Computational Statistics and Data Analysis 88 (2015) 53–74

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Unbiased regression trees for longitudinal and clustered data

Wei Fu, Jeffrey S. Simonoff*

New York University Leonard N. Stern School of Business, NY, USA

ARTICLE INFO

Article history:
Received 12 February 2014
Received in revised form 8 December 2014
Accepted 6 February 2015
Available online 14 February 2015

Keywords:
Clustered data
Longitudinal data
Mixed models
Regression tree

ABSTRACT

A new version of the RE–EM regression tree method for longitudinal and clustered data is presented. The RE–EM tree is a methodology that combines the structure of mixed effects models for longitudinal and clustered data with the flexibility of tree-based estimation methods. The RE–EM tree is less sensitive to parametric assumptions and provides improved predictive power compared to linear models with random effects and regression trees without random effects. The previously-suggested methodology used the CART tree algorithm for tree building, and therefore that RE–EM regression tree method inherits the tendency of CART to split on variables with more possible split points at the expense of those with fewer split points. A revised version of the RE–EM regression tree corrects for this bias by using the conditional inference tree as the underlying tree algorithm instead of CART. Simulation studies show that the new version is indeed unbiased, and has several improvements over the original RE–EM regression tree in terms of prediction accuracy and the ability to recover the correct tree structure.

1. Introduction

The regression tree is a nonparametric method for estimating a regression function. Assume the data set consists of a response variable y and one or more predictor (covariate) variables $\mathbf{X} = (X_1, X_2, \ldots, X_k)$. The regression tree algorithm splits the data set into subsets based on the values of its covariate variables $\mathbf{X}$. The process is repeated on each derived subset recursively until halted based on some stopping rule. One common approach is to split until the subset at a node has all the same value of the response variable y or predictor variable values $\mathbf{x}$, or the node has sample size less than a threshold determined beforehand. The tree is then "pruned back" to make it less complex. After the tree building is complete, the response sample mean $\bar{y}$ of each node serves as the predicted response value associated with the covariates' values in that node.

The earliest effort to extend regression tree methodology to longitudinal and clustered data was made by Segal (1992). Zhang (1998) extended the CART algorithm of Breiman et al. (1984) to the multiple binary response situation. De'Ath (2002) modified the CART algorithm to the multivariate response case and their algorithm, the multivariate regression tree (MRT), is implemented in the $\mathcal{R}$ package `mvpart`. All of these approaches share the restriction of not allowing time-varying covariates.

More recently, Hajjem et al. (2011) took a mixed-effects models approach and extended regression tree algorithms to the case of clustered data for continuous outcomes. The essential idea of his mixed-effects regression tree (MERT) algorithm is to dissociate the fixed from the random effects. They used CART as the standard regression tree to model the fixed effect and a node-invariant linear structure to model the random effects, and implement the algorithm within the framework of the EM algorithm.

* Correspondence to: 44 West 4th Street, New York, NY 10012-1126, USA. Tel.: +1 212 998 0452; fax: +1 212 995 4003.
E-mail addresses: wfu@stern.nyu.edu (W. Fu), jsimonof@stern.nyu.edu (J.S. Simonoff).

http://dx.doi.org/10.1016/j.csda.2015.02.004
0167-9473/© 2015 Elsevier B.V. All rights reserved.

Independently, Sela and Simonoff (2012) proposed a similar estimation method that uses a tree structure to estimate the fixed effects. Since neither the random effects nor the fixed effects are known, they alternate between estimating the regression tree, assuming that estimates of the random effects are correct, and estimating the random effects, assuming that the regression tree for the fixed effects is correct. Such a procedure is reminiscent of the EM algorithm and hence they termed it a Random Effects/EM tree, or RE–EM tree. As is true for MERT, the algorithm is implemented using CART as the underlying regression tree. Both MERT and RE–EM can easily accommodate time-varying covariates.

Despite being the most popular regression tree method, CART has two fundamental problems: overfitting and a selective bias towards covariates with many possible splits. Although overfitting can be handled by a pruning procedure, the interpretation of the trees is affected by the biased variable selection. Hothorn et al. (2006, Section 6.1), noted that "An algorithm for recursive partitioning is called unbiased when, under the conditions of the null hypothesis of independence between a response y and covariates $X_1, \ldots, X_m$, the probability of selecting covariate X_j is $1/m$ for all $j = 1, \ldots, m$ regardless of the measurement scales or number of missing values". Bias corresponds to certain types of covariates being more or less likely to be split on, and is introduced because the tree is constructed based on maximization of a splitting criterion over all possible splits simultaneously. This has been identified as a problem and studied by many researchers.

White and Liu (1994) demonstrated that the usual splitting measures, such as information gain and gain ratio, are biased in favor of covariates with larger numbers of values. Suppose we randomly partition a covariate A to produce a derived covariate A', which has a larger number of values. Quinlan (1986) showed that

$$H_T\left(A'\right) \geq H_T\left(A\right),$$

where H_T is called the transmitted information, an information measure corresponding to the reduction in impurity gained by splitting the parent node into its daughter nodes in CART. That means, in general, that the derived covariate will transmit more information about class membership than the original one. However, as the additional partition is random, A' cannot be reasonably preferred over A as a candidate for splitting. Hence, the information measure, H_T, is not comparable between covariates with different numbers of values. In fact, any information-based measure that is calculated without taking the number of levels of covariates (degrees of freedom) into account is inappropriate for comparison between two covariates that have different numbers of levels. White and Liu (1994) concluded that approaches using the chi-squared distribution are preferable because they compensate automatically for differences between covariates in the number of levels they take.

For continuous responses, GUIDE (Loh, 2002) controls bias by employing chi-squared analysis of residuals. For example, at each node, a constant (namely, the sample mean of the response variable for all observations in that node) is fitted and the residuals are computed. Next, for each numerical-valued predictor variable, the data are divided into four groups at the sample quantiles. A 2×4 contingency table is then constructed with the signs of the residuals (positive versus non-positive) as rows and the groups as columns. The number of observations in each cell is determined and the χ^2 test of independence and its theoretical p-value from a χ^2_3 distribution are computed. This is also done for each categorical variable, using the categories of the variable to form the columns of the contingency table and omitting columns with zero column totals. Finally, the predictor variable associated with the smallest p-value is selected as the basis of splitting at the node. Loh and Zheng (2013) extend this idea to longitudinal and multiple response data based on simultaneous patterns of positive and negative residuals for all of the response variables, but this approach cannot directly handle time-varying covariates.

Eo and Cho (2014) adapted the GUIDE approach, focusing on splits related to changes in the response over time within an object rather than the response itself. They could only handle time-varying covariates indirectly, by assuming there was a lower-order polynomial relationship between such a covariate and time, and using the estimated coefficients of that relationship as potential time-invariant covariates. Note that focusing on changes related to time has the advantage of allowing for direct dependence on time through growth curves at each node (in contrast, the methods of Hajjem et al., 2011, Sela and Simonoff, 2012, and Loh and Zheng, 2013, can model dependence on time as part of the tree by including time as a potential covariate), but still does not allow for dependence on arbitrary time-varying covariates.

Hothorn et al. (2006) address the bias problem for data without a longitudinal structure using conditional distributions and permutation tests (randomization). In their paper, they present a framework embedding recursive partition into a well-defined theory of permutation tests developed by Strasser and Weber (1999). The conditional distribution of statistics measuring the association between responses and covariates is the basis for an unbiased selection among covariates measured at different scales.

The paper focuses on regression models describing the conditional distribution of a response variable y given the status of k covariates by means of tree-structured recursive partitioning. The response y from some sample space $\mathcal{Y}$ may be multivariate as well. The k-dimensional covariate vector $\mathbf{X} = (X_1, \ldots, X_k)$ is taken from a sample space $\chi = \chi_1 \times \cdots \times \chi_k$. Both the response variable and covariates may be measured at arbitrary scales. They assume that the conditional distribution $D(y|\mathbf{X})$ of the response y given the covariates $\mathbf{X}$ depends on a function f of the covariates

$$D(y|\mathbf{X}) = D(y|X_1, \ldots, X_k) = D(y|f(X_1, \ldots, X_k)).$$

A generic algorithm for recursive binary partitioning for a given learning sample can then be formulated. Assume there is one dependent variable y and k covariates $(X_j,\ j = 1, 2, \ldots, k)$. If y and X_j are independent, under null hypothesis H_0^j, the conditional distribution $D(y|X_j) = D(y)$. The recursion will stop if the global hypothesis $H_0 = \bigcap_{j=1}^{k} H_0^j$ cannot be rejected (this is the stopping criterion). If it can be rejected, they measure the association between y and each covariate X_j by a test statistic T_j. The distribution of T_j under H_0^j depends on the joint distribution of y and X_j, which is unknown. However, under H_0, by fixing the covariate X_j and conditioning on all possible permutations of response y, one can obtain the conditional expectation μ_j and covariance Σ_j of T_j under H_0 given all permutations, which was derived by Strasser and Weber (1999). With conditional expectation μ_j and covariance Σ_j, they are able to standardize the statistics T_j and obtain the corresponding p-value for each T_j.

The bias correction idea is that instead of directly comparing the measures T_j for all X_j, $j = 1, \ldots, k$, they construct a conditional distribution for each T_j first. Such a conditional distribution is constructed based on the assumption that y and X_j are independent (permutation will force them to be independent). They then calculate the p-value for T_j from their distribution to measure how far it deviates from its mean, i.e. how far it deviates from independence. Such statistics (p-values) are no longer affected by the scale or dimension of X_j (adjusted through their own distribution), and are thus directly comparable among different X_j. With the smallest p-value, one can select the variable X_j to split, while the split itself can be determined by any criterion, including those established by Breiman et al. (1984). Moreover, unlike CART, which uses a pruning procedure to find the correct tree structure, they stop the recursion when they determine there is no significant association between any of the covariates and the response through multiple testing procedures. The algorithm that implements this method is called conditional inference trees (*ctree* in the $\mathcal{R}$ package `party`). Note that in all of the discussion on bias, it is considered as a property under the null hypothesis; that is, CART prefers to split on variables with more possible splits when the variables have no predictive power. The situation when a variable actually has predictive power is different, and will be discussed in Section 3.6.

In our proposed method, we use *ctree* instead of CART in the original RE–EM tree algorithm initially proposed by Sela and Simonoff (2012), in order to gain unbiasedness. We lay out the details of the RE–EM tree algorithm by Sela and Simonoff (2012) and propose our new version in Section 2; in Section 3, we use simulations to demonstrate the unbiasedness property of our proposed method. We also evaluate its performance in terms of prediction accuracy and the ability to recover the correct tree structure compared to the original RE–EM tree algorithm. Section 4 illustrates the application of the method to real data.

2. The unbiased RE–EM tree

Before illustrating the details of the RE–EM tree, we must first formalize notation and terminology. We follow those used in Sela and Simonoff (2012). We observe a panel of objects or individuals $i = 1, \ldots, I$ at times $t = 1, \ldots, T_i$. We will refer to a member of the panel, i, as an object, and a single observation period for an object, (i, t), as an observation. That is, one object is associated with multiple observations. For each observation, we observe a vector of covariates, $\mathbf{x}_{it} = (x_{it1}, \ldots, x_{itk})'$. The covariates may be constant over time, constant across objects, or varying across time and objects. To account for the differences between objects across time periods, we include a known design matrix (which is actually either a scalar or a row vector), Z_{it}, which may vary each period and depends on the covariates, and a vector of unknown time-constant, object-specific effects, $\mathbf{b}_i$. In the case where only the intercept varies across objects, Z_i is a matrix of ones and b_i is the object-specific intercept. This then implies a general effects model with additive errors:

$$y_{it} = Z_{it}\mathbf{b}_i + f\,(x_{it1}, \ldots, x_{itk}) + \varepsilon_{it}, \qquad (1)$$

$$\begin{pmatrix} \varepsilon_{i1} \\ \vdots \\ \varepsilon_{iT_i} \end{pmatrix} \sim Normal\,(\mathbf{0}, R_i)$$

and

$$\mathbf{b}_i \sim Normal\,(\mathbf{0}, D)\,.$$

We assume that the errors, ε_{it}, are independent across objects and are uncorrelated with the effects, $\mathbf{b}_i$. Note, however, that an autocorrelation structure within the errors for a particular object is allowed. To do this, we allow R_i to be a non-diagonal matrix. If f is a known function that is linear in the parameters and the $\mathbf{b}_i$ are taken as fixed or potentially correlated with the covariates, then this is a linear fixed effects model. Under the same assumptions about f, if the $\mathbf{b}_i$ are assumed to be random and uncorrelated with the covariates, then the model is a linear mixed effects model. Traditional mixed effects models, such as the linear mixed effects model (where $f = X\boldsymbol{\beta}$), assume a parametric form for f, which might be a too restrictive assumption. Instead, a regression tree is used to estimate f. If the random effects, $\mathbf{b}_i$, were known, (1) implies that we could fit a regression tree to $y_{it} - Z_{it}\mathbf{b}_i$ to estimate f. If the population-level effects, f, were known, then we could estimate the random effects using a traditional mixed effects linear model with population-level effects corresponding to the values $f\,(x_i)$. Since neither the random effects nor the fixed effects are known, we alternate between estimating the regression tree, assuming that our estimates of the random effects are correct, and estimating the random effects, assuming that the regression tree is correct. More formally, the estimation method is given as follows:

1. Initialize the estimated random effects, $\hat{\mathbf{b}}_i$, to zero.
2. Iterate through the following steps until the estimated random effects, $\hat{\mathbf{b}}_i$, converge (based on change in the likelihood or restricted likelihood function being less than some tolerance value):

(a) Estimate a regression tree approximating f, based on the target variable, $y_{it} - Z_{it}\hat{\mathbf{b}}_i$, and covariates, $\mathbf{x}_{it} = (x_{it1}, \ldots, x_{itk})$, for $i = 1, \ldots, I$ and $t = 1, \ldots, T_i$. Use this regression tree to create a set of indicator variables, $I\left(\mathbf{x}_{it} \in g_p\right)$, where g_p ranges over all of the terminal nodes in the tree.

