Mixed Effects Trees and Forests for Clustered Data

Denis Larocque\(^{(1)}\)
joint work with Ahlem Hajjem\(^{(2)}\) and François Bellavance\(^{(1)}\)

(1) Department of Management Sciences, HEC Montreal
(2) Department of Marketing, Université du Québec à Montréal (UQAM)

March 2014
Talk based on:


Outline I

1 Introduction
   - Regression Trees and Random Forests
   - Clustered Data
   - Problem Statement
   - Linear Mixed Models
   - Previous Work on Trees and Forests for Correlated and Multivariate Data

2 Mixed Effects Regression Tree (MERT) and Forest (MERF)
   - Model
   - EM-Algorithm for LMM
   - EM-Algorithm for MERT (MERF)

3 Simulation Study: Part 1
   - MERT vs Standard Tree
Outline II

- MERF vs RF

4 Data Example 1
  - Description of the Data
  - Results

5 Generalized Mixed Effects Regression Tree (GMERT)
  - Generalized Linear Mixed Models
  - Penalized Quasi–Likelihood (PQL)
  - Generalized Mixed Effects Regression Tree (GMERT)

6 Simulation Study: Part 2
  - GMERT vs Standard Tree

7 Data Example 2
  - Description of the Data
  - Results
Trees

- Tree–based methods like CART (Breiman et al. 1984) and GUIDE (Loh 2011) are valuable alternatives to parametric methods. Their ability to automatically detect certain types of interactions and their ease of interpretation and visualization makes them tools of choice for practitioners.

- The basic idea is to recursively partition the covariates space by improving a performance criterion at each step.

- Usually restrict the search to two–way splits with one covariate at a time.
For a continuous (or at least ordinal) covariate $x$, the possible splits take the form $x \leq c$ where $c$ is a specified cutpoint.

For a categorical covariate $x$, the possible splits take the form $x \in \{c_1, \ldots, c_l\}$ where $\{c_1, \ldots, c_l\}$ is a subset of the possible values of $x$.

CART (Classification and Regression Trees) proceeds by an exhausting search through all possible two–way splits. A large tree is built and then pruned back with a cross-validation scheme, to avoid over-fitting.
For a continuous outcome, the best split can be the one minimizing the least-squares criterion:
\[ \sum_{i \in t^L}(y_i - \bar{Y}_L)^2 + \sum_{i \in t^R}(y_i - \bar{Y}_R)^2, \]
where \( t^L \) (\( t^R \)) is the subset of indices of the observations that go in the left (right) node, and \( \bar{Y}_L \) (\( \bar{Y}_R \)) is the mean of the outcome in the left (right) node.

For a binary outcome, the best split can be the one minimizing the Gini total impurity in the two nodes. The Gini impurity in a node is \( \hat{\pi}_0(1 - \hat{\pi}_0) + \hat{\pi}_1(1 - \hat{\pi}_1) \), where \( \hat{\pi}_k \) is the proportion of class \( k \) observations in the node.
(Root)
\[ x_1 \leq 50 \]

\[ x_2 \leq 3 \]

Terminal (1)
Terminal (2)
Terminal (3)
Other approach: GUIDE (Generalized, Unbiased, Interaction Detection and Estimation) to protect against possible selection bias in the choice of the covariate.

Trees have been extended to more complicated settings: survival data, count data, multivariate data...

However, the prediction performance of a single tree can often be improved by using ensemble of trees with methods like Boosting and Random Forests.
Random Forests (RF) (Breiman, 2001). Fast, versatile and has the ability to work with large data sets. It has been tested and tried in a wide array of domains with real and simulated data sets and has proven to yield very accurate results. Basic algorithm:

1. Draw $K$ bootstrap samples from the original data.
2. For each bootstrap sample, grow an unpruned regression tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample $p_0 \ (0 < p_0 \leq p)$ of the $p$ predictors and choose the best split from among those variables.
3. Predict new data by averaging the predictions of the $K$ trees.

