RE-EM Trees: A New Data Mining Approach to Longitudinal Data

Rebecca J. Sela
Stern School of Business, New York University
Joint work with Jeffrey Simonoff

Conference on Quantitative Social Science Research Using R
June 19, 2009
Fordham University

Longitudinal (or panel) datasets include multiple observations per individual.

- We observe individuals $i = 1, ..., I$ at times $t = 1, ..., T_i$.
- This structure allows us to follow the path of the target variable as it changes for each individual.
- The unobserved characteristics that are constant for an individual mean that observations for the same individual will not be independent.

Longitudinal structures have been studied extensively in linear models, models for count data, and models for multinomial data.

The relationship between observations for the same individual should not be ignored.

- In the case of a linear model where the individual-specific effects are uncorrelated with the predictors,
  - The parameter estimates will be consistent, but inefficient.
  - The standard errors will be wrong.
  - Predictions for future observations for the individuals in the sample will be inefficient.
- If the effects are correlated with the predictors, the parameter estimates will be biased as well.

The linear model has been particularly well-studied.

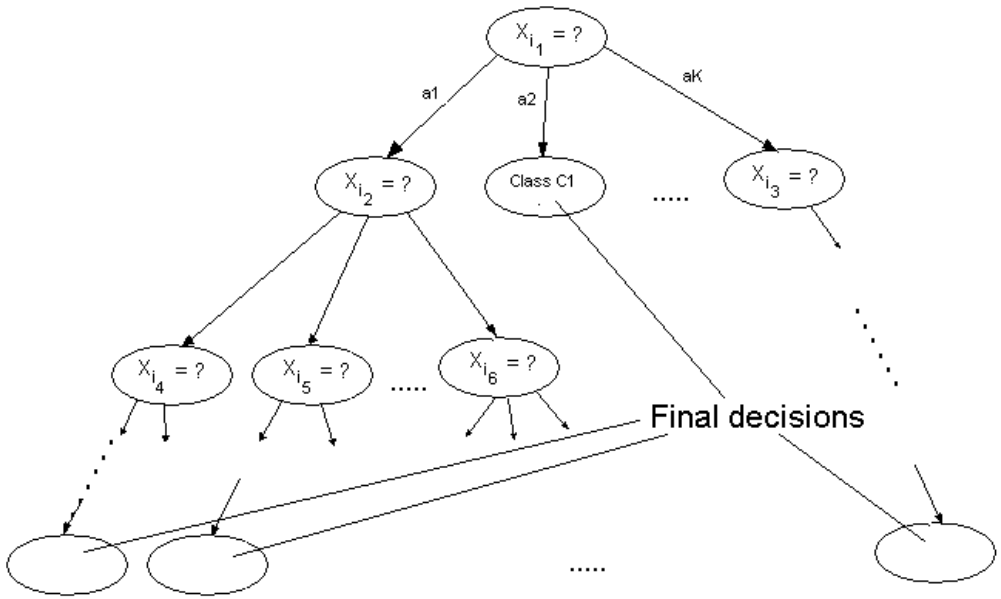
$$
\begin{aligned}
y_{it} &= Z_{it}\mathbf{b}_i + \beta_1 x_{it1} + \cdots + \beta_K x_{itK} + \varepsilon_{it} \\
\begin{pmatrix} \varepsilon_{i1} \\ \dots \\ \varepsilon_{iT_i} \end{pmatrix} &\sim Normal(0, R_i)
\end{aligned}
$$

- If $Z_{it} = 1$ and the $\mathbf{b}_i$ are fixed scalars, this is a *fixed effects* linear model.
- If $Z_{it} = 1$ and the $\mathbf{b}_i$ are randomly distributed scalars that are uncorrelated with the predictors, this is a *random effects* linear model.
- If $Z_{it} = (1, x_{itk})$ for some k, this is a *random parameters* model.

Regression trees, such as CART (Breiman, et. al., 1984), have been studied less for longitudinal data.

- Regression trees are based on recursive partitioning, where the space of all observations is split into subsets based on one predictor at a time.
- Trees partition the data recursively to minimize the "impurity" in each node.
- The splits could continue until the target variable takes on a single value in each node.
- Control parameters of the fitting process are chosen to avoid overfitting the tree.
- To predict future observations, we follow the splits to the correct node and use the mean response at that node.

The tree splits on the predictors to make the values of the target similar in each node.



Trees have some advantages over parametric models.

- They allow for the fitting of complex, flexible, nonlinear structures to a set of data.
- They produce easy-to-understand rules for prediction.
- They can be fit to data with numerical and categorical predictors and with missing values for some predictors.
- Efficient algorithms exist for them that allow for fitting to massive data sets.
- They work, particularly for datasets that are large or have complex structure.

We will be estimating:

$$
\begin{aligned}
y_{it} &= Z_{it}\mathbf{b}_i + f(x_{it1}, ..., x_{itK}) + \varepsilon_{it} \\
\begin{pmatrix} \varepsilon_{i1} \\ \dots \\ \varepsilon_{iT_i} \end{pmatrix} &\sim Normal(0, R_i) \\
\mathbf{b}_i &\sim Normal(0, D)
\end{aligned}
$$

- We observe covariates, x_{itk}, $k = 1, ..., K$, and a response, y_{it}.
- We also observe a known design matrix, Z_{it} (which may vary each period and depend on the covariates).
- There is a vector of unobserved, time-constant, individual-specific effects, $\mathbf{b}_i$.
- We will model f using a regression tree.

