dataset was erroneously partitioned was very small for GLMMtree (.04; Table 1), and much larger for GLMtree (.33; Table 1).

Table 1: Tree size distributions for GLMtree and GLMMtree for datasets with a main treatment effect only.

	tree size						
	1	3	5	7	9	11	total
GLMMtree	20625	932	43	0	0	0	21,600
	(.96)	(.04)	(< .01)	(.00)	(.00)	(.00)	(1.00)
GLMtree	14501	4202	2013	802	79	3	21,600
	(.67)	(.20)	(.09)	(.04)	(< .01)	(< .01)	(1.00)

Note. Bracketed values are proportions. Tree sizes are expressed as the total number of nodes in a tree. A tree with a total of k nodes has has $(k+1)/2$ terminal nodes ; the true tree size in datasets with a main treatment effect only was 1.

A linear model tree (Figure 5) indicated that the main predictors of tree size in datasets without treatment-subgroup interactions were sample size and magnitude of σ_{b_m}. When random effects were absent (i.e., $\sigma_{b_m} = 0$), both GLMtree and GLMMtree tend to create trees of size 1. In the presence of random effects, GLMMtree also tends to create trees of size 1, but GLMtree created larger trees. For GLMtree, tree size increased with both sample size, and magnitude of σ_{b_m}.

Tree size in datasets with treatment-subgroup interactions

In datasets with treatment-subgroup interactions, GLMMtrees were also smaller than GLMtrees. For these datasets, the true tree size was 7 (4 terminal nodes and 3 inner nodes; Figure 4). The sizes of the GLM- and GLMMtrees for datasets with treatment-subgroup interactions are presented in Table 2. The average size of GLMMtrees was 7.15 (SD=0.61), and the average size of GLMtrees was 8.11 (SD=2.05). The estimated probability that a datasets was erroneously not partitioned was 0, for both GLM- and GLMMtree. However, Table 2 shows that a proportion of .91 of GLMMtrees matched the true tree size, whereas a proportion of only .64 of GLMtrees matched the true tree size (Table 2).

Table 2: Tree size distributions for GLMtree and GLMMtree for datasets with treatment-subgroup interactions.

	tree size							
	3	5	7	9	11	13	15	total
GLMMtree	3	227	29556	2472	89	3	0	32,400
	(< .01)	(< .01)	(.91)	(< .01)	(< .01)	(< .01)	(.00)	(1.00)
GLMtree	145	1002	20578	4443	4178	1665	389	32,400
	(< .01)	(.03)	(.64)	(.14)	(.13)	(.05)	(.01)	(1.00)

Note. Bracketed values are proportions. Tree sizes are expressed as the total number of nodes in the tree. A tree with a total of k nodes has has $(k + 1)/2$ terminal nodes; the true tree size in datasets with treatment-subgroup interactions was 7.

A linear model tree (Figure 6) indicated a three-way interaction between sample size, value of σ_{b_m}, and whether values of b_m are correlated with a potential partitioning variable. Overall, both algorithms created trees of size ≈ 7, with the following two exceptions:

When sample size is small (i.e., $N = 200$), the variance of b_m is large (i.e., $\sigma_{b_m} = 10$), and values of b_m are not correlated with one of the U_k variables (Figure 6). This was the case for 2,400 datasets (7.41%), in which mean tree size was 6.12 (SD=1.37) for GLMtree and 7.01 (SD=0.48) for GLMMtree. Because in these cases, GLMtree cannot account for the variance in y due to cluster-specific effects, it has difficulty detecting partitions when σ_{b_m} is large and sample size is low. However, the large variability for GLMtree indicates

