Regression Trees for Longitudinal and Clustered Data: Methods, Applications, and Extensions

Jeffrey S. Simonoff (New York University)

Joint work with Denis Larocque, Rebecca J. Sela, and Wei Fu

Modern Modeling Methods (M^3) Conference, May 25, 2016

Outline of talk

Longitudinal data

Panel or longitudinal data, in which we observe many individuals over multiple periods, offers a particularly rich opportunity for understanding and prediction, as we observe the different paths that a variable might take across individuals. Such data, often on a large scale, are seen in many applications:

- test scores of students over time
- blood levels of patients over time
- transactions by individual customers over time
- tracking of purchases of individual products over time

Longitudinal data

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. Conversely, "big data" often come in clustered and longitudinal form, a structure that is typically ignored (or at least underutilized) in modern machine learning methods.

We observe a panel of *individuals* $i = 1, ..., I$ at times $t = 1, ..., T_i$. A single observation period for an individual (i, t) is termed an *observation*; for each observation, we observe a vector of covariates, $\mathbf{x}_{it} = (x_{it1}, ..., x_{itK})'$, and a numerical response, y_{it}.

Longitudinal data models

Because we observe each individual multiple times, we may find that the individuals differ in systematic ways; e.g., y may tend to be higher for all observation periods for individual i than for other individuals with the same covariate values because of characteristics of that individual that do not depend on the covariates. This pattern can be represented by an "effect" specific to each individual (for example, an individual-specific intercept) that shifts all predicted values for individual i up by a fixed amount.

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

Fixed and random effects

- If f is linear in the parameters and the $\mathbf{b}_i$ are taken as fixed or potentially correlated with the predictors, then this is a linear *fixed effects model* (analysis of covariance).
- If f is linear in the parameters and the $\mathbf{b}_i$ are assumed to be random and uncorrelated with the predictors, then the model is a linear *mixed effects model* (with random effects $\mathbf{b}_i$).

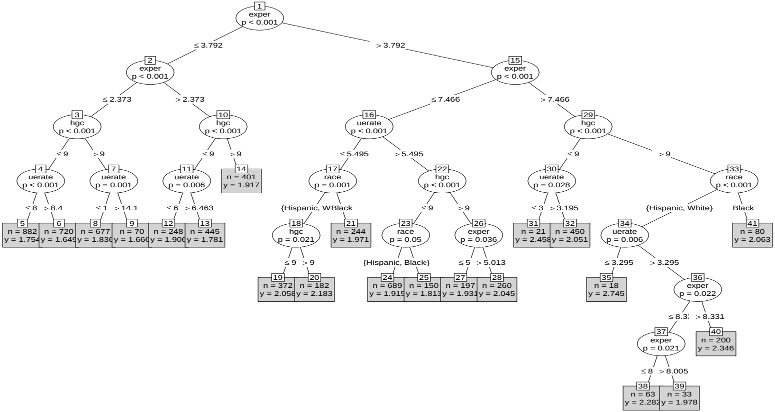
Conceptually, random effects are appropriate when the observed set of individuals can be viewed as a sample from a large population of individuals, while fixed effects are appropriate when the observed set of individuals represents the only ones about which there is interest.

Modeling for large data sets

The linear mixed effects model assumes a simple parametric form for f, which might be too restrictive an assumption; when there is a large number of individuals, a more complex functional form could be supported. Furthermore, K may be very large, requiring model selection, and linear models cannot include variables with missing values as easily as many data mining methods can.

We focus on regression trees. A regression tree is a binary tree, where each non-terminal node is split into two nodes based on the values of a single predictor. This method allows for interactions between variables and can represent a variety of functions of the predictors.

Regression tree for National Longitudinal Survey of Youth (NLSY) logged wages data



Previous research

Historically most approaches to extending tree models to longitudinal or clustered data were based on concepts from multivariate response data (the repeated responses for a particular individual are treated as a multivariate response from that individual, and the splitting criterion is modified accordingly):

- Gillo and Shelly (1974)
- Segal (1992)
- De'Ath (2002) (`mvpart`)
- Larsen and Speckman (2004)
- Loh and Zheng (2013) (`GUIDE`)

Previous research

This approach has several challenges:

- It requires the same number of time points for all individuals.
- It uses a single set of predictors for all of the observation periods, which means that either time-varying (observation-level) predictors cannot be used, or predictor values from later time periods can potentially be used to predict responses from earlier ones even though that is probably contextually unrealistic.
- It cannot be used for the prediction of future periods for the same individuals in a direct way.
- Missing data is a challenge.