There are many R packages for trees and forests:

- `rpart`: Classic CART
- `randomForest`: Breiman’s original RF
- `party`: A computational toolbox for recursive partitioning
- `randomSurvivalForest`: RF for survival data
- `mvpart`: Multivariate regression trees

Also, there is GUIDE, a stand-alone program with multi-purposes machine learning algorithms for constructing classification and regression trees, maintained by Wei-Yin Loh (http://www.stat.wisc.edu/~loh/guide.html).
Clustered Data

Many kind of data, either observational or from designed experiments have a clustered structure:

- Students in a school
- Patients at a clinic
- Workers in a department
- Repeated measurements on an individual

Each school (clinic, department or individual) is a cluster. Observations from the same cluster are possibly correlated while observations from distinct clusters are independent.
Problem Statement

- Training sample. We have $n$ clusters of size $m_1, \ldots, m_n$ for a total of $N = \sum_{i=1}^{n} m_i$ observations.
- We have an outcome of interest $Y$ and $p$ covariates $X_1, \ldots X_p$.
- The goal is to develop a model with the training sample in order to predict new $Y$ observations, based on the covariates.
- The new observations to predict can come from known clusters (present in the training sample), or from new clusters not part of the training sample.
For a gaussian continuous outcome, the **linear mixed model** (LMM) is often written in the following form:

\[ y_i = X_i \beta + Z_i b_i + \epsilon_i, \]
\[ b_i \sim N(0, D), \quad \epsilon_i \sim N(0, R_i), \quad i = 1, 2, ..., n \text{ (clusters)} \]

- \( y_i = (y_{i1}, \ldots, y_{im_i}) \) is the vector of continuous response for cluster \( i \) \((m_i \times 1)\).
- \( X_i \) is the matrix of fixed–effects covariates \((m_i \times p)\).
- \( Z_i \) is the matrix of random–effects covariates, usually a subset of the columns of \( X_i \) \((m_i \times q)\).
- \( b_i \) is the unknown random effects vector for cluster \( i \) \((q \times 1)\).
- \( \beta \) is the unknown parameter vector for the fixed effects \((p \times 1)\).
- \( \epsilon_i \) is the vector of individual errors \((m_i \times 1)\).
- The total number of observations is \( N = \sum_{i=1}^{n} m_i \).
- For simplicity, we assume that the correlation is induced solely via the between-clusters variation, that is, \( R_i = \sigma^2 I_{m_i} \).
Previous Work: Trees

Ciampi, du Berger, Taylor and Thiffault (1991): Proposed to treat multiple continuous outcomes using a maximum likelihood criterion, under normality assumption, within their recursive partition and amalgamation process (RECPAM). The goal is to partition the population into a number of classes such that the distribution of the outcome vector is homogeneous on each class and varies across classes.
Segal (1992): Extended the regression tree methodology to repeated measures and longitudinal data by modifying the split function to accommodate multiple responses. One of his objectives was the identification of cluster subgroups, i.e., subgroups of growth curves. Hence, all the observations in a cluster end up in the same terminal node and describe the growth curve corresponding to that terminal node. All clusters must have the same number of observations.

Abdolell, LeBlanc, Stephens and Harrison (2002): By using a likelihood ratio test statistic from a mixed model as the splitting criterion, they were able to lift the requirements that subjects have an equal number of repeated observations.
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Lee (2005): Tree-based method that can analyze any type of multiple responses. His tree algorithm fits a marginal regression tree at each node using the generalized estimating equations, then separates clusters into two subgroups based on the sign of their Pearson's residual average.</td>
</tr>
<tr>
<td>5</td>
<td>Eo and Cho (2013) proposed using a model with a random intercept and a fixed time effect as the basic model in a node. The goal of their method is to find meaningful patterns over time as a function of the covariates.</td>
</tr>
</tbody>
</table>
Loh and Zheng (2013) proposed an extension of the GUIDE approach to multiple responses and show how to adapt it to the case of longitudinal data.

With these six methods, clusters (or vectors) are not separated during the splitting process. Also, they can only handle cluster-level covariates and can not include random covariate effects. Basically, for longitudinal data, the goal of these methods is to model and predict the trend of the subject’s responses. Our goal is to predict the individual responses. Hence, these methods are not aiming at solving the same problem as ours.
The last article, independently developed, is closely related to part of our work.

Sela and Simonoff (2012): RE–EM trees which are single trees with mixed effects for a gaussian outcome. Very similar to our method, MERT, presented next. They also have a R package REEMtree.

We also developed forests (MERF) and extensions (GMERT) to other types of outcomes (binary, Poisson,...).
Previous Work: Forests

1. Karpievitch, Hill, Leclerc, Dabney and Almeida (2009): Proposed the RF++ method which performs cluster based bootstrapping to create learning data for single trees in a standard random forest predictor.

2. Adler, Potapov and Lausen (2011): Found that resampling of clusters and then sampling one observation from them is better compared to sampling entire clusters.