So far, there have been two approaches to regression trees with longitudinal data.

- One approach models all of the responses for an individual as a single response variable.
- Another approach modifies the split function to try to account for the covariance structure.

The first approach was suggested by Segal (1992) and De'Ath (2002).

- Suppose $T_i = T$ for all i.
- The response variable is the vector $y_i = (y_{i1}, ..., y_{iT})$.
- Their goal is to compute a vector of means at each node.
- Splits are based on a weighted sum of squares, since the target variable is now a vector.

This approach has two weaknesses.

- It requires a single set of predictor values for each individual for all time periods, which rules out time-varying covariates.
- It explicitly models only the set of time periods that are in the training data, which means that the resulting tree cannot be used to forecast values from future time periods for individuals in the training data (within-individual forecasts).

Galimberti and Montanari (2002) suggest a different approach.

- They model each response as a function of its own covariates.
- They modify the split function to account for possible correlation across observations for an individual, even if they are in different nodes.

This has some drawbacks:

- The covariances of the errors and random effects must be estimated prior to fitting the regression tree.
- The random effects are not estimated, which means they cannot be used for prediction.
- This modification of the split function is not generally available in software and cannot handle missing observations.

We offer an alternative.

$$
\begin{aligned}
y_{it} &= Z_{it}\mathbf{b}_i + f(x_{it1}, ..., x_{itK}) + \varepsilon_{it} \\
\begin{pmatrix} \varepsilon_{i1} \\ \dots \\ \varepsilon_{iT_i} \end{pmatrix} &\sim Normal(0, R_i) \\
\mathbf{b}_i &\sim Normal(0, D)
\end{aligned}
$$

We suggest an estimation method that uses a tree structure to estimate f, but also incorporates individual-specific random effects, $\mathbf{b}_i$.

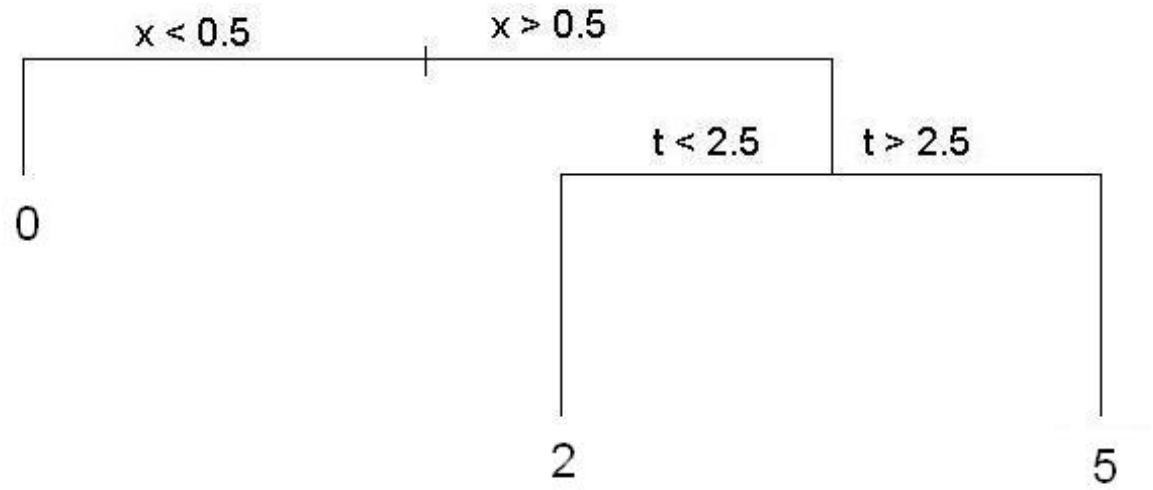
In our approach,

- As in Galimberti and Montanari's approach, the nodes may split based on any covariate, so that different observations for the same individual may be placed in different nodes, with different (time-varying) covariate values.
- Our estimation method is based on the Expectation-Maximization (EM) algorithm of Laird and Ware (1982), where the M-step is based on using a regression tree instead of traditional parametric maximum likelihood methods.
- This method creates a *Random Effects-Expectation Maximization* (RE-EM) Tree.

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:
   - 2.1 Estimate a regression tree approximating f, based on the target variable, $y_{it} - Z_{it}\hat{\mathbf{b}}_i$, and predictors, $\mathbf{x}_{it\cdot} = (x_{it1}, ..., x_{itK})$, for $i = 1, ..., I$ and $t = 1, ..., T_i$.
   - 2.2 Use this regression tree to create a set of indicator variables, $I(\mathbf{x}_{it\cdot} \in g_p)$, where g_p ranges over all of the terminal nodes in the tree.
   - 2.3 Fit the linear random effects model, $y_{it} = Z_{it}\mathbf{b}_i + I(\mathbf{x}_{it\cdot} \in g_p)\mu_p + \varepsilon_{it}$. Extract $\hat{\mathbf{b}}_i$ from the estimated model.