(b) Fit the linear mixed effects model, $y_{it} = Z_{it}\mathbf{b}_i + \sum_p I\left(\mathbf{x}_{it} \in g_p\right)\mu_p + \varepsilon_{it}$. Extract $\hat{\mathbf{b}}_i$ from the estimated model.

3. Replace the predicted response at each terminal node of the tree with the estimated population level predicted response $\hat{\mu}_p$ from the linear mixed effects model fit in 2b.

The algorithm makes clear why time-varying (observation-level) covariates are easily handled in the MERT/RE–EM tree approach, since it is observations that are being split, not objects. The fitting of the tree in Step 2a can be achieved using any tree algorithm. Sela and Simonoff (2012) used the implementation of the CART algorithm in the `rpart` $\mathcal{R}$ package in their implementation in the `REEMtree` package. The algorithm splits a node where it maximizes the reduction in sum of squares for the node. Such recursive splitting continues as long as the proportion of variability accounted for by the tree (called the complexity parameter, cp) increases at least 0.001 and the number of observations in the candidate splitting node is greater than 20. After the initial tree is built, it is pruned back based on 10-fold cross-validation. First, the algorithm obtains the tree with final splits corresponding to the cp value with minimized 10-fold cross-validation error. Then, the tree with final split corresponding to the largest cp value that has 10-fold cross-validation error less than one standard error above the minimized value is determined as the final tree (the so-called "one-SE" rule). The linear model with random effects in Step 2b can be estimated using maximum likelihood or restricted maximum likelihood (REML).

As was mentioned earlier, using CART as the regression tree will lead to the algorithm being biased. We propose to use the conditional inference tree of Hothorn et al. (2006) instead of CART in Step 2a above; since the conditional inference tree is unbiased, the new version of the RE–EM tree using *ctree* should also be unbiased. We shall refer to this new version of the RE–EM tree as the unbiased RE–EM tree. Its unbiasedness properties, as well as other properties, will be studied in the next section using simulations. It is important to note that in principle any unbiased regression tree method, such as the nonlongitudinal version of GUIDE, could be used in Step 2a of the algorithm to achieve unbiasedness; we briefly discuss using GUIDE in this way (instead of *ctree*) in the next section. The main advantage of using *ctree* is that it is easy to implement in $\mathcal{R}$ as a simple adjustment to the `REEMtree` package; details are provided at the paper's associated web site discussed in the Conclusions section.

3. Simulation results

3.1. Unbiasedness property of trees

We first use simulations to assess the unbiasedness of the proposed RE–EM tree method. For comparison, the results for the CART-based RE–EM tree are also included for each simulation setting. We follow the approach of Hothorn et al. (2006), who demonstrated the unbiasedness of *ctree*. Five uniformly distributed random variables $X_1, \ldots, X_5 \sim \mathcal{U}[0, 1]$ serve as numeric covariates. In covariate X_4, 25% of the values are drawn missing at random, and the values of X_5 are rounded to one digit, that is, we induce 11 unique realizations. An additional nominal covariate X_6 is measured at two levels, with 50% of the observations being equal to zero (Hothorn et al., 2006).

In the linear mixed effects model, for each individual unit i,

$$\mathbf{y}_i = X_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{b}_i + \boldsymbol{\varepsilon}_i,$$

where $\boldsymbol{\varepsilon}_i$ is distributed as $N(\mathbf{0}, R_i)$ (normal with mean $\mathbf{0}$ and covariance matrix R_i). Here R_i is a $T_i \times T_i$ positive-definite covariance matrix; it depends on i through its dimension T_i, but the set of unknown parameters in R_i will not depend on i (Laird and Ware, 1982). In our simulations each individual unit i has same dimension, that is, $T_i = T$ for $i = 1, 2, \ldots, I$, and we generate $\boldsymbol{\varepsilon}_i$ from $N(\mathbf{0}, R)$. The $\mathbf{b}_i$ are distributed as $N(\mathbf{0}, D)$, independently of each other and of $\boldsymbol{\varepsilon}_i$. In our design, we set

$$\mathbf{y}_i = \mathbf{Z}_i\mathbf{b}_i + \boldsymbol{\varepsilon}_i,$$

in order to make the response variable y independent from the Xs. In this setting, y should not split on any covariate X (that is, this is a null situation). If forced to split, the unbiased regression tree should split on each covariate X with equal probability as they are all independent of y. $\mathbf{Z}_i$ is a known design matrix whose structure determines the complexity of the random effect. In the case where only the intercept varies across individuals (the random intercept model), $\mathbf{Z}_i$ is a vector of ones. Our simulation models three settings:

1. Random intercept model, where $\mathbf{Z}_i$ is a vector of ones;
2. One random effect vector, i.e. $\mathbf{Z}_i = (\mathbf{1}, \mathbf{t})$, where $\mathbf{t} = (1, 2, \ldots, T)'$ (note that this corresponds to different linear growth curves for different objects);
3. Two random effect vectors, i.e. $\mathbf{Z}_i = (\mathbf{1}, \mathbf{t}, \mathbf{u})$, where $\mathbf{u} \sim N(0, 1)$.

Despite the different structures of the random effect, an unbiased RE–EM tree algorithm should select each predictor variable $(X_1, \ldots, X_6)$ as the split of the root node with the same probability 1/6. We evaluate the results using the Pearson Chi-Squared Test, whose null hypothesis is the underlying uniform population probability vector (1/6, 1/6, 1/6, 1/6, 1/6, 1/6).

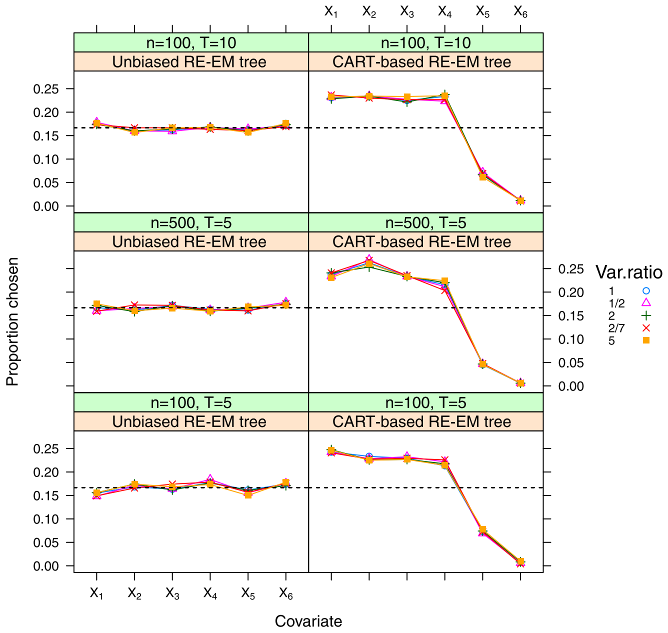

Fig. 1. Unbiasedness plot for random intercept case.

3.1.1. Random intercept model

In the random intercept model, $\mathbf{Z}_i$ is a vector of ones, so $\mathbf{Z}_i\mathbf{b}_i$ is a vector of identical elements, i.e. the same number for each observation of individual i. Hence, we generate element $(Zb)_i \sim N\left(0, \sigma_1^2\right)$ and generate $\varepsilon_{it} \sim N\left(0, \sigma_2^2\right)$ ($R_i = \sigma_2^2 I$, I is the identity matrix, and ε_{it} is the tth element of $\boldsymbol{\varepsilon}_i$) for each individual i, where σ_1 and σ_2 are specified. Fig. 1 gives the simulation results.

Note that the root split is forced in the simulations, which has 1200 replications for each set of parameters. Since it is well-known that the mix of number of individuals and number of observations make a difference in the properties of mixed effects models, different sets of sample size and group size are tested. Our base set is $\{n = 100, T = 5\}$, i.e. 100 individuals and 5 observations each individual, in order to keep results consistent with the simulation structure in Hajjem et al. (2011), which we will follow in Section 3.2. Variations made upon this base set of $\{n, T\}$ are tested in our simulation, such as $\{n = 500, T = 5\}$ and $\{n = 100, T = 10\}$ in the random intercept model setting, and $\{n = 20, T = 50\}$ and $\{n = 10, T = 50\}$ in both the one random effect vector and two random effect vectors settings. One can see that regardless of the set of $\{n, T\}$ and the value of the ratio σ_1^2/σ_2^2, the proposed unbiased RE–EM tree returns unbiased results. In contrast, the original CART-based RE–EM tree consistently returns biased results, which is shown on the right side of Fig. 1. Tail probabilities for chi-squared tests of uniformity are all greater than 0.218 for the unbiased RE–EM tree, and less than 2.2×10^{-16} for the CART-based tree. Note that for the CART-based RE–EM tree the bias is as expected, with fewer splits as the number of distinct values of the predictor decreases. Note also that the missingness in predictor X_4 does not affect the bias properties of the proposed method.

Since bias in the RE–EM tree comes from the use of *rpart* in Step 2a of the algorithm, using any unbiased nonlongitudinal tree should result in unbiasedness of the RE–EM tree. This was investigated by exploring the properties of a tree using the nonlongitudinal version of GUIDE (Loh, 2002) instead of *rpart*. As would be expected the resultant RE–EM tree is indeed unbiased, but because GUIDE is not available in $\mathcal{R}$ code and must be executed using the `system()` command, it is far slower than using *ctree*, and will thus not be pursued further.

3.1.2. One random effect vector, i.e. $\mathbf{Z}_i = (\mathbf{1}, \mathbf{t})$

Under our design of a random effect with one vector, $\mathbf{t}$ is the vector $(1, 2, \ldots, T)'$ in the matrix $\mathbf{Z}_i = (\mathbf{1}, \mathbf{t})$. Since $\mathbf{b}_i \sim N(\mathbf{0}, D)$, to account for the potential influence of the covariance matrix of $\mathbf{b}_i$, 3 different covariance matrices in our simulation are tested,

$$D_1 = \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{pmatrix}, \qquad D_2 = \begin{pmatrix} 2.0 & 0.5 \\ 0.5 & 2.0 \end{pmatrix} \quad \text{and} \quad D_3 = \begin{pmatrix} 5.0 & 0.1 \\ 0.1 & 5.0 \end{pmatrix}.$$

We generate ε_{it} from $N(0, 1)$, where ε_{it} is the tth element of $\boldsymbol{\varepsilon}_i$. From Fig. 2, one can see that the proposed unbiased RE–EM tree is unbiased for each covariance matrix (with p-values for uniformity of splits all greater than 0.432) while the CART-based RE–EM tree is biased in general (with p-values all less than 2.2×10^{-16}).

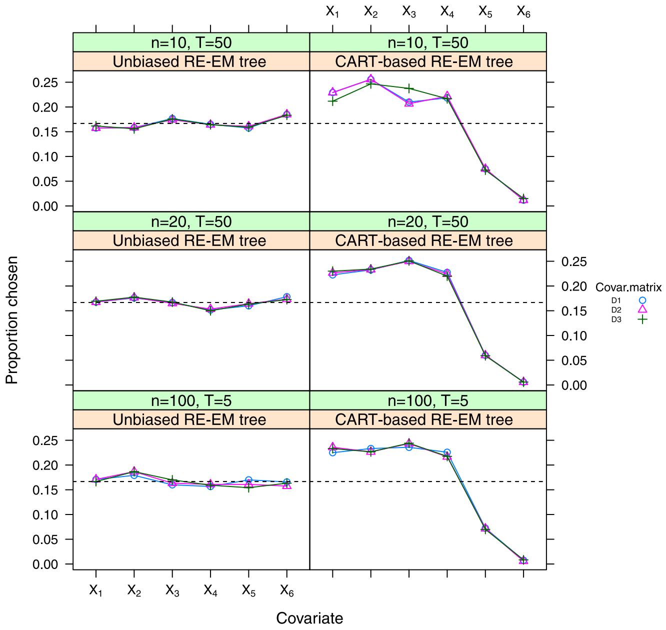

Fig. 2. Unbiasedness plot for one random effect vector.