These methods do not provide predictions of the random effects. They are basically adjusting the sampling method for clustering but do not incorporate random effects in the predictions.
Our method has the following characteristics:

1. It can handle clusters with different numbers of observations (unbalanced clusters).
2. It allows observations from a same cluster to be separated during the splitting process. Our goal is to predict the individual outcomes.
3. It allows the covariates to have random effects (at the cluster level). We will see that using them can greatly improve the predictions.
4. It can incorporate covariates both at the cluster–level and at the observation–level (which are time-varying covariates in the context of longitudinal data).
The following two examples illustrate some key points.

In the first example, patients are nested within clinics (clusters). In the second one, repeated measures are taken on subjects (clusters).
LONGITUDINAL DATA

Subject 1 (M)

Subject 2 (M)

Subject 3 (W)

Subject 4 (W)

SEX = M

SEX = W
Mixed Effects Regression Tree (MERT) and Forest (MERF)

The model behind the proposed mixed effects regression tree method is:

\[ y_i = f(X_i) + Z_i b_i + \epsilon_i, \]
\[ b_i \sim N(0, D), \epsilon_i \sim N(0, \sigma^2 I_{m_i}), \]
\[ i = 1, 2, \ldots, n, \]

where all quantities are defined as in a classical linear mixed effects model except that a more general and unspecified fixed part, \( f(X_i) \), now replaces the usual linear part \( X_i \beta \). It will be estimated with either a single tree or a forest. The random part, \( Z_i b_i \), is still assumed linear.
The major cycle for the ML-based EM-algorithm, as described in §2.2.5 of Wu and Zhang (2006) is as follows:

**Step 0.** Set \( r = 0 \). Let \( \hat{\sigma}^2(0) = 1 \), and \( \hat{D}(0) = I_q \).

**Step 1.** Set \( r = r + 1 \). Update \( \hat{\beta}(r) \) and \( \hat{b}_i(r) \)

\[
\hat{\beta}(r) = \left( \sum_{i=1}^{n} X_i^T \hat{V}^{-1}_{i(r-1)} X_i \right)^{-1} \left( \sum_{i=1}^{n} X_i^T \hat{V}^{-1}_{i(r-1)} y_i \right),
\]

\[
\hat{b}_i(r) = \hat{D}_{(r-1)} Z_i^T \hat{V}^{-1}_{i(r-1)} (y_i - X_i \hat{\beta}(r)), \quad i = 1, 2, \ldots, n,
\]

where \( \hat{V}_{i(r-1)} = Z_i \hat{D}_{(r-1)} Z_i^T + \hat{\sigma}^2_{r-1} I_{n_i}, \quad i = 1, 2, \ldots, n \).

**Step 2.** Update \( \hat{\sigma}^2(r) \), and \( \hat{D}(r) \) using

\[
\hat{\sigma}^2(r) = N^{-1} \sum_{i=1}^{n} \left\{ \hat{e}_{i(r)}^T \hat{e}_{i(r)} + \hat{\sigma}^2_{(r-1)} [n_i - \hat{\sigma}^2_{(r-1)} tr(\hat{V}_{i(r-1)})] \right\},
\]

\[
\hat{D}(r) = n^{-1} \sum_{i=1}^{n} \left\{ \hat{b}_i(r) \hat{b}_i^T(r) + [\hat{D}_{(r-1)} - \hat{D}_{(r-1)} Z_i^T \hat{V}^{-1}_{i(r-1)} Z_i \hat{D}_{(r-1)}] \right\},
\]

where \( \hat{e}_{i(r)} = y_i - X_i \hat{\beta}_{(r)} - Z_i \hat{b}_i(r), \quad N = \sum_{i=1}^{n} n_i \).

**Step 3.** Repeat Steps 1 and 2 until convergence.
The mixed effects tree algorithm is the ML-based EM-algorithm in which we replace the linear structure used to estimate the fixed part of the model by a single tree (MERT) or a forest (MERF).

**Step 0.** Set \( r = 0 \). Let \( \hat{b}_{i(0)} = 0 \), \( \hat{\sigma}^2_{(0)} = 1 \), and \( \hat{D}(0) = I_q \).

**Step 1.** Set \( r = r + 1 \). Update \( y^*_i(r) \), \( \hat{f}(X_i)(r) \), and \( \hat{b}_i(r) \)

i) \( y^*_i(r) = y_i - Z_i \hat{b}_{i(r-1)} \), \( i = 1, \ldots, n \),

ii) Let \( \hat{f}(X_i)(r) \) be estimated from a tree (or forest) algorithm with \( y^*_i(r) \) as responses and \( X_i \) as covariates,

iii) \( \hat{b}_i(r) = \hat{D}_{i(r-1)} Z_i^T \hat{V}^{-1}_{i(r-1)} \left( y_i - \hat{f}(X_i)(r) \right) \), \( i = 1, 2, \ldots, n \),

where \( \hat{V}_{i(r-1)} = Z_i \hat{D}_{i(r-1)} Z_i^T + \hat{\sigma}^2_{r-1} I_n \), \( i = 1, 2, \ldots, n \).

