Mach Learn (2012) 86:169–207
DOI 10.1007/s10994-011-5258-3

RE-EM trees: a data mining approach for longitudinal and clustered data

Rebecca J. Sela · Jeffrey S. Simonoff

Received: 9 February 2010 / Accepted: 14 June 2011 / Published online: 13 July 2011

Abstract Longitudinal data refer to the situation where repeated observations are available for each sampled object. Clustered data, where observations are nested in a hierarchical structure within objects (without time necessarily being involved) represent a similar type of situation. Methodologies that take this structure into account allow for the possibilities of systematic differences between objects that are not related to attributes and autocorrelation within objects across time periods. A standard methodology in the statistics literature for this type of data is the mixed effects model, where these differences between objects are represented by so-called "random effects" that are estimated from the data (population-level relationships are termed "fixed effects," together resulting in a mixed effects model). This paper presents a methodology that combines the structure of mixed effects models for longitudinal and clustered data with the flexibility of tree-based estimation methods. We apply the resulting estimation method, called the RE-EM tree, to pricing in online transactions, showing that 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. We also apply it to a smaller data set examining accident fatalities, and show that the RE-EM tree strongly outperforms a tree without random effects while performing comparably to a linear model with random effects. We also perform extensive simulation experiments to show that the estimator improves predictive performance relative to regression trees without random effects and is comparable or superior to using linear models with random effects in more general situations.

Keywords Clustered data · Longitudinal data · Panel data · Mixed effects model · Random effects · Regression tree · CART

Editor: Johannes Fürnkranz.

R.J. Sela (✉)
J.P. Morgan Chase & Co., Columbus, OH, USA
e-mail: rebeccapaul@yahoo.com

J.S. Simonoff
Statistics Group, Information, Operations, and Management Sciences Department, Leonard N. Stern School of Business, New York University, New York, NY, USA

1 Introduction

Some response data are one dimensional: observations over time or across objects. However, panel or longitudinal data, in which we observe many objects over multiple periods, offers a particularly rich opportunity for understanding and prediction, as we observe the different paths over time that a response variable might take across objects. Such data, often on a large scale, are seen in many applications, including business. Good examples are the tracking of transactions by individual customers over time and the tracking of purchases of individual products over time; the latter forms the basis of analyses in Sects. 4 and 5. The analysis of longitudinal data is especially rewarding with large amounts of data, as this allows the fitting of complex or highly structured functional forms to the data. Clustered data, wherein multiple observations can be viewed as being sampled within objects (for example, students within classes, classes within schools, etc.), also reflect this type of hierarchical structure. In this paper, we present a data mining approach that is specialized for longitudinal and clustered data with a numerical response variable. This method combines the flexibility of a data mining method with the specific nature of a longitudinal or clustered data set. In what follows we will generically refer to data of this type as longitudinal data, but the discussion applies equally well to other kinds of hierarchical structure.

Consider the following situation, which is based on one of the examples discussed in Sect. 4. A set of software titles is offered for sale by third party sellers on an online web site. The goal is to model or predict the price at which a software title sells (or, as in the example in Sect. 4, the price premium, which is the difference between the sale price and the average price in the market). Each title can have multiple sales from possibly different sellers at possibly different prices. Each sale has a set of attributes associated with it, including characteristics of the seller and market that could differ both between titles and between different sales for a given title (that is, they might be time-varying).

There are two types of tasks that an analyst might be interested in in this context: modeling and prediction. The modeling task is at the population level; that is, trying to understand the overall relationship between average prices or price premiums and attributes of the software titles from, for example, an economic point of view (examining the ways the market reacts to properties of the sellers, for example). The simplest solution to this problem would presumably be to fit a linear regression treating each individual sale as an independent observation, with price as the response (dependent) variable and the different attributes as (potential) independent variables. It is natural to suppose, however, that a more flexible relationship than a linear one could be supported by the data, particularly if the sample is large, which suggests consideration of methods such as nonparametric regression, regression trees, multivariate adaptive regression splines (MARS), model trees, neural networks, and so on.

Unfortunately, while any of these methods can be applied to these hypothetical software title data in this naive way, doing so would violate a fundamental assumption all share—that the observations (or, more precisely, the random errors relative to the expected response associated with the observations) be statistically independent of each other. For repeated sales data of this type (or more generally, repeated measurement (longitudinal) or clustered data) this will not be the case. Knowing that the price is higher than expected (based on the attributes) for one sale of a particular title, for example, provides information about prices for other sales of that title, because of characteristics of the title that do not depend on the attributes being used as predictors at the population level (that is, the errors within a title are apparently correlated). In other words, prices might be systematically higher or lower for a given title for reasons that are not part of the attributes used to predict prices at the population level. This could reflect a simple shift in price upwards or downwards on average for a given title (perhaps for reasons unknown or unavailable to the analyst), or it could be a function of other known properties of the particular sale, such as the type of software, the time of year of the sale, the manufacturer's suggested retail price of the software at the time of the sale, and so on (these might not be part of the population-level model because they do not reflect general market economics). Ignoring this induced correlation can result in overstatement of the strength of the relationship between the dependent and independent variables, and can result in the decision to choose models that are too complex. An additional possible source of correlation is autocorrelation of prices over time within a title even after taking the title effects into account (that is, knowing that a given title sold for more than expected could imply that its next sale will also be at a price higher than expected); ignoring the presence of this conditional autocorrelation can also affect inferential decisions.

An even more direct need for methods that account for the longitudinal structure of the data is in prediction. It is clear that a prediction of a future sale price based on hypothesized future attribute values for a title for which sales are already in the data set should take into account evidence in those previous sales that the title sells for systematically higher or lower prices than expected, something that is not possible using methods that treat each sale as an independent observation. Similarly, prediction of a future sale price for a new title (not in the original data set) for which information on past sales becomes available should evaluate evidence for systematic effects at the title level using those past sales and the already-fit longitudinal model, and then take those effects into account when predicting a future sale price, something that is once again not possible for methods that do not account for the longitudinal structure. A third possible prediction is at the population level: what is the "typical" price for a given set of attribute values over all possible titles? Since this prediction is not a function of an individual title, it is likely that this type of prediction from a longitudinal model would be similar to that from a corresponding model that ignores longitudinal structure, although the recognition of within-title structure should improve the accuracy of predictions somewhat.

A generalization of the linear regression model designed to address these issues is termed a linear mixed effects model. The goal of this paper is to generalize the linear mixed effects model to tree-based models. We first formalize notation and terminology. We observe a panel of objects $i = 1, \dots, I$ at times $t = 1, \dots, T_i$ (such objects are often called individuals in the longitudinal data literature, as they often correspond to individual patients in a medical trial; in the example above, these are the software titles). Throughout this paper, 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 attributes, $\mathbf{x}_{it} = (x_{it1}, \dots, x_{itK})'$ (in the example above the properties of the seller, for example), and a response, y_{it} (the sale price or price premium above). The attributes 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, Z_{it}, which may vary each period and depend on the attributes, and a vector of unknown time-constant, object-specific effects, $\mathbf{b}_i$. In the case where only the intercept varies across objects (in the example above, the only systematic difference in prices between software titles is a simple shift upwards or downwards on average), Z_i is a matrix of ones and b_i is the object-specific intercept, but in the more general situation where differences in prices for a particular sale of a particular title could depend on other attributes (such as time of year of the sale), the columns of Z_{it} would correspond to these attributes. This then implies a general effects model with additive errors:

$$y_{it} = Z_{it}\mathbf{b}_i + f(x_{it1}, \dots, x_{itK}) + \varepsilon_{it} \tag{1}$$

$$\begin{pmatrix} \varepsilon_{i1} \\ \vdots \\ \varepsilon_{iT_i} \end{pmatrix} \sim Normal(0, R_i) \tag{2}$$

$$\mathbf{b}_i \sim Normal(0, D) \tag{3}$$

Throughout this paper, we assume that the errors, ε_{it}, are independent across objects and are uncorrelated with the effects, $\mathbf{b}_i$. Note, however, that 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 b_i are taken as fixed or potentially correlated with the attributes, then this is a linear fixed effects model. Under the same assumptions about f, if the b_i are assumed to be random and uncorrelated with the attributes, then the model is a linear mixed effects model. Mixed effects models, when appropriate, are more efficient than fixed effects models, because the number of parameters estimated in a fixed effects model increases with the addition of more objects. This is especially important when T_i is small and I is large, as would often be the case in data mining applications. Furthermore, fixed effects models with object-specific intercepts (by far the most common kind) do not allow the inclusion of attributes that are always constant for objects, such as gender (when objects are people) or product type (when objects are products), because of collinearity, a serious drawback since such demographic-type variables are often of great interest to businesses and researchers. Finally, because the distribution of fixed effects b_i is not estimated, we have no basis for modeling the properties of the object-specific effects in predictions for objects not in the sample. For these reasons we will focus here on mixed effects models (that is, those that include random effects at the object level).

There are several approaches to fitting models with random effects in the literature. The two-stage approach, described by Harville (1977), yields estimates of the random effects, $\mathbf{b}_i$, instead of including them in the error terms as an alternative, the generalized least squares estimation method, would. These estimated random effects can be useful for prediction for new sales of objects already in the sample as described above, and are also crucial for the construction of the proposed tree estimator, so they are estimated in the methodology discussed here. We focus on the EM algorithm for two-stage mixed effects models given by Laird and Ware (1982). For more information on mixed effects models, including modified estimation procedures and extensions, see Patterson and Thompson (1971), Harville (1977), Laird and Ware (1982), and Verbeke and Molenberghs (2000).

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 too restrictive an assumption. The functional form of f is frequently unknown, and assuming a linear model may not be the best option. Furthermore, K may be very large, so that including all of the attributes directly may lead to overfitting and therefore poor predictions. In addition, linear models cannot include variables with missing values as many data mining methods can. A variety of nonparametric and data mining methods exist to estimate f in the case where $\mathbf{b}_i$ is constant across objects (that is, when random effects are unnecessary). We focus on regression trees, as described by Breiman et al. (1984), using the implementation of regression trees in the `rpart` package (Therneau and Atkinson 2010) of the statistical software package `R` (R Development Core Team 2009). Tree-based methods have been widely studied and applied in the statistics and data mining literature for 25 years, as discussed in Witten and Frank (2000, Sect. 3.7), Hastie et al. (2001, Sect. 9.2) Liu and Bozdogan (2004), Berk (2008), and many other references. An rpart regression tree is a binary tree, where each non-terminal node is split into two nodes based on the values of a single attribute. To find the predicted value for a response, one finds the correct terminal node based on the attributes and then takes the mean of all the response values in that node. This method allows for interactions between variables and can represent a variety of functions of the attributes. One could fit a regression tree to a longitudinal data set, ignoring the longitudinal data structure and assuming that $\mathbf{b}_i = \mathbf{0}$ for all i (that is, when random effects do not affect predictive performance), but as noted above, when such effects exist applying a nonparametric method designed for cross-sectional data directly to longitudinal data can be misleading and inefficient. Instead, we discuss a method that accounts for the additional longitudinal structure in the data.

We continue in Sect. 2 with a review of the existing literature on data mining methods for longitudinal data. In Sect. 3, we present and motivate the estimation method. In Sect. 4, we provide case studies of the analysis of Amazon third party transactions and of state-level traffic fatalities. These case studies demonstrate that the tree incorporating random effects can improve on both linear mixed effects models and ordinary regression trees in out-of-sample predictions for new observations and new objects. In Sect. 5, we use simulated data sets to explore the efficacy of the method, showing that these properties carry over to general situations. Section 6 concludes with a discussion of potential future work.

2 Previous applications of trees to numerical longitudinal data

Segal (1992) and De'Ath (2002), apparently independently, proposed the first application of regression trees to longitudinal data, in the case where $T_i = T$ for all i. Both created trees in which the response variable was the vector $\mathbf{y}_i = (y_{i1}, \ldots, y_{iT})$. At each node, a vector of means, $\mu(g)$, is produced, where $\mu_t(g)$ is the estimated value for y_{it} at node g. Note that these trees cannot be used for the prediction of future periods for the same objects. That is, if we observe $y_{i1}, \ldots, y_{iT}$ for each i, this method will not be able to predict $y_{i,T+1}$, since the means for period $T + 1$ must be constructed based on observations for that period. Notice that this approach uses a single set of attributes for all of the observation periods, since all of the elements of $\mathbf{y}_i$ lie in a single node. This prevents prediction of any observation using the values of time-varying attributes observed after the first period. This could easily lead to a loss of information and therefore poorer predictions. Alternatively, all of the periods of time-varying attributes could be used for predicting every observation; this would likely not make sense in many situations, since that would allow for attribute values from future time periods to be used in predicting response values from earlier time periods (for example, a model that requires knowing what the market will look like in the future is of little use to a seller or buyer who wishes to estimate an object's price now, and would be difficult to justify from an economic point of view). Given the central importance of predictive performance in data mining applications, these two limitations are quite serious in many practical applications, as we will see in Sect. 4. De'Ath's version of the tree is available as the R package mvpart (De'Ath 2006). Various authors, such as Larsen and Speckman (2004) and Hsiao and Shih (2007), have proposed alternative versions of this estimation method.

Work by Galimberti and Montanari (2002) developed a way to create trees that include both time-varying attributes and a longitudinal data structure. While their underlying model is similar to ours, their implementation is much more complex. They first assumed that the covariances of the errors and the random effects were estimated outside their procedure.

They then modified the split function to account for the correlation structure. Because they allowed for time-varying attributes, different observations for the same group could appear in different nodes; this made the split function particularly complicated (the method proposed here also allows different observations for the same group to appear in different nodes, but in a much more straightforward manner). Their algorithm is not generally available in software. Furthermore, they did not propose a way to handle observations with missing attribute values. Finally, because the group-specific effects are never estimated, one cannot predict future observations for objects already included in the sample, which is (as noted above) a serious deficiency. This paper will present an algorithm that accomplishes their goal in a more direct way, while also overcoming these weaknesses.

Other papers have also applied the tools of data mining to longitudinal data. Some followed the approach of Segal (1992), applying his method to other types of responses. Zhang (1998) considered the case of binary response variables; these are classification trees instead of regression trees. Lee (2005, 2006) and Lee et al. (2005) used generalized estimating equations to fit trees for general types of response variables. Their trees were not the traditional regression trees; instead, they estimated a parametric model using maximum likelihood at each node and then split based on the residuals from estimation. These methods also depend on a single set of attributes for all periods and cannot predict future observations for objects in the sample. Abdolell et al. (2002) discussed the use of trees to find clusters based on a single attribute and a longitudinal outcome variable. Ritschard et al. (2008) discussed data mining applications in the somewhat-related topic of event histories (although in that context the event responses are categorical rather than numerical). Ritschard and Oris (2005) applied classification trees to such data, taking lagged response values as potential predictors, but still not treating the response variable as inherently multidimensional. Other papers have considered data mining methods other than trees for longitudinal data. Zhang (1997) used adaptive splines to fit longitudinal data models, while Evgeniou et al. (2007) used ridge regression to fit models of consumer heterogeneity. We do not pursue either of these methods further.