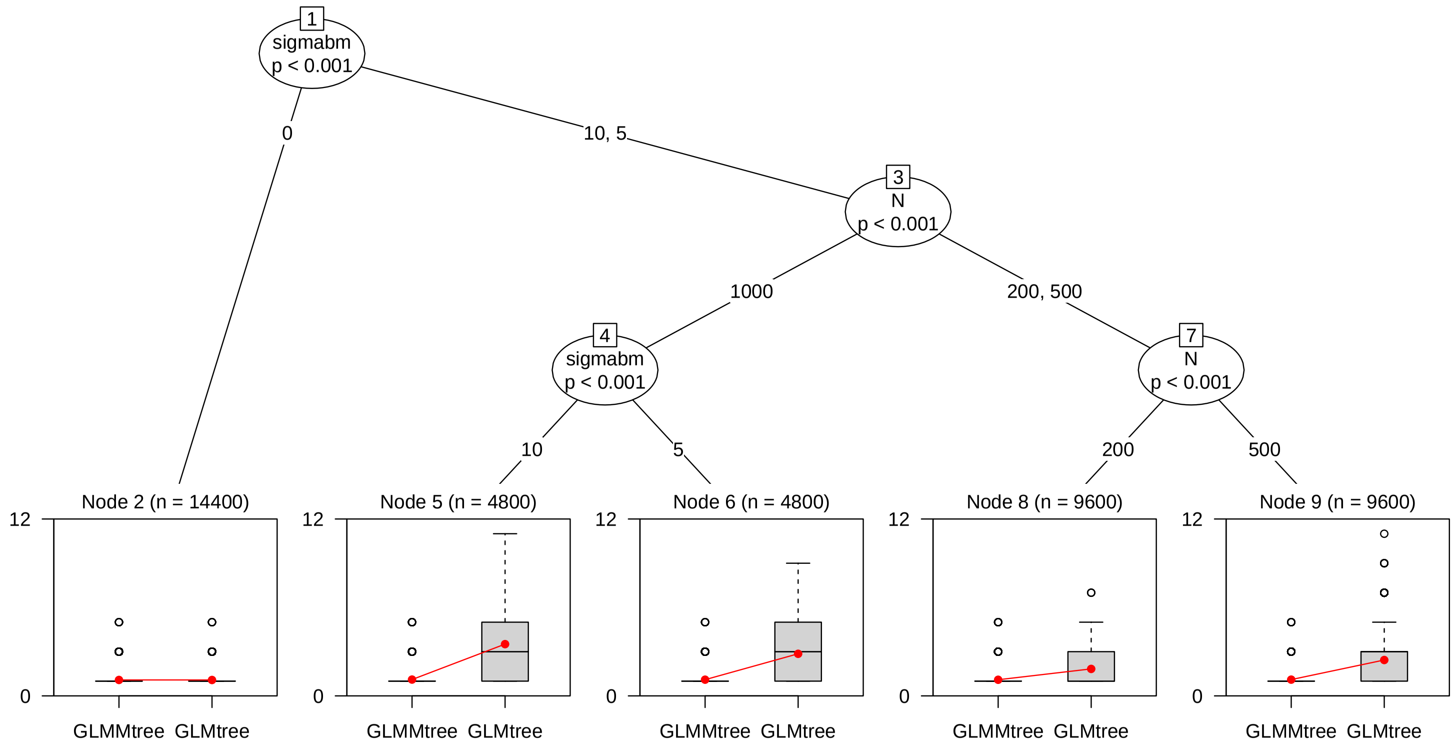

Figure 5. Linear model tree of the size of GLMM- and GLMtrees in datasets without treatment-subgroup interactions. The y-axes of the terminal nodes represent tree size (total number of nodes in a tree). Circles represent outliers (values below $Q_1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$). N = total sample size; sigmabm = σ_{b_m}.