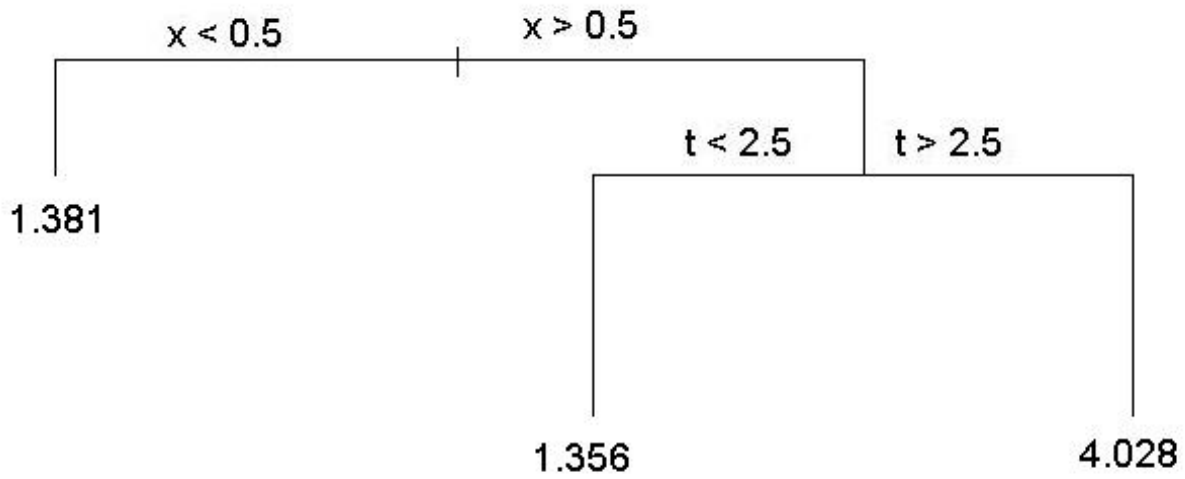
We illustrate the algorithm with a simple example.



$$y_{it} = b_i + 2x_i + 3I(t > 2 \cap x_i = 1) + \varepsilon_{it}$$

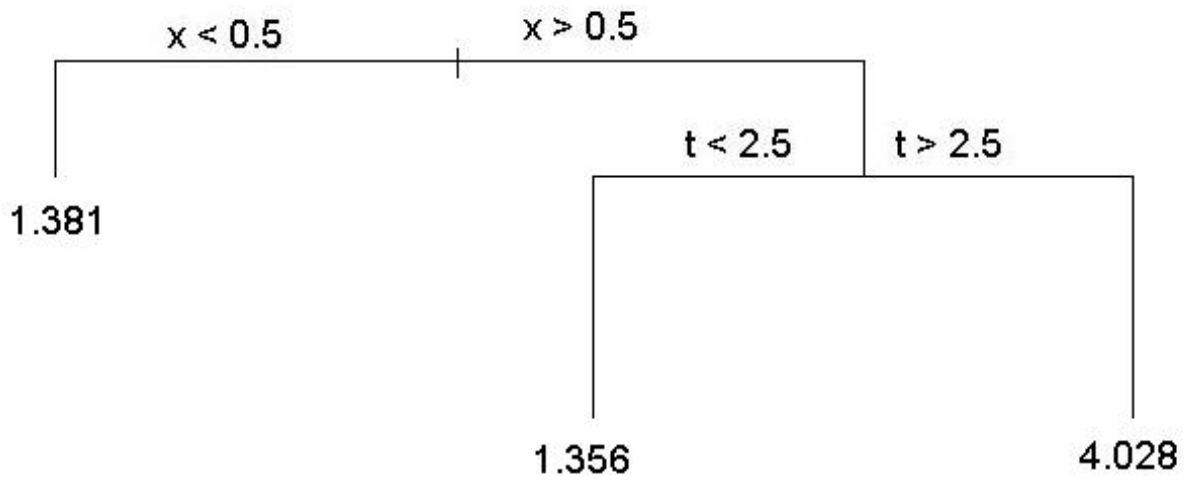
$$x_i = \begin{cases} 1 & i = 1, 2, 3 \\ 0 & i = 4, 5, 6 \end{cases}$$

The initial tree estimate is assumes that all the random effects are equal to zero.



$$\hat{y}_{it} = \hat{b}_i + 1.38 - 0.02I(x_{it\cdot} \in g_4) + 2.65I(x_{it\cdot} \in g_5)$$

In this simple case, the tree structure does not change after the random effects are estimated.