Hajjem et al. (2011) and Sela and Simonoff (2012) independently proposed an approach that accounts for the longitudinal structure of the data while avoiding these difficulties.

"EM"-type algorithm

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

If the random effects, $\mathbf{b}_i$, were known, the model implies that we could fit a regression tree to $y_{it} - Z_{it}\mathbf{b}_i$ to estimate f, using for example CART. If the fixed effects, f, were known and can be represented as a linear function, then we could estimate the random effects using a traditional mixed effects linear model with fixed effects corresponding to the fitted values, $f(x_i)$. This alternation between the estimation of different parameters is reminiscent of (although is not) 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**. Hajjem et al. refer to this as the MERT (mixed effects regression tree) method.

Estimation of a RE-EM Tree

- The fitting of the regression tree uses built-in methods for missing data, such as probabilistic or surrogate split.
- The fitting of the random effects portion of the model can be based on either independence within individuals, or a specified autocorrelation structure.
- Multilevel hierarchies (e.g., classrooms within schools within school districts within counties) are easily handled.
- Nodes are defined at the *observation* level, not the *individual* level; that is, different observations of the same individual end up in different (terminal) nodes. This is why observation-level (time-varying) covariates are easily accommodated.

Transaction data set

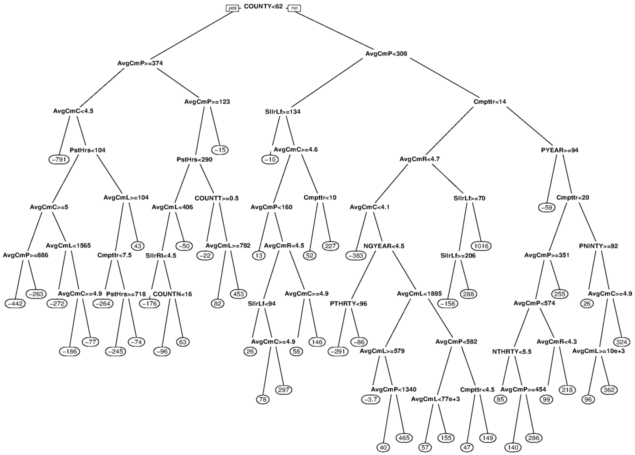
We apply this method to a dataset on third-party sellers on Amazon Web Services to predict the prices at which software titles are sold based on the characteristics of the competing sellers (Ghose, 2005). The goal is to use the tree structure of the RE-EM tree to describe the factors that appear to influence prices. We also use the dataset to compare the predictive performance of the RE-EM tree to that of alternative methods through two types of leave-one-out cross validation.

The data consist of 9484 transactions for 250 distinct software titles; thus, there are $I = 250$ individuals in the panel with a varying number of observations T_i per individual.

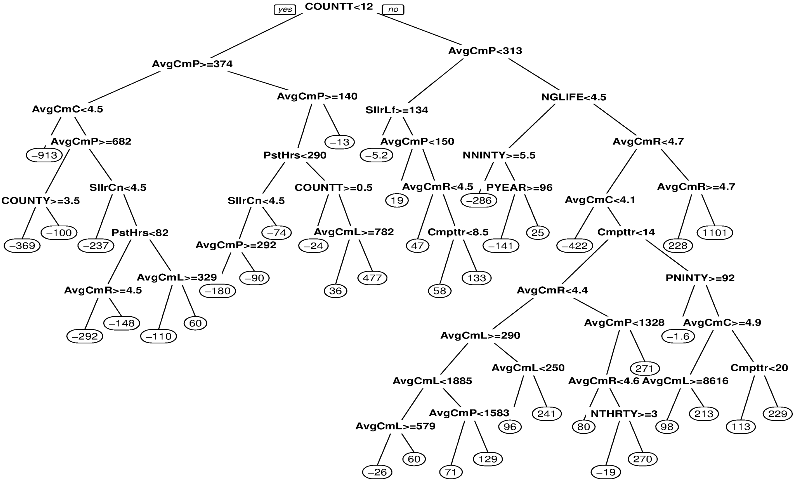
Transaction data set

- Target variable: the price premium that a seller can command (the difference between the price at which the good is sold and the average price of all of the competing goods in the marketplace).
- Predictor variables
  - The seller's own reputation (total number of comments, the number of positive and negative comments received from buyers, the length of time that the seller has been in the marketplace)
  - The characteristics of its competitors (the number of competitors, the quality of competing products, and the average reputation of the competitors, and the average prices of the competing products).