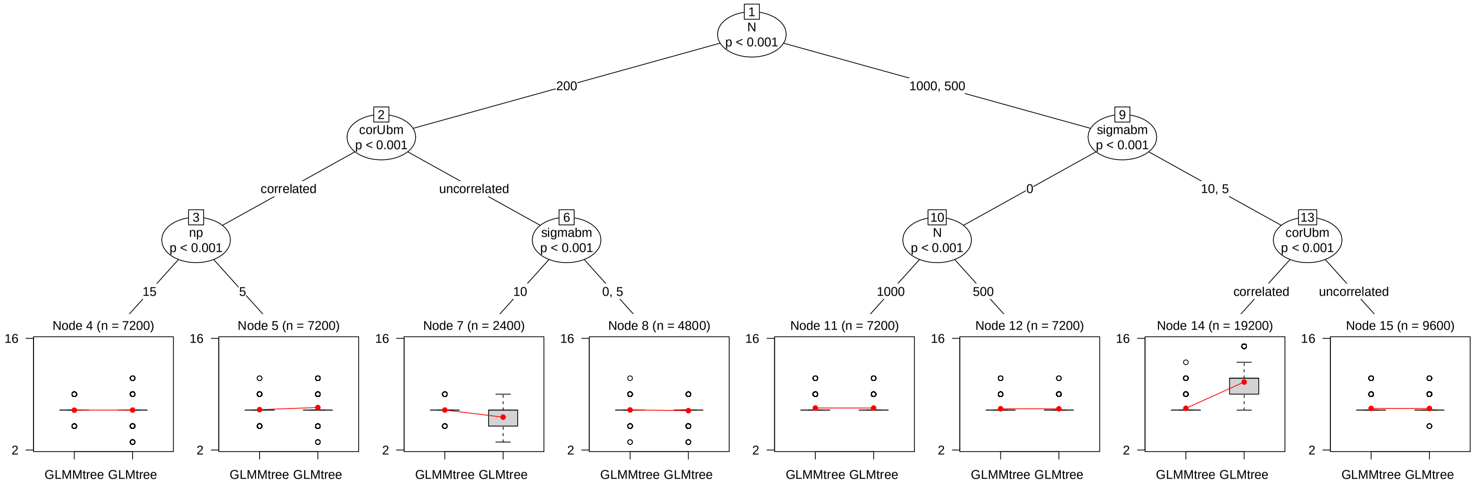

Figure 6. Linear model tree of tree sizes of GLMM- and GLMtrees in datasets with treatment-subgroup interactions. The y-axes of the terminal nodes represent tree size (total number of nodes in a tree). Circles represent outliers (values below $Q_1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$). N = total sample size; sigmabm = σ_{b_m}; corUbi = correlation between random intercepts and one of th e potential partitioning variables; np = number of potential partitioning variables.

that it may still create spurious splits.

When sample size is larger (i.e., $N = 500$ or $N = 1000$), when the variance of b_m is non-zero, and values of b_m are not correlated with one of the U_k variables (Figure 6). This was the case for 19,200 datasets (59.26%), in which mean tree size was 10.53 (SD=2.04) for GLMtree, and 7.23 (SD=0.66) for GLMMtree. Obviously, in these cases, GLMtree is more likely to create spurious splits. Because GLMMtree can more adequately deal with the additional variance caused by non-zero values of σ_{b_m}, the size of GLMMtrees seems not to be influenced much by values of σ_{b_m}.

Tree accuracy in datasets with treatment-subgroup interactions

To assess accuracy, we inspected the partitioning variables and values selected in every dataset. For the first split, both GLMtree and GLMMtree always selected the true partitioning variable (U_2). The true splitting value for U_2 was 30 (Figure 4), and the mean splitting value selected for the first split was 29.94 for both GLMMtree and GMLtree. However, GLMtree showed somewhat higher variability in recovering the splitting value than GLMMtree (SD=0.155 and SD=0.126, respectively).

GLMMtree performed well in recovering treatment-subgroup interactions, accurately recovering the tree in 90.19% of datasets. GLMtree accurately recovered the treatment-subgroup interactions in only 61.44% of datasets. A generalized linear model tree (Figure 7) indicated that performance of both algorithms was predicted by a three-way interaction between sample size, value of σ_{b_m}, and whether values of b_m are correlated with a potential partitioning variable. Overall, both algorithms recovered the true tree with about equal probabilities (i.e., all probabilities $> .85$), with three notable exceptions:

When $\sigma_{b_m} > 0$, and values of b_m are correlated with a potential partitioning variable, GLMMtree clearly outperformed GLMtree. When sample size was low (i.e., $N = 200$), the probability of accurately recovering the tree was .59 for GLMtree, and .91 for GLMMtree (Figure 7). With larger sample sizes (i.e., $N = 500$ or $1,000$), the probability of accurately recovering the tree was .13 for GLMtree, and .89 for GLMMtree (Figure 7).