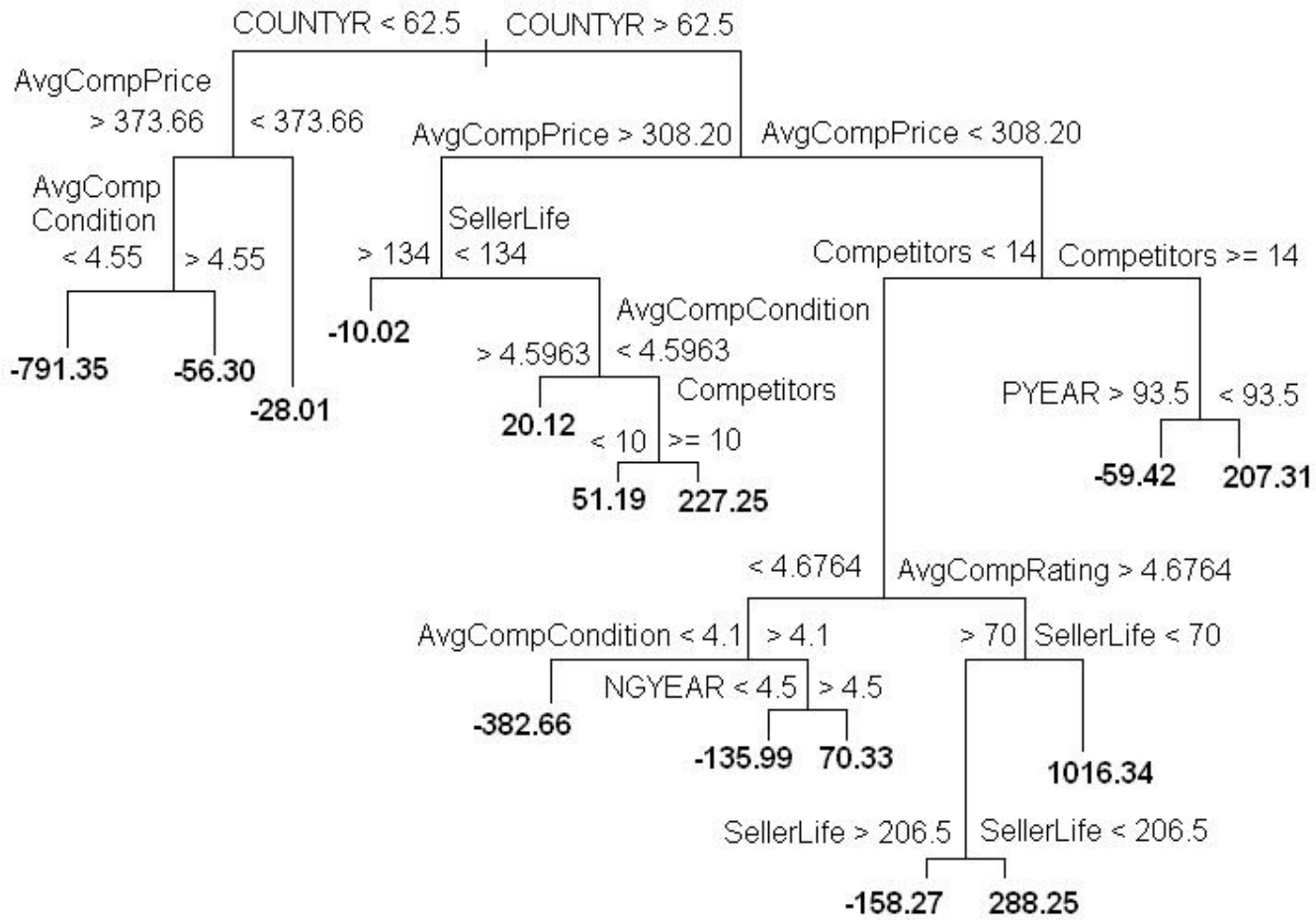
- Since the tree structure did not change, the algorithm converged in one step.
- For data with more structure, many steps may be required, and the fitted tree may change at each step.

Example: Amazon Transactions Data

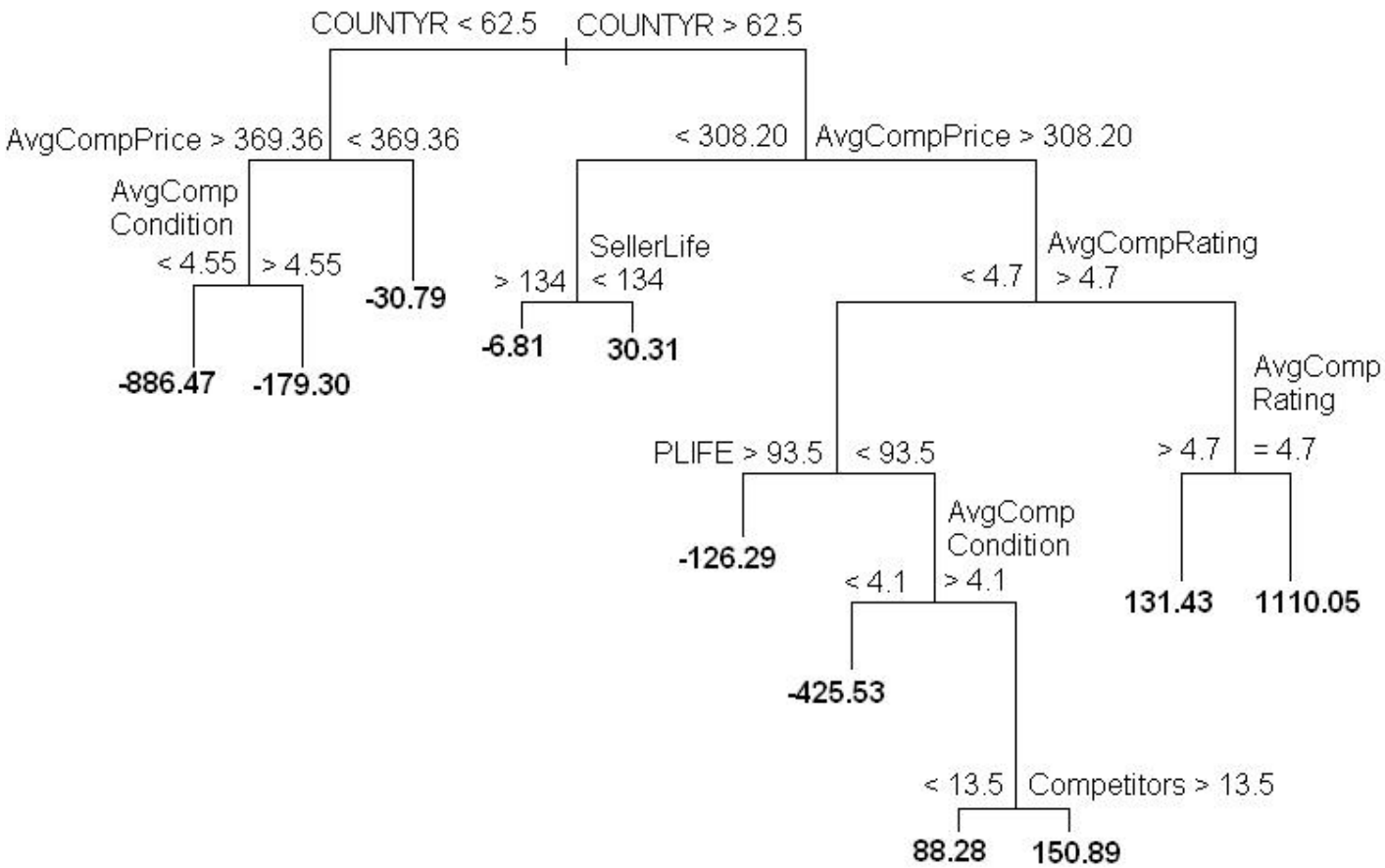
We now apply RE-EM trees to data on software transactions for third-party sellers on Amazon Web Services, from Ghose, et. al. (2005).