**Step 2.** Update \( \hat{\sigma}^2(r) \), and \( \hat{D}(r) \) using

\[
\hat{\sigma}^2(r) = N^{-1} \sum_{i=1}^n \left\{ \hat{\epsilon}^T_i(r) \hat{\epsilon}_i(r) + \hat{\sigma}^2_{(r-1)} [n_i - \hat{\sigma}^2_{(r-1)} tr(\hat{V}_{i(r-1)})] \right\}
\]

\[
\hat{D}(r) = n^{-1} \sum_{i=1}^n \left\{ \hat{b}_i(r)^T \hat{b}_i(r) + [\hat{D}_{i(r-1)} - \hat{D}_{i(r-1)} Z_i^T \hat{V}^{-1}_{i(r-1)} Z_i \hat{D}_{i(r-1)}] \right\},
\]

where \( \hat{\epsilon}_i(r) = y_i - \hat{f}(X_i)(r) - Z_i \hat{b}_i(r) \).
To predict the response for a new observation $j$ that belongs to a cluster $i$ among those used to fit the MERT (MERF) model, we use both its corresponding population-averaged tree (forest) prediction, $\hat{f}(x_{ij})$, and the predicted random part corresponding to its cluster, $Z_i \hat{b}_i$. For a new observation that belongs to a cluster not included in the sample used to train the model, we can only take the corresponding population–averaged tree (forest) prediction, $\hat{f}(x_{ij})$. Hence,

1. For a known cluster: Prediction $= \hat{f}(x_{ij}) + Z_i \hat{b}_i$.
2. For a new cluster: Prediction $= \hat{f}(x_{ij})$. 
With parametric models (e.g. LMM), one goal behind using random effects is to model the covariance structure in order to have a valid inference about the parameters. But here, we don’t want to test hypotheses or build confidence intervals. One might wonder why bother with all that. Why not simply add a fixed effect categorical covariate to represent the clusters and build an ordinary tree (or forest) with this additional covariate?

The reason is that typically the number of clusters is very large (may have hundreds of clusters or more). rpart and randomForest are limited to 32 levels for a categorical covariate. Moreover, we may have to predict observations from a new cluster. This would be a problem if the cluster was modeled as a fixed effect.
There is at least another way to build a forest. We could build directly many MERT trees with bootstrap samples and aggregate them. However this has two drawbacks:

1. Since the original observations are possibly correlated, taking a standard bootstrap sample may not be the best choice. Bootstrapping directly clustered data can be done in different ways (Field and Welsh, 2007), but this adds a difficulty level. The proposed method avoids this problem because the forest is build with “de-correlated” data (i.e. after removing the random effects) that we treat as if they were independent.

2. The computation time is a lot larger because we need to run the EM-algorithm for each tree. The proposed method (MERF) runs the EM-algorithm only once with the forest inside it. Very fast code exists for forests (randomForest).
Nevertheless, we still tried this other approach for forest building with limited simulations. We experimented with three resampling strategies:

1. Resampling individual observations
2. Resampling entire clusters
3. Resampling of clusters and then of observations within them (two-stage-bootstrap)

In our limited experience, the proposed approach is better than building such a forest of MERT trees.
Simulation Study: MERT vs Standard Tree

Design:

1. 14 data generating processes (trees with 4 terminal nodes) with and without random effects.

2. Training sample: 500 observations nested within 100 clusters. Balanced (5 obs. per cluster) or unbalanced (1, 3, 5, 7 or 9 obs. per cluster).