When σ_{b_m} was non-zero, and values of b_m are not correlated with a potential partitioning variable, GLMMtree also outperformed GLMtree, but only when sample size was small. In these datasets, the probability of accurately recovering the tree was .72 for GLMtree and .92 for GLMMtree (Figure 7). When sample size was larger (i.e., $N = 500$ or $1,000$), the probability of accurately recovering the tree was .90 for both algorithms (Figure 7).

Predictive accuracy on test data

Overall, the treatment differences predicted by GLMMtree were closer to the true differences than the treatment differences predicted by GLMtree. The average correlation between the true and predicted treatment differences over all 32,400 datasets was .88

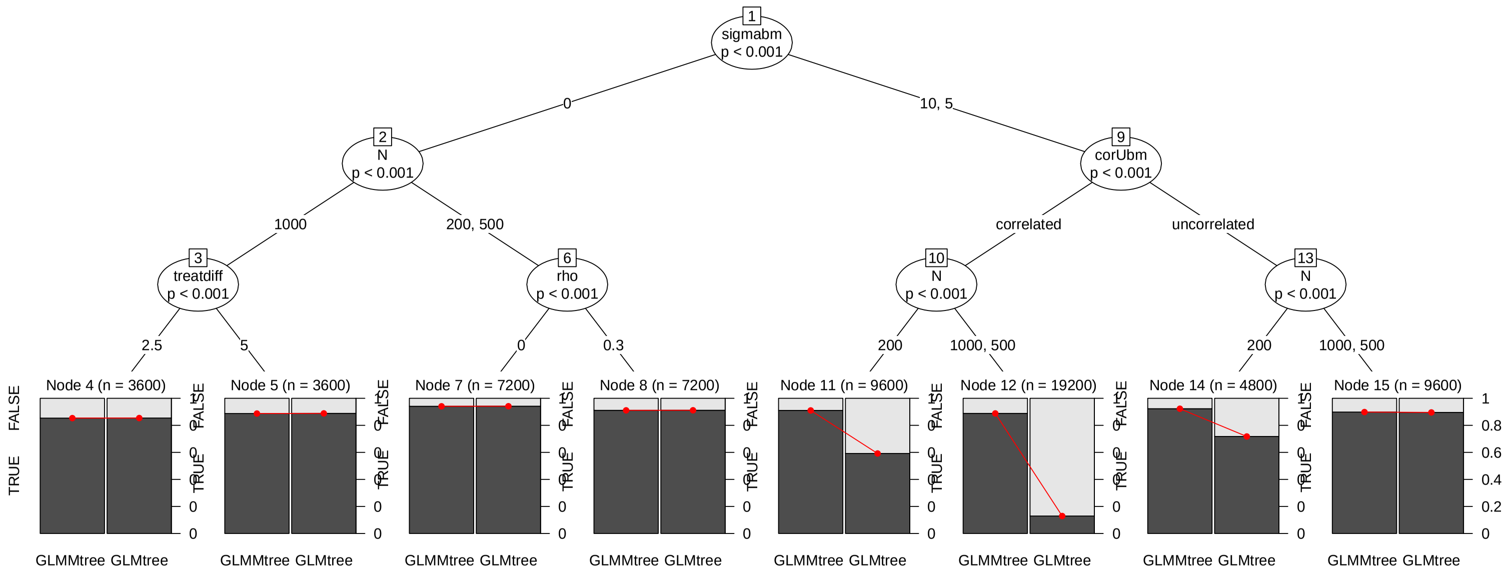

Figure 7. Generalized linear model tree of accurate recovery of treatment-subgroup interactions by GLMM- and GLMtree in datasets with treatment-subgroup interactions. Bar plots in the terminal nodes represent the proportion of datasets in which the treatment-subgroup interactions were accurately recovered. Circles represent outliers (values below $Q_1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$). N = total sample size; sigmabm = σ_{b_m}; corUbi = correlation between random intercepts and one of th e potential partitioning variables; rho = correlation between potential partitioning variables; treatdiff = β_{j1}, the unstandardized mean difference in treatment outcomes in subgroups with differential effects for treatment 1 and treatment 2.

(SD=0.20) for GLMtree, and .94 (SD=0.10) for GLMMtree.