- Target variable: The price premium that a seller can command.
- Predictors: Seller's reputation (measured by a variety of variables), characteristics of the competitors, quality of the product sold.
- This dataset consists of 9484 transactions for 250 distinct software titles.

The estimated tree without random effects



The estimated RE-EM tree



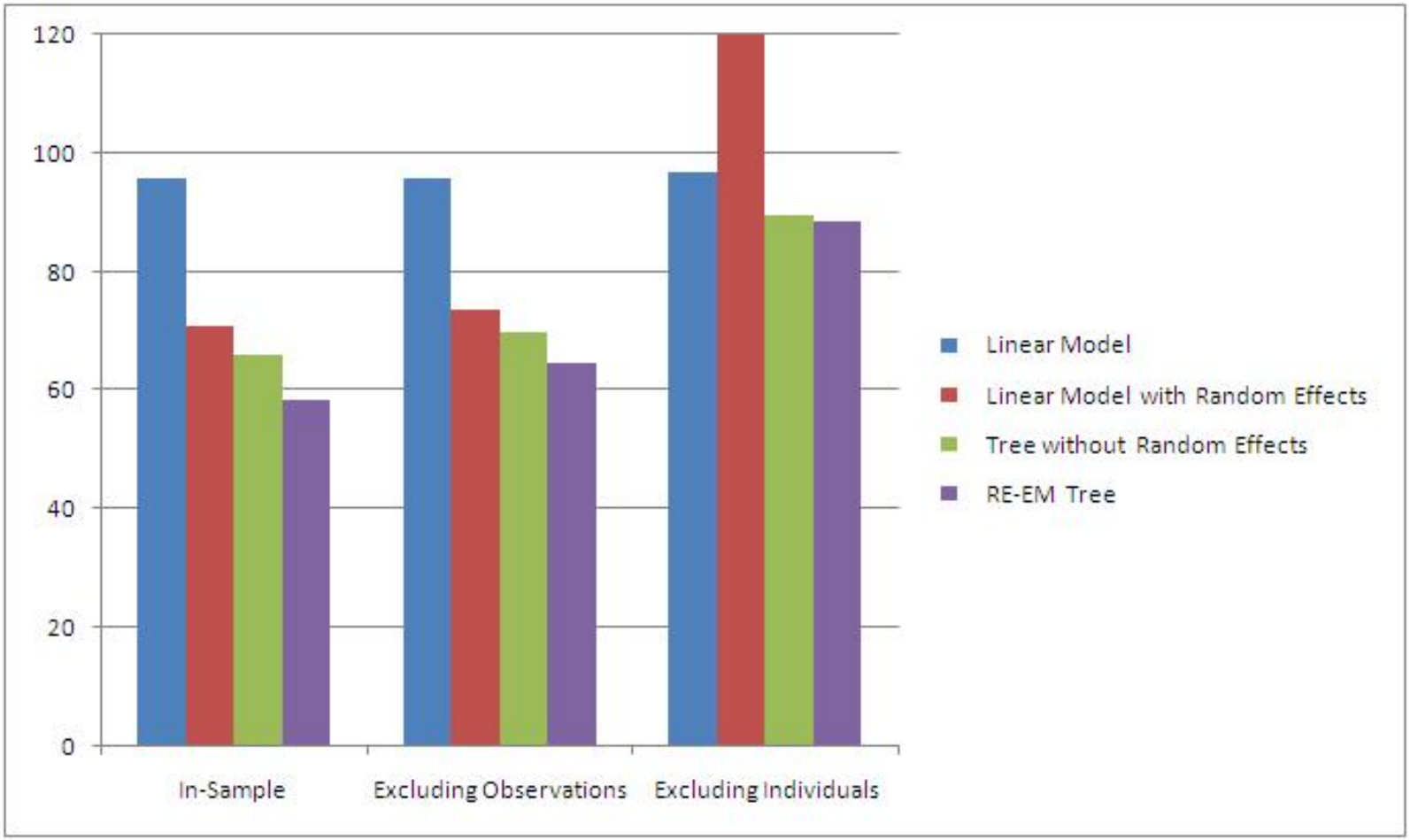
There are different types of out-of-sample predictions.