Tree ignoring longitudinal structure



RE-EM Tree


yes
COUNTT<12
no
AvgCmP>=374
AvgCmP<313
AvgCmC<4.5
AvgCmP>=140
SllrLf>=134
NGLIFE<4.5
AvgCmP>=682
-913
PstHrs<290
-13
-5.2
AvgCmP<150
NNINTY>=5.5
AvgCmR<4.7
COUNTY>=3.5
SllrCn<4.5
SllrCn<4.5
COUNTT>=0.5
19
AvgCmR<4.5
-286
PYEAR>=96
AvgCmC<4.1
AvgCmR>=4.7
-369
-100
-237
PstHrs<82
AvgCmP>=292
-74
-24
AvgCmL>=782
47
Cmpttr<8.5
-141
25
-422
Cmpttr<14
228
1101
AvgCmR>=4.5
AvgCmL>=329
-180
-90
36
477
58
133
AvgCmR<4.4
PNINTY>=92
-292
-148
-110
60
AvgCmL>=290
AvgCmP<1328
-1.6
AvgCmC>=4.9
AvgCmL<1885
AvgCmL<250
271
AvgCmR<4.6
AvgCmL>=8616
Cmpttr<20
AvgCmL>=579
AvgCmP<1583
96
241
80
NTHRTY>=3
98
213
113
229
-26
60
71
129
-19
270

Cross-validated RMSE accuracy

Method	Excluding Observations	Excluding Titles
Linear Model	95.88	96.92
LM with RE	73.62	461.48
LM with RE - AR(1)	74.75	387.18
rpart	69.66	89.38
RE-EM Tree	64.54	88.53
RE-EM Tree - AR(1)	63.88	87.90
FE-EM Tree	65.67	91.10

Properties of RE-EM trees

- When the true data generation process is a tree
  - RE-EM tree is best
- When the true data generation process is a linear model
  - linear mixed effects model is best for small samples
  - RE-EM tree is as good as the linear model with random effects when T or I are large for most types of predictions
- When the true data generation is a complex polynomial model with interactions the relative performance of the tree and linear model methods are similar to when it is a linear model.

Properties of RE-EM trees

- RE-EM tree provides more accurate estimates of random effects in almost all situations.
- When the true data generation process is a tree RE-EM tree provides best estimates of true fixed effects.
- When the true data generation process is a linear or polynomial model
    - linear mixed effects model provides best estimates of true fixed effects for small samples
    - RE-EM tree is as good as linear model when T or I are large
- Autocorrelation hurts all models, but hurts linear models more.

Variable selection bias

Tree methods like CART suffer from a variable selection (splitting) bias, in that the algorithm is more likely to split on variables with a larger number of possible split points. This bias is introduced because the tree is constructed based on maximization of a splitting criterion over all possible splits simultaneously; that is, the choice of which variable to split on and where the split should be are made in a single step. As a result of this, in general, standard measures of impurity will prefer a variable that has been randomly partitioned into a larger number of values as a candidate for splitting, even though the additional partition is random.

Avoiding variable selection bias

Several authors have proposed approaches that avoid this bias. In the multivariate response / longitudinal framework `GUIDE` (Loh and Zheng, 2013) and `MELT` (Eo and Cho, 2014) use χ^2 goodness-of-fit tests based on residuals to assess whether a variable should be split, with the best split set then found for that variable. In the RE-EM tree formulation, any variable selection bias comes from the use of CART as the underlying tree method using $y_{it} - Z_{it}\hat{\mathbf{b}}_i$ as the responses, but there is no requirement that CART be used for this; if a tree method that has unbiased variable selection is used instead, the resultant RE-EM tree should inherit that lack of bias.