The linear model tree depicting the relationship between the various data-generating parameters and predictive accuracy of GLMtree and GLMMtree, is presented in Figure 8. Figure 8 shows clear main effects of sample size N, treatment difference size, and value of σ_{b_m}. With larger sample sizes ($N = 500$ or $1,000$), both GLMtree and GLMMtree perform better than with smaller sample sizes ($N = 200$). With larger treatment differences ($d = 5$), the difference between the performance of GLMtree and GLMMtree becomes smaller, and the performance of GLMtree shows less variation. When the random intercepts are sampled from a population distribution with larger variance (i.e., $\sigma_{b_m} = 10$), the difference in accuracy between GLMMtree and GLMtree is more pronounced, than when there are no random intercept differences, or when these differences are small (i.e., $\sigma_{b_m} = 0$ or 5).

As the boxplots in Figure 8 show, for some simulated datasets, correlations between predicted and true differences were obtained that were clear outliers. It should be noted that low or negative correlations between the true and predicted treatment differences were much more often found for GLMtree than for GLMMtree: a correlation $< .40$ was obtained in 930 out of 32,400 datasets for GLMtree, and in 175 out of 32,400 datasets for GLMMtree.

Application to real data

Method

To illustrate the application and potential differences in the results of GLMtree and GLMMtree, we applied both algorithms to a dataset from a meta-analytic study of Cuijpers et al. (2014). This meta-analysis was based on individual-patient data from 14 RCTs, comparing the effects of psychotherapy (cognitive behavior 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 either predicted or moderated treatment outcomes. Further details on the dataset are provided 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 linear predictor was treatment type (0=CBT and 1=PHA). An indicator for study was used as the cluster indicator.

In RCTs, treatment effects are often estimated after controlling posttreatment values on the outcome measure for the linear effect of pretreatment values on the same measure. Therefore, we included the predictions of a linear regression of HAM-D posttreatment on HAM-D pretreatment scores, as an offset variable in all models. An offset variable is a linear

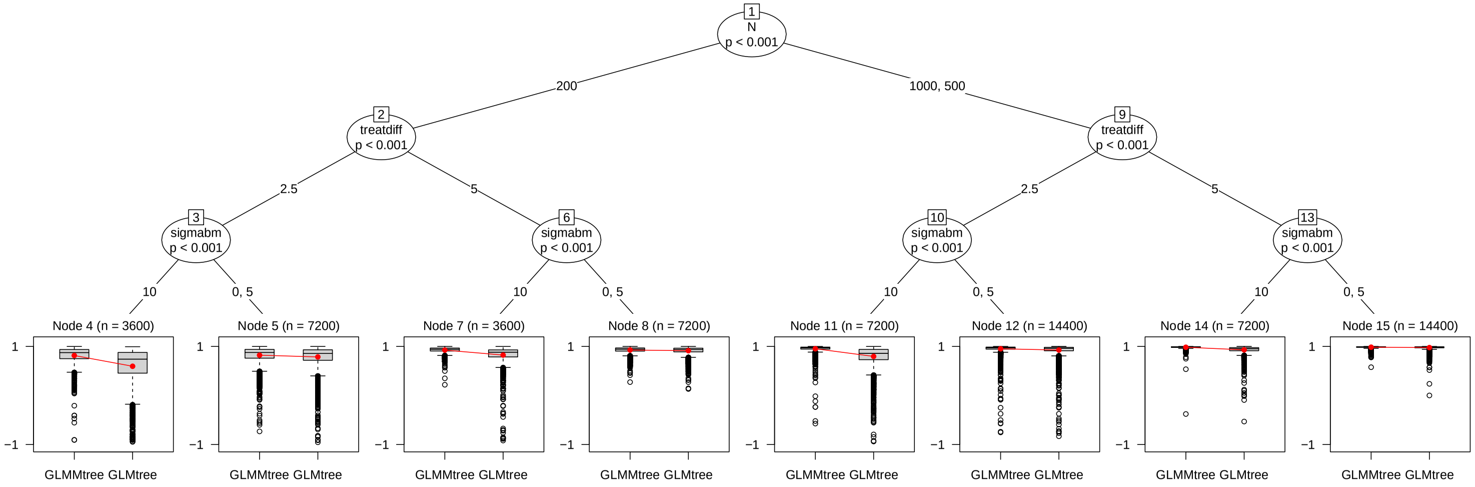

Figure 8. Linear model tree of correlations with true treatment differences for GLMMtrees and GLMtrees. The y-axes in the terminal nodes represent the correlation between the true and predicted treatment effect differences. Circles represent outliers (values below $Q_1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$). N = total sample size; treatdiff = β_{j1}, the unstandardized mean difference in treatment outcomes in subgroups with differential effects for treatment 1 and treatment 2; sigmabm = σ_{b_m}.