- Future observations for individuals in the sample: Predict $f(x_{it1}, ..., x_{itK})$ using the estimated tree and then add on $Z_i\hat{\mathbf{b}}_i$.
- Individuals for whom there are no observations of the response: Set $\hat{\mathbf{b}}_i = 0$, so the best predictor is $f(x_{it1}, ..., x_{itK})$.

We now compare root mean squared errors of various types of prediction.

- We compare the root mean squared errors in three cases:
  - In-sample fits
  - Leave-one-out cross-validation in which one observation is excluded at a time.
  - Leave-one-out cross-validation in which all of the observations for one individual are excluded at a time.
- We compare four models:
  - Linear model without random effects (`lm`)
  - Linear model with random effects (`lme` from the package `nlme`)
  - Regression tree without random effects (`rpart` from the package `rpart`)
  - RE-EM tree

The RE-EM tree performs well in all of the cases.



In this paper:

- We have presented a new technique for applying regression trees to longitudinal data.
- We have shown its applicability to a dataset about software transactions.

The study of RE-EM trees is just beginning and can go in many directions.

- Fitting trees requires choosing a variety of parameters; thus far, we have considered only the defaults.
- Boosting, bagging, and other tools that have been shown to improve tree performance, and may improve longitudinal data models.
- The RE-EM methodology may be applied to classification trees.
- RE-EM trees may be used in a wide variety of applications.

If you are interested in using RE-EM trees with your data:

`http://pages.stern.nyu.edu/~rsela/REEMtree/code.html`

E-mail: `rsela@stern.nyu.edu`

References I

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone (1984), *Classification and Regression Trees*, Wadsworth.

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

G. Galimberti and A. Montanari (2002), "Regression trees for longitudinal data with time-dependent covariates," in *Classification, Clustering and Data Analysis*, Springer.

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

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

References II

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

FE-EM Trees

One could replace the random effects linear model by a fixed effects linear model. Potential problems include:

- Perfect collinearity if the trees split only on predictors that are constant for an individual.
- Lost degrees of freedom relative to random effects linear models.
- Prediction for new individuals.

Even when the true DGP is a linear random effects model, the RE-EM tree performs well in predicting future observations.

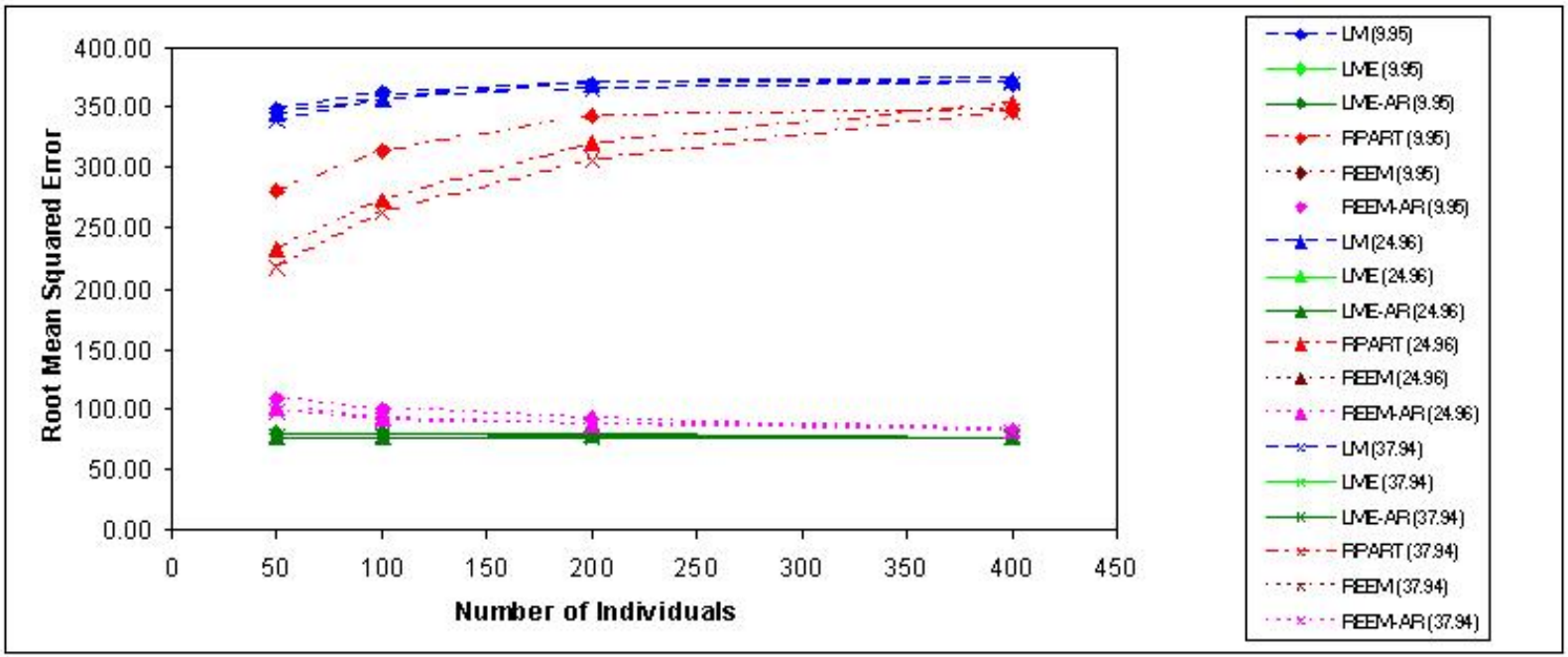


Figure: Root mean squared errors of estimated values of $f(x)$ when the true data generating process is a linear model.