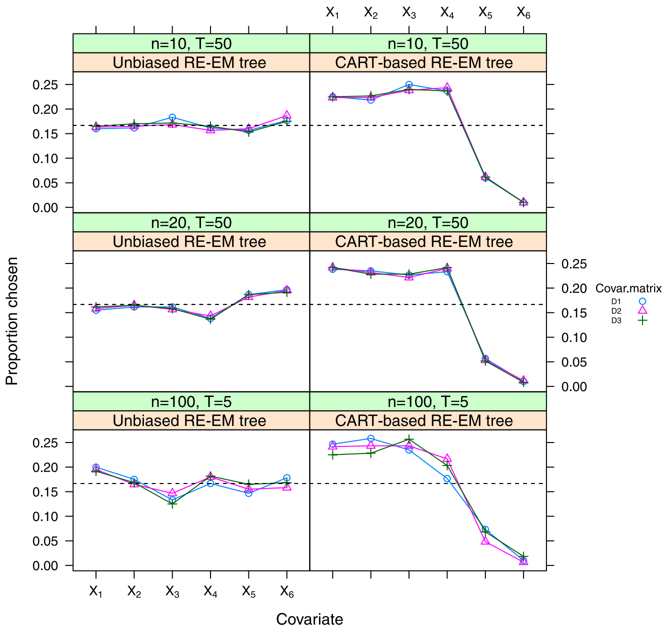

Fig. 3. Unbiasedness plot for two random effect vectors.

3.1.3. *Two random effect vectors, i.e.* $\mathbf{Z}_i = (\mathbf{1}, \mathbf{t}, \mathbf{u})$

For these simulations, $\mathbf{t}$ is the same vector $(1, 2, \ldots, T)'$ as above, while $\mathbf{u}$ is a vector with values randomly generated from $N(0, 1)$. We again investigated performance for three different covariance matrices, which are

$$D_1 = \begin{pmatrix} 1.0 & 0.5 & 0.2 \\ 0.5 & 1.0 & 0.5 \\ 0.2 & 0.5 & 1.0 \end{pmatrix}, \qquad D_2 = \begin{pmatrix} 3.0 & 1.0 & 0.5 \\ 1.0 & 2.0 & 0.2 \\ 0.5 & 0.2 & 1.0 \end{pmatrix} \quad \text{and} \quad D_3 = \begin{pmatrix} 0.5 & 0.1 & 0.2 \\ 0.1 & 2.0 & 1.0 \\ 0.2 & 1.0 & 7.0 \end{pmatrix}.$$

The results of the simulations are given in Fig. 3. In this situation, simulations are based on 600 trials, because the algorithm becomes time consuming for the setting $\mathbf{Z_i} = (\mathbf{1}, \mathbf{t}, \mathbf{u})$. One can see that the unbiased RE–EM tree is unbiased

Table 1
Ability to recover correct tree structure.

$(\mu_1, \mu_2, \mu_3, \mu_4)$	d_{11}	d_{22}	d_{12}	Unbiased RE–EM tree			CART-based RE–EM tree		
				Uniform	11-valued	Binary	Uniform	11-valued	Binary
(11, 12, 13, 14)	0.00	0.00	0.00	75	73	89	92	90	88
(11, 12, 13, 14)	0.25	0.00	0.00	88	80	84	85	82	82
(11, 12, 13, 14)	0.50	0.00	0.00	84	81	84	83	75	83
(11, 12, 13, 14)	0.25	0.25	0.00	75	71	85	50	49	52
(11, 12, 13, 14)	0.50	0.50	0.00	72	75	83	51	59	39
(11, 12, 13, 14)	0.25	0.25	0.125	72	77	87	56	55	30
(11, 12, 13, 14)	0.50	0.50	0.25	64	70	85	45	56	23

$(\mu_1, \mu_2, \mu_3, \mu_4)$	d_{11}	d_{22}	d_{12}	σ^2_δ	Unbiased RE–EM tree			CART-Based RE–EM tree		
					RS X_1	RS X_4	Correct tree	RS X_1	RS X_4	Correct tree
(11, 12, 13, 14)	0.50	0.00	0.00	1	100	0	85	40	60	35
(11, 12, 13, 14)	0.50	0.00	0.00	5	100	0	85	100	0	85
(11, 12, 13, 14)	0.50	0.00	0.00	25	100	0	85	100	0	85
(11, 12, 13, 14)	0.50	0.50	0.00	1	96	1	84	16	48	7
(11, 12, 13, 14)	0.50	0.50	0.00	5	99	0	89	58	3	20
(11, 12, 13, 14)	0.50	0.50	0.00	25	100	0	89	70	0	42
(11, 12, 13, 14)	0.50	0.50	0.25	1	98	1	88	15	38	8
(11, 12, 13, 14)	0.50	0.50	0.25	5	99	0	89	37	2	18
(11, 12, 13, 14)	0.25	0.25	0.125	1	100	0	89	22	71	11
(11, 12, 13, 14)	0.25	0.25	0.125	5	100	0	88	91	2	33

The top part of the table gives simulation results without X_4 in the regression; the bottom part of table gives results when including X_4 in the regression.

for each set of parameters, although there is marginal evidence of nonuniform splits for D_1 ($p = 0.07$) and D_3 ($p = 0.096$) when $n = 100$ and $T = 5$. The CART-base RE–EM tree is biased as expected ($p < 2.2 \times 10^{-16}$ in all situations).

3.2. Ability of trees to recover the true tree structure

These simulations follow the simulation structure used in Hajjem et al. (2011). The simulation design used has a hierarchical structure of 100 balanced clusters with 55 observations generated in each cluster. The first five observations in each cluster form the training sample, and the other 50 observations are left for the test sample. Consequently, the tree is built with 500 observations (100 clusters of 5 observations) in each simulation run. Three random variables, X_1, X_2, and X_3, are first generated independently with a uniform distribution in the interval [0, 10]. They serve as predictors (note that each observation has a unique set of predictor values, corresponding to time-varying covariates unrelated to time itself). The response variable y is generated based on the following rules:

1. Leaf 1. If $x_{it1} \leq 5$ and $x_{it2} \leq 5$ then $y_{it} = \mu_1 + z'_{it}\mathbf{b}_i + \varepsilon_{it}$;
2. Leaf 2. If $x_{it1} \leq 5$ and $x_{it2} > 5$ then $y_{it} = \mu_2 + z'_{it}\mathbf{b}_i + \varepsilon_{it}$;
3. Leaf 3. If $x_{it1} > 5$ and $x_{it3} \leq 5$ then $y_{it} = \mu_3 + z'_{it}\mathbf{b}_i + \varepsilon_{it}$;
4. Leaf 4. If $x_{it1} > 5$ and $x_{it3} > 5$ then $y_{it} = \mu_4 + z'_{it}\mathbf{b}_i + \varepsilon_{it}$,

where $\mathbf{b}_i$ and $\boldsymbol{\varepsilon}_i$ are generated according to $N(\mathbf{0}, D)$ and $N(\mathbf{0}, I)$ respectively, for $i = 1, \dots, 100$ and $t = 1, \dots, 5$. Each observation t in cluster i falls into only one of the four terminal nodes with mean response value equal to μ_1, μ_2, μ_3, or μ_4, respectively.

In the simulations, the means of the four terminal nodes are set to be $\mu_1 = 10$, $\mu_2 = 11$, $\mu_3 = 12$, and $\mu_4 = 13$. The random components are generated based on the following three different scenarios:

1. No random effect, i.e. $D = 0$;
2. Random intercept (RI), i.e. $z_{ij} = 1$ for $i = 1, \dots, 100$ and $t = 1, \dots, 5$, and $D = d_{11} > 0$;
3. Random intercept and covariate (RIC) which is a RI with linear random effect for X_1. More precisely, $z_{it} = [1, x_{it1}]$ for $i = 1, \dots, 100$ and $t = 1, \dots, 5$, and $D = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}$, $d_{11} > 0$ and $d_{22} > 0$.

In all cases, the within-cluster variance σ^2 is set to 1. We consider two levels of the between-clusters covariance matrix D. In the RI case, we use $D = d_{11} = 0.25$ and 0.5 which are equivalent to an intra-cluster correlation coefficient of 0.20 and 0.33 respectively. In the RIC case, we have two additional conditions based on the value of the correlation between the random components, $d_{12}/\sqrt{d_{11} + d_{22}} = 0$ and $d_{12}/\sqrt{d_{11} + d_{22}} = 0.5$; in the first correlation scenario, $d_{11} = d_{22} = 0.25$, and in the second $d_{11} = d_{22} = 0.5$. We also test on variations where the root split variable, X_1, is categorical with 11 values (rounding the original uniformly distributed X_1 to closest integer) and when X_1 is binary taking values 0 and 5.77 with probability 0.5 (this way X_1 has the same variance as a Uniform(0, 10)). Table 1 gives results for 100 simulation runs in terms of the percentage of the time the method recovers the true tree structure. The default parameter settings are used in both the unbiased RE–EM tree and CART-based RE–EM tree algorithms.

One can see from the top of Table 1 that the CART-based RE–EM tree is biased against splitting on a variable with fewer possible splits; as a result, when X_1 is binary, it is less likely to find the correct tree structure compared to when X_1 is uniform. This bias is more obvious when the random effect has a complex structure corresponding to random slope and intercept. On the other hand, the performance of the unbiased RE–EM tree is relatively unaffected by the random effect's complexity.

In this design, the three predictors are independent of each other. The possibility of correlated predictors is much more realistic and interesting. Hence, we add another variable $X_4 = X_1 + \delta$ to the regression, where $\delta \sim N\left(0, \sigma_\delta^2\right)$. If X_1 is binary $\{0, 5.77\}$, X_1 and X_4 will be correlated with correlation equal to 0.95 when $\sigma_\delta^2 = 1$, 0.8 when $\sigma_\delta^2 = 5$, and 0.5 when $\sigma_\delta^2 = 25$. Note that in this situation, splitting on X_4 is incorrect, but when X_4 is highly correlated with X_1, such false splitting is not surprising. The bottom part of Table 1 gives the simulation results when we include X_4 in the regression.

In this part of the table, "RS" means root split, so "RS X_1" gives the counts of the number of times the root split was on X_1 (the correct split). From the simulation results, we can see the CART-based RE–EM tree will falsely split on X_4 instead of the correct variable X_1 if those two variables have high correlation, and hence recover less often the correct tree structure. Thus, the CART-based RE–EM tree will have trouble identifying the correct variable on which to split if the correct predictor has a highly correlated "competitor", especially when the "competitor" has more possible splits than the correct predictor. In contrast, in these simulations the unbiased RE–EM tree almost never picks X_4 instead of X_1 as the root split variable, and thus finds the correct tree structure more often in general.

3.3. Prediction performance of the trees

In this section, we compare the prediction performance of the CART-based RE–EM tree to that of the unbiased RE–EM tree. Data are generated by the method used in Section 3.2, i.e. 100 balanced clusters with 55 observations generated in each cluster. The first five observations in each cluster form the training sample, and the other 50 observations are left for the test sample. We evaluate the predictive accuracy by the predictive mean square error:

$$\text{PMSE} = \frac{\sum_{i=1}^{100}\sum_{t=1}^{50}\left(y_{it} - \hat{y}_{it}\right)^2}{5000},$$

where $\hat{y}_{it}$ is the predicted response by the tree algorithm for observation t in cluster i in the test set. We also compare the PMSE of the fixed effect estimates and PMSE of the random effect estimates given by the two methods in order to have a comprehensive view on the details of prediction. To accommodate a more general setting, we modify the simulation design in the following ways.

1. We include 9 predictors, $\{X_1, X_2, X_3, X_5, X_6, X_7, X_8, X_9, X_{10}\}$ in all simulation runs. If X_4 is included, it is correlated with X_1 in the way described in the previous section. X_5 and X_6 are continuous Uniform(0, 10), X_7 and X_8 are categorical with 11 values, and X_9 and X_{10} are binary $\{0, 5.77\}$.
2. Only X_1, X_2 and X_3 are used to define the fixed effect as in the previous section; $X_5 - X_{10}$ are independent from X_1, X_2, X_3 and each other.
3. We consider the Random intercept (RI) case and Random intercept and covariate (RIC) case here, with $z_{it} = [1, x_{it5}]$ in the RIC case.
4. Two situations of the root split variable X_1 are considered, Uniform(0, 10) and Binary$\{0, 5.77\}$.
5. In the Random intercept (RI) case, we consider $D = d_{11}$ with 6 different values $\{0.25, 0.75, 1.25, 1.75, 2.25, 3.00\}$, in the Random intercept and covariate (RIC) case, $D = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}$ and we consider 6 different matrices: $D_1 = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix}$, $D_2 = \begin{pmatrix} 0.50 & 0 \\ 0 & 0.50 \end{pmatrix}$, $D_3 = \begin{pmatrix} 1.50 & 0 \\ 0 & 1.50 \end{pmatrix}$, $D_4 = \begin{pmatrix} 3.00 & 0 \\ 0 & 3.00 \end{pmatrix}$, $D_5 = \begin{pmatrix} 0.50 & 0.25 \\ 0.25 & 0.50 \end{pmatrix}$ and $D_6 = \begin{pmatrix} 2.00 & 1.75 \\ 1.75 & 3.00 \end{pmatrix}$.