3. Test set: 5000 observations nested within the same clusters with the same proportions.

4. Competitors: Standard tree vs MERT.

5. Criteria: Predictive mean square error (PMSE) on the test set and finding the right tree structure.
Figure: Mixed effects regression tree structure used for the simulation study.

\[ y_{ij} = \mu_1 + z_{ij}^T b_i + \epsilon_{ij} \]

\[ y_{ij} = \mu_2 + z_{ij}^T b_i + \epsilon_{ij} \]

\[ y_{ij} = \mu_3 + z_{ij}^T b_i + \epsilon_{ij} \]

\[ y_{ij} = \mu_4 + z_{ij}^T b_i + \epsilon_{ij} \]
Results:

1. MERT is as good as a standard tree when there are no random effects (i.e. independent data).
2. MERT is always better (PMSE and for recovering the true tree structure) than a standard tree when random effects are present.
Table: Data generating processes (DGP) for the simulation study.

<table>
<thead>
<tr>
<th>DGP</th>
<th>Fixed Component</th>
<th>Random Component</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Effect</td>
<td>$\mu_1$</td>
</tr>
<tr>
<td>1 2</td>
<td>Large</td>
<td>-20</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>10</td>
</tr>
<tr>
<td>3 4</td>
<td>Large</td>
<td>-20</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>10</td>
</tr>
<tr>
<td>5 6</td>
<td>Large</td>
<td>-20</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>10</td>
</tr>
<tr>
<td>7 8</td>
<td>Large</td>
<td>-20</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>10</td>
</tr>
<tr>
<td>9 10</td>
<td>Large</td>
<td>-20</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>10</td>
</tr>
</tbody>
</table>
**Simulation Study: Part 1**

**Data Example 1**

**Generalized Mixed Effects Regression Tree (GMERT)**

**Simulation Study: Part 2**

**Data Example 2**

### Table: Results of the 100 simulation runs in the unbalanced scenarios.

<table>
<thead>
<tr>
<th>DGP</th>
<th>Fixed effect</th>
<th>Random effect</th>
<th>Fitted tree model*</th>
<th>% of trees with the right tree structure</th>
<th>PMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Large</td>
<td>No random effect</td>
<td>STD</td>
<td>100</td>
<td>1.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>100</td>
<td>1.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>100</td>
<td>1.97</td>
</tr>
<tr>
<td>2</td>
<td>Small</td>
<td></td>
<td>STD</td>
<td>94</td>
<td>0.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>95</td>
<td>0.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>95</td>
<td>0.94</td>
</tr>
<tr>
<td>3</td>
<td>Large</td>
<td></td>
<td>STD</td>
<td>100</td>
<td>2.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>100</td>
<td>2.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>100</td>
<td>2.12</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Random intercept</td>
<td>STD</td>
<td>100</td>
<td>2.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>100</td>
<td>2.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>100</td>
<td>2.04</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Large random intercept</td>
<td>STD</td>
<td>81</td>
<td>1.18</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>91</td>
<td>1.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>93</td>
<td>1.04</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Small</td>
<td>STD</td>
<td>61</td>
<td>1.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>85</td>
<td>1.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>84</td>
<td>1.07</td>
</tr>
</tbody>
</table>

* STD: Standard tree model; RI: Random intercept tree model; RIC: Random intercept and covariate tree model
### Table: Results of the 100 simulation runs in the unbalanced scenarios.

<table>
<thead>
<tr>
<th>DGP</th>
<th>Fixed effect</th>
<th>Random effect</th>
<th>Fitted tree model*</th>
<th>% of trees with the right tree structure</th>
<th>PMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td></td>
<td>Large</td>
<td>STD</td>
<td>100</td>
<td>10.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>100</td>
<td>4.34</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>100</td>
<td>2.26</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Random intercept and covariate with 0.5 correlation</td>
<td>STD</td>
<td>100</td>
<td>20.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>100</td>
<td>6.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>100</td>
<td>2.31</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>Small</td>
<td>STD</td>
<td>0</td>
<td>10.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>3</td>
<td>3.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>75</td>
<td>1.27</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td>STD</td>
<td>0</td>
<td>20.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>0</td>
<td>6.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>71</td>
<td>1.36</td>
</tr>
</tbody>
</table>

* STD: Standard tree model; RI: Random intercept tree model; RIC: Random intercept and covariate tree model
Simulation Study: MERF vs RF