3 The RE-EM tree estimation method

Consider again the general mixed effects model given in (1). Hajjem et al. (2008, 2011) and Sela and Simonoff (2009) independently proposed an estimation method that uses a tree structure to estimate f, but also incorporates object-specific random effects, $\mathbf{b}_i$, which we discuss further here. In this method, the nodes may split based on any attribute, so that different observations for the same object may be placed in different nodes. However, the method ensures that the longitudinal structure in the errors is preserved.

3.1 Longitudinal tree estimation

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 $f(x_i)$. Estimation methods for such models are included in most statistical packages. 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. This alternation between the estimation of different parameters is reminiscent of the EM algorithm, as used by Laird and Ware (1982); for this reason, we call the resulting estimator a Random Effects/EM Tree, or RE-EM Tree. Notice that regression trees are not fitted through traditional maximum likelihood methods; this means that this is not a true EM algorithm, so that the usual properties of the EM algorithm do not necessarily apply. More formally, the estimation method is given as follows:

Method Estimation of a RE-EM Tree

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 attributes, $\mathbf{x}_{it\cdot} = (x_{it1}, \ldots, x_{itK})$, for $i = 1, \ldots, I$ and $t = 1, \ldots, T_i$. Use this regression tree to create a set of indicator variables, $I(\mathbf{x}_{it} \in g_p)$, where g_p ranges over all of the terminal nodes in the tree.
   (b) Fit the linear mixed effects model, $y_{it} = Z_{it}\mathbf{b}_i + I(\mathbf{x}_{it} \in g_p)\mu_p + \varepsilon_{it}$. 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 fitting of the tree in Step 2a can be achieved using any tree algorithm, based on any tree growing and pruning rules that are desired, such as, for example, GUIDE (Loh 2002). In all of the examples and simulations performed here, tree building is based on the `R` function `rpart`, which is an implementation of the CART tree algorithm proposed in Breiman et al. (1984). The tree is a binary recursive splitting algorithm, in which splitting is based on maximizing the reduction in sum of squares for the node. Splitting continues as long as the increase in the proportion of variability accounted for by the tree (termed the complexity parameter, cp) is at least 0.001 and the number of observations in the node being considered for splitting is at least 20. Once the initial tree is formed, it is pruned based on 10-fold cross-validation. First, the tree with final split corresponding to the cp value with minimized 10-fold cross-validated error is obtained. Then, the tree with final split corresponding to the largest cp value with 10-fold cross-validated error that is no more than one standard error above the minimized value is determined; this is the final tree.

The linear model with random effects in Step 2b can be estimated using maximum likelihood or using restricted maximum likelihood (REML). In most of the results we present, we estimate the linear model with REML, because it yields unbiased estimates for the variance, R_i. Simulation results show that using maximum likelihood instead of REML has a very small effect on the resulting estimates. Basing the algorithm on a linear model with random effects also allows us to account for autocorrelation of errors within objects using existing estimation methods for linear models (by allowing for non-diagonal R_i in the model fitting), if necessary. Many statistical packages contain code to estimate linear mixed effects models; the `lme` function of the `R` `nlme` package (Pinheiro et al. 2009) is used here. It fits the model using a combination of the ECME algorithm (Liu and Rubin 1994), a modification of the EM algorithm designed to speed its convergence, and the Newton-Raphson algorithm.

A faster alternative that will also be explored here is to limit Step 2 above to one iteration. That is, an initial tree is fit ignoring the longitudinal structure, a mixed effects model is fit based on the resultant tree structure, and a final population-level tree is reported with the same structure, but with predicted responses that reflect the estimated random effects. This one-step approach only requires one linear mixed effects model fit above the computational cost of the original tree. If the fully iterated version is used, convergence is based on the change in the (restricted) log-likelihood being small enough (less than 0.001 in all results reported here).

A useful property of tree algorithms is that they typically include automatic procedures to handle missing values in the attributes X; for example, `rpart` uses surrogate split (Breiman et al. 1984). This results in the ability to produce estimated responses for objects with missing attribute values. This means that the RE-EM tree also can be fit when there are missing values in the attributes, since the tree fitting in Step 2a proceeds using (for example) surrogate split, while the estimating of $\mathbf{b}_i$, R_i, and D in Step 2b does not use X (Laird and Ware 1982), and hence is unaffected by the missing attribute values.

3.2 Allowing for autocorrelation within individuals

A simple test for whether autocorrelation should be included in the linear mixed effects model is to compare the predictive power of the model with and without autocorrelation. To test for autocorrelation in a linear mixed effects model more formally, we can use a likelihood ratio test. This test compares the log-likelihoods of the mixed effects fits in-sample with and without autocorrelation, correcting for the additional degrees of freedom (and therefore potential for overfitting) that the linear mixed effects model with autocorrelation uses. The two models used in this likelihood ratio test must have the same attributes. Generalizing the likelihood ratio test to RE-EM trees is not entirely straightforward because different trees will imply linear models with different attributes whenever the tree structures differ. Since the estimation method is iterative, the inclusion or exclusion of autocorrelation in the linear model can affect the estimated tree after the first iteration, so that the final estimated tree structures differ. Because of this, we conduct two likelihood ratio tests for autocorrelation: one where the attributes correspond to the RE-EM tree where autocorrelation is not allowed and one where the attributes correspond to the RE-EM tree where autocorrelation is allowed. In the examples we consider in Sect. 4, the two tests lead to identical conclusions.

3.3 Out-of-sample prediction

Given a RE-EM tree, the associated random effects, and the estimated covariance matrices, the out-of-sample predictions discussed earlier are straightforward. Suppose the tree is estimated on data for objects $i = 1, \dots, N_1$ for periods $t = 1, \dots, T_1$; for notational simplicity, we are assuming that all objects have the same number of observations, though this is not required. As was noted earlier, based on this training data set, we may be interested in three types of prediction:

1. Predicting observations for new objects for whom there are no past observations of the response: $i > N_1$ (that is, a population-level prediction).
2. Predicting future observations for objects in the sample: $t > T_1$ for $1 \le i \le N_1$.
3. Predicting future observations for new objects for which past observations are available: $i > N_1$ with the target observed for $t = 1, \dots, T_1$ and predictions for $t > T_1$.

For the first sort of prediction, we have no basis for estimating $\mathbf{b}_i$, so we set it to its expected value of $\mathbf{0}$, yielding the value $\hat{f}(x_{it1}, \dots, x_{itK})$. In this case, we might expect that methods that do not incorporate random effects would have comparable performance to those that do, as long as the sample is large enough so that $f(x_{it1}, \dots, x_{itK})$ is well-estimated by those methods. For the second type of prediction, we predict $f(x_{it1}, \dots, x_{itK})$ using the estimated tree and then add on $Z_i\hat{\mathbf{b}}_i$, which is known from the estimation process. In the third case, we can use the observations in the first T_1 periods to estimate $\hat{\mathbf{b}}_i$ based on the fitted $\hat{f}(x_{it1}, \ldots, x_{itK})$. Estimating the new random effect applies (3.2) of Laird and Ware (1982), with Z_i equal to the design matrix for the new object and R_i equal to the covariance matrix for the object based on the estimated parameters from the original model. We then proceed with prediction as when the random effects had already been estimated.

4 Applications to real data

4.1 Transactions data

In order to illustrate the use of the RE-EM tree, we now apply this method to two real-world data sets. The first example refers to data on third-party sellers on Amazon Web Services to predict the prices at which software titles are sold based on the characteristics of the competing sellers. See Ghose et al. (2005) for background on this data set and its first use. We will use the tree structure of the RE-EM tree to describe the factors that appear to influence prices. We also use the data set to compare the predictive performance of the RE-EM tree to that of alternative methods through two types of leave-one-out cross validation.

Our data consist of 9484 transactions for 250 distinct software titles; thus, there are 250 objects in the panel with a varying number of observations per object. (While there are also a few sellers who are included more than once, our longitudinal structure is based only on the products.) In this analysis, our target variable is the price premium that a seller can command; this is the difference between the price at which the good is sold and the average price of all of the competing goods in the marketplace. We also analyze the logged relative price premium, which is the logarithm of the ratio of those two quantities. Attributes include both the seller's own reputation and the characteristics of its competitors. The seller's reputation is measured by the total number of comments and the number of positive and negative comments received from buyers over different time periods. The length of time that the seller has been in the marketplace is also an attribute. Other attributes include the number of competitors, the quality of competing goods in the marketplace, the average reputation of the competitors, and the average prices of the competing goods. These variables allow us to see the effect of seller reputation and other characteristics on the prices that consumers will pay, which may allow sellers to set prices in a way that will encourage customers to buy from them.

We first fit a tree without random effects and a RE-EM tree to the data. The estimated regression tree without random effects is shown in Fig. 1, while the RE-EM tree is shown in Fig. 2. All trees presented here are plotted using the package `rpart.plot` (Milborrow 2011), with abbreviated variable names used to improve readability of the plot. The trees split on a variety of variables, and the structures of the two trees are noticeably different. For these data, a RE-EM tree that allows for autocorrelation, shown in Fig. 3, has very similar structure to a RE-EM tree that does not allow for autocorrelation. The two tests for autocorrelation lead to the same conclusion. The autocorrelation parameter is estimated to be 0.185 and the model without autocorrelation is strongly rejected ($p < 10^{-50}$) when we use either tree to compute the mixed effects models. The one-iteration version of the RE-EM tree without autocorrelation is given in Fig. 4; the structure of the tree is identical to that in Fig. 1 (as it must be), but the estimated population-level price premiums are different, since those in Fig. 4 take the title random effects into account.

For comparison, we fit linear models with and without random effects. Because some of the attributes have missing values, we cannot directly fit linear models that include all

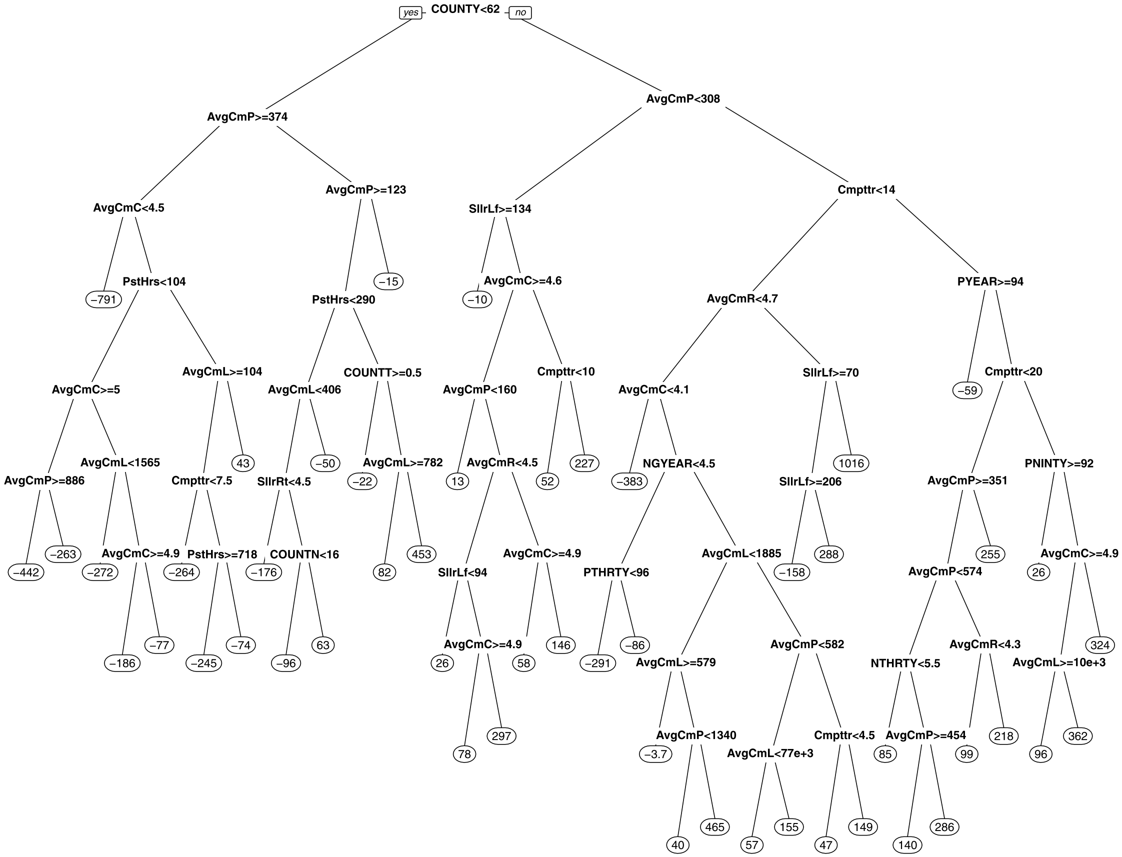


Fig. 1 Estimated tree without random effects for the price premium in the transactions data