The results are summarized in Figs. 4–7.

1. **Setting when X_4 is not included among the candidate covariates**

   Fig. 4 shows the results of the simulations in terms of performance in recovering the correct tree structure. The horizontal axis indexes the six different random effect d_{11} values (RI) and D matrices (RIC), respectively, and the vertical axis gives the percentage of correct tree structures recovered based on 100 simulation replications. We investigate two scenarios: when the root split variable X_1 is Uniform(0, 10) and when it is Binary$\{0, 5.77\}$. The figure shows that regardless of the root split variable type, the performance of the biased RE–EM tree is worse in the RIC case compared to RI case. In contrast, the unbiased RE–EM tree has similar performance in either situation.

   In Fig. 5, we compare the predictive accuracy of the two RE–EM tree methods for two scenarios: when the root split variable is binary and when it is uniform. One can see that the biased RE–EM tree consistently has greater PMSE value than the unbiased tree. Signed-rank tests on each pair of PMSEs returned by the biased and unbiased tree methods for each random effect value or matrix find that for most RIC cases, the difference between the PMSE of the biased tree and PMSE of the unbiased tree is statistically significant at 0.05 level, implying that the unbiased RE–EM tree performs better than the biased version in terms of predicting the response value y when the underlying random effect has a more complex structure.

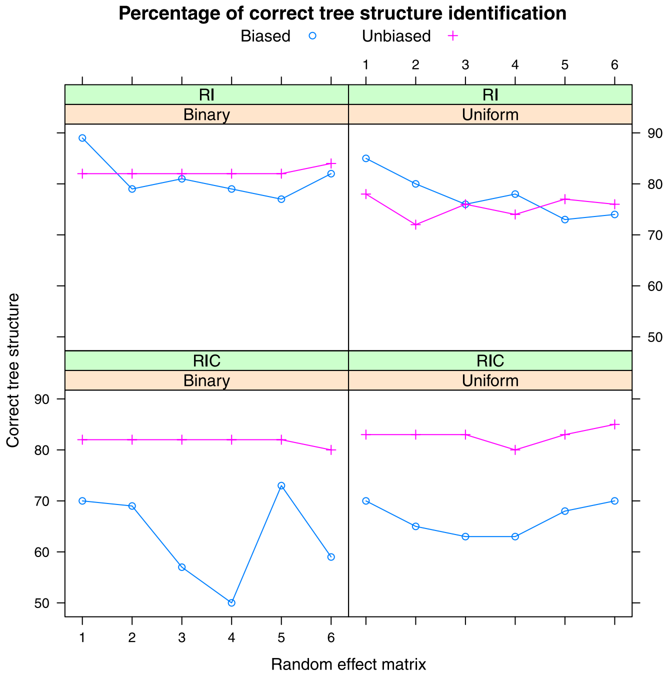

Fig. 4. The ability to recover the correct tree structure by the unbiased and biased RE–EM trees when X_4 is not included; Binary and Uniform stand for the root split variable (X_1) type.

The second and third plots of Fig. 5 show that the difference in PMSE performance comes from estimation of the fixed effects (second plot) rather than the random effects (third plot). Signed rank tests confirm this, as they are statistically significant for differences in estimation accuracy in all fixed effects comparisons, and are not statistically significant in any of the random effects comparisons.

2. **Setting when X_4 is included, and correlated with X_1 with correlation 0.95**

In Fig. 6, we see how split bias can hurt performance in terms of finding the correct tree structure. When the root split variable is binary and its highly correlated competitor (X_4 here) is uniform, the biased tree will tend to split on X_4 instead of X_1. This is why we see that the biased tree finds the correct tree structure significantly less often when X_1 is binary compared to when X_1 is uniform, and compared to the unbiased tree result. This translates into worse performance in estimating the fixed effect compared to the unbiased tree and worse predictive accuracy for the response valuable y (Fig. 7). All paired comparisons for fixed effect estimation in the Binary scenario are statistically significant, and most pairs in the Uniform scenario are significantly different when the random effect is RIC. The same pattern holds for comparison of PMSE values for the overall response. It should be noted, however, that the PMSE of y is usually smaller than the sum of the PMSEs of the fixed effect and the random effect for both RE–EM tree methods, indicating that the errors made in the course of estimating the fixed effect and the random effect tend to compensate for each other in the RE–EM tree algorithm.

3.4. Time-related structure

Most of the simulations thus far do not actually use time structure at all, as the observations are actually generated as clustered, rather than longitudinal. A time effect can be incorporated into the model in two different ways. One possibility is that time is part of the fixed effect; that is, the time point associated with each observation is used as a potential split variable in the tree (obviously values of any covariate from earlier time periods, including time itself, also can be used as a covariate). Since time is merely acting as another predictor this is no different from the simulations already examined, with time being one of the X_j variables. In particular, if the true effect follows a tree structure it can be recovered quite successfully. If the relationship with time does not take the form of a tree, estimation of the fixed effect degrades as would be expected, but with enough replications (time points) the tree can approximate non-tree relationships reasonably well, as was demonstrated in Sela and Simonoff (2012). A second possibility is to include time as a covariate in the random effect (an RIC model). Once again there is nothing special about time in this context, as was seen earlier for RIC models. As time is simply treated as any predictor would be the occurrence of observations at unequally-spaced time points has no effect on performance. We provide here some details on these results.

Details regarding the performance of the tree methods when the true relationship with time is actually linear (that is, the model is misspecified) can be found in the supplemental material available at the paper's associated web site.

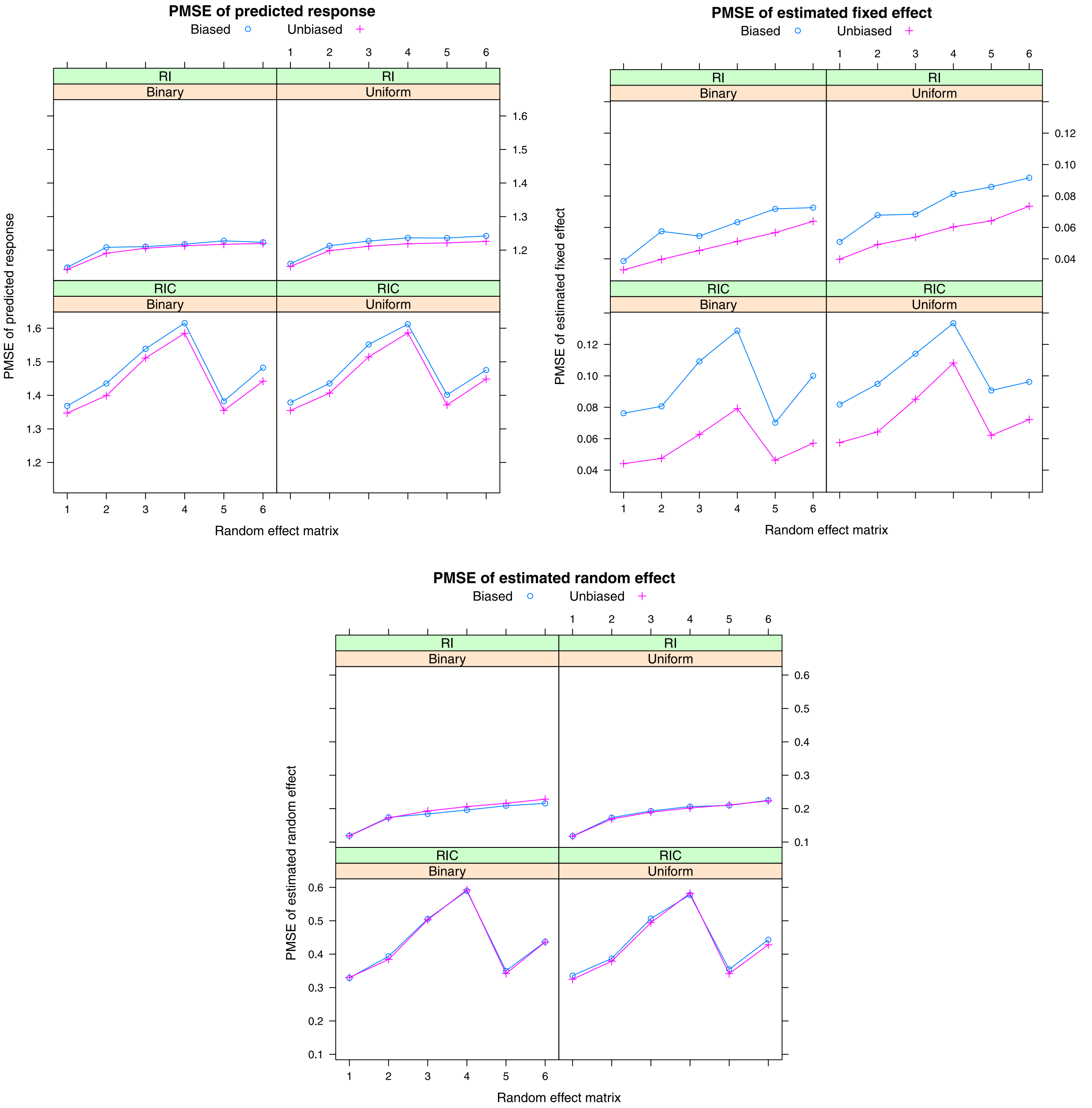

Fig. 5. PMSE of response y, fixed effect and random effect, respectively, when X_4 is not included.

3.4.1. *Evenly-spaced time points*

To make the training/testing set more consistent with a longitudinal setting, we repeat the simulations in Sections 3.2 and 3.3 with the following changes in simulation design:

- Add a time index variable T to the regression, which has values $\{1, 2, 3, 4, 5\}$ for observations within each individual of the training set, and has 50 values evenly spaced on the interval $[1, 5]$ for each individual in the testing data.
- Make X_3 a time-invariant variable.
- In the RIC cases, the linear random effect is based on T rather than X_1, i.e. $Z_{it} = [1, T_{it}]$ for $i = 1, \ldots, 100$ and $t = 1, \ldots, 5$.

More specifically, the model has covariates X_1, X_2, X_3, T, where X_1, X_2, X_3 are uniform[0, 10], with X_1, X_2 time-varying covariates and X_3 a time-invariant variable, and T has the values given above. While X_1, X_2, X_3 determine the fixed effect, T has an effect on the random effect in the RIC cases. The variable X_4 is a "competitor" covariate that is constructed to be correlated with X_1 as before.

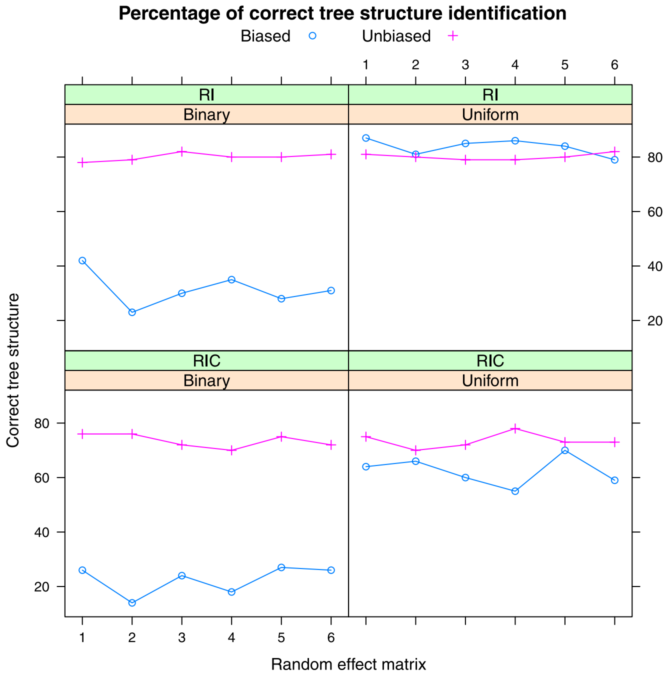

Fig. 6. The ability to recover right tree structure by unbiased and biased RE–EM tree when X_4 is included; Binary and Uniform stand for the root split variable (X_1) type.

Table 2
Ability to recover the correct tree structure when time is included in the random effect in the RIC model.