The RE-EM tree also performs quite well in predicting observations for new individuals.

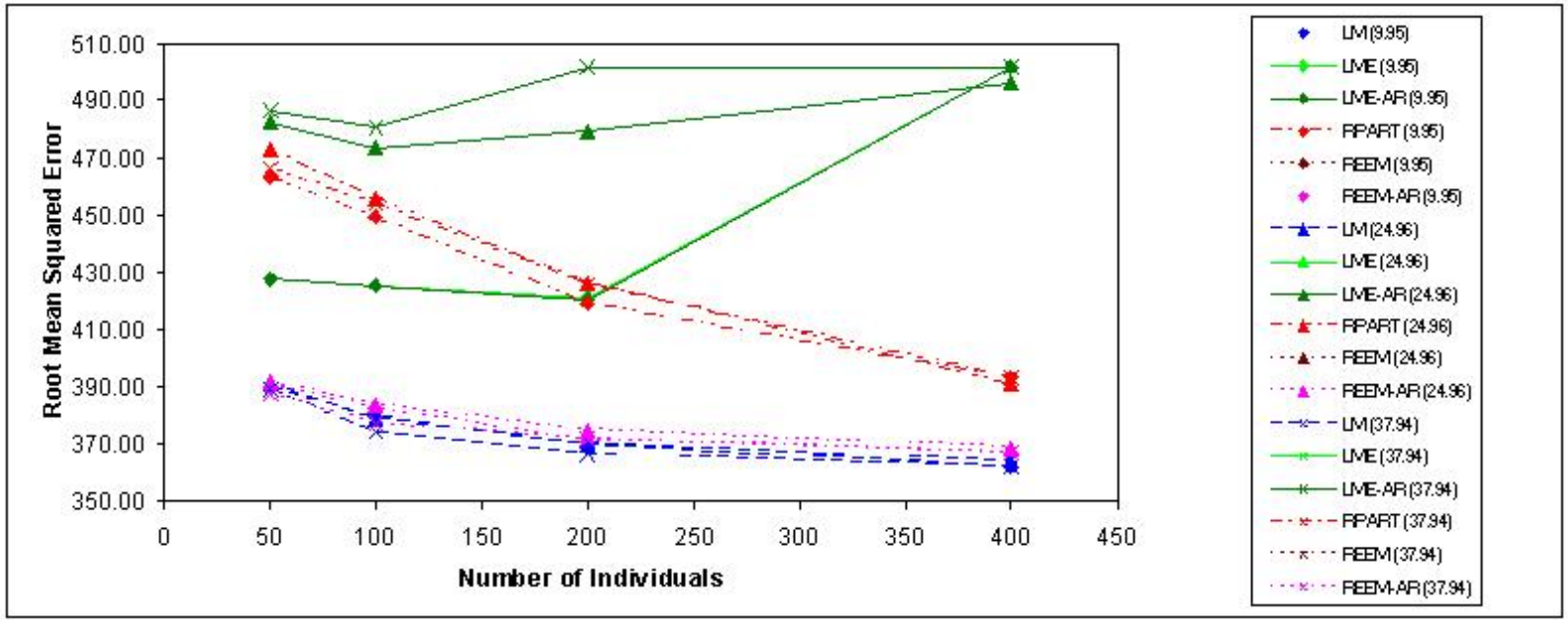


Figure: Root mean squared errors of estimated values of $f(x)$ when the true data generating process is a linear model.

The RE-EM tree is successfully estimates the value of f for each observation.