Conditional inference trees

We replace CART with the *conditional inference tree* proposed by Hothorn et al. (2006). This method is based on a hypothesis testing approach, in which the process of choosing variables on which to split is stopped when the hypothesis that all of the conditional distributions of y given X_j equal the unconditional distribution cannot be rejected. The testing is based on a permutation version of each conditional distribution, addressing the bias problem (since the p-value for the test of association of y and X_j is not related to the number of potential splitting points of X_j). The split point itself can be determined by any criterion, and unlike CART, no pruning procedure is necessary (avoiding the randomness of the 10-fold cross-validation pruning procedure).

The algorithm that implements this method is available in the R packages `party` and `partykit`.

Properties of unbiased RE-EM trees

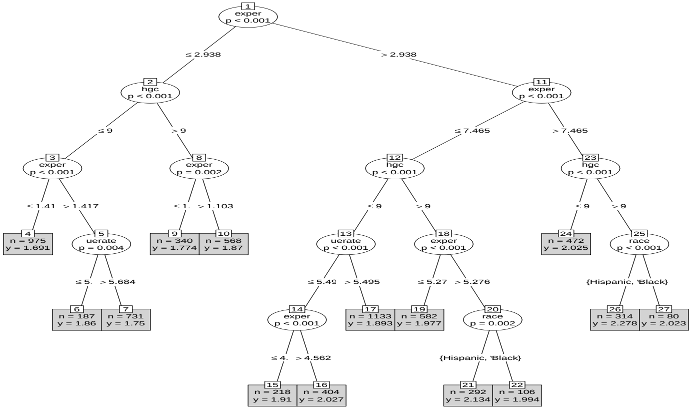
- CART-based RE-EM trees inherit the tendency to split on variables with more possible split points, but the unbiased RE-EM tree completely corrects for that phenomenon.
- The unbiased tree has lower error in estimating fixed effects.
- The unbiased tree has much better performance at recovering structure when the true first split variable is binary and there is a correlated continuous predictor present in the data.

NLSY logged wages data

We examine data from the National Longitudinal Survey of Youth (NLSY), focusing on wage data that were also analyzed by Singer and Willett (2003). The data consist of 888 high school dropouts, ages 14-17, with the goal being to model hourly wages (in constant 1990 dollars). Predictors include `race` (White, Black, or Hispanic), `hgc` (highest grade of schooling completed), which are time-invariant, and `exper` (duration of work experience in years), `ged` (whether the respondent had earned a high school equivalency degree at the time) and `uerate` (unemployment rate at the time).

The CART-based RE-EM tree splits only on experience at roughly 3 and 7.5 years, which seems an unlikely "true" result.

NLSY logged wages data; unbiased RE-EM tree



Drawback of standard regression trees

A potential weakness of these methods is that they restrict the expected response at each terminal node to be a constant. Eo and Cho (2014) proposed MELT, a regression tree method for longitudinal data that provides an estimated slope at each terminal node for a linear function of time. This method cannot handle time-varying covariates in a direct fashion, and since it does not provide an estimate of the intercept cannot be used for prediction.

Functionals at the terminal nodes

A modification of the basic MERT/RE-EM idea that allows for linear functions of predictors at terminal nodes rather than simply mean responses was proposed by Larocque and Simonoff (2015) and Fokkema et al. (2015). This is done through the use of model-based partitioning, as is discussed in Zeileis et al. (2008). Bürgin and Ritschard (2015) proposed a similar idea for ordinal multinomial responses.

Implementing a linear function of time at the nodes

We will focus on the most natural situation for longitudinal data, in which the functional form is a linear function of time, producing different growth curves for different subsets of observations. The underlying model is