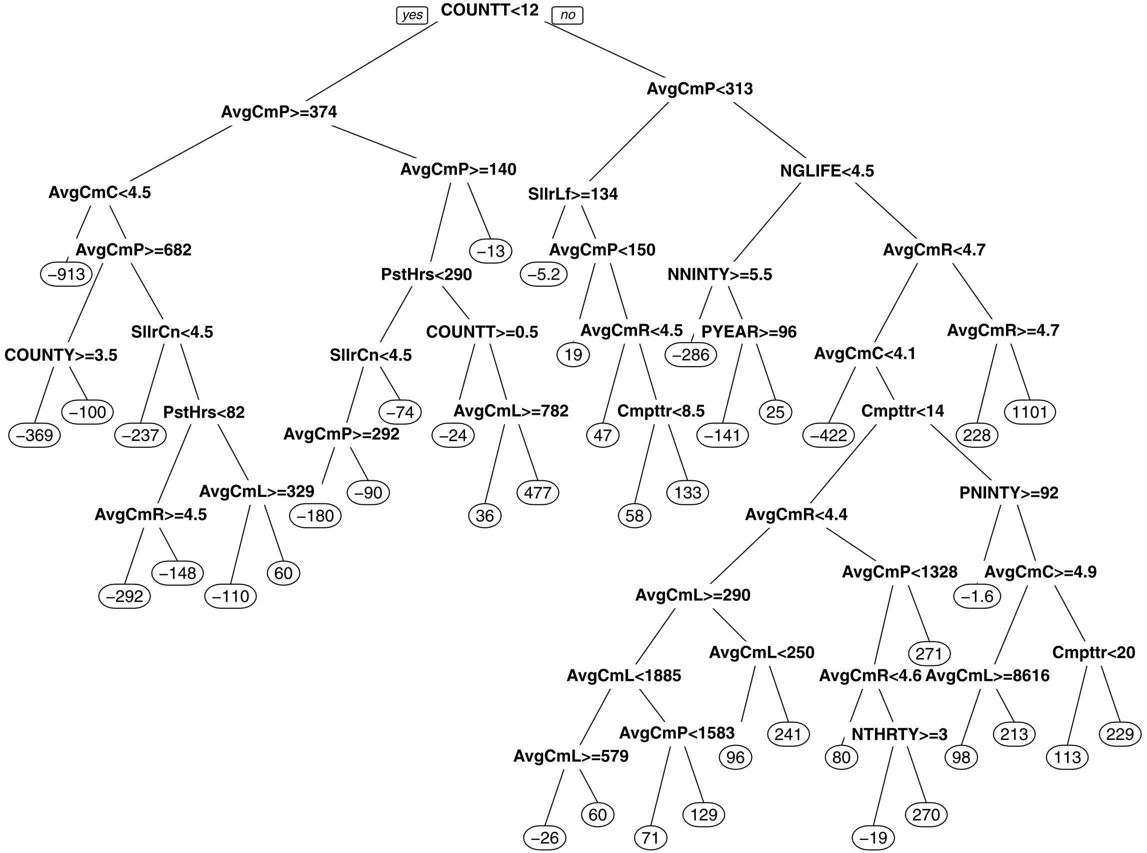


Fig. 2 Estimated RE-EM tree for the price premium in the transactions data

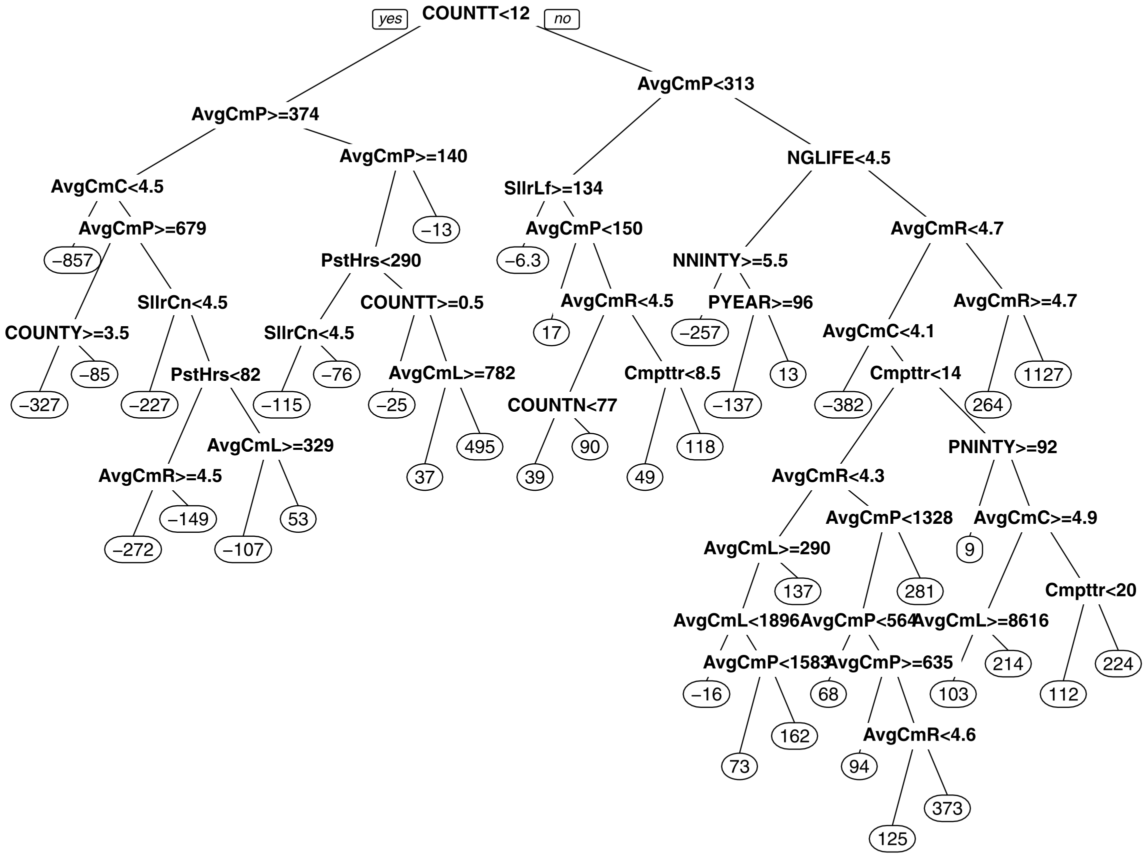


Fig. 3 Estimated RE-EM tree with autocorrelation for the price premium in the transactions data

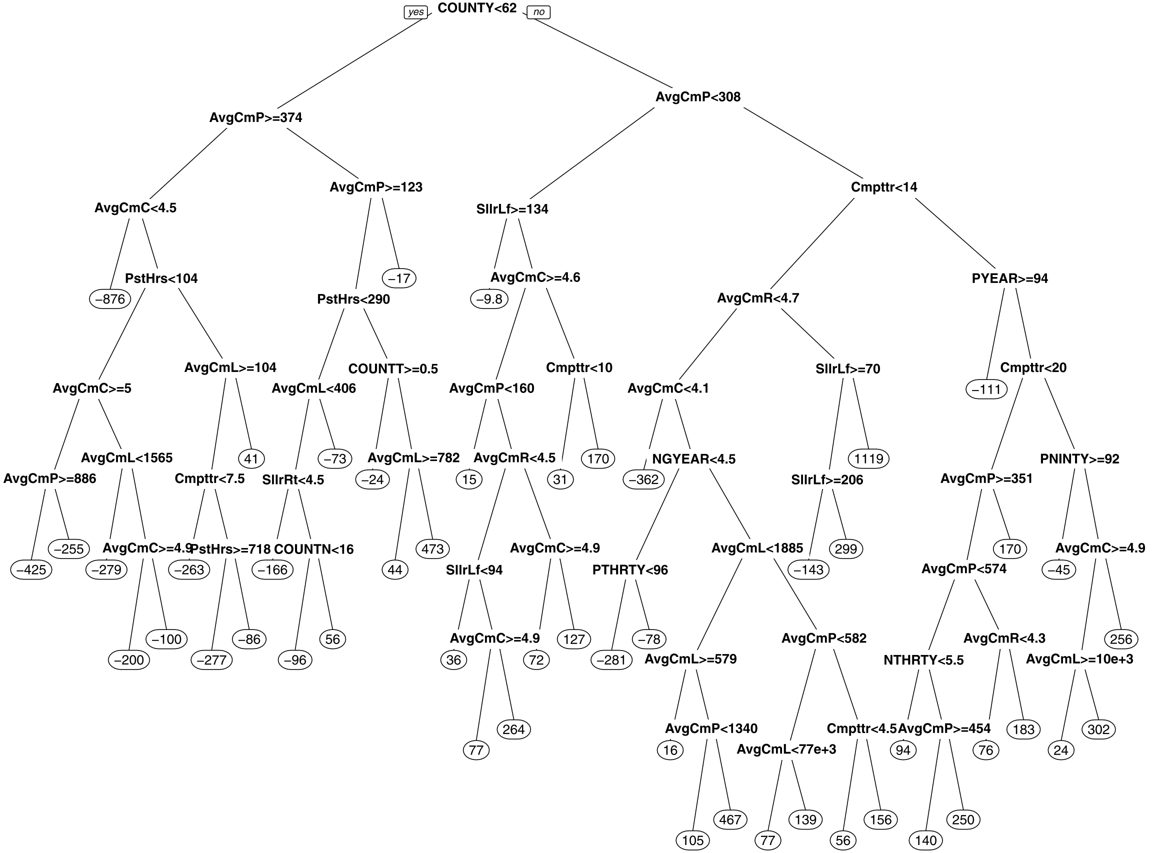


Fig. 4 Estimated RE-EM tree with one iteration for the price premium in the transactions data

of the possible attributes. Instead, we fit two versions of linear models: first, one that includes all of the attributes that appear in the RE-EM tree, since it happened that none of the attributes chosen for the RE-EM tree had missing values, and second, based on all of the independent variables after missing values are imputed. The variables with occasional missing values correspond to the proportion of comments about the seller that are positive, neutral, or negative, respectively, over time periods of the previous 30 days, 90 days, and one year, respectively. Within each time period the three comment variables can be considered a trinomial probability vector, so imputation proceeds by first fitting a multinomial logistic regression to the complete data with the counts of the different types of comments as the response, and the other attributes as the independent variables (Simonoff 2003, Chap. 10) and then estimating those proportions when they are missing using the fitted model and available independent variable values. The parameter estimates from the different linear models are given in Table 1. Few variables are statistically significant in the simpler linear model without random effects, while all of the variables are at least marginally statistically significant when random effects are included. Two of the variables that are statistically significant in the model without random effects, the average competitor price and the number of competitors, are statistically significant with the opposite signs when random effects are included; similar reversals can be seen in the models using imputed missing values. This underscores the importance of including random effects in the estimation of parameters. The average competitor price appears in the RE-EM tree several times; in one branch, lower competitor prices are associated with higher premiums, while in the other branch lower prices are associated with lower premiums. This ambiguous effect is impossible for a linear model without interactions to pick up and may explain why the coefficient changed sign from the linear model without random effects to the linear model with random effects.

We compare the trees and linear models using two different types of root mean squared errors $RMSE = [\sum(y_i - \hat{y}_i)^2/n]^{1/2}$; both are reported in Table 2, using leave-one-out cross-validation to measure out-of-sample prediction performance. To measure the performance when a random effect can be estimated, we exclude one transaction (observation) at a time, using the tree to estimate a random effect corresponding to an observation based on the other observations for that object. To measure the performance for new objects (where random effects are not used), we repeat the leave-one-out cross-validation by now excluding all of the observations for a single software title at each replication. For each type of cross-validation, we measure performance by the *RMSE* of prediction for the omitted observation(s). It can be seen that when single transactions are excluded, the linear models not including random effects have the largest *RMSE*, while the one-iteration RE-EM tree has the smallest *RMSE*. The difference in *RMSE* for the one-iteration RE-EM tree versus the `rpart` tree (without random effects) is of greater practical importance than might be supposed from the values in the table, as the former method has smaller absolute predictive error than the latter for 65% of the cross-validated observations. When all transactions are excluded for one title at a time, the linear models with random effects perform much worse. We believe that this reflects the fact that a linear population-level functional form is not appropriate here (as is apparent from the higher *RMSE* values for the linear models), which has hurt estimation of the population-level f in the linear mixed model more than in the ordinary linear model without random effects. When individual transactions are omitted the estimated random effect for that transaction's title (based on the other transactions for that title) can help recover to some extent from the poor population-level estimate of f, but when all transactions for that title are omitted the prediction is only based on the more poorly-estimated population level f. Again, the one-iteration RE-EM tree performs best, though its *RMSE* is not very different from the *RMSE* of a regression tree without random effects (recall that this is to be expected, since no estimated random effect is used for the "new" title).

Table 1 Parameter estimates for the linear models for the price premium with and without random effects. Standard errors are reported in parentheses

Variable	Linear model	Mixed effects model	Mixed effects model with autocorr.	Linear model (imputed)	Mixed effects model (imputed)	Mixed effects model with autocorr. (imputed)
(Intercept)	88.800**	501.756***	330.62***	−134.475***	228.226***	151.753***
	(34.895)	(52.742)	(44.60)	(40.72)	(56.730)	(48.847)
Average competitor price (`AvgCompPrice`)	0.064***	−1.654***	−1.367***	0.063***	−1.678***	−1.399***
	(0.004)	(0.031)	(0.027)	(0.004)	(0.031)	0.027
Average condition of competing goods (`AvgCompCondition`)	−0.218	12.231*	14.760**	−12.816***	32.659***	25.345***
	(4.943)	(7.323)	(6.292)	(4.922)	(7.432)	(6.474)
Average rating of competitors (`AvgCompRating`)	7.168	−22.043***	−17.078***	14.131***	−27.951***	−19.389***
	(4.764)	(6.044)	(4.985)	(4.697)	(5.955)	(5.009)
Life of the seller (`SellerLife`)	0.001	0.002**	0.001*	0.001	0.002**	0.001*
	(0.001)	(0.001)	(0.0006)	(0.001)	(0.001)	(0.001)
Number of competitors (`Competitors`)	2.115***	−1.099***	−0.864**	2.216***	−0.653	−0.885**
	(0.160)	(0.418)	(0.345)	(0.159)	(0.423)	(0.357)
Lifetime positive comments (`PLIFE`)	−1.659***	−1.615***	−0.661***	−1.874***	−2.490***	−0.929**
	(0.099)	(0.084)	(0.084)	(0.602)	(0.472)	(0.431)
Number of comments in the last year (`COUNTYR`)	−0.001	−0.002*	−0.0015	0.002	−0.001	−0.003
	(0.001)	(0.001)	(0.001)	(0.002)	(0.002)	(0.002)
Average lifetime of competitors (`AveCompLife`)				0.00006*	0.0002***	0.0001**
				(0.00003)	(0.00006)	(0.00005)
Hours item was posted for sale (`PostHours`)				0.009	0.004	0.016***
				(0.006)	(0.005)	(0.004)
Item condition (`SellerCond`)				24.810***	23.401***	21.405***
				(2.407)	(2.070)	(1.959)
Seller rating (`SellerRating`)				9.325	10.615	12.231
				(12.653)	(9.777)	(8.912)
30-Day positive comments (`PTHRTY`)				0.880***	0.926***	0.504
				(0.194)	(0.149)	(0.127)
90-Day positive comments (`PNINTY`)				1.736***	1.777***	0.696***
				(0.315)	(0.242)	(0.218)
365-Day positive comments (`PYEAR`)				−1.586***	−1.512***	−1.200***
				(0.508)	(0.397)	(0.350)
30-Day neutral comments (`NTHRTY`)				−1.213***	−1.610***	−1.185***
				(0.304)	(0.244)	(0.207)

Table 1 (*Continued*)

Variable	Linear model	Mixed effects model	Mixed effects model with autocorr.	Linear model (imputed)	Mixed effects model (imputed)	Mixed effects model with autocorr. (imputed)
90-Day neutral comments (NNINTY)				1.747^{***} (0.568)	2.415^{***} (0.436)	1.388^{***} (0.373)
365-Day neutral comments (NYEAR)				0.728 (0.997)	0.290 (0.759)	−0.253 (0.679)
Lifetime neutral comments (NLIFE)				0.330 (0.972)	−0.675 (0.742)	0.292 (0.686)
Number of comments in the last 30 days (COUNTTH)				0.003 (0.011)	0.013 (0.008)	0.009 (0.007)
Number of comments in the last 90 days (COUNTNY)				−0.007 (0.006)	-0.008^{*} (0.005)	−0.001 (0.004)

*Significantly different from zero at the 10% level; **significantly different from zero at the 5% level; ***significantly different from zero at the 1% level

Table 2 *RMSE*s from cross-validation leaving out one observation or one software title at a time, using the transactions data, using the price premium

Method	Excluding observations	Excluding titles
Linear model	95.88	96.92
Linear model with random effects	73.62	461.48
Linear model with random effects—AR(1)	74.75	387.18
Linear model (imputed)	94.28	96.00
Linear model with random effects (imputed)	73.27	465.38
Linear model with random effects—AR(1) (imputed)	74.09	393.50
Tree without random effects	54.42	87.34
RE-EM tree	55.96	90.03
RE-EM tree—AR(1)	55.13	89.44
RE-EM tree (1 iteration)	51.12	86.27
RE-EM tree—AR(1) (1 iteration)	51.19	85.39

Diagnostic plots for the RE-EM tree and linear model (not shown here, but available online at http://www.stern.nyu.edu/~jsimonof/REEMtree) highlight some potential violations of the mixed effects model assumptions, including possible heteroscedasticity and fat tails in the residuals. Because of this, we consider an alternative functional form of the target variable, the logged relative price premium, which is the logarithm of the sale price divided by the average price of the competing goods (note that the presence of heteroscedasticity is a potential issue for two reasons: first, the tree algorithm is based on an unweighted reduction in sum of squares, when a weighted one would be appropriate, and second, the linear mixed effects fit is based on an assumption of constant variance in the errors). The fitted trees without random effects, with random effects, with random effects and autocorrelation, and without autocorrelation based on one iteration of the algorithm are plotted in Figs. 5, 6, 7, and 8, respectively. The likelihood ratio test for autocorrelation rejects the hypothesis of no autocorrelation ($p < 10^{-50}$).