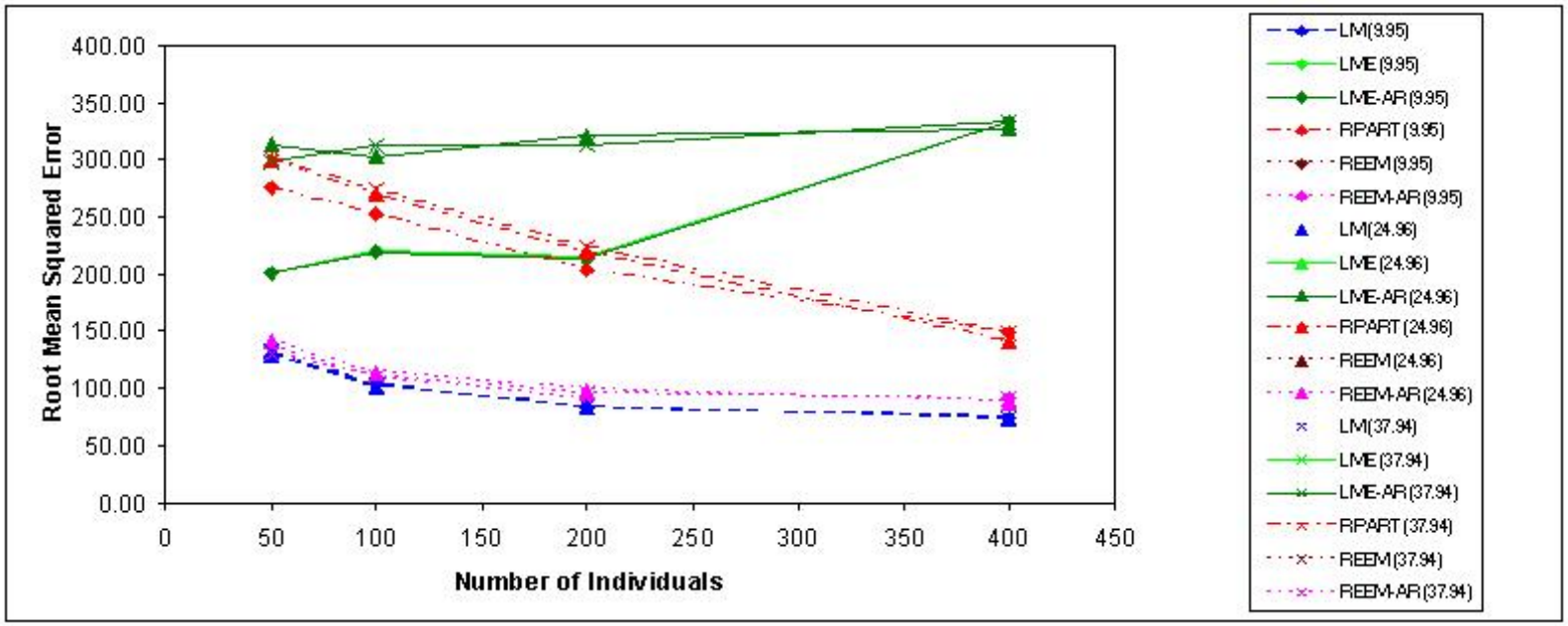


Figure: Root mean squared errors of estimated values of $f(x)$ when the true data generating process is a linear model.

The RE-EM tree also estimates the values of the random effects.

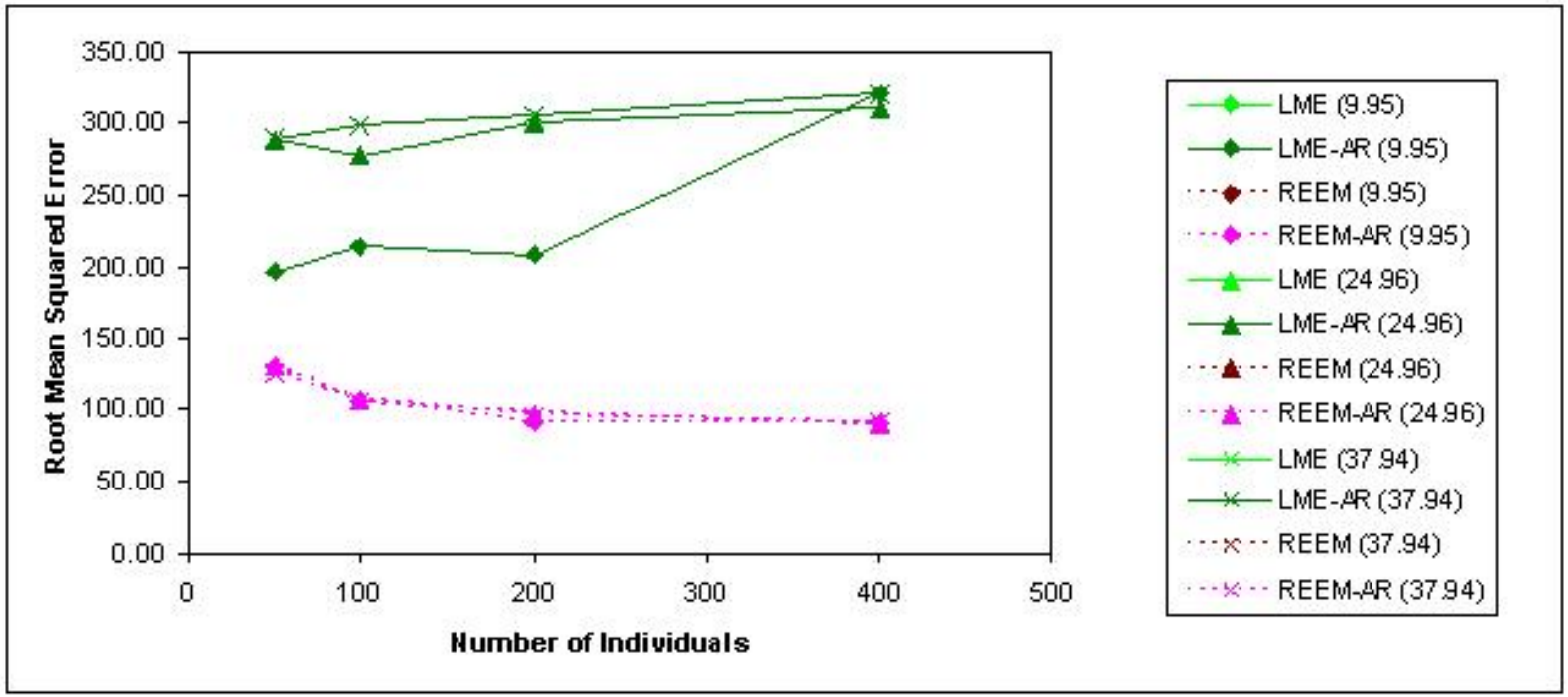


Figure: Root mean squared errors of estimated random effects when the true data generating process is a linear model.