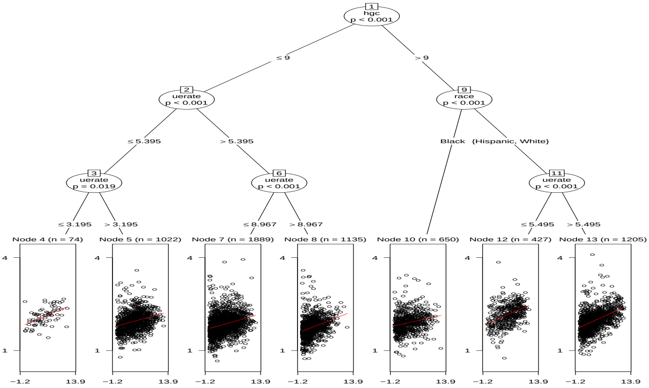
$$y_{it} = Z_{it}\mathbf{b}_i + f\left(x_{it1}, \ldots, x_{itk}, \mathtt{time}_{it}\right) + \varepsilon_{it},$$

The algorithm proceeds by alternating between estimating a regression tree (splitting on the $\mathbf{x}$ variables) with a linear function of time ($\beta_0 + \beta_1 \times \mathtt{time}$) at each node, assuming that our estimates of the random effects are correct, and estimating the random effects, assuming that the model-based regression tree is correct. This results in a **MOD**el-bas**E**d **R**a**N**dom effects tree, or a **MODERN** tree.

Properties of trees

- When it can be applied MELT's slope estimates are far more variable than MODERN's, and it cannot be used for prediction, so it falls short in routine applications.
- MODERN is competitive with REEM/MERT when slopes are all zero, but much better when slopes are nonzero.
- Performance of MODERN is relatively insensitive to the values of β_0 and (nonzero) β_1.
- Longitudinal trees (including MODERN) are more effective when there are time-varying splitting variables, as they result in lower variability of intercept estimates (this is related to the estimation error of $\tilde{\mathbf{b}}$). Slope estimates are unaffected.

MODERN tree for NLSY logged wages data (experience as time)



Linear mixed effects models and goodness-of-fit

Recall the general mixed effects model

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

The most common choice of f is of course the linear model

$$y_{it} = Z_{it}\mathbf{b}_i + X_{it}\boldsymbol{\beta} + \varepsilon_{it},$$

assuming errors ε that are normally distributed with constant variance. This model has the advantage of simplicity of interpretation, but as is always the case, if the assumptions of the model do not hold inferences drawn can be misleading. Such model violations include *nonlinearity* and *heteroscedasticity*. If specific violations are assumed, tests such as likelihood ratio tests can be constructed, but omnibus goodness-of-fit tests would be useful to help identify unspecified model violations.

Regression trees and goodness-of-fit

The idea discussed here is a simple one that has (perhaps) been underutilized through the years: since the errors are supposed to be unstructured if the model assumptions hold, examining the residuals using a method that looks for unspecified structure can be used to identify model violations. A natural method for this is a regression tree.

Miller (1996) proposed using a CART regression tree for this purpose in the context of identifying unmodeled nonlinearity in linear least squares regression, terming it a **diagnostic tree**.

Regression trees and goodness-of-fit

Su, Tsai, and Wang (2009) altered this idea slightly by simultaneously including both linear and tree-based terms in one model, terming it an **augmented tree**, assessing whether the tree-based terms are deemed necessary in the joint model. They also note that building a diagnostic tree using squared residuals as a response can be used to test for heteroscedasticity.

The diagnostic trees are not meant to replace examination of residuals or more focused (and powerful) tests of specific model violations; rather, they are an omnibus tool to add to the data analyst's toolkit to try to help identify unspecified mixed effects model violations.

Proposed method

We propose adapting the diagnostic tree idea to longitudinal/clustered data using RE-EM trees as follows:

- Fit the linear mixed effects model.
- Fit a RE-EM tree to the residuals from this model to explore nonlinearity.
- Fit a RE-EM tree to the absolute residuals from the model to explore heteroscedasticity (squared residuals are more non-Gaussian and lead to poorer performance).

A final tree that splits from the root node rejects the null model. The structure of the tree can help suggest the form of the violation of assumptions.