As before, we fit linear models with and without random effects to these data, using the attributes chosen by the RE-EM tree (as with the price premium, none of the chosen attributes has missing values) and using all of the attributes after imputing missing values (Table 3). Many of the attributes chosen by the RE-EM tree have coefficients that are not significantly different from zero in the linear models.

Diagnostic plots (not shown) for the estimates for the logged relative price premium show that taking the logarithm has reduced the heteroscedasticity somewhat, but that non-normality and outliers remain. Plots of the residuals versus the fitted values for the RE-EM tree and linear mixed effects model show a large negative outlier, but little evidence of heteroscedasticity. Omitting the outlier and re-estimating has little effect on the estimates.

We again compute the *RMSE* for predictions using leave-one-out cross validation in which we omit one observation at a time and then one title at a time. The results are given in Table 4. Once again, the trees outperform the linear models, and the RE-EM trees outperform the `rpart` tree when omitting observations. All of the tree methods have very similar *RMSE* when we exclude all the observations for the title (again, the closeness of performance for this measure is not surprising). Thus, for these data, the benefits of using a tree-based model occur for both population and object-level predictions, and accounting for longitudinal structure is beneficial for predictions at the object level; combining both in the RE-EM tree provides best performance overall.

4.2 Accident fatality data

In this section we describe the analysis of a smaller data set. The data are described and discussed in Dee and Sela (2003), and refer to the highway fatality rate in states of the U.S. from 1982–1999, and how they relate to changes in driving laws (65 or 75 mile per hour speed limit, mandatory seat belt, blood alcohol limit) and state unemployment rate (a proxy for business activity). The response variable is the logged traffic fatality rate per 100,000 population of all drivers, while predictors include the year, the state speed limit for that year (consisting of the five categories 55 MPH, 65 MPH, 70 MPH, 75 MPH, or no speed limit), the drinking age, the driving age, the presence in the state of a mandatory seat belt law, the presence of a zero tolerance law for drivers under the age of 21 related to consuming alcohol, the minimum blood alcohol level (BAC) at which it is illegal to drive (0.10 or 0.08), the presence of an administrative license revocation law, whereby the state licensing authority is allowed to suspend a driver's license prior to any court action, and the state unemployment rate. The data thus consist of 48 objects (states, excluding Alaska and Hawaii), each measured 18 times, and is thus much smaller than the transactions data set analyzed previously; for this reason, it would not be surprising for a linear model to be comparatively effective in this case.

Dee and Sela (2003) fit a fixed effects linear model (including 47 indicator variables to account for state effects), but we will use a mixed effects model (fitting state using random effects) here for comparative purposes. Table 5 gives the results of fitting linear models to the data.It can be seen that in the ordinary linear model (where no state effects are accounted for) speed limit (which is fit using four indicator variables, with the 55 MPH speed limit being the reference category) has a very strong effect on (logged) fatality rate, with the higher speed

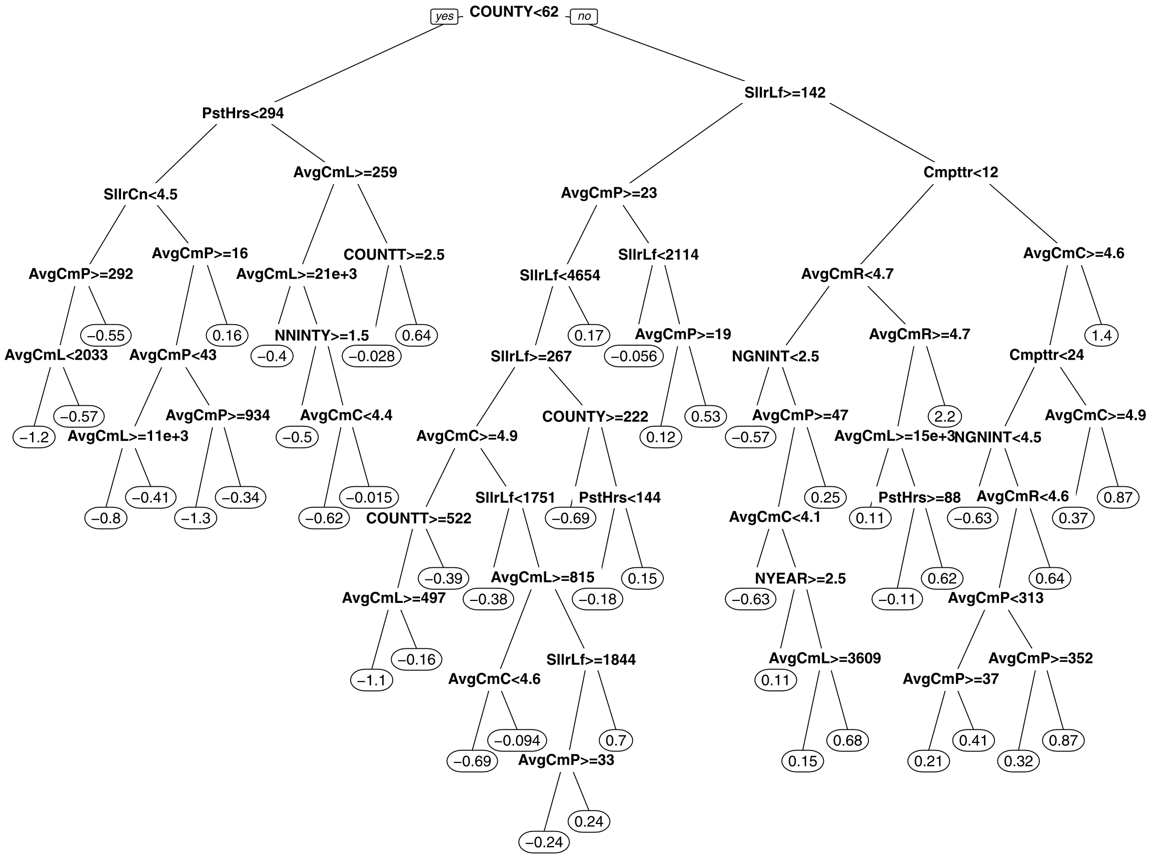


Fig. 5 Estimated tree without random effects for the logged relative price premium in the transactions data

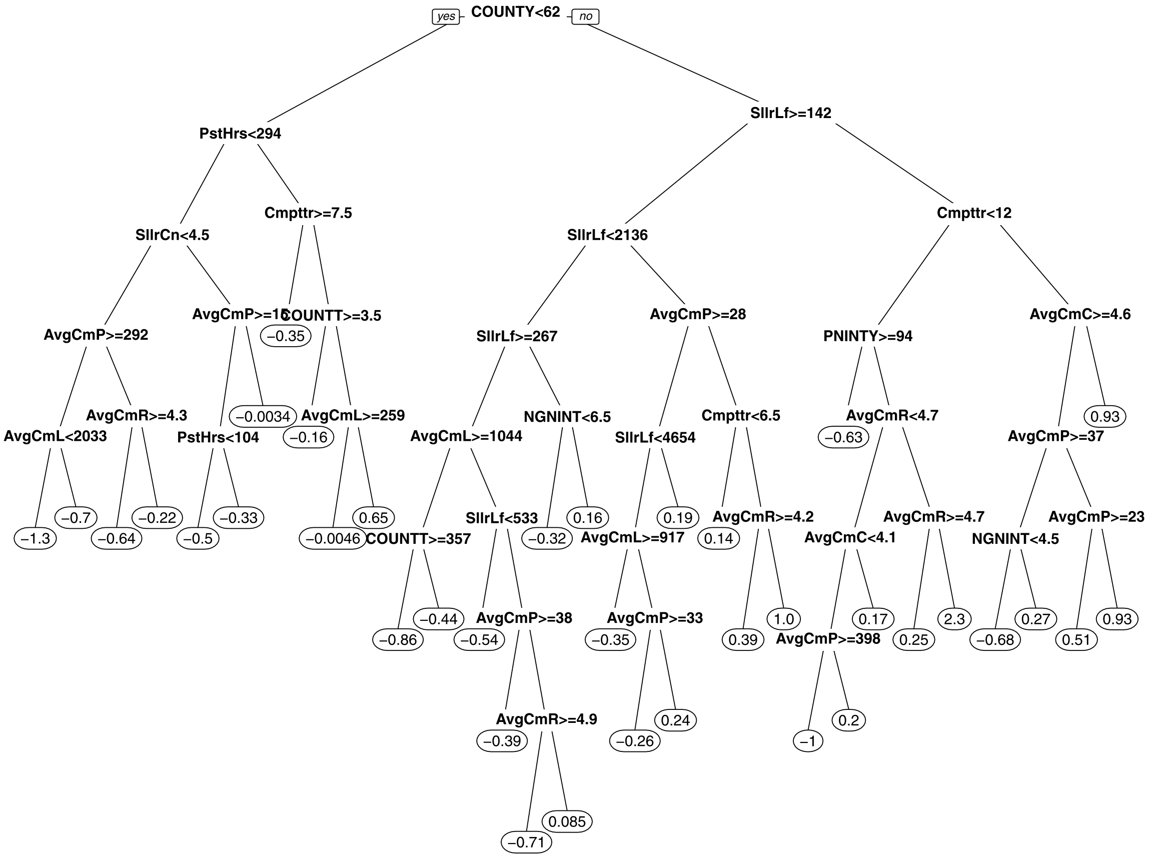


Fig. 6 Estimated RE-EM tree for the logged relative price premium in the transactions data

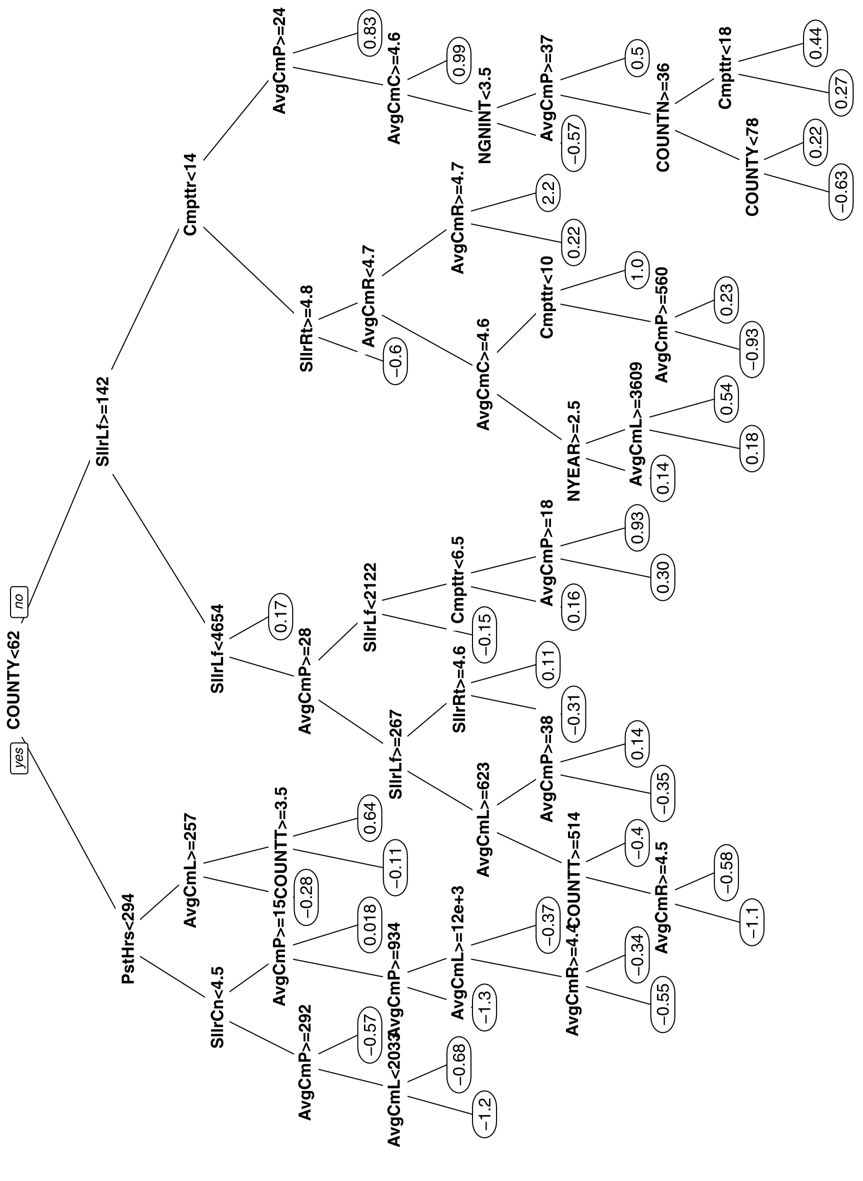


Fig. 7 Estimated RE-EM tree with autocorrelation for the logged relative price premium in the transactions data