predictor with an a-priori determined coefficient of one. Including the linear regression predictions as an offset has the same effect as statistically controlling for the linear effects of a of pretreatment scores, as is often done in ANCOVA.

The `glmtree` function deals with missing data by listwise deletion. Therefore, we build the trees using data of a subset of 694 patients from 7 studies, as complete observations (i.e., observations with non-missing values for potential partitioning variables, and pre- and posttreatment HAM-D score) for these patients were available. Results of our analysis may therefore not be representative of the complete dataset of the meta-analysis by (Cuijpers et al., 2014).

Predictive accuracy of GLMtree and GLMMtree was assessed by calculating the average correlations between observed and predicted HAM-D scores, based on 50-fold cross validation.

Results

We applied GLMtree and GLMMtree to the dataset with complete observations. The resulting trees are presented in Figure 9 and 10. Note that the GLMtree in Figure 9 is also the tree that is created in the first iteration of the GLMMtree algorithm.

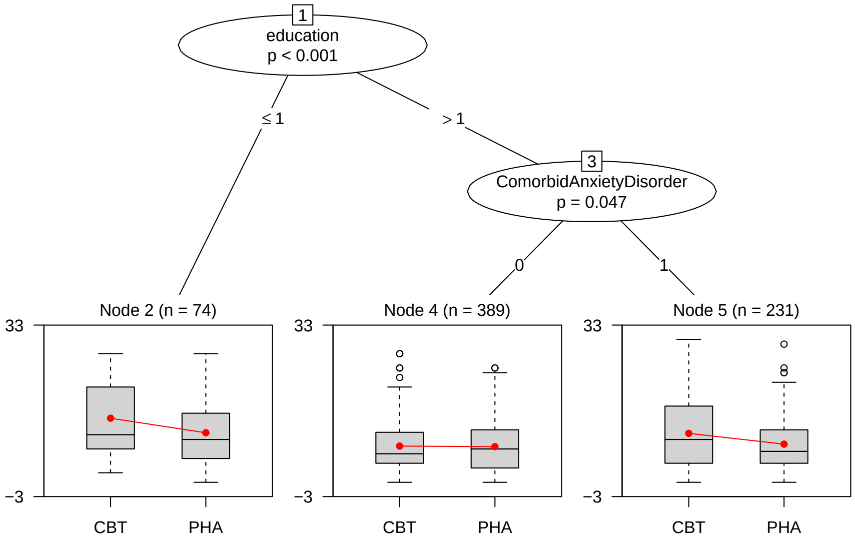

Figure 9. GLMtree for prediction of posttreatment total scores on the Hamilton Rating Scale for Depression (HAM-D). The y-axes of the boxplots represent posttreatment HAM-D scores, and the x-axes represent treatment levels: cognitive behavior therapy (CBT) vs. pharmacotherapy (PHA).

The GLMtree (Figure 9) 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. Terminal node 2 of Figure 9 indicates that for patients with a low level of education, antidepressant medication provides the greatest reduction in HAM-D scores. Terminal node 4 indicates that for patients with a higher level of education, and no comorbid anxiety disorder, the reduction in HAM-D scores is about the same for CBT and antidepressant mediation. Terminal node 5 indicates, that for patients with a higher level of education and a comorbid anxiety disorder, the reduction in HAM-D scores is greatest for pharmacotherapy.