Design:
9 random variables are first generated from a multivariate normal distribution \( (X_1, \ldots, X_9) \sim N(0, \Sigma) \) with \( \Sigma \) chosen such that all variables have unit variance and are equicorrelated. Then, the continuous response variable \( y \) is generated according to the following non linear model, using only the first three random variables:

\[
y_{ij} = m \times g(x_{ij}) + b_i + \varepsilon_{ij},
\]

\[
g(x_{ij}) = 2x_{1ij} + x_{2ij}^2 + 4(x_{3ij} > 0) + 2 \log |x_{1ij}|x_{3ij},
\]

\[
b_i \sim N(0, \sigma_b^2), \varepsilon_{ij} \sim N(0, \sigma_{\varepsilon}^2),
\]

\[
i = 1, \ldots, 100, j = 1, \ldots, m_i,
\]

where \( m \times g(x_{ij}) \) represents the response fixed part with a variance \( \sigma_{\text{Fixed}}^2 = m^2 \sigma_g^2 \). The parameter \( m \) serves as a tuning parameter to control the magnitude of \( \sigma_{\text{Fixed}}^2 \).
1. Training sample: 500 observations nested within 100 unbalanced clusters having 1, 3, 5, 7, or 9 observations.
2. Test sample 1 (*known* clusters): 4500 observations nested within the same clusters as the training sample and in the same proportions.
3. Test sample 2 (*new* clusters): 4500 observations nested within 100 new clusters with the same characteristics. But new random effects $b_i$ are generated.
Table: Data generating processes (DGP) for the simulation study.

<table>
<thead>
<tr>
<th>DGP</th>
<th>ρ</th>
<th>PTEV*</th>
<th>PREV**</th>
<th>$\sigma^2_{\text{Fixed}}$</th>
<th>$\sigma^2_b$</th>
<th>m</th>
<th>ICC***</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>30</td>
<td>50</td>
<td>8.1</td>
<td>0.8</td>
<td>0.9</td>
<td>47.4</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>50</td>
<td>50</td>
<td>6.3</td>
<td>0.7</td>
<td>2.7</td>
<td>73.0</td>
</tr>
<tr>
<td>3</td>
<td>90</td>
<td>50</td>
<td>50</td>
<td>4.5</td>
<td>0.6</td>
<td>4.5</td>
<td>81.8</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>1.4</td>
<td>0.3</td>
<td>0.2</td>
<td>13.0</td>
</tr>
<tr>
<td>5</td>
<td>60</td>
<td>10</td>
<td>10</td>
<td>1.1</td>
<td>0.3</td>
<td>0.5</td>
<td>31.0</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>0.8</td>
<td>0.2</td>
<td>0.8</td>
<td>42.9</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>8.1</td>
<td>0.7</td>
<td>0.9</td>
<td>47.4</td>
</tr>
<tr>
<td>8</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>6.3</td>
<td>0.6</td>
<td>2.7</td>
<td>73.0</td>
</tr>
<tr>
<td>9</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>4.5</td>
<td>0.5</td>
<td>4.5</td>
<td>81.8</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>1.4</td>
<td>0.3</td>
<td>0.2</td>
<td>13.0</td>
</tr>
<tr>
<td>11</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>1.1</td>
<td>0.3</td>
<td>0.5</td>
<td>31.0</td>
</tr>
<tr>
<td>12</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>0.8</td>
<td>0.2</td>
<td>0.8</td>
<td>42.9</td>
</tr>
</tbody>
</table>

* Proportion of Total Effects Variability = $\frac{\sigma^2_{\text{Fixed}} + \sigma^2_b}{\sigma^2_{\text{Fixed}} + \sigma^2_b + \sigma^2_\varepsilon} \times 100$

** Proportion of Random Effects Variability = $\frac{\sigma^2_b}{\sigma^2_{\text{Fixed}} + \sigma^2_b} \times 100$

*** Intra Cluster Correlation = $\frac{\sigma^2_b}{\sigma^2_b + \sigma^2_\varepsilon} \times 100$
Results:

Relative difference (RD) in PMSE between MERF and RF.

\[ RD = \frac{PMSE_{RF} - PMSE_{MERF}}{PMSE_{RF}} \times 100. \]
Relative difference in PMSE between MERF and RF for known clusters

Data Generating Process

Relative Difference in PMSE

Independent predictors, i.e. $\rho = 0$

Dependent predictors, with $\rho = .4$

PTEV = .9

PTEV = .6

PREV = .1

PREV = .3

PREV = .5

Relative difference in PMSE between MERF and RF for known clusters
Data Generating Process
Relative difference in PMSE between MERF and RF for new clusters

PREV = .1 .3 .5
PTEV = .9
Independent predictors, i.e. $\rho = 0$

PREV = .1 .3 .5
PTEV = .6
Dependent predictors, with $\rho = .4$

Relative difference in PMSE between MERF and RF for new clusters
Average relative difference in PMSE between MERF and all other methods for known clusters
Average relative difference in PMSE between MERF and all other methods for new clusters.
Box-office data example

1. Data set consists of first-week box office revenues of 60,175 screens nested within 2,656 new movies presented in the province of Qu´ebec in Canada from 2001 to 2008.