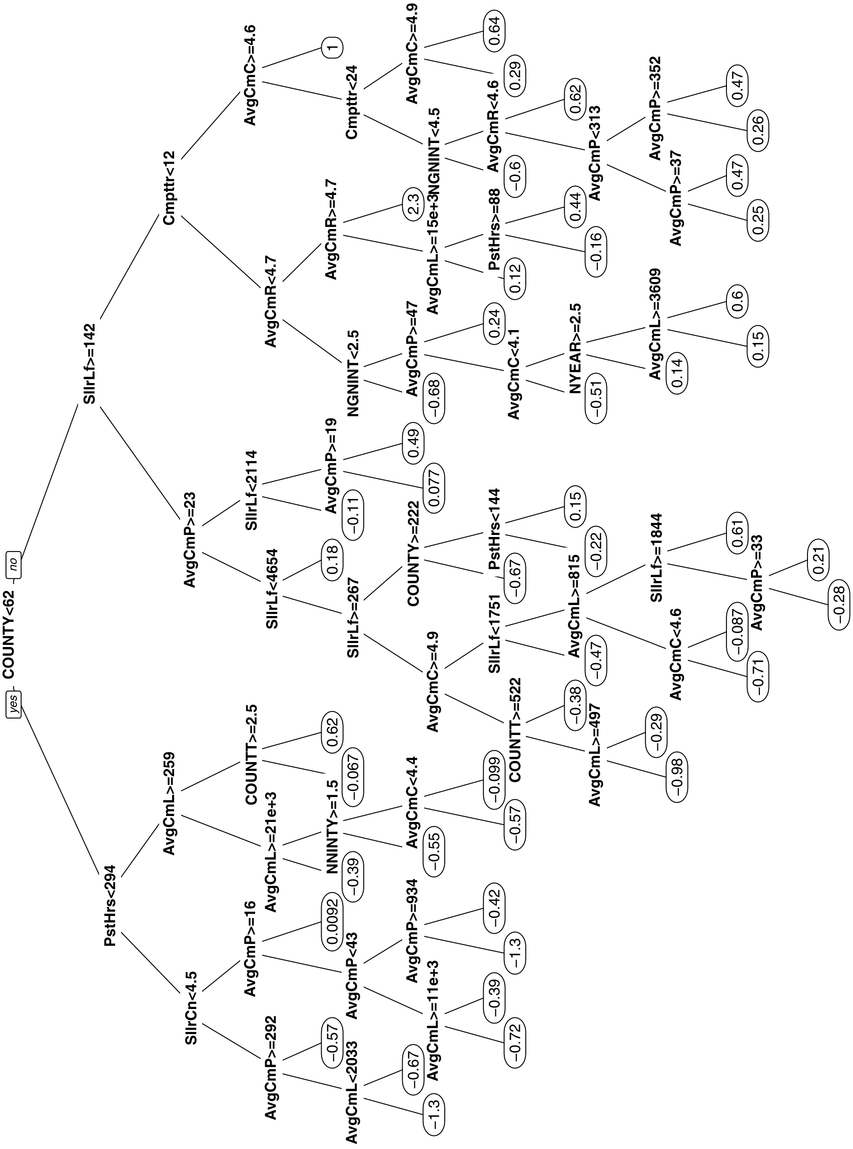


Fig. 8 Estimated RE-EM tree with one iteration for the logged relative price premium in the transactions data

Table 3 Parameter estimates for the linear models for the logged relative price premium with and without random effects. Standard errors are reported in parentheses

Variable	Linear model	Mixed effects model	Mixed effects model with autocorr.	Linear model (imputed)	Mixed effects model (imputed)	Mixed effects model with autocorr. (imputed)
(Intercept)	−0.024 (0.103)	0.354** (0.151)	0.436*** (0.120)	−0.143 (0.188)	0.694*** (0.250)	0.217 (0.212)
Number of comments in the last year (`COUNTYR`)	−3.338E–6 (1.195E–5)	6.183E–6 (1.08E–5)	2.00E–6 (9.49E–6)	2.701E–5** (1.127E–5)	2.343E–5** (1.012E–5)	5.722E–6 (9.586E–6)
Number of hours posted (`PostHours`)	−3.049E–4*** (2.833E–5)	−2.291E–4*** (2.632E–5)	9.551E–5*** (2.13E–5)	−2.532E–5 (2.792E–5)	−2.811E–5 (2.792E–5)	8.579E–5*** (2.5510E–5)
Seller life (`SellerLife`)	8.415E–6* (5.061E0–6)	3.99E–6 (4.52E–6)	3.11E–6 (3.68E–6)	5.035E–6 (4.975E–6)	6.437E–6 (4.389E–6)	4.727E–6 (3.916E–6)
Number of competitors (`Competitors`)	3.149E–3*** (7.686E–4)	4.963E–3** (2.141E–3)	5.634E–3*** (1.689E–3)	5.802E–3*** (7.362E–4)	0.010*** (1.723E–3)	6.904E–3*** (1.723E–3)
Number of comments in the last year (`COUNTNY`)	−1.711E–5 (1.914E–5)	−2.632E–5 (1.73E–5)	−1.356E–5 (1.56E–5)	−9.015E–5*** (2.781E–5)	−1.075E–4*** (2.444E–5)	5.563E–3*** (2.092E–5)
Average competitor price (`AvgCompPrice`)	6.862E–5*** (1.886E–5)	−1.399E–3*** (9.71E–5)	−1.576E–3*** (9.663E–5)	4.758E–7 (1.813E–5)	−1.935E–3*** (1.046E–4)	−1.536E–3*** (9.249E–5)
Average rating of competitors (`AvgCompRating`)	6.058E–3 (0.023)	−0.026 (0.032)	−0.049* (0.025)	0.064*** (0.022)	−0.139*** (0.030)	−0.062** (0.025)
Average lifetime of competitors (`AveCompLife`)				−1.629E–7 (1.565E–7)	−1.812E–7 (2.897E–7)	−6.251E–7*** (2.377E–7)
Average condition of competing goods (`AvgCompCondition`)				−0.180*** (0.023)	−0.056 (0.036)	−0.086*** (0.031)
Item condition (`SellerCond`)				0.192*** (0.011)	0.191*** (0.011)	0.192*** (0.010)

Table 3 (*Continued*)

Variable	Linear model	Mixed effects model	Mixed effects model with autocorr.	Linear model (imputed)	Mixed effects model (imputed)	Mixed effects model with autocorr. (imputed)
Seller rating (`SellerRating`)				0.139** (0.058)	0.118** (0.051)	0.051 (0.045)
30-Day positive comments (`PTHRTY`)				6.127E–3*** (8.964E–4)	6.306E–3*** (7.764E–4)	3.601E–3*** (6.431E–4)
90-Day positive comments (`PNINTY`)				0.011*** (0.001)	0.010*** (1.265E–3)	1.802E–3 (1.107E–3)
365-Day positive comments (`PYEAR`)				−6.461E–3*** (2.346E–3)	4.510E–3** (2.066E–3)	−3.558E–3** (1.777E–3)
Lifetime positive comments (`PLIFE`)				−0.022*** (0.003)	−0.024*** (2.457E–3)	−7.629E–3*** (2.195E–3)
30-Day neutral comments (`NTHRTY`)				−2.858E–3** (1.405E–3)	2.307E–3* (1.271E–3)	2.594E–5 (1.049E–3)
90-Day neutral comments (`NNINTY`)				6.954E–3*** (2.622E–3)	7.029E–3*** (2.274E–3)	−2.049E–4 (1.891E–3)
365-Day neutral comments (`NYEAR`)				2.621E–3 (4.604E–3)	1.978E–3 (3.962E–3)	−3.821E–3 (3.454E–3)
Lifetime neutral comments (`NLIFE`)				−2.355E–3 (4.486E–3)	−5.818E–3 (3.872E–3)	4.533E–3 (3.496E–3)
Number of comments in the last 30 Days (`COUNTTH`)				2.793E–5 (4.988E–5)	7.958E–5* (4.368E–5)	7.134E–5* (3.683E–5)

*Significantly different from zero at the 10% level; **significantly different from zero at the 5% level; ***significantly different from zero at the 1% level

Table 4 *RMSE*s from cross-validation leaving out one observation or one software title at a time, using the transactions data, using the logged relative price premium

Method	Excluding observations	Excluding titles
Linear model	0.4566	0.4712
Linear model with random effects	0.4087	0.6478
Linear model with random effects—AR(1)	0.3957	0.6567
Linear model (imputed)	0.4358	0.4455
Linear model with random effects (imputed)	0.3787	0.6478
Linear model with random effects—AR(1) (imputed)	0.3873	0.6567
Tree without random effects	0.3186	0.3933
RE-EM tree	0.2880	0.3906
RE-EM tree—AR(1)	0.2881	0.3907
RE-EM tree (1 iteration)	0.2876	0.3917
RE-EM tree—AR(1) (1 iteration)	0.2879	0.3861

limits associated with progressively higher fatality rates holding all else fixed, but otherwise the only variables that are statistically significant at a 0.05 level are administrative license revocation and unemployment rate, with each having counterintuitive signs (with a license revocation law associated with higher fatality rate and high unemployment, and hence less economic activity, associated with a higher fatality rate, holding all else fixed). Accounting for state effects changes the picture dramatically, however, with speed limit effects much smaller, blood alcohol laws statistically significant, and administrative license revocation and unemployment rate now having coefficients with intuitive signs.

Figures 9, 10, 11, and 12 give the tree estimates for these data. Not surprisingly given the relatively small sample, the trees are much simpler than those for the transactions data. The tree without state effects (Fig. 9) splits twice on speed limit, with large differences in fatality rates between speed limits (for example, for years before 1990, the fatality rate is estimated to be 90% higher when the speed limit is 70 MPH or there is no speed limit compared to when it is 55 MPH). When state effects are taken into account in the RE-EM trees, however, speed limit either does not appear in the tree at all (Fig. 10) or is associated with much smaller effects (Figs. 11 and 12). This is particularly apparent when comparing the `rpart` tree (Fig. 9) to the one-iteration RE-EM tree (Fig. 12), since they must have the same structure; in the latter tree the difference in estimated fatality rate between a 55 MPH speed limit and a 70 MPH speed limit or no speed limit only corresponds to a 10% difference, clearly showing that taking state effects into account dramatically weakens any evidence of an effect of speed limits on traffic fatalities.

Table 6 summarizes the cross-validated *RMSE* for the different methods, omitting one state at a time and one observation at a time. It can be seen that all of the methods have comparable performance omitting one state at a time, although that of the linear mixed effects model lags behind. When predicting at the individual observation level, however, the benefit of using a method that accounts for the longitudinal (repeated years within states) structure in the data is apparent, as the ordinary linear and tree models fare far worse than either the linear mixed effects or RE-EM estimates. In this case the linear mixed effects model is slightly more effective than the RE-EM tree, but all of the versions of the RE-EM tree are much better than the ordinary `rpart` tree. Thus, even in a case where a population-level tree structure is not an improvement over a linear model, the benefits of estimating the longitudinal structure when making predictions at the observation level are clear.

Table 5 Parameter estimates for the linear models for the logged fatality rate with and without random effects. Standard errors are reported in parentheses

Variable	Linear model	Mixed effects model	Mixed effects model with autocorr.
(Intercept)	7.1445^{***}	5.2217^{***}	5.0739^{***}
	(0.396)	(0.170)	(0.214)
Year	-0.0376^{***}	-0.0241^{***}	-0.0216^{***}
	(0.004)	(0.002)	(0.002)
Speed limit = 65	0.3656^{***}	−0.0096	−0.0061
	(0.030)	(0.012)	(0.014)
Speed limit = 70	0.6616^{***}	0.0724^{***}	0.0525^{**}
	(0.054)	(0.021)	(0.025)
Speed limit = 75	0.7950^{***}	0.0846^{***}	0.0602^{**}
	(0.057)	(0.022)	(0.027)
No speed limit	0.9908^{***}	0.1066^{**}	0.0317
	(0.145)	(0.052)	(0.072)
Drinking age	−0.0636	−0.0058	0.0046
	(0.064)	(0.021)	(0.018)
Driving age	0.0149	0.0210	0.0036
	(0.064)	(0.021)	(0.019)
Seatbelt law	−0.0169	0.0009	0.0040
	(0.014)	(0.006)	(0.007)
Zero tolerance	-0.0511^{*}	0.0109	0.0155
	(0.028)	(0.011)	(0.013)
Illegal at BAC $\geq$ 0.10	−0.0334	-0.0307^{**}	-0.0290^{**}
	(0.029)	(0.013)	(0.014)
Illegal at BAC $\geq$ 0.08	−0.0608	-0.0428^{**}	−0.0354
	(0.041)	(0.020)	(0.023)
Administrative license revocation	0.0524^{**}	-0.0599^{***}	-0.0472^{***}
	(0.023)	(0.012)	(0.014)
Unemployment rate	1.6422	-3.7574^{***}	-2.874^{***}
	(0.532)	(0.245)	(0.304)

*Significantly different from zero at the 10% level; **significantly different from zero at the 5% level; ***significantly different from zero at the 1% level

5 Simulated data sets

5.1 Design of simulations

We now use simulations to assess the usefulness and effectiveness of the RE-EM tree method. (Comparing performance on a suite of additional large-scale, real-world longitudinal data would be highly desirable, but such a suite is not available. Instead, we turn to simulated data sets as a workable alternative. Simulated data sets also allow us to measure

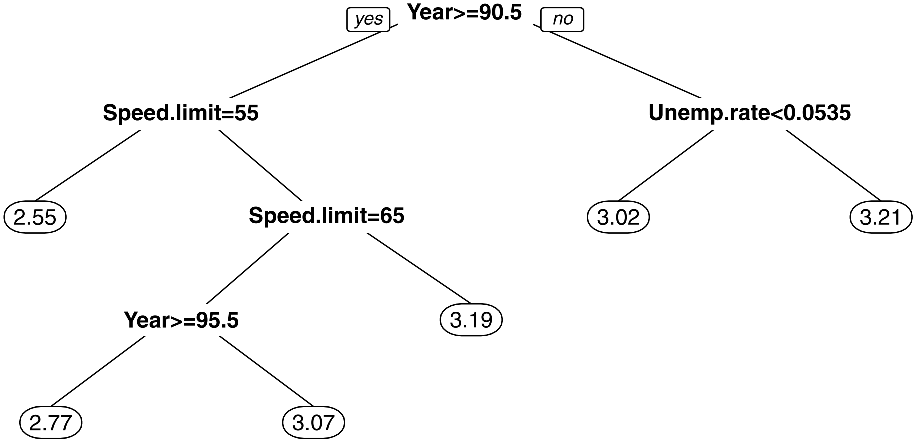
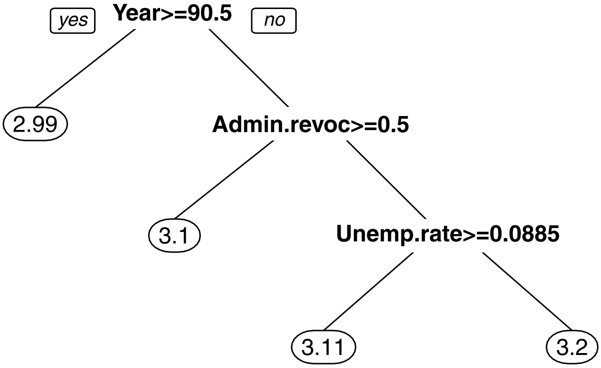