$(\mu_1, \mu_2, \mu_3, \mu_4)$	d_{11}	d_{22}	d_{12}		Unbiased RE–EM tree			CART-based RE–EM tree		
					Uniform	11-valued	Binary	Uniform	11-valued	Binary
(11, 12, 13, 14)	0.00	0.00	0.00		86	90	87	96	84	91
(11, 12, 13, 14)	0.25	0.00	0.00		78	77	87	79	66	78
(11, 12, 13, 14)	0.50	0.00	0.00		78	76	83	68	70	70
(11, 12, 13, 14)	0.25	0.25	0.00		75	76	81	28	31	26
(11, 12, 13, 14)	0.50	0.50	0.00		69	69	78	19	15	21
(11, 12, 13, 14)	0.25	0.25	0.125		74	80	83	22	19	29
(11, 12, 13, 14)	0.50	0.50	0.25		65	72	77	19	14	27
$(\mu_1, \mu_2, \mu_3, \mu_4)$	d_{11}	d_{22}	d_{12}	σ_δ^2	Unbiased RE–EM tree			CART-Based RE–EM tree		
					RS X_1	RS X_4	Correct tree	RS X_1	RS X_4	Correct tree
(11, 12, 13, 14)	0.50	0.00	0.00	1	100	0	90	41	59	29
(11, 12, 13, 14)	0.50	0.00	0.00	5	100	0	90	100	0	65
(11, 12, 13, 14)	0.50	0.00	0.00	25	100	0	90	100	0	65
(11, 12, 13, 14)	0.50	0.50	0.00	1	100	0	71	37	63	4
(11, 12, 13, 14)	0.50	0.50	0.00	5	100	0	71	100	0	19
(11, 12, 13, 14)	0.50	0.50	0.00	25	100	0	71	100	0	22
(11, 12, 13, 14)	0.50	0.50	0.25	1	100	0	74	36	64	8
(11, 12, 13, 14)	0.50	0.50	0.25	5	100	0	73	100	0	21
(11, 12, 13, 14)	0.25	0.25	0.125	1	100	0	70	36	64	3
(11, 12, 13, 14)	0.25	0.25	0.125	5	100	0	70	100	0	25

The top part of the table gives simulation results without X_4 in the regression; the bottom part of table gives results when including X_4 in the regression.

With these changes, after repeating the experiment in Section 3.2, we obtain the results given in Table 2. One can easily see that these results are very similar to the ones in Table 1, which indicates that the conclusions in the paper with respect to the ability to recover the correct tree structure hold in the general longitudinal setting.

Fig. 8 examines predictive performance. The procedure is similar to the one in Section 3.3, except noise variables $X_5, \ldots, X_{10}$ are not included. Also, note that in the RIC cases, the linear part of random effect is based on the time index variable T instead of X_5 in the paper. The other settings of the simulation are identical to what is used in Section 3.3.

Once again, results are similar to those given earlier, and the unbiased RE–EM tree consistently outperforms the CART-based RE–EM tree. Fig. 9 gives corresponding results after including the correlated "competitor" variable X_4 in the regression. Results are similar to those in Fig. 8 and Section 3.3.

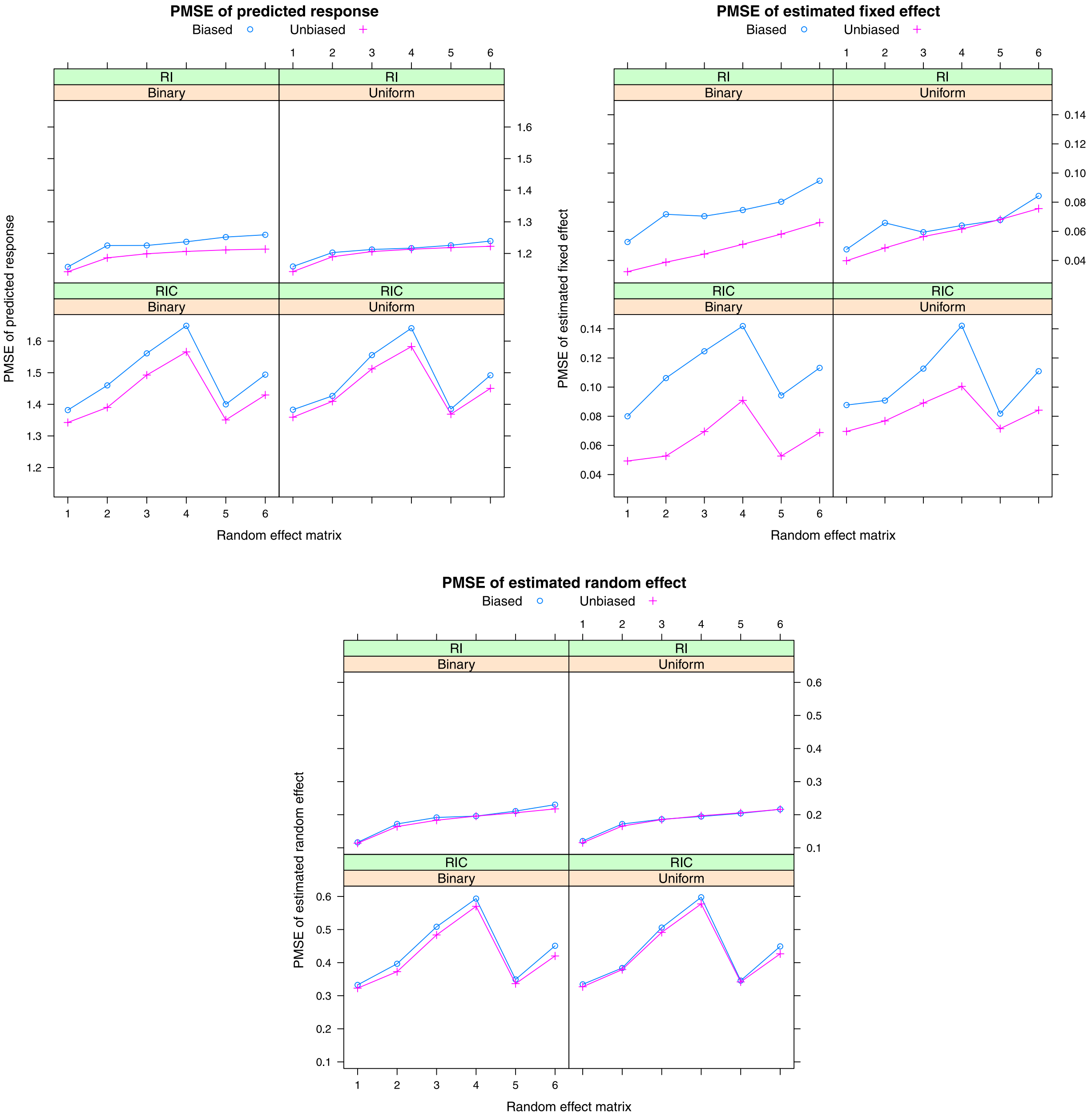

Fig. 7. PMSE of response y, fixed effect and random effect, respectively, when X_4 is included.

3.4.2. Unevenly-spaced time points

In the previous subsection, the time index variable T has the same equally-spaced values for all individuals. To incorporate unevenly-spaced time points, we modify the values of the time index valuable T in the following way:

- In the training set, T values for each individual are obtained by randomly sampling 5 integer values from 1 to 10 and ordering them in ascending order. Note that each individual has unevenly-spaced time points and potentially different time points from any other individual.
- In the testing data, for each individual, sample 50 values from the 100 evenly-spaced grid points on the interval [1, 10] and assign the ordered values to the time index variable values for that individual.

Predictive performance is summarized in Fig. 10. Comparing Fig. 10 to Fig. 8 shows that performance of the RE–EM tree is not affected by the presence of unevenly-spaced time points.

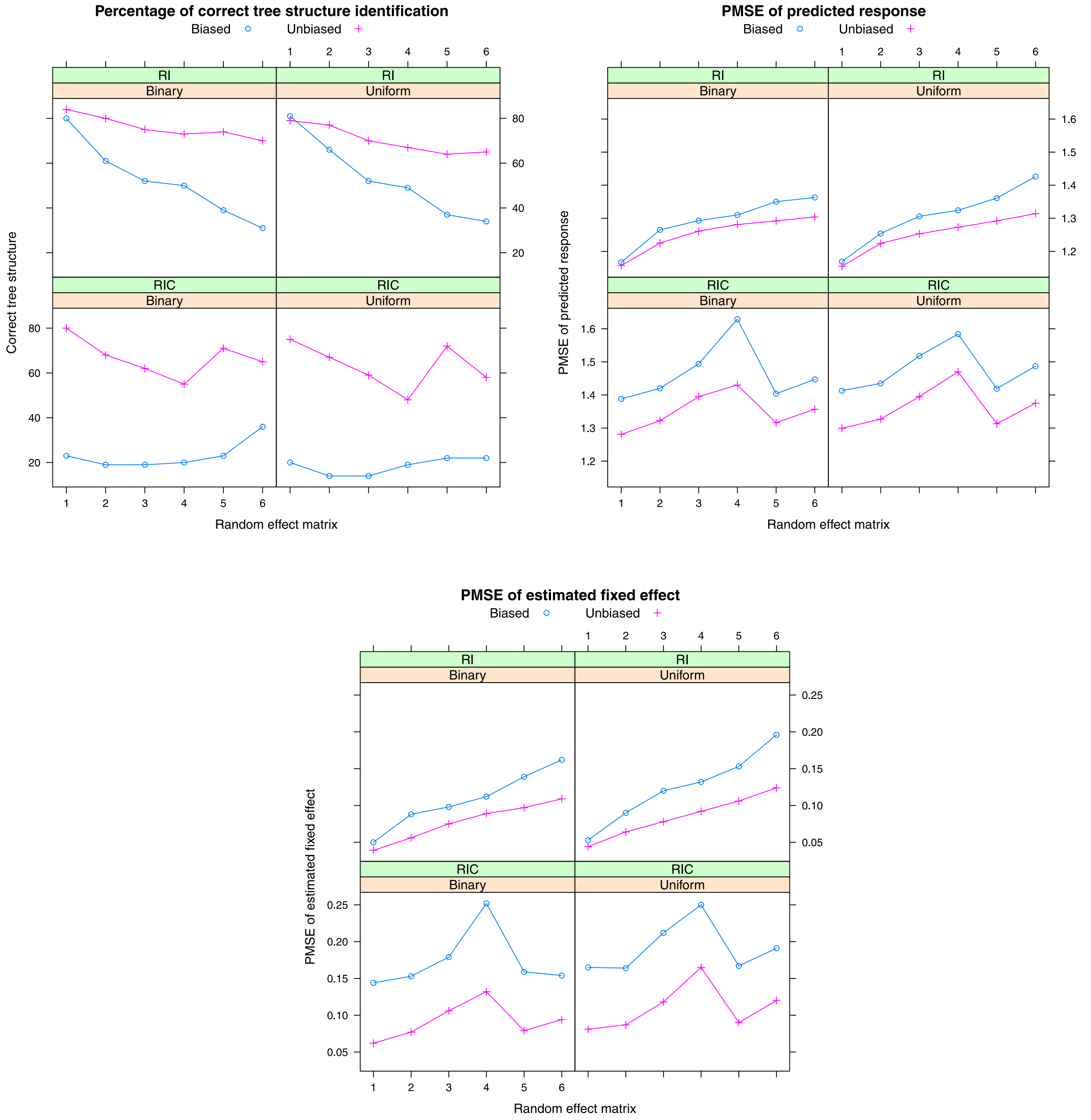

Fig. 8. Proportion of the simulation runs that the correct tree structure is recovered, PMSE of the response y, and PMSE of the fixed effect, respectively, when time is included in the random effect of the RIC model and X_4 is not included.

3.5. Unequal observations per object and time-invariant covariates

The simulations of the previous sections are all based on an equal number of observations for each individual. In practice, of course, there can easily be different numbers of observations for different objects (that is, an unbalanced design), so it is of interest to explore the behavior of the unbiased tree in that situation. The RE–EM tree methods only take into account the balance of the design in the estimation of the random effects from the linear mixed model fit in Step 2b of the algorithm, so it would be expected that they are relatively insensitive to lack of balance, and this is in fact the case. Indeed, in simulations in Sela and Simonoff (2012) the CART-based RE–EM tree was less sensitive to lack of balance than the linear mixed model itself. Simulations indicate that this insensitivity carries over to the unbiased tree. The ability to correctly recover and estimate the tree structure is quite insensitive to lack of balance, and overall predictive ability only begins to degrade when there is extreme lack of balance and a complex random effect structure, in that case associated with higher variance in predictions. Details can be found in the supplemental material available at the paper's associated web site.