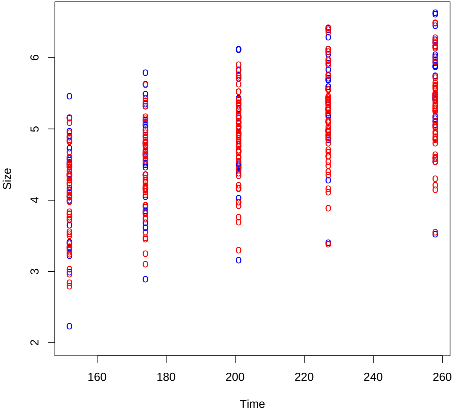
Properties of the tests

- The test based on the standard pruning rule for RE-EM has roughly .05 (or lower) Type I error. The unbiased tree based on the conditional inference tree by definition has the correct size.
- Simulations indicate good power to identify both different slopes and polynomial terms when that nonlinear structure is unknown to the analyst.
- The test for heteroscedasticity has good power to identify it when it is related to a predictor or to the response even though that structure is not theorized.

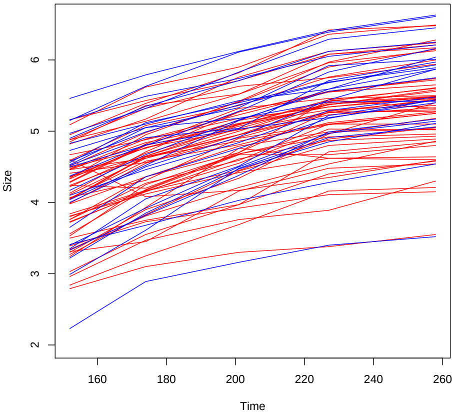
Spruce tree growth

Diggle, Liang, and Zeger (1994) and Venables and Ripley (2002) discuss a longitudinal growth study. The response is the log-size of 79 Sitka spruce trees, two-thirds of which were grown in ozone-enriched chambers, measured at five time points.

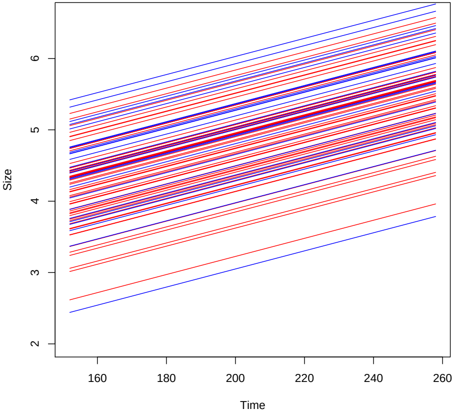
The data



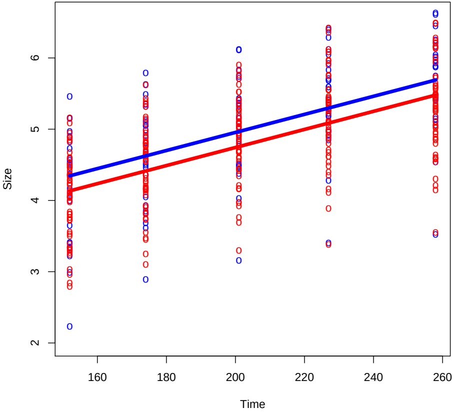
The data



Linear mixed effects model

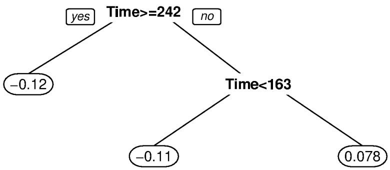


Linear mixed effects model



Diagnostic tree for lack of fit

The tree-based nonlinearity test indicates lack of fit of the linear mixed effects model, related to time.



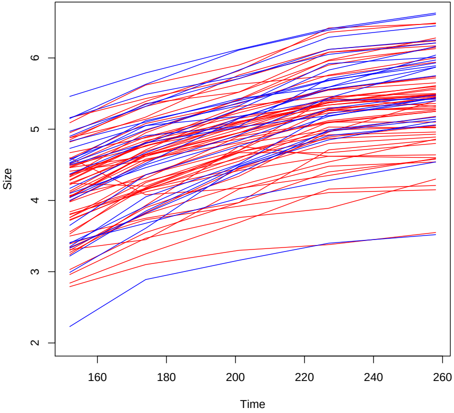
A natural alternative model is one allowing for different slopes for the treatment and control groups, but that does not correct the lack of fit.