Fig. 9 Estimated tree without random effects for the logged fatality rate in the traffic data

Fig. 10 Estimated RE-EM tree for the logged fatality rate in the traffic data



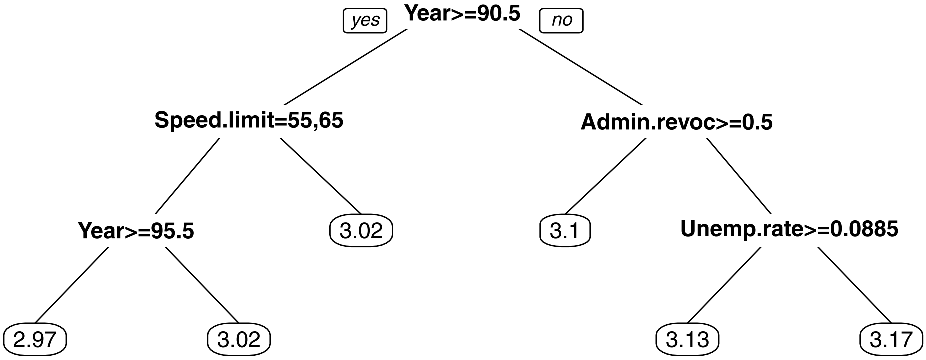


Fig. 11 Estimated RE-EM tree with autocorrelation for the logged fatality rate in the traffic data

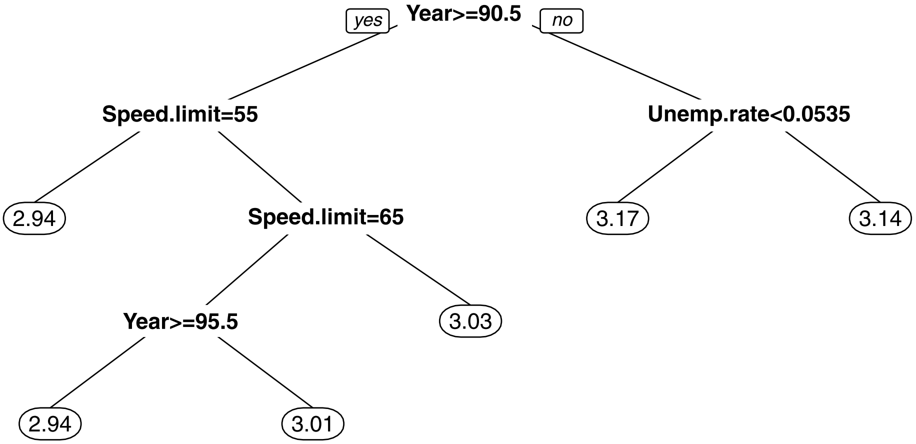


Fig. 12 Estimated RE-EM tree with one iteration for the logged fatality rate in the traffic data

Table 6 *RMSE*s from cross-validation leaving out one observation or one state at a time, using the traffic fatality data, using the logged fatality rate

Method	Excluding observations	Excluding states
Linear model	0.2798	0.3013
Linear model with random effects	0.0904	0.3913
Linear model with random effects—AR(1)	0.0920	0.3369
Tree without random effects	0.2821	0.3014
RE-EM tree	0.1016	0.3320
RE-EM tree—AR(1)	0.1031	0.3271
RE-EM tree (1 iteration)	0.1020	0.3330
RE-EM tree—AR(1) (1 iteration)	0.1031	0.3286

the success of the estimation methods in estimating the random effects and fixed effects separately.) These simulations consider data sets that contain $I = 50$, 100, 200, 400, 1000 or 2000 objects, with $T = 10$, 25, 50 or 100 observations per object. We consider three data generating processes, to allow for cases in which the tree is only an approximation to reality. In each experiment, we compare the performance of the RE-EM tree with a tree that does not account for random effects and with parametric linear models that do and do not include random effects, as well as (when feasible) a different longitudinal tree method.

Our data generation procedure for attributes is based on the values in the transactions data, while the response variable is based on the estimated RE-EM trees and linear models for variables from the logged price premium fit to the full transactions data set discussed in Sect. 4, fit to the price premium. This simulates complex yet realistic data patterns in both attributes and response. Specifically, the “true” models are the RE-EM tree fit to the price premium in the first set of experiments, the linear model with scalar random effects fit to the price premium in the second set, and a more complicated non-tree, mixed effects model in the third. In the third case, we define f by estimating the price premium using a linear model including all possible products of the eight continuous variables that appeared in the trees, listed in Table 1, together with the squares of `AvgCompPrice`, `AvgCompLife`, AvgCompCondition, and AvgCompRating. All but the last of the squared variables has a statistically significant coefficient, and some of the product terms have statistically significant coefficients as well. Each method is estimated based on the full data set. This estimation yields a prediction for any set of attributes as well as a list of estimated random effects, $\hat{b}_i$, and estimated observation errors, $\hat{\varepsilon}_{it}$, for each object. For each sample size, I, we use the attributes from a random sample (with replacement) of I objects to compute the expected value, $E(y_{it})$, of the target variable given the true model. When T is larger than the number of observations for the randomly chosen object, we use the attributes from the next object(s) in the sample. We generate a random effect $\hat{b}_i$ and errors $\hat{\varepsilon}_{it}$ for $t = 1, \dots, T$ as normally distributed with zero mean and standard deviations equal to the observed values from the linear mixed effects fit to the data. Then, the new observed data consist of $y_{it} = E(y_{it}) + \hat{b}_i + \hat{\varepsilon}_{it}$ together with the attributes. Data are created in the same way for an additional 50 objects who are used as the hold-out sample. For each group of $I + 50$ objects, we resample 50 times in this way, which allows us to check for any effects of the attributes on predictive performance. We then move on to a new sample of size $I + 50$ and repeat the experiment for 50 different samples of objects.

There is little guidance in the literature for the assessment of predictive power for methods for longitudinal data. One exception is Afshartous and de Leeuw (2005), who found that a mixed effects fit for new observations of objects in the sample was most effective among the methods they studied (this corresponds to the prediction methods used here). Afshartous and de Leeuw (2005) also examined the results of methods fit to one object at a time (treating the observations for that object as a complete sample); in our simulations this approach performed quite poorly, particularly for prediction, as will be discussed in Sect. 5.2.

To measure out-of-sample performance for both objects already in the sample and new objects, we fit each method to the first 75% of observations for I objects. We then predict the future observations for those objects to estimate the out-of-sample performance of the methods for future observations for objects used in estimation. For the additional sample of 50 objects, we predict the first 75% of their observations using just $\hat{f}$; this allows us to measure the prediction performance for new objects. Finally, we use the original fitted model and the first 75% of observations for the new objects to predict the last 25% of observations for those objects. This allows us to measure the predictive performance for future observations of new objects. We start with examination of the accuracy of estimates of the underlying population-level function $f(x_{it})$ and the random effects b_i in Sect. 5.3. These underlying values are of interest in their own right, but also go a long way to accounting for predictive performance, which is discussed in Sect. 5.2. In that section, we also test whether the *RMSE*s from RE-EM trees differ significantly from those of other methods, using the Wilcoxon signed-rank test. We generalize to the case of unbalanced panels, where T varies across objects, in Sect. 5.4. We also explore the effects of changing the parameters of the model or the estimation method in Sect. 5.4. In each figure, results based on fits using the RE-EM tree are given using a solid line and circles (REEM), the RE-EM tree based on one iteration a short dashed line and triangle (REEM-I1), a single rpart tree fit to the entire data set (ignoring the longitudinal structure) using a dotted line and plus (RPART), separate rpart trees fit to each observation using a dotted and short dashed line and x (RPART-Obj), a linear mixed effects model using a long dashed line and diamond (LME), a linear model (ignoring the longitudinal structure) a dotted and long dashed line and inverted triangle (LM), and an mvpart tree a solid line and square (MVPART). The figures given are trellis displays (Becker et al. 1996), where the vertical axis in each panel of the display is the appropriate *RMSE*, the horizontal axis is the number of objects I, and the panels correspond to increasing time periods T (10, 25, 50, and 100, respectively) moving from left to right.

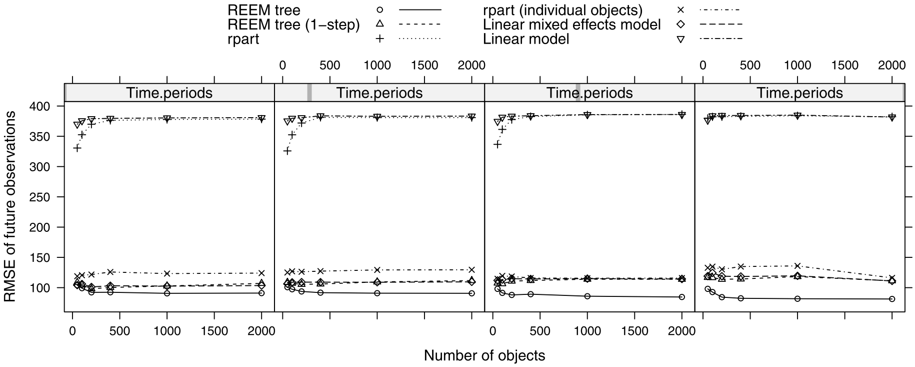


Fig. 13 *RMSE*s of predictions of future observations when the true data generating process is a RE-EM tree. In this and all figures symbols are as follows: RE-EM—*solid line* and *circle*; RE-EM (1 iteration)—*short dashed line* and *triangle*; rpart—*dotted line* and *plus*; separate rpart trees—*dotted* and *short dashed line* and x; linear mixed effects model—*long dashed line* and *diamond*; linear model—*dotted* and *long dashed line* and *inverted triangle*; mvpart—*solid line* and *square*

Note that all of the linear models are fit using all of the available attributes (but without products or quadratics) without any attempt to simplify the models. MVPART is fit using the mvpart package of De'Ath (2006) in R.

5.2 Predictive performance

We first consider in this section the prediction error for future observations. Figure 13 refers to the situation when the true data generating process is a RE-EM tree. As can be seen, the display is dominated by the clear separation between methods that work comparably poorly and those that work comparably well. The methods that work poorly are LM and RPART, the two methods that ignore the longitudinal structure in the data. This is not surprising, since prediction of future observations for a given object should take into account the random effect associated with that object, and these methods ignore that. For this reason, in all figures involving prediction of future observations these methods will be omitted, since they always trail badly behind. Figure 14 gives results only for the other four methods (recall that MVPART cannot be used to predict future observations, so it does not appear in the display). The figure makes clear that the RE-EM tree provides best performance for the prediction of future observations. The two other methods that use random effects (REEM-I1 and LME) are next, and have similar performance. The *RMSE* values of these two methods are statistically significantly higher than that of REEM for all combinations of I and T based on Wilcoxon tests. The construction of separate rpart trees for each object is better than using a single tree for all observations (since it accounts for structure within an object by being based only on data for that object), but since there are only T data points for each object the small sample makes the predictions less accurate (statistically significantly so for all I and T).

Figure 15 gives results for the four longitudinal methods when the true model is a linear mixed effects model. Not surprisingly, in this case LME is the best performer, as it is the correct model. Once again trees on separate objects lag behind (doing worse for larger T), while the two RE-EM estimators are noticeably better performers (and perform similarly to each other, although the fully iterated RE-EM tree is usually statistically significantly better).

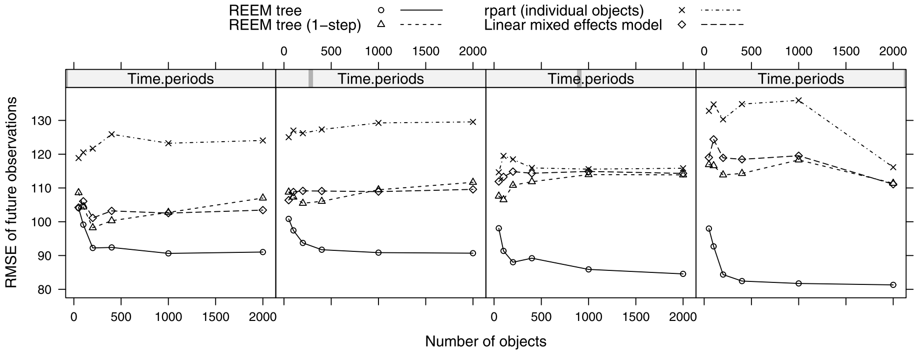


Fig. 14 *RMSE*s of predictions of future observations when the true data generating process is a RE-EM tree, omitting results for LM and RPART

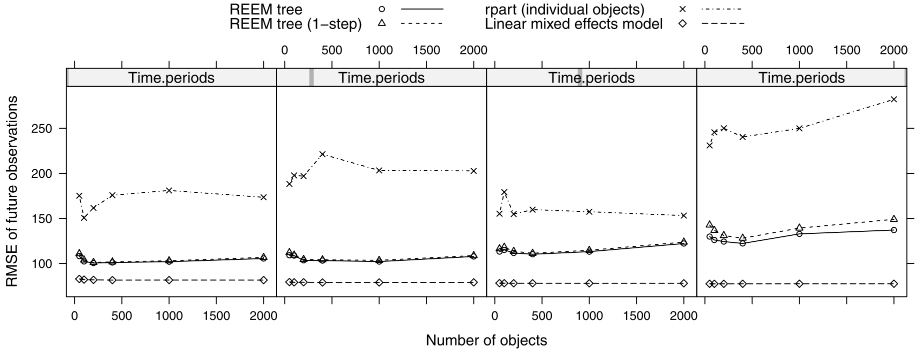


Fig. 15 *RMSE*s of predictions of future observations when the true data generating process is a linear mixed effects model, omitting results for LM and RPART

The more complicated model is an interesting test case, since it corresponds to a true relationship that is not at all a tree (being based on a linear model), yet is not the simple linear model being fit by LME, and includes product terms more analogous to an interaction effect. Figure 16 gives results for this case. It can be seen that the RE-EM trees and LME are much closer in performance than when the true model is a linear model (the performances of the two versions of the RE-EM tree are often not statistically significantly different from each other, but both significantly lag behind LME). Further, for larger T the performance of the trees is very similar to that of LME (particularly for smaller I), indicating that with enough replications the tree can recover the signal well even when the true relationship is not a tree, while also accounting for the random effects.