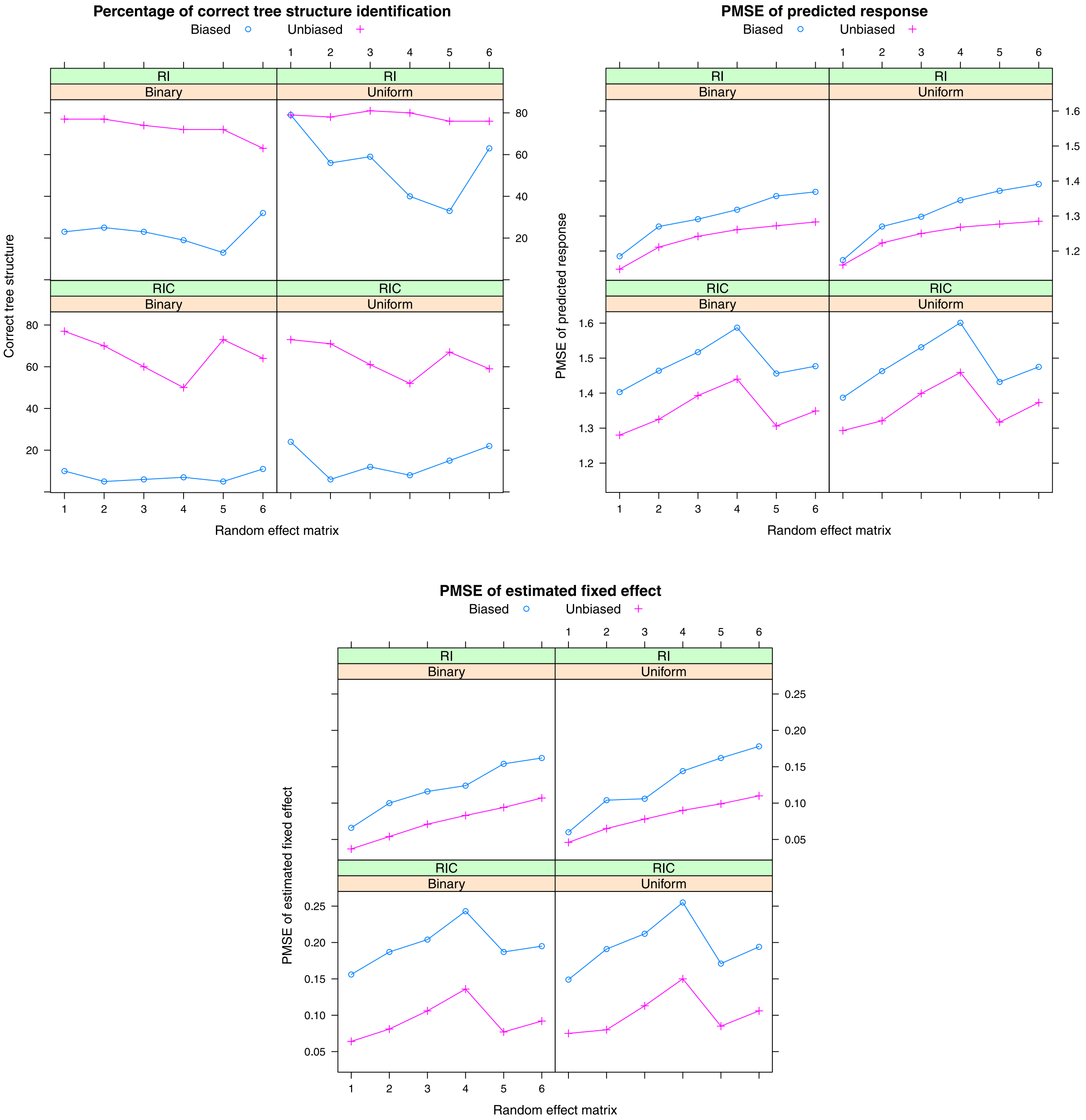

Fig. 9. Proportion of the simulation runs that the correct tree structure is recovered, PMSE of the response y, and PMSE of the fixed effect, respectively, when time is included in the random effect of the RIC model and X_4 is included.

As was noted earlier, whether covariates are time-invariant or time-varying is not relevant for RE–EM trees, since calculations are done at the observation level rather than the object level. Simulations that restrict predictors to be time-invariant do have the advantage of then allowing for a direct comparison with the longitudinal version of GUIDE (Loh and Zheng, 2013). The model here is similar to one used in Loh and Zheng (2013) that includes both a tree structure and a linear term in time as the true fixed effect, with

$$y_{it} = f\,(X_{i1}, X_{i2}, X_{i3}) + 0.5T_{it} + [1, T_{it}]\mathbf{b} + \varepsilon_{it},$$

where the random effects and $f\,()$ are the same as are used for the RIC model in Section 3.2, except that the covariates are time-invariant. Table 3 gives MSE values of the estimated fixed effects for each algorithm, along with ±1 standard deviation limits, based on 100 simulation replications (note that GUIDE does not provide estimates of random effects, and hence cannot be used for predictions at the observation level). It is apparent that both the unbiased RE–EM tree and GUIDE outperform the CART-based RE–EM tree. GUIDE and the unbiased tree have comparable performance, with GUIDE apparently doing better

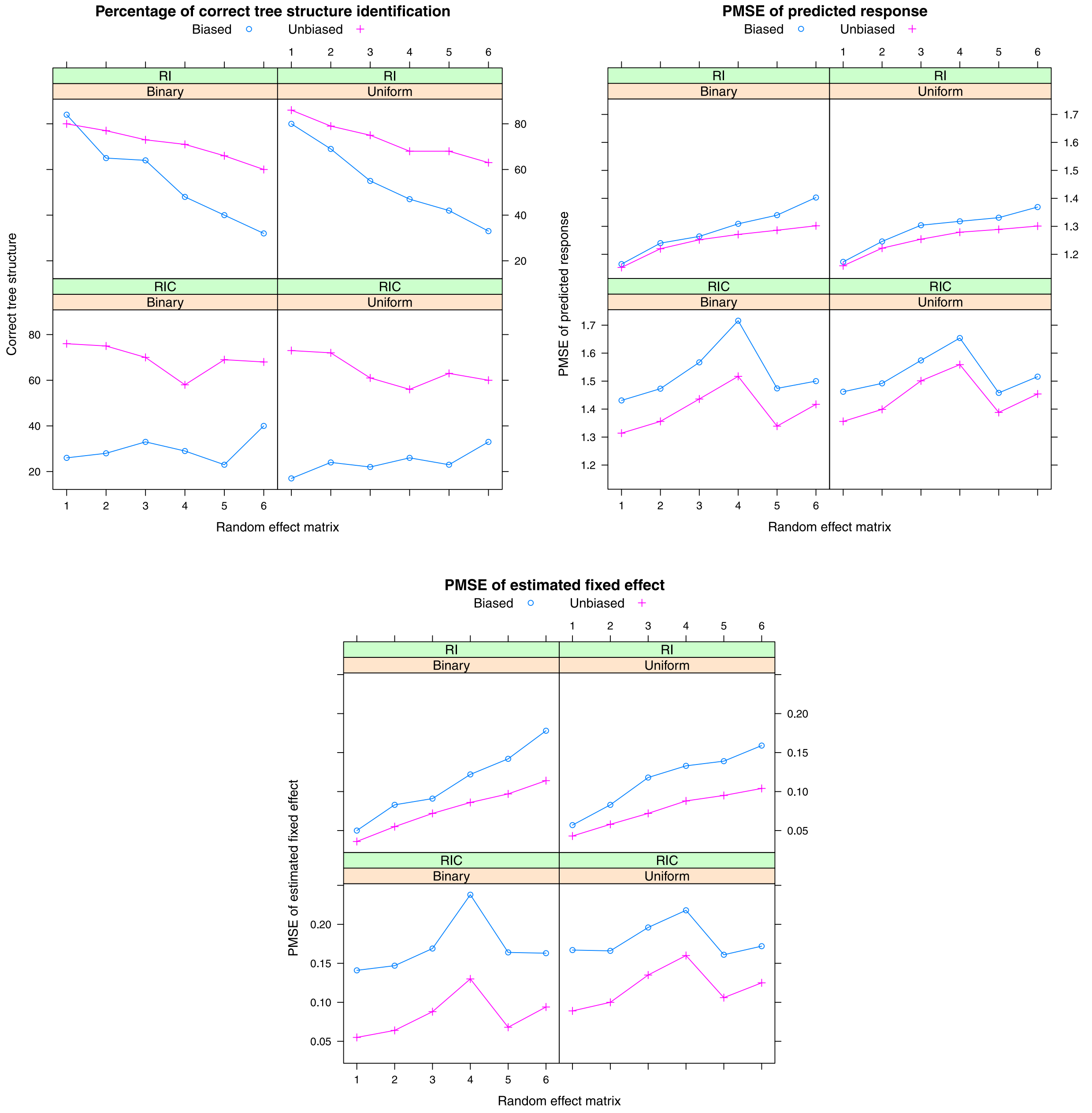

Fig. 10. Proportion of the simulation runs that the correct tree structure is recovered, PMSE of the response y, and PMSE of the fixed effect, respectively, when time is included in the random effect of the RIC model and X_4 is not included, for unevenly-spaced time points.

Table 3
Comparison of fixed effects MSE values between RE–EM trees and GUIDE for time-invariant covariates.

Covariance matrix	CART-based RE–EM tree	Unbiased RE–EM tree	GUIDE
D_1	1.1793 ± 0.5326	0.6675 ± 0.3545	0.6552 ± 0.6425
D_2	1.9967 ± 0.8537	1.2267 ± 0.5633	1.2620 ± 0.9040
D_3	3.4975 ± 1.3199	2.4406 ± 1.0025	2.2136 ± 0.6042
D_4	4.5796 ± 1.6911	3.3165 ± 1.0362	2.6422 ± 1.2586
D_5	2.4270 ± 1.0615	1.4730 ± 0.6586	1.6375 ± 0.9117
D_6	4.4712 ± 1.5577	4.0425 ± 1.3329	2.7460 ± 0.9880

when the coefficients of the random effects have higher variability. Of course, the advantage of the unbiased RE–EM tree over GUIDE is its ability to handle time-varying covariates.

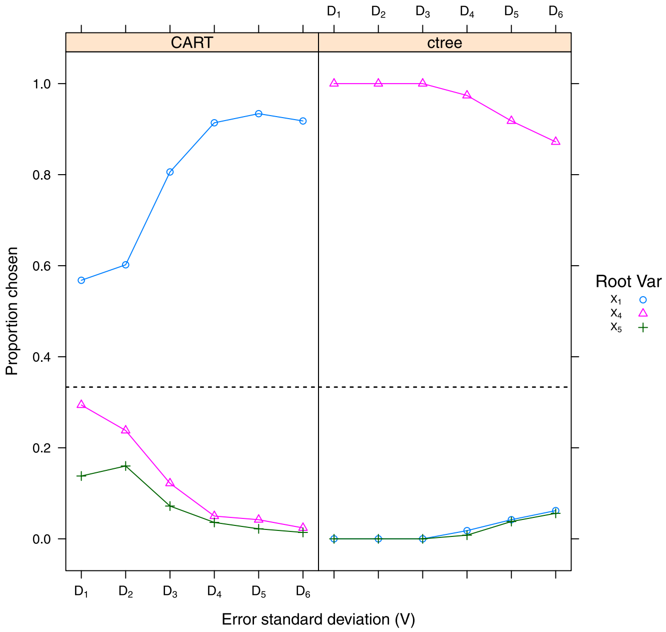

Fig. 11. Proportion of the simulation runs each predictor is selected as the root split variable in the non-null situation by the two tree algorithms, separated by variance of ε, for non-longitudinal data.

3.6. The "bias" of unbiased trees when associations are present in the data

As was noted earlier, in the literature an unbiased tree is defined as a recursive partition that selects each covariate with equal probability regardless of the measurement scales, when they are independent from response variable y. That is, the unbiasedness property of trees is considered under the null situation of independence between y and each covariate $X_1, \ldots, X_k$. A question that seems not to have been explored, however, is if the unbiased algorithm still has such "unbiasedness" under the non-null scenario, that is, when associations are present between y and the Xs. This section briefly explores the properties of both nonlongitudinal and longitudinal trees under non-null situations using simulation.

We first consider the nonlongitudinal situation. Data are generated similarly to the simulations described in Section 3.2. Three independent random variables X_1, X_2, X_3 are generated with uniform distributions in the interval [0, 10]. The response variable y is determined as follows:

- Leaf 1. If $x_{i1} \leq 5$ and $x_{i2} \leq 5$ then $y_i = 11 + \varepsilon_i$;
- Leaf 2. If $x_{i1} \leq 5$ and $x_{i2} > 5$ then $y_i = 12 + \varepsilon_i$;
- Leaf 3. If $x_{i1} > 5$ and $x_{i3} \leq 5$ then $y_i = 13 + \varepsilon_i$;
- Leaf 4. If $x_{i1} > 5$ and $x_{i3} > 5$ then $y_i = 14 + \varepsilon_i$,

where $\varepsilon_i \sim N(0, V^2)$. If $x_{i1} \leq C$ for some value of C, then $x_{i4} = 0$; If $x_{i1} > C$, then $x_{i4} = 1$. X_5 is ceiling of X_1, making X_5 ordinal with 10 values. C takes on the values {4.0, 4.2, 4.3, 4.5, 4.8, 5.0, 5.2, 5.5, 5.7, 5.8, 6.0} in the simulations. The tree fitting is based on using all of the predictors, with 500 simulation replications. For these data the correct root split is on X_1, with X_4 and X_5 providing alternatives with fewer possible split points. Prediction error is measured by the mean of *PMSE* on independent test data.