Alternatives to the linear mixed model

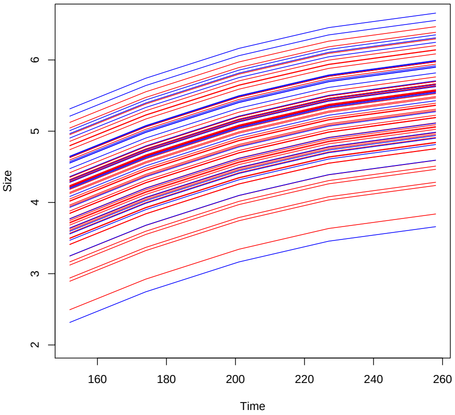
As the diagnostic trees suggest, the problem is in the linear formulation of the effect of time. If time is treated as a categorical predictor, the apparent lack of fit disappears, as the diagnostic tree has no splits.

In fact, the pattern of responses in the diagnostic tree suggests that a quadratic term in time could account for the nonlinearity, and the diagnostic tree based on the quadratic model indicates no lack of fit.

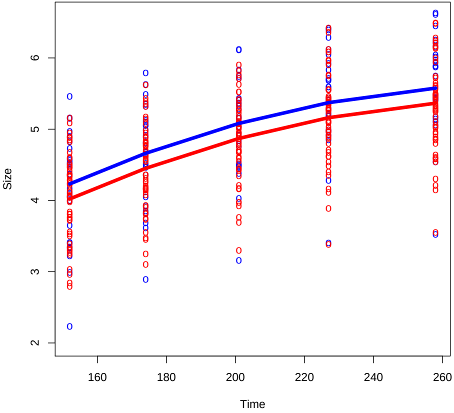
The data



Quadratic mixed effects model



Quadratic mixed effects model



Treating time as categorical

An additional interaction of the treatment and (categorical) time effects is statistically significant, but has higher AIC and BIC values than the additive model, reinforcing that from a practical point of view the fit of the simpler model is adequate.

Heteroscedasticity diagnostic trees for all models do not split.

Conclusion and future work

Random forests for these methods are discussed in Hajjem et al. (2014) and Larocque and Simonoff (2015).

Longitudinal data often come with a time-to-event (survival) aspect; for example, repeat visits of patients to their doctor, with particular interest in how changes in a patient's health relate to survival time.

- Longitudinal data are inherently time-varying, but existing work on time-varying survival trees is very limited. Can such trees be formulated in an effective way? Yes — see forthcoming Arxiv paper Fu and Simonoff (2016).
- Can tree-based methods be built that jointly model longitudinal and time-to-event data?
- Can time-to-event trees be based directly on survival time, rather than indirectly on hazard functions?

References

- Fokkema, M., Smits, N., Zeileis, A., Hothorn, T., and Kelderman, H. (2015), "Detecting Treatment-Subgroup Interactions in Clustered Data With Generalized Linear Mixed-Effects Model Trees," *University of Innsbruck Working Papers in Economics and Statistics No. 2015-10*.
- Fu, W. and Simonoff, J.S. (2015), "Unbiased Regression Trees for Longitudinal Data," *Computational Statistics and Data Analysis*, **88**, 53-74.
- Hajjem, A., Bellavance, F., and Larocque, D. (2011), "Mixed Effects Regression Trees for Clustered Data," *Statistics and Probability Letters*, **81**, 451-459.
- Hajjem, A., Bellavance, F. and Larocque, D. (2014), "Mixed Effects Random Forest for Clustered Data," *Journal of Statistical Computation and Simulation*, **84**, 1313-1328.

References

- Larocque, D. and Simonoff, J.S. (2015), "Model-Based Mixed Effects Trees for Longitudinal and Clustered Data," *Proceedings of the 30th International Workshop on Statistical Modelling, Volume 1*, Linz, Austria, 279-284.
- Sela, R.J. and Simonoff, J.S. (2012), "RE-EM Trees: A Data Mining Approach for Longitudinal and Clustered Data," *Machine Learning*, **86**, 169-207.
- Simonoff, J.S. (2013), "Regression Tree-Based Diagnostics for Linear Multilevel Models," *Statistical Modelling*, **13**, 459-480.

The `R` package `REEMtree` used to construct RE-EM trees based on `rpart` is available from CRAN. A function that adapts this to unbiased trees based on `ctree` is available at `people.stern.nyu.edu/jsimonof/unbiasedREEM/`. The `glmertree` package on R-Forge fits model-based regression trees.