Next, we consider predictions for observations of new objects. Since object-specific regression trees produce I different trees, the average prediction over all trees is used as the prediction for that method. The MVPART tree is included in this case, since future observations are not being predicted. When the true data generating process is a RE-EM tree (Fig. 17), prediction using RE-EM trees has the lowest mean squared errors (only by 2–3%, but this is always statistically significant), with the other methods similar when I is large enough ($I \geq 400$ or so). For small values of I, RPART, RPART on individual objects, and MVPART seriously lag behind, illustrating that merely fitting a tree does not necessarily

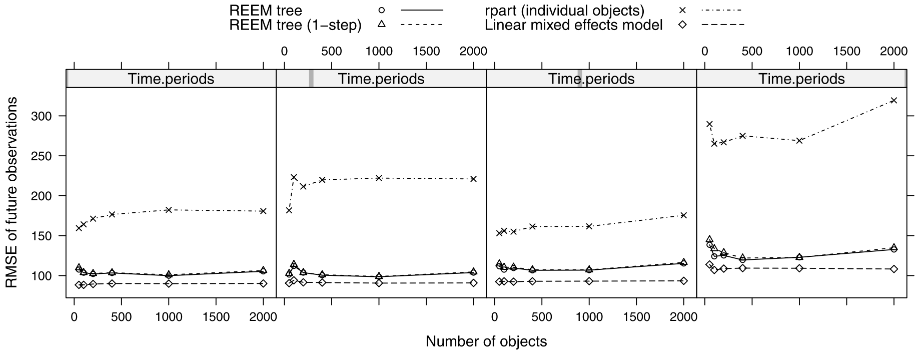


Fig. 16 *RMSE*s of predictions of future observations when the true data generating process is the more complicated mixed effects model, omitting results for LM and RPART

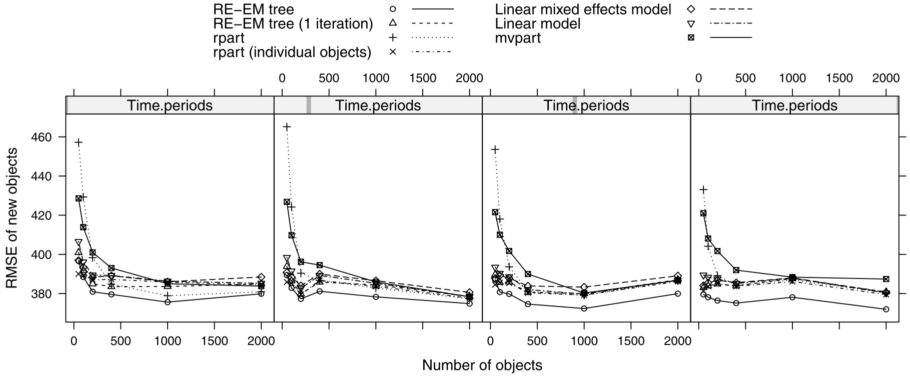


Fig. 17 *RMSE*s of predictions of new objects when the true data generating process is a RE-EM tree

lead to good estimation in the RE-EM situation. When the true process is a linear model with random effects (Fig. 18), LME performs best (as expected); since prediction of new objects does not involve the random effects the performance of LM is similar to that of LME, especially when I is larger. The *RMSE* values for the RE-EM trees are roughly 5–10% higher than those of LME (and usually not significantly different from each other), with RPART being a little worse. MVPART and the average of separate RPART trees on each object lag behind badly. When the true generating process is the more complicated model (Fig. 19) the performance of all of the methods is very similar (although LME is best), other than that of MVPART and the average of individual RPART trees.

Finally, we examine the predictions of future observations for objects that were not in the original sample, using some of their observations to estimate random effects. Once again MVPART cannot be used here, and once again the two methods that do not account for the random effects (LM and RPART), and the one that uses them inefficiently (RPART on individual objects) lag behind badly and are not included. The results parallel those when predicting future observations of objects in the original sample: when the true model is a RE-EM tree the RE-EM tree method is best (Fig. 20), when the true model is LME the LME method is best (Fig. 21), with REEM improving for larger I, and when the true model is

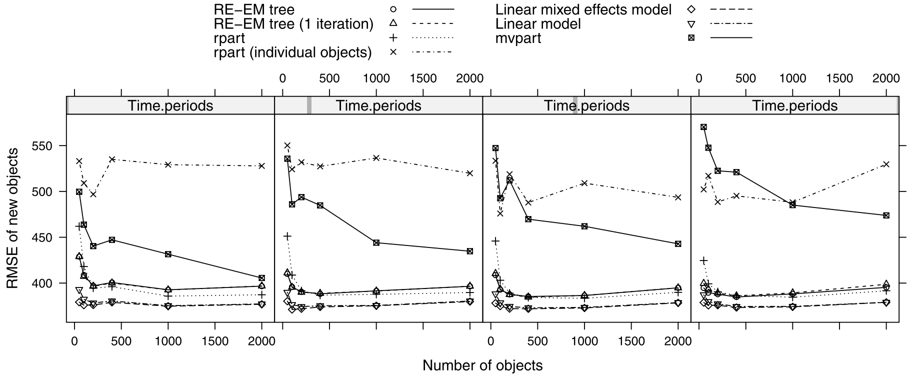


Fig. 18 *RMSE*s of predictions of new objects when the true data generating process is a linear mixed effects model

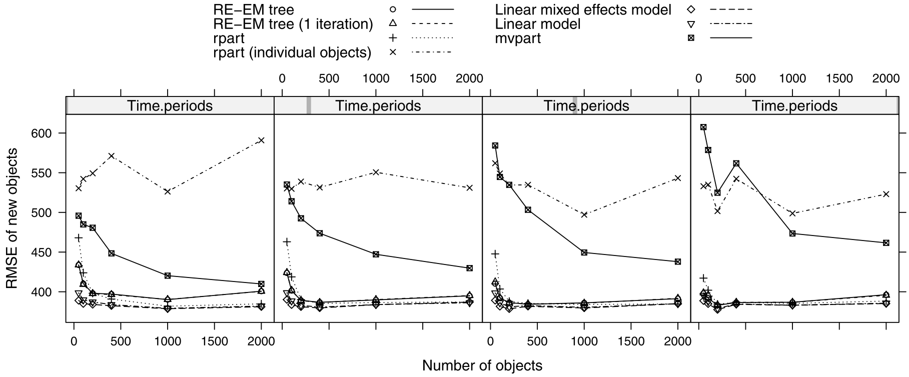


Fig. 19 *RMSE*s of predictions of new objects when the true data generating process is the more complicated mixed effects model

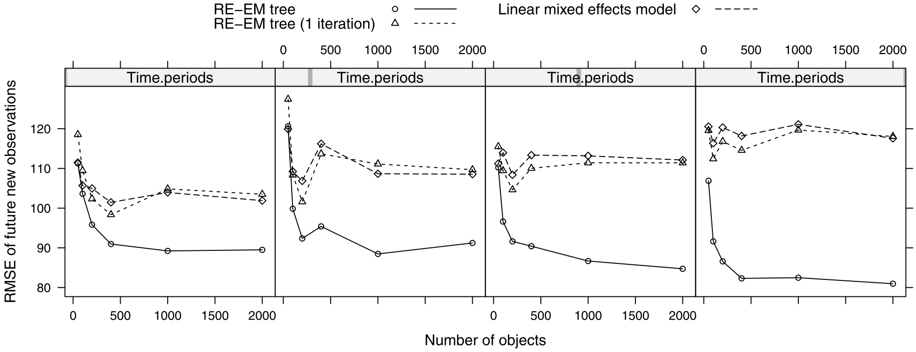


Fig. 20 *RMSE*s of predictions of future observations of new objects when the true data generating process is a RE-EM tree, omitting results for LM, RPART, and RPART on individual objects

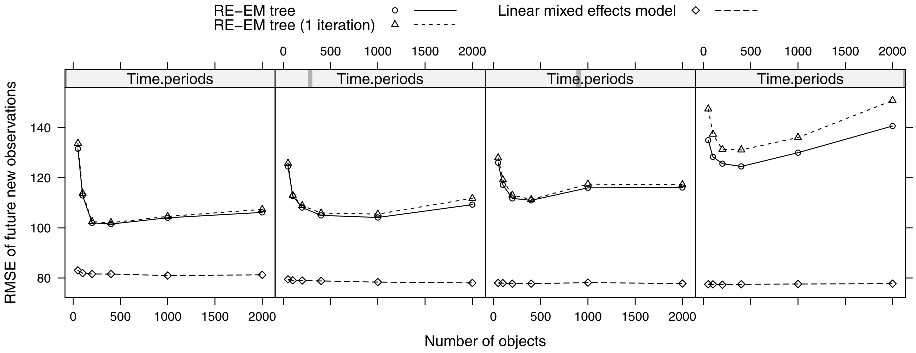


Fig. 21 *RMSE*s of predictions of future observations of new objects when the true data generating process is a linear mixed effects model, omitting results for LM, RPART, and RPART on individual objects

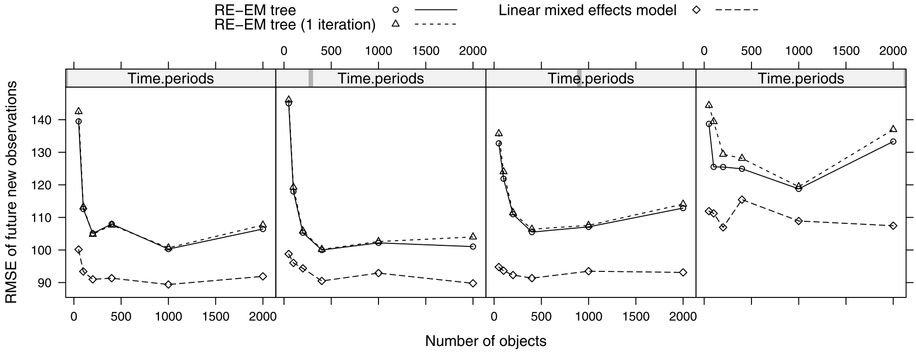


Fig. 22 *RMSE*s of predictions of future observations of new objects when the true data generating process is the more complicated mixed effects model, omitting results for LM, RPART, and RPART on individual objects

the more complicated linear model LME is best but REEM becomes more competitive for larger I (Fig. 22).

In all of the different types of prediction, the RE-EM tree estimation has the best predictive performance when it is the true model and good performance otherwise, especially for larger sample sizes. The RE-EM tree is clearly the most effective tree-based estimator. The success of the RE-EM tree when it is not the correct model allows us to apply it to situations when the model is unknown and is likely to be complicated, such as was the case for the transactions data.

5.3 Estimation of the underlying function and random effects

Although in many contexts the predictive performance discussed in the previous section is most important, we also investigate the ability of the different methods to estimate the population-level expected response $f(\cdot)$ and true random effects $\mathbf{b}$, using *RMSE* to measure performance. The results underscore the patterns in the previous section as would be expected. When estimating the underlying function (Figs. 23, 24, 25) the method that is fitting the correct model does best, while methods that fit the correct structure without accounting

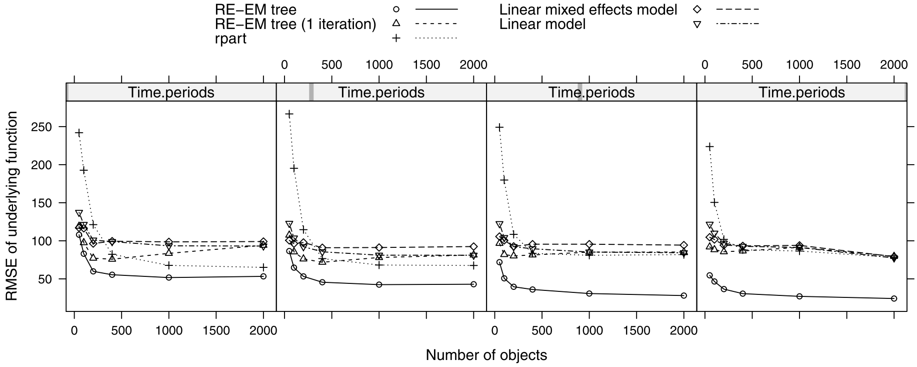


Fig. 23 *RMSE*s of estimates of underlying function when the true data generating process is a RE-EM tree

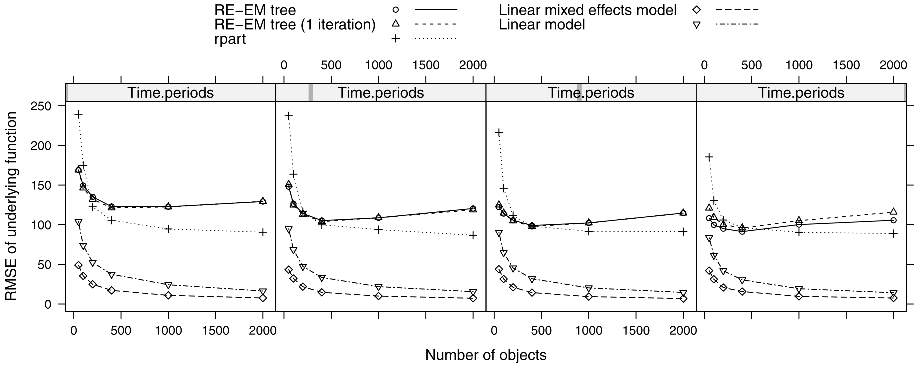


Fig. 24 *RMSE*s of estimates of underlying function when the true data generating process is a linear mixed effects model

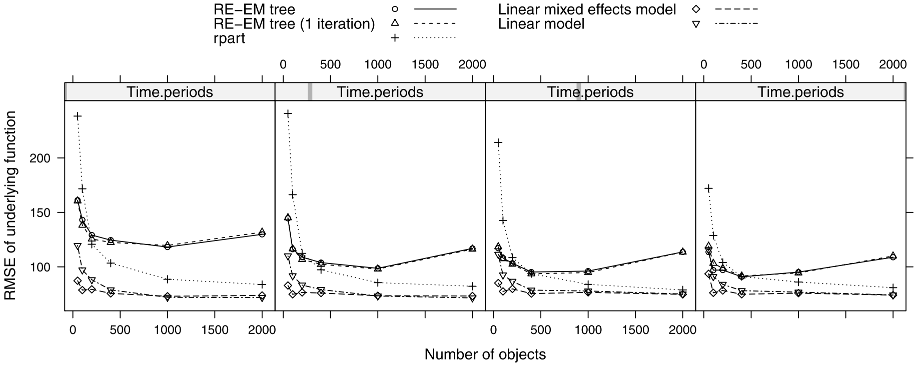


Fig. 25 *RMSE*s of estimates of underlying function when the true data generating process is the more complicated mixed effects model