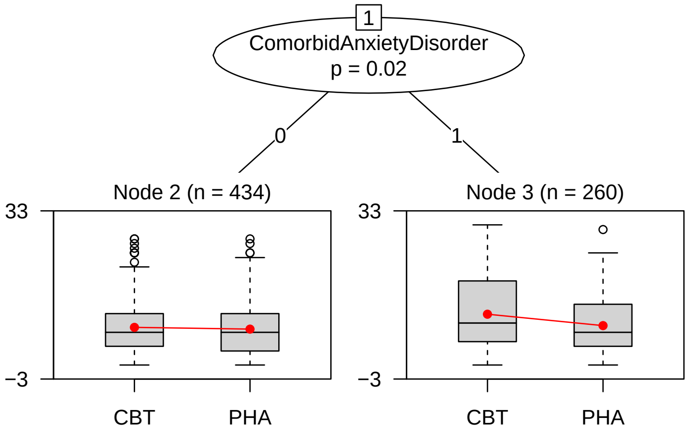

Figure 10. GLMMtree for prediction of posttreatment total scores on the Hamilton Rating Scale for Depression (HAM-D). The y-axes of the boxplots represent posttreatment HAM-D scores, and the x-axes represent treatment levels: cognitive behavior therapy (CBT) vs. pharmacotherapy (PHA).

By taking into account the study-specific intercepts, the final GLMMtree (Figure 10) indicates that the first split made by GLMtree is a spurious split. The GLMMtree selected only presence of a comorbid anxiety disorder as a partitioning variable. The terminal nodes of Figure 10 show only a single treatment-subgroup interaction: for patients without a comorbid anxiety disorder, CBT and antidepressant medication provide more or less the same reduction in HAM-D scores, whereas for patients with a comorbid anxiety disorder, antidepressant medication provides a greater reduction in HAM-D scores than CBT. The estimated variance of the random intercept term was 2.12, with an estimated intraclass correlation coefficient of .05.

Assessment of predictive accuracy by means of 50-fold cross validation showed that the GLMMtree had higher predictive accuracy than the GLMtree. The correlation between true and predicted posttreatment HAM-D total scores, averaged over the 50 folds, was .39 (SD=.20) for GLMMtree, and .31 (SD=.24) for GLMtree. This indicates that GLMMtree not only provided higher predictive accuracy, on average, but also had somewhat lower variability of predictive accuracy than GLMtree.

Discussion

The results of our simulation study show that GLMMtree performed very well in recovering treatment-subgroup interactions, by recovering the true tree structure in 90% of the simulated datasets with treatment-subgroup interactions. In the absence of treatment-subgroup interactions, GLMMtree erroneously detected subgroups in only 4% of the datasets. GLMtree performed less accurate than GLMMtree: in datasets with treatment-subgroup interactions, GLMtree recovered the true tree structure in 61% of the simulated datasets. In datasets without treatment-subgroup interactions, GLMtree erroneously detected subgroups in 33% of the datasets.

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

Also, predictive accuracy of GLMMtree was higher than that of GLMtree. The average correlation between the true treatment differences and those predicted by GLMMtree was .94. The average correlation between the true treatment differences and those predicted by GLMtree was .88. In terms of predictive accuracy, GLMMtree clearly outperformed GLMtree when random effects in the datasets were sizable, and random intercepts were correlated to potential partitioning variables.

As expected, when random effects were absent from the simulated datasets, GLMtree and GLMMtree showed high and equal predictive accuracy. This finding indicates that GLMMtree can be applied, whenever cluster-specific random effects are expected. In the absence of random effects, GLMtree and GLMMtree are expected to perform equally well, and in the presence of random effects, GLMMtree will outperform GLMtree. This is especially the case with large sample sizes ($N > 500$), as the increased power will likely cause GLMtree to create spurious splits in the presence of random effects.

Not surprisingly, for both algorithms, accuracy of predicted treatment differences was somewhat less when sample size was low (i.e., $N = 200$). Sample size influenced performance of GLMtree and GLMMtree similarly, suggesting that the larger number of estimated parameters for GLMMtree did adversely influence accuracy with low sample sizes. However, our simulation results do warrant caution for the detection of treatment-subgroup interactions or treatment moderators in small datasets (e.g., single RCTs), irrespective of the algorithm used.

Although these findings are encouraging for the use of GLMMtree in the detection of treatment-subgroup interactions in datasets with clustered structures, some limitations of our study and challenges for future research should be noted.