3.6.1. $C = 5$

When $C = 5$, a binary split on root node by $X_1 \leq 5$ is equivalent to a binary split by $X_4 \leq 0$, as well as $X_5 \leq 5$. Thus, an ideal "unbiased" algorithm should split on each of these variables at the root node with equal probability.

Fig. 11 presents the proportion of the time each covariate is selected as root split variable out of 500 trials. One can see that CART is biased towards the continuous variable X_1 as expected. What might be surprising is that the ordinal variable X_5 is split less than is the binary variable X_4, particularly when the error variance is smaller, which cannot be explained by the algorithm's bias towards covariates with more possible splits.

What is even more striking is that the unbiased *ctree* is, in fact, biased towards splitting on the binary variable over the other two (equivalent) possible split variables, presumably because the underlying test has more power to identify the (correct) split when it only has one split point to examine. Unsurprisingly, this does not negatively affect predictive performance (since splitting on X_4 is correct), and the *PMSE* of *ctree* is on average up to 10% smaller than that of CART in these simulations, with higher gains corresponding to higher error variance situations.

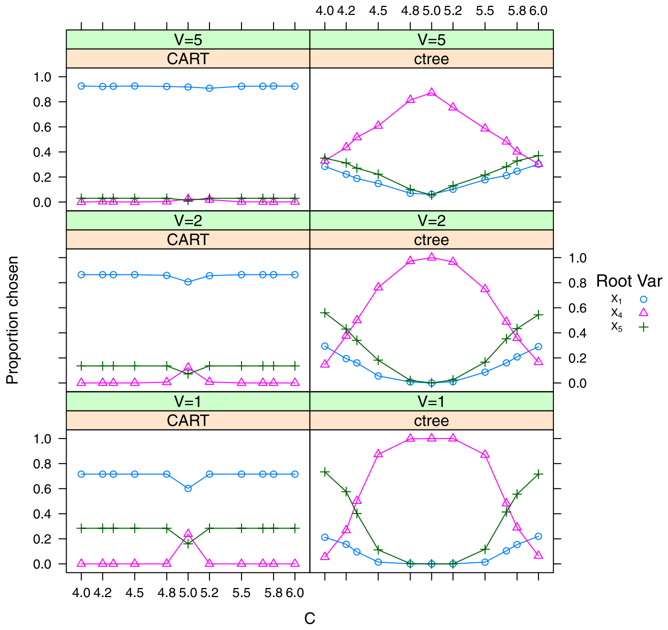

Fig. 12. Proportion of the simulation runs each predictor is selected as the root split by the two tree algorithms separated on varying C value and noise level for non-longitudinal data.

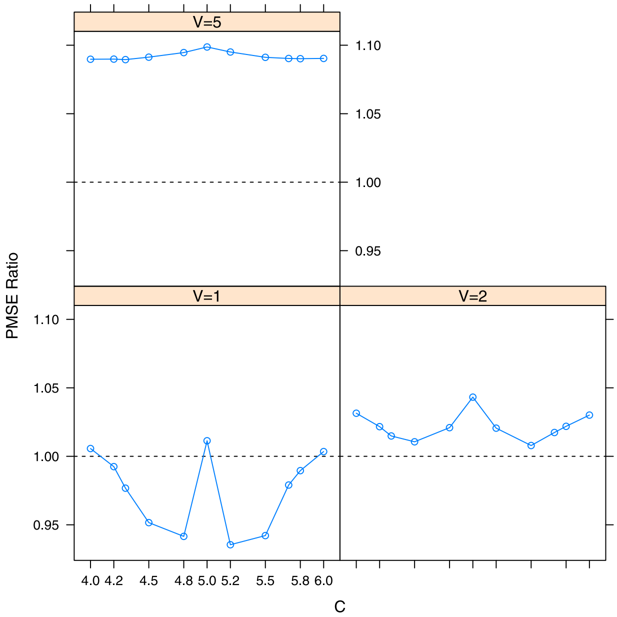

Fig. 13. *PMSE* ratio with varying C value and noise level for CART versus *ctree*.

3.6.2. $C \neq 5$

Varying C allows for a more detailed examination of the non-null situation, since when $C \neq 5$, a root split by $X_1 \leq 5$ or $X_5 \leq 5$ is still correct, while splitting by $X_4 \leq 0$ is not. It is possible that such a split (a root split on X_4) could still return an equivalent tree structure to the true data generating process, but this is not likely, especially when the noise level is high, i.e. V is large.

Fig. 12 shows the proportion of the time each covariate is selected as the root split variable by the two tree algorithms with varying C and noise level. One can see that while *ctree* tends to select the binary X_4 as the root split variable when C is close to 5, this tendency correctly decreases as C deviates from 5. For values of C close to 5, the "bias" of *ctree* in favor of a binary variable does affect the *ctree*'s ability to recover the correct tree structure. Fig. 13 summarizes the predictive performance of the two tree algorithms, plotting the ratio of CART *PMSE* to *ctree PMSE* (with values less than 1 favoring

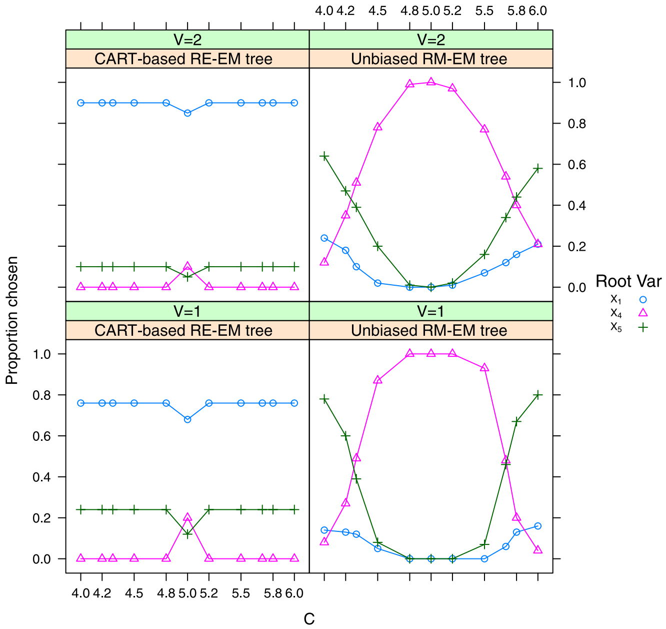

Fig. 14. Proportion of the simulation runs each predictor is selected as the root split variable in the non-null situation by the two tree algorithms, separated by variance of ε, for longitudinal data.

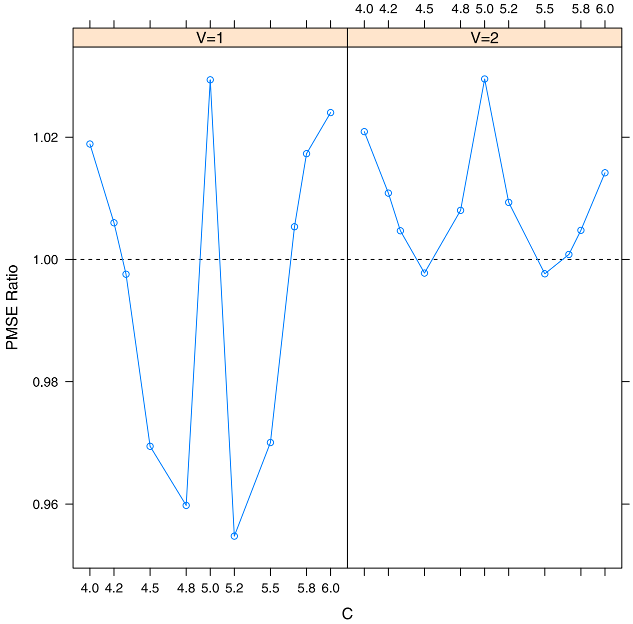

Fig. 15. *PMSE* ratio with varying C value and noise level, longitudinal data case.

CART). One can see that only in very limited cases is the predictive performance of *ctree* undermined, and even in those cases, the prediction accuracy of *ctree* is around 95% of that given by CART (or greater). Thus, the tendency for *ctree* to split on "wrong" binary variables tends to have little practical impact from a predictive point of view.

3.6.3. *The longitudinal case*

We now examine the effect of the "bias" of *ctree* under the non-null situation for longitudinal data. The setup is similar to that of Section 3.2, once again including binary (X_4) and ordinal (X_5) versions of X_1 in the fitting. Only the random intercept model is considered, $b_i \sim N(0, D)$ and $\varepsilon_i \sim N(0, V^2)$ as usual. Here, we fix $D = 1$ and vary the values of V. Results are given in Figs. 14 and 15.

Table 4
Proportion of the simulation runs that X_1 and X_4, respectively, are chosen as the root split variable, and the correct tree structure is recovered, using RE–EM trees for longitudinal data.

$(\mu_1, \mu_2, \mu_3, \mu_4)$	d_{11}	d_{22}	d_{12}	σ_δ^2	Unbiased RE–EM tree			CART-Based RE–EM tree		
					RS X_1	RS X_4	Correct tree	RS X_1	RS X_4	Correct tree
(11, 12, 13, 14)	0.50	0.00	0.00	1	99	1	76	100	0	63
(11, 12, 13, 14)	0.50	0.00	0.00	5	100	0	85	100	0	41
(11, 12, 13, 14)	0.50	0.00	0.00	25	100	0	81	100	0	49
(11, 12, 13, 14)	0.50	0.50	0.00	1	96	1	68	93	0	26
(11, 12, 13, 14)	0.50	0.50	0.00	5	87	0	58	97	0	16
(11, 12, 13, 14)	0.50	0.50	0.00	25	62	0	40	97	0	26
(11, 12, 13, 14)	0.50	0.50	0.25	1	97	1	66	96	0	16
(11, 12, 13, 14)	0.50	0.50	0.25	5	82	1	55	96	0	16
(11, 12, 13, 14)	0.25	0.25	0.125	1	99	1	74	100	0	17
(11, 12, 13, 14)	0.25	0.25	0.125	5	97	0	66	99	0	9

Here, X_4 is binary and X_1 is continuous. The correct root split is on X_1.

Table 5
Comparison of cross-validation results for CART-based and unbiased RE–EM tree methods for UCLA wage data and traffic fatality data. For each type of cross-validation the *p*-value of the signed rank test comparing the absolute errors of the two methods is given, along with the proportion of observations for which the absolute cross-validated predictive error was smaller for the unbiased tree.

	Leave-out-observation		Leave-out-individual	
	Signed rank *p*-value	Proportion smaller	Signed rank *p*-value	Proportion smaller
Wages data (Singer and Willett, 2003)				
Two covariates	0.170	0.507	0.068	0.505
Three covariates	$<2.2 \times 10^{-16}$	0.555	$<2.2 \times 10^{-16}$	0.529
Ten covariates	1.9×10^{-7}	0.535	1.1×10^{-11}	0.553
Traffic fatality data (Dee and Sela, 2003)				
	0.002	0.548	0.143	0.516

We can see that the simulation results for the longitudinal data cases are quite similar to the previous non-longitudinal cases, with the "bias" of *ctree* under the non-null situation only undermining its predictive performance in limited situations, supporting its use.

Another way to demonstrate that the "bias" of *ctree* has limited effect is using correlated predictors. Recall that in Sections 3.2 *and* 3.3 we have shown that when the root split variable is binary and a highly correlated competitor is uniform, the biased RE–EM tree based on CART will tend to falsely split on its competitor instead of the correct binary variable, which causes it to find the correct tree structure significantly less often, and translates into worse predictive performance. We can test to see if the same thing holds for *ctree*, i.e. if the root split variable is uniform and a highly correlated competitor is binary, will the "bias" of *ctree* cause it to falsely split on the binary competitor and worsen its predictive performance? We take $X_1 = X_4 + \delta$, where X_4 is binary $\{0, 5.77\}$ and $\delta \sim N\left(0, \sigma_\delta^2\right)$, with the root split on X_1. That is, everything is the same as in Section 3.2 except that X_4 is binary here instead of X_1 having been binary in Section 3.2.

From Table 4, we can see that when there exists a binary competitor variable X_4 in the regression, the *ctree*-based unbiased RE–EM tree rarely falsely splits on X_4 instead of the correct root split variable X_1. Further, the predictive performance of the *ctree*-based RE–EM tree continues to be better than that of the CART-based tree (results not given). This implies that the "bias" of *ctree* towards binary variables does not have the serious effects that the bias of CART towards continuous variables does.

4. Real data examples

We now compare the two RE–EM tree methods on a real data set. The wages data obtained from the UCLA Academic Technology Service website gives 888 individuals' hourly log wage (response variable) information and corresponding covariate values. It was previously studied by Singer and Willett (2003) and Eo and Cho (2014). The number of observations of each individual range from 1 to 13, so the data are highly unbalanced, with a total of 6402 observations. Eo and Cho (2014) were limited to time-invariant covariates in their applications of GUIDE and MELT to these data, using race (White, Black and Hispanic) and *hgc* (highest degree completed by each individual). The CART-based RE–EM tree is also given in Eo and Cho (2014), and it only splits once, on the *hgc* variable. Fig. 16 (produced using the `partykit` $\mathcal{R}$ package, Hothorn and Zeileis, 2014) gives the unbiased RE–EM tree, which has a more complex structure, broadly similar to that of GUIDE (as presented in Eo and Cho, 2014).