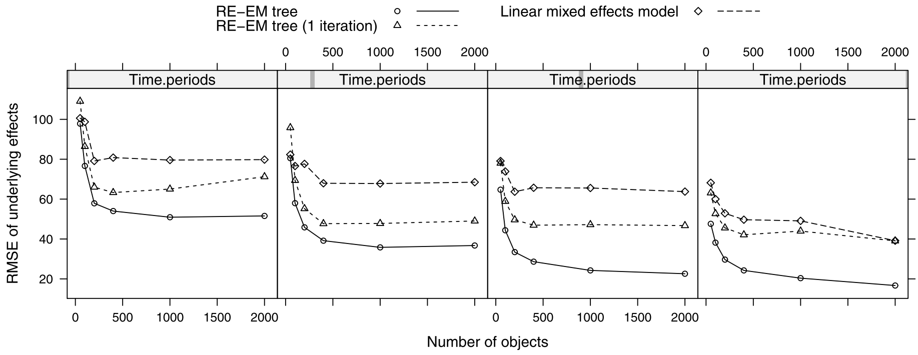


Fig. 26 *RMSE*s of estimates of random effects when the true data generating process is a RE-EM tree

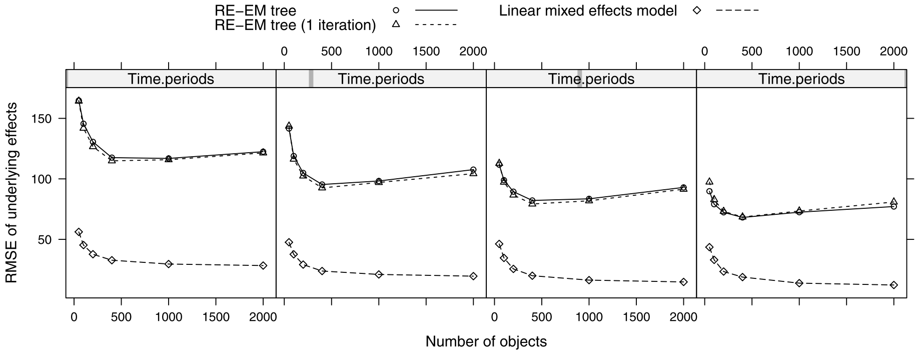


Fig. 27 *RMSE*s of estimates of random effects when the true data generating process is a linear mixed effects model

for the random effects do less well, but still reasonably. When the model being fit is the more complicated linear model LME performs best, but REEM gets closer for larger I. The pattern is similar when estimating the random effects (Figs. 26, 27, 28). As expected, the fully iterated version of the RE-EM tree outperforms the one-iteration version when the true model is a RE-EM tree, with similar performance otherwise, and random effects are best estimated using the method fitting the correct model. Thus, performance when estimating the underlying function parallels the results in the previous section when predicting a new object, since the latter prediction is based only on the underlying function. In contrast, performance when estimating the random effects also affects performance when predicting new observations of objects for which response information is available.

5.4 Autoregression, unbalanced panels, and estimation methods

In this section we briefly discuss performance when changing the simulation structure in other ways. Since the results are very similar to those already presented, we do not provide figures, but merely summarize the results. We first explore the effect of an error term with an autoregressive component of order one (so that $\mathrm{Corr}(\varepsilon_{i,t}, \varepsilon_{i,t-1}) = \rho$). This turns out to have a consistent, but relatively small, effect on performance. Methods that are fitting the correct functional form are slightly more effective at predicting new observations when

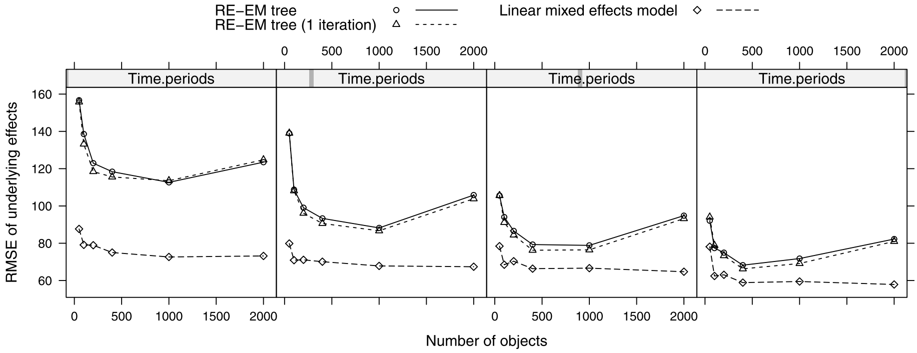


Fig. 28 *RMSE*s of estimates of random effects when the true data generating process is the more complicated mixed effects model

the autocorrelation structure is accounted for, but autoregressive methods that fit the wrong population-level functional form are slightly less effective. That is, if the true model is a RE-EM tree with autoregressive errors, a RE-EM tree that accounts for autoregressive errors performs slightly better than a RE-EM tree that does not, but a linear mixed effects model that accounts for autoregressive errors performs slightly worse than one a linear mixed effects model that does not. The corresponding opposite pattern occurs if the true model is a linear mixed effects model with autoregressive errors. Thus, accounting for the second-order effect of correlation in the errors is only helpful if the first-order effect of fitting the right function is taken care of. The more complicated linear model (which corresponds to an incorrect functional form for both estimators) occupies the expected middle ground: the RE-EM tree accounting for autoregressive errors is worse than one without autoregressive errors, and so is the linear mixed effects model (particularly for small I and T). For all underlying functional forms and both methods the choice of including or not including autoregressive errors in the fitting has virtually no effect on prediction of new objects.

A balanced panel is one where the number of time periods with observed responses for each object i (T_i) is the same for all i, as was the case in all of the simulations reported thus far. We also examined the performance of the estimation methods in unbalanced panels where T_i may vary across objects, with averages of approximately 10, 25, or 38 observations per object.[1] Note that we cannot use the `mvpart` estimation method in this case, because the method only applies when T_i is constant. Furthermore, separate linear regressions and separate regression trees for each object in the resulting data set are sometimes not feasible, since the simulated data can include objects for which T_i is too small to fit a linear regression with eight attributes or a meaningful tree. All of the results for different values of $E(T_i)$ are very similar to those with balanced panels for corresponding values T, with one notable exception: when $E(T_i) \approx 10$ (and thus some data sets have very few observations within some objects) LME can perform quite poorly (even when the linear mixed effects model is the true model), especially when $I \leq 200$. Thus, the RE-EM tree appears to be less sensitive to a small number of time periods than is the linear mixed effects model.

We also assess the stability of our tree estimates by starting estimation with alternative initial values for the random effects. The results we have presented so far fit RE-EM trees

[1] The average number of observations per object in the underlying price premium data set on which the simulations are based is 38.

with initial values of 0 for all of the random effects. We also fit trees in which we vary the initial values for the random effects; specifically, we fit a RE-EM tree with initial values of the random effects set to 0 and then use initial values that are those estimated effects in random order or in reverse order. As additional comparisons, we fit trees using maximum likelihood instead of restricted maximum likelihood when we estimate the linear model. The estimated fitted values are generally similar across the different estimation possibilities. Changing the initial values of the random effects has a small impact, and the difference between performance with different initial values declines steadily as the sample size grows. The change in estimates based on using maximum likelihood instead of REML to estimate the random effects is even smaller, as there was almost no difference in estimates when either optimization method is used for estimating the underlying tree.

5.5 Computation time for the RE-EM tree

An advantage of the RE-EM tree method is that it is based on two parts (a regression tree algorithm and a linear mixed effects regression algorithm) for which there are many alternative methods; although all of the calculations here are based on the `R` packages `rpart` and `nlme`, respectively, any alternative tree and mixed modeling methods could be used instead. If a data set is large, the dominant contributor to computing time is the mixed model portion of the fitting, and since different packages use different computational algorithms (see West et al. 2007, pp. 30–33), it is possible that an algorithm using a package other than `R`, or a function other than `nlme`, might be more computationally efficient.

Based on timings for data sets with 50 to 5000 objects (I), 10 to 500 time periods (T), and 10 to 50 attributes (K), the CPU time in seconds when running the single-iteration version of the RE-EM tree on a PC running Windows XP using a 3.20 GHz Pentium 4 processor and 2.0 GB of RAM roughly followed the relationship

$$\text{CPU time} \approx 0.42 \times I^{1.15} T^{1.12} K^{0.32}$$

Thus, the complexity of the algorithm appears to be roughly linear in the number of objects and the number of time periods, and much less than linear in the number of attributes being used in the modeling.

6 Conclusion and future work

In this paper, we have presented a tool for data mining with longitudinal data and demonstrated its usefulness in simulations and with two real data sets. The RE-EM tree accounts for the structure of longitudinal or clustered data while allowing for unbalanced panels and prediction of future time periods, while also providing the ability to use time-varying attributes in the construction of a flexible representation for the underlying relationship between the response and the attributes; indeed, by including time as a potential attribute, it is possible to fit completely different tree structures for different time periods if the tree splits on time. Using data sets on web transactions and traffic fatalities, we have shown that RE-EM trees can improve predictive performance over standard trees and allow the modeling of target variables without assuming that linear models hold. In simulation experiments, we have found that RE-EM trees outperform trees that do not allow for random effects, are more effective than other methods when the true relationship takes the form of a tree, and are comparable to linear models that include random effects, even when a tree is not the underlying model.

RE-EM trees also outperform multivariate regression trees and generally outperform regression trees that are fit separately to each object. This is true for different types of prediction and in a wide variety of scenarios. We have also demonstrated that the RE-EM tree can be more successful at estimating the underlying functional form and random effects than is a linear mixed effects model, especially when the number of observations per object is small.

This paper has explored the basics of the RE-EM tree method. A number of possible issues remain to be explored. First, methods such as bagging and boosting build on a tree structure as a way to improve predictive performance (see for example, Hastie et al. 2001, Sect. 8.7 and Chap. 10). We expect that the improvements from these methods would carry over when they are applied to RE-EM trees as well. Further, these methods might generalize to classification trees, which would extend their use to another class of response variables. Finally, one could explore the extension of the existing consistency results for regression trees and mixed effects models to RE-EM trees, checking whether f or the random effects are estimated consistently.

An `R` package to implement the RE-EM tree, called `REEMtree`, is available on CRAN.

Acknowledgements The authors thank Foster Provost and Anindya Ghose for many helpful comments and Norman White for assistance with computational issues. The authors also thank several anonymous reviewers for comments that greatly improved the paper.

References

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

Afshartous, D., & de Leeuw, J. (2005). Prediction in multilevel models. *Journal of Educational and Behavioral Statistics*, *30*, 109–139.

Becker, R. A., Cleveland, W. S., & Shyu, M.-J. (1996). The visual design and control of trellis display. *Journal of Computational and Graphical Statistics*, *5*, 123–155.

Berk, R. A. (2008). *Statistical learning from a regression perspective*. New York: Springer.

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

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

De'Ath, G. (2006). *mvpart: multivariate partitioning*. R package version 1.2-4.

Dee, T. S., & Sela, R. J. (2003). The fatality effects of highway speed limits by gender and age. *Economics Letters*, *79*, 401–408.

Evgeniou, T., Pontil, M., & Toubia, O. (2007). A convex optimization approach to modeling consumer heterogeneity in conjoint estimation. *Marketing Science*, *26*, 805–818.

Galimberti, G., & Montanari, A. (2002). Regression trees for longitudinal data with time-dependent covariates. In K. Jajuga, A. Sokolowski, & H.-H. Bock (Eds.), *Classification, clustering and data analysis* (pp. 391–398). New York: Springer.

Ghose, A., Ipeirotis, P., & Sundararajan, A. (2005). *The dimensions of reputation in electronic markets* (Technical Report 06-02). NYU CeDER Working Paper.

Hajjem, A., Bellavance, F., & Larocque, D. (2008). *Mixed-effects regression trees for clustered data*. Les Cahiers du GERAD G-2008-57.

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

Harville, D. A. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, *72*, 320–340.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer.

Hsiao, W.-C., & Shih, Y.-S. (2007). Splitting variable selection for multivariate regression trees. *Statistics and Probability Letters*, *77*, 265–271.

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

Larsen, D. R., & Speckman, P. L. (2004). Multivariate regression trees for analysis of abundance data. *Biometrics*, *60*, 543–549.

Lee, S. K. (2005). On generalized multivariate decision tree by using GEE. *Computational Statistics & Data Analysis*, *49*, 1105–1119.

Lee, S. K. (2006). On classification and regression trees for multiple responses and its application. *Journal of Classification*, *23*, 123–141.

Lee, S. K., Kang, H.-C., Han, S.-T., & Kim, K.-H. (2005). Using generalized estimating equations to learn decision trees with multivariate responses. *Data Mining and Knowledge Discovery*, *11*, 273–293.

Liu, Z., & Bozdogan, H. (2004). Improving the performance of radial basis function (RBF) classification using information criteria. In H. Bozdogan (Ed.), *Statistical data mining and knowledge discovery* (pp. 193–216). Boca Raton: Chapman and Hall/CRC.

Liu, C., & Rubin, D. B. (1994). The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence. *Biometrika*, *81*, 633–648.

Loh, W.-Y. (2002). Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, *12*, 361–386.

Milborrow, S. (2011). *rpart.plot: plot rpart models*. R package version 1.2-2.

Patterson, H. D., & Thompson, R. (1971). Recovery of inter-block information when block sizes are unequal. *Biometrika*, *58*, 545–554.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., & the R Core team (2009). *nlme: linear and nonlinear mixed effects models*. R package version 3.1-93.

R Development Core Team (2009). *R: a language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing. ISBN 3-900051-07-0. URL http://www.R-project.org.

Ritschard, G., & Oris, M. (2005). Life course data in demography and social sciences: statistical and data mining approaches. In R. Levy, P. Ghisletta, J.-M. Le Goff, D. Spini, & E. Widmer (Eds.), *Towards an interdisciplinary perspective on the life course, advances in life course research* (pp. 289–320). Amsterdam: Elsevier.

Ritschard, G., Gabadinho, A., Müller, N. S., & Studer, M. (2008). Mining event histories: a social science perspective. *International Journal of Data Mining, Modelling and Management*, *1*, 68–90.

Segal, M. R. (1992). Tree-structured models for longitudinal data. *Journal of the American Statistical Association*, *87*, 407–418.

Sela, R. J., & Simonoff, J. S. (2009). *RE-EM trees: a new data mining approach for longitudinal data*. NYU Stern Working Paper SOR-2009-03.

Simonoff, J. S. (2003). *Analyzing categorical data*. New York: Springer.

Therneau, T. M., & Atkinson, B. (2010). *rpart: recursive partitioning*. R port by Brian Ripley. R package version 3.1-46.

Witten, I. H., & Frank, E. (2000). *Data mining*. New York: Morgan Kauffman.

Verbeke, G., & Molenberghs, G. (2000). *Linear mixed models for longitudinal data*. New York: Springer.

West, B. T., Welch, K. B., & Galecki, A. T. (2007). *Linear mixed models: a practical guide using statistical software*. Boca Raton: Chapman and Hall/CRC.

Zhang, H. (1997). Multivariate adaptive splines for analysis of longitudinal data. *Journal of Computational and Graphical Statistics*, *6*, 74–91.

Zhang, H. (1998). Classification trees for multiple binary responses. *Journal of the American Statistical Association*, *93*, 180–193.