As noted earlier, simulations in the current study were confined to random-intercept models. GLMMtree allows for the estimation of random slopes as well, but estimation of random treatment effects is currently not possible, as treatment effects are estimated with local linear fixed-effects models. Our simulations also did not include models with multiple fixed-effects predictor variables in X. Multiple fixed-effects predictor variables can be easily included in GLMtree and GLMMtree, but it should be noted that the parameters corresponding to these variables will then be included in tests for parameter instability as well, which may be undesirable. For example, in RCTs, ANCOVA will often be used to control for the linear effects of pretreatment values on the treatment outcome variable. Whether or not such parameters should be included in parameter stability tests, or should be allowed to vary over partitions, should be decided by the researcher.

One challenge for further research is the development of parameter stability tests for random-effects parameters. In the current study, random-effects parameters were estimated globally, using all observations in the dataset, and fixed-effects parameters were estimated locally, using the observations in a single node. This would allow, for example, for estimation of random treatment effects, instead of fixed treatment effects.

A second challenge is the development of more adequate ways to deal with missing data in treatment-subgroup interaction detection. GLMtree, like most tree-based algorithms for treatment-subgroup interaction detection, handles missing data by listwise deletion. However, missing data commonly occurs in clinical trails, and listwise deletion is obviously not the preferred method for dealing with missing data (e.g., Wood, White, & Thompson, 2004).

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

References

Bates, D., Maechler, M., & Bolker, B. (2012). *lme4: Linear mixed-effects models using S4 classes.* Retrieved from `http://CRAN.R-project.org/package=lme4`

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

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

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

Cuijpers, P., Weitz, E., Twisk, J., Kuehner, C., Cristea, I., David, D., . . . Hollon, S. (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.

DerSimonian, R., & Laird, N. (1986). Meta-analysis in clinical trials. *Controlled Clinical Trials*, *7*(3), 177–188.

Doove, 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*, 1–23.

Driessen, E., Smits, N., Peen, J., Don, F., Kool, S., Westra, D., . . . Van, H. (2014). *Differential efficacy of cognitive behavioral therapy and psychodynamic therapy for major depression: a study of prescriptive factors.* Manuscript under review.

Dusseldorp, E., & Meulman, 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.

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

Friedenreich, C. (1993). Methods for pooled analyses of epidemiologic studies. *Epidemiology*, *4*(4), 295–302.

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

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

Hothorn, T., & Zeileis, A. (2014, March). *partykit: A modular toolkit for recursive partytioning in R* (Working Paper No. 2014-10). Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universität Innsbruck. Retrieved from `http://EconPapers.RePEc.org/RePEc:inn:wpaper:2014-10`

Hox, J. (1998). Multilevel modeling: When and why. In *Classification, data analysis, and data highways* (pp. 147–154). Springer.

Koopman, L., Van der Heijden, G., Glasziou, P., Grobbee, D., & Rovers, 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., Frank, E., & Kupfer, D. (2006). Moderators of treatment outcomes: clinical, research, and policy importance. *Journal of the American Medical Association*, *296*(10), 1286–1289.

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.

R Core Team. (2014). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `http://www.R-project.org/`

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

Simpson, E. (1951). The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society. Series B (Methodological)*, *13*, 238–241.

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

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

Wood, A., White, I., & Thompson, S. (2004). Are missing outcome data adequately handled? a review of published randomized controlled trials in major medical journals. *Clinical Trials*, *1*(4), 368–376.

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.

Appendix: Notation

$1, ..., i, ..., N$	observation number
$1, ..., j, ..., J$	terminal node number in a tree
$1, ..., k, ..., K$	partitioning variable number
$1, ..., m, ..., M$	cluster number
β_j	column vector of fixed-effects coefficients in terminal node j
b_m	column vector of random-effects coefficients in cluster m
d_j	$\beta_{j1}/\sigma_\epsilon$; effect size of predicted differences in treatment outcome y between treatments 1 and 2, in terminal node j
ϵ	deviation of observed treatment outcome y from its expected value
σ_{b_m}	square root of variance of b_m
σ_ϵ	square root of the variance of ϵ
U	(potential) partitioning variables
U_k	(potential) partitioning variable k
x_i	row vector of fixed-effects predictor variable values for observation i
y_i	treatment outcome for observation i
z_i	row vector of random-effects predictor variable values for observation i