Table 5 compares the CART-based and unbiased RE–EM trees using predictive error based on two versions of validation: leaving each observation out and estimating the omitted log wage based on the rest of the data, and doing the same omitting all of the observations for each individual. The table gives the *p*-value of the signed rank test comparing the set of absolute cross-validated predictive errors, and the proportion of predictive errors in which the absolute error was smaller for the

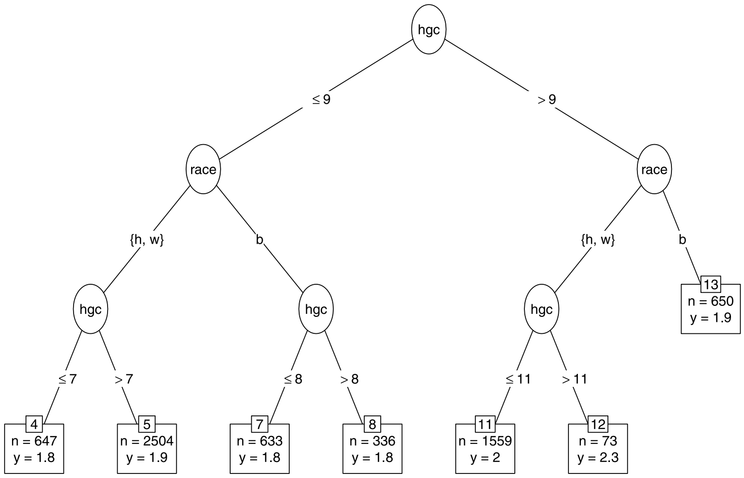

Fig. 16. Tree structure estimated by the unbiased RE–EM tree method for the wages data with two predictors.

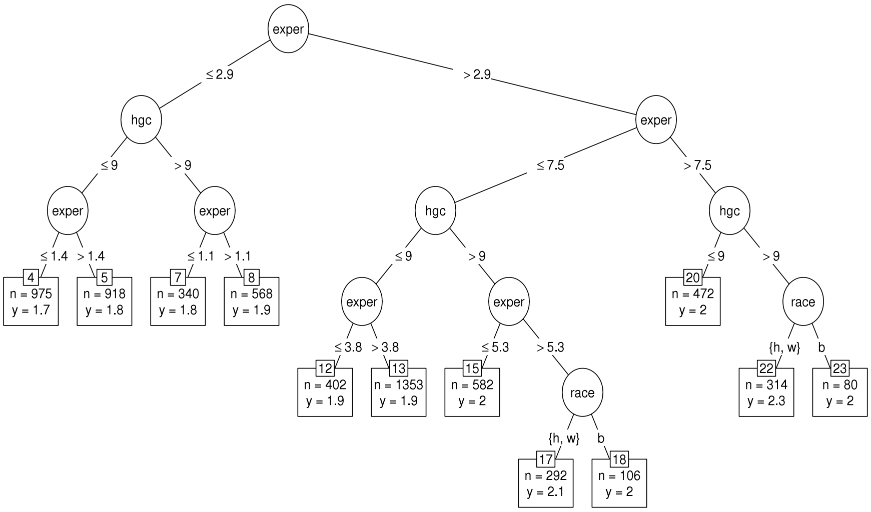

Fig. 17. Tree structure estimated by the unbiased RE–EM tree method for the wages data with three predictors.

unbiased RE–EM tree. The first line of the table shows that performance for the unbiased tree is better, but only marginally so, with barely more than 50% of the absolute predictive errors smaller and the *p*-values for the signed rank tests not statistically significant at a 0.05 level.

The RE–EM trees are not restricted to time-invariant predictors, of course, so Fig. 17 and the second line of the table, respectively, give results when adding *exper* (years of experience of the worker) to the potential predictors. The CART-based RE–EM tree splits twice on *exper*, partitioning the sample based only on the three *exper* groups defined by $\{<2.9, [2.9, 7.5), \geq 7.5\}$. The unbiased tree splits on all three variables, and its absolute cross-validated predictive errors are highly statistically significantly smaller than those of the CART-based tree, being smaller 55.5% (leave-out-observation) and 52.9% (leave-out-individual) of the time, respectively.

The original data set includes seven other potential covariates, and Fig. 18 and the third line of the table, respectively, summarize results using all 10 potential covariates. The CART-based RE–EM tree is identical to that when only three predictors are used, but the unbiased tree uses additional unemployment-related covariates. Table 5 shows that the unbiased tree's absolute cross-validated predictive errors are highly statistically significantly smaller than those of the CART-based tree, being smaller 53.5% (leave-out-observation) and 55.3% (leave-out-individual) of the time, respectively (that is, relative performance is slightly worse for leave-out-observation and slightly better for leave-out-individual when including all covariates).

Table 5 also summarizes cross-validated performance for the traffic fatality data set discussed in Sela and Simonoff (2012). The data set comes from Dee and Sela (2003), and are state-by-year data representing logged highway fatality rates per 100,000 population in 48 states (Alaska and Hawaii excluded) from 1982 to 1999 obtained from the Fatality Analysis Reporting System (FARS), for a total of 864 observations. Fixed effect predictors include the year, the speed limit, the drinking

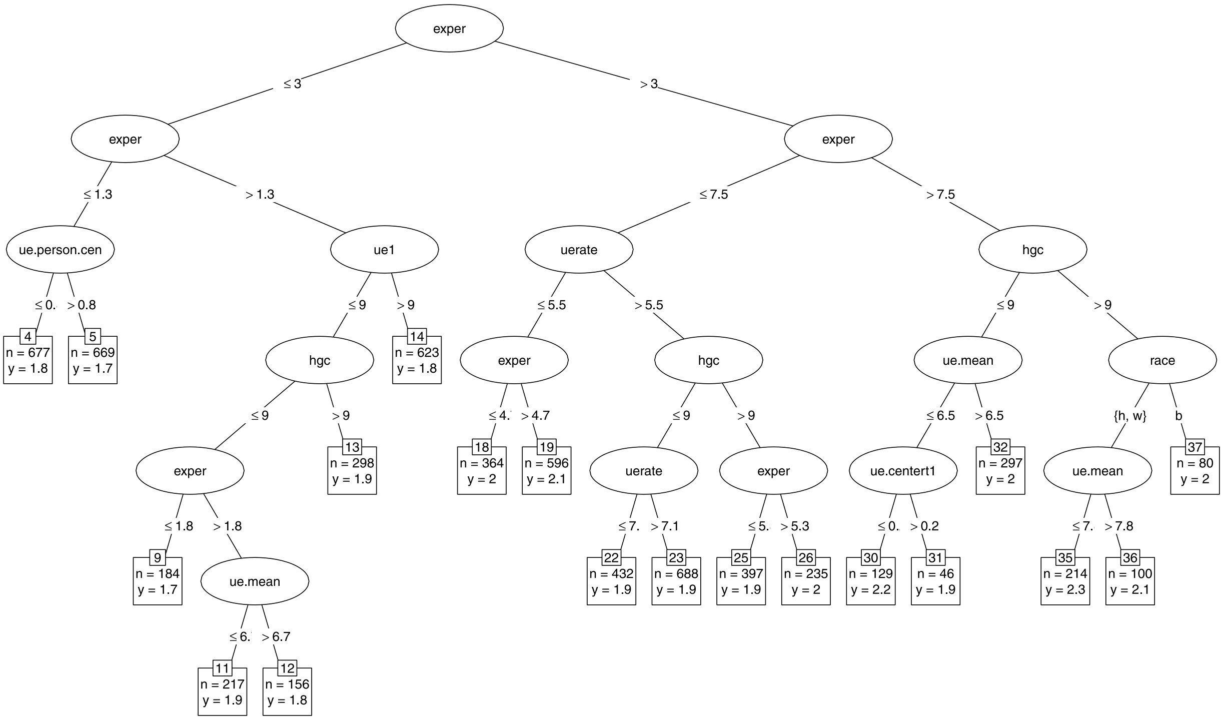

Fig. 18. Tree structure estimated by the unbiased RE–EM tree method for the wages data with 10 predictors.

age, the driving age, the existence of a mandatory seat belt law, the existence of a zero tolerance law for drivers to consume alcohol under age 21, the maximum blood alcohol level (BAC) at which it is legal to drive, the existence of an administrative license revocation law and the state unemployment rate. It can be seen that once again the unbiased RE–EM tree outperforms the CART-based tree, significantly so for leave-out-observation, where it has smaller absolute error 54.8% of the time.

5. Conclusions

We propose an unbiased RE–EM tree algorithm based on the original RE–EM tree method of Sela and Simonoff (2012). Through simulation study, we show that the proposed method is indeed apparently unbiased, not falsely splitting on a variable with more possible splits instead of the correct one. Simulation studies also show that the unbiased RE–EM tree has better performance in terms of recovering the correct tree structure and its predictive accuracy compares well to that of the original CART-based RE–EM tree method. Its performance is comparable to that of the longitudinal version of GUIDE when GUIDE can be applied, but is applicable for data with time-varying covariates (when GUIDE cannot be applied). The unbiased algorithm shares the desirable property of the underlying conditional inference tree that the selected tree is not based on a random (cross-validation) pruning algorithm. It is also possible that diagnostic tests based on whether a tree splits or not (such as those proposed in Simonoff, 2013) will have better control of null size, since the conditional inference tree is based on a hypothesis testing construction. An interesting topic for future research would be to see if the algorithm can be adapted to allow for functional relationships (such as growth curves related to time) at each terminal node, as is done in Eo and Cho (2014).

More information on the unbiased tree method, including supplemental material and implementation in $\mathcal{R}$, can be found at the associated web site at http://people.stern.nyu.edu/jsimonof/unbiasedREEM.

Acknowledgments

We would like to thank Patrick Perry for helpful discussion of this material, and Soo-Heang Eo for sharing computer code for GUIDE. We would also like to thank the referees for suggestions that led to improvements in the paper.

CART® is a registered trademark of California Statistical Software, Inc.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.csda.2015.02.004.

References

Breiman, L., Friedman, J., Olshen, R., Stone, C., 1984. Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA.

De'Ath, G., 2002. Multivariate regression trees: a new technique for modeling species–environment relationships. Ecology 83, 1105–1117.

Dee, T.S., Sela, R.J., 2003. The fatality effects of highway speed limits by gender and age. Econom. Lett. 79, 401–408.

Eo, S.-H., Cho, H., 2014. Tree-structured mixed-effects regression modeling for longitudinal data. J. Comput. Graph. Statist. 23, 740–760.

Hajjem, A., Bellavance, F., Larocque, D., 2011. Mixed effects regression trees for clustered data. Statist. Probab. Lett. 81, 451–459.

Hothorn, T., Hornik, K., Zeileis, A., 2006. Unbiased recursive partitioning: a conditional inference framework. J. Comput. Graph. Statist. 15, 651–674.

Hothorn, T., Zeileis, A., 2014. `partykit`: a modular toolkit for recursive partytioning in $\mathcal{R}$. Working Papers in Economics and Statistics, Research Platform Empirical and Experimental Economics, Universität Innsbruck, 2014-10 (http://EconPapers.RePEc.org/RePEc:inn:wpaper:2014-10).

Laird, N.M., Ware, J.H., 1982. Random-effects models for longitudinal data. Biometrics 38, 963–974.

Loh, W.Y., 2002. Regression trees with unbiased variable selection and interaction detection. Statist. Sinica 12, 361–386.

Loh, W.-Y., Zheng, W., 2013. Regression trees for longitudinal and multiresponse data. Ann. Appl. Stat. 7, 495–522.

Quinlan, J.R., 1986. Induction of decision trees. Mach. Learn. 1, 81–106.

Segal, M.R., 1992. Tree-structured methods for longitudinal data. J. Amer. Statist. Assoc. 87, 407–418.

Sela, R.J., Simonoff, J.S., 2012. RE-EM trees: a data mining approach for longitudinal and clustered data. Mach. Learn. 86, 169–207.

Simonoff, J.S., 2013. Regression tree-based diagnostics for linear multilevel models. Stat. Model. 13, 459–480.

Singer, J.D., Willett, J.B., 2003. Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence. Oxford University Press, Oxford, UK.

Strasser, H., Weber, C., 1999. On the asymptotic theory of permutation statistics. Math. Methods Statist. 2, 220–250.

White, A.P., Liu, W.Z., 1994. Technical note: bias in information-based measures in decision tree induction. Mach. Learn. 15, 321–329.

Zhang, H., 1998. Classification trees for multiple binary responses. J. Amer. Statist. Assoc. 93, 180–193.