2. On average, there are 22.7 screens per movie \((minimum = 1; \ first \ quartile = 1; \ median = 8; \ third \ quartile = 47; \ maximum = 93)\).

3. Each movie is treated as a cluster.
**Outcome**: log transform of the first-week box office revenues. There are three screen–level covariates:

1. **Language** (1-French Version; 2-Original English Version; 3-Original French Version; 4-Original Version with Subtitles).
2. **Region** (1-Montréal; 2-Montérégie; 3-Québec City; 4-Laurentides; 5-Lanaudière; 6- Others).
3. **Theater owner** (1-Independent; 2-Cinéplex; 3-Guzzo; 4-Ciné-entreprise; 5-Famous Players; 6-Cinémas R.G.F.M.; 7-Cinémas Fortune; 8-AMC).
There are eight movie-level covariates:

1. **Movie critics’ rating**, an ordinal covariate taking on values from 1 (the best) to 7 (the worst).
2. **Movie length**, a continuous covariate ranging between 70 to 227 minutes.
3. **Movie genre** (1-Comedy; 2-Drama; 3-Thriller; 4-Action/Adventure; 5-Science fiction; 6-Cartoons; 7-Others).
4. **Visa**. (1- General; 2-Thirteen years old; 3-Sixteen years old; 4-Eighteen years old).
5. **Month** of movie release.
Movie *distributor* (1-Vivafilm; 2-Sony; 3-Warner; 4-Fox; 5-Universal; 6-Paramount; 7-Disney; 8-Chrystal Films; 9-Films Séville; 10-DreamWorks; 11-MGM; 12-TVA Films; 13-Equinoxe; 14-Others).

Country of origin (1-USA; 2-Québec; 3-France; 4-Rest of Canada; 5-Other countries).

Size, total number of screens for a movie in its first-week (this is a common measure that approximates the marketing effort).
1. Training sample: 30,018 screens within 2,656 movies
2. Test sample: 30,157 screens within 1,920 movies.
3. Models:
   1. Standard random forest (RF).
   2. Random intercept random forest (MERF).
   4. Random intercept tree (MERT).
   5. Linear model (LM).
**Table:** Results (PMSE) for the first–week box office revenues example.

<table>
<thead>
<tr>
<th></th>
<th>PMSE</th>
<th>Estimated ICC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Random Forest</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MERF</td>
<td>0.47</td>
<td>0.54</td>
</tr>
<tr>
<td>RF</td>
<td>0.60</td>
<td>–</td>
</tr>
<tr>
<td><strong>Single Tree</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MERT</td>
<td>0.53</td>
<td>0.51</td>
</tr>
<tr>
<td>RT</td>
<td>0.90</td>
<td>–</td>
</tr>
<tr>
<td><strong>Linear Model</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMM</td>
<td>0.62</td>
<td>0.42</td>
</tr>
<tr>
<td>LM</td>
<td>1.00</td>
<td>–</td>
</tr>
</tbody>
</table>

Note: MERT has 28 leaves while CART has 44 leaves.
The convergence of the algorithm is monitored by computing at each iteration the following generalized log-likelihood (GLL) criterion:

\[
GLL(f, b_i | y) = \sum_{i=1}^{n} \left\{ [y_i - f(X_i) - Z_i b_i]^T R_i^{-1} [y_i - f(X_i) - Z_i b_i] + b_i^T D^{-1} b_i + \log |D| + \log |R_i| \right\}
\]

(2)

For the movie data, here is the graph of GLL by iterations for MERF.
Figure 7: Behavior of the generalized log-likelihood (GLL) through the iteration process for fitting the random intercept random forest (MERF) model for the example on first-week box office revenues.
Generalized linear mixed models (GLMMs) are extensions of the LMM for outcomes that are not necessarily gaussian. Recall that for a LMM, \( E[Y_{ij}|b_i] = X'_{ij}\beta + Z'_{ij}b_i \) and \( V[Y_{ij}|b_i] = \sigma^2 \). For a GLMM (Fitzmaurice, Laird and Ware, 2011):

1. \( Y_{ij}|b_i \) belongs to the exponential family of distribution.
2. \( g(E[Y_{ij}|b_i]) = \eta_{ij} = X'_{ij}\beta + Z'_{ij}b_i \), for some known link function \( g \).
3. \( V[Y_{ij}|b_i] = \nu(E[Y_{ij}|b_i])\phi \), where \( \nu \) is a known variance function.
4. Given the \( b_i \)'s, the \( Y_{ij} \)'s are independent.
5. \( b_i \sim N(0, D) \) (and independent of the \( X_{ij} \)'s).

The choices 1) normal, 2) \( g(u) = u \), 3) \( \nu(u) = 1 \), \( \phi = \sigma^2 \) give the LMM.
Estimating the parameters in a GLMM is not a straightforward task. Direct likelihood require numerical integration methods. Approximate methods are also available. The **Penalized Quasi–Likelihood** (PQL) is one of them. The idea is to linearize the problem and use the existing estimation methods for the LMM. A first–order Taylor series expansion of the conditional mean function around current estimates $\hat{\beta}$ and $\hat{b}_i$ gives the approximate model:

$$ Y_{ij}^* = v^{-1}(\hat{\mu}_{ij}^b)(Y_{ij} - \mu_{ij}^b) = X_{ij}'\hat{\beta} + Z_{ij}'\hat{b}_i + v^{-1}(\hat{\mu}_{ij}^b)\epsilon_{ij}, $$

where $\mu_{ij}^b = g^{-1}(X_{ij}'\beta + Z_{ij}'b_i)$, is the conditional mean of $Y_{ij}$ given $b_i$. 
The estimation proceeds by iterating the following two steps:

1. Fit a LMM to the linearized outcomes $Y_{ij}^*$ to get updated estimates of $\beta$, $D$ and predictions of the $b_i$’s. The model is fitted using weights that are inversely proportional to the variance of $\nu^{-1}(\hat{\mu}_{ij})\epsilon_{ij}$.

2. Use the updated estimates to update the linearized outcomes $Y_{ij}^*$.

Iterate until convergence.
**Generalized Mixed Effects Regression Tree (GMERT)**

The model behind the proposed generalized mixed effects regression tree method replaces the linear fixed effect by a more flexible effect that will be estimated with a tree:

$$ g(E[Y_{ij}|b_i]) = \eta_{ij} = f(X_{ij}) + Z_i' b_i. $$

The basic idea is the replace the fitting of the LMM in the algorithm above by a MERT. Namely,
1. Fit a MERT to the linearized outcomes $Y_{ij}^*$ to get updated estimates of $f(X_{ij})$, $D$ and predictions of the $b_i$'s. The tree is fitted using weights that are inversely proportional to the variance of $\nu^{-1}(\hat{\mu}_{ij}^b)\epsilon_{ij}$.

2. Use the updated estimates to update the linearized outcomes $Y_{ij}^*$.

Iterate until convergence of the $\hat{\eta}_{ij}$.

The MERT itself is fitted as before with the EM algorithm.
So far, we have implemented the binary (logit link) and Poisson (log link) outcome cases. For example, in the binary case, the predicted probability that $Y_{ij} = 1$ is:

$$
\frac{1}{1 + \exp(-\hat{f}(X_{ij}) - Z_{ij}'\hat{b}_i)}
$$

where $\hat{f}(x_{ij})$ is the predicted fixed component that results from the tree and $Z_{ij}'\hat{b}_i$ is its predicted random part corresponding to its cluster.

If the observation comes from a new cluster, than we just fix $\hat{b}_i = 0$. 
Simulation Study: GMERT vs Standard Tree

**Design:** For binary responses and Poisson responses.

1. 10 data generating processes (tree with 6 terminal nodes) with and without random effects.
2. Training sample: 500 observations nested within 100 clusters (5 obs. per cluster).
3. Test set: 5000 observations nested within the same clusters with the same proportions.
4. Competitors: Standard tree vs GMERT vs GLMM.
5. Criteria (binary response): 1) Predictive mean absolute deviation (PMAD) in terms of the estimated probability and 2) Predictive misclassification rate (PMCR):

\[
PMAD = (5000)^{-1} \sum_{i=1}^{100} \sum_{j=1}^{50} |\mu_{ij} - \hat{\mu}_{ij}|,
\]

\[
PMCR = (5000)^{-1} \sum_{i=1}^{100} \sum_{j=1}^{50} |y_{ij} - \hat{y}_{ij}|
\]
**Table:** Data generating processes (DGP) according a tree structure for the simulation study with binary responses.

<table>
<thead>
<tr>
<th>DGP</th>
<th>Effect</th>
<th>Data Structure</th>
<th>Random Component</th>
<th>Effect</th>
<th>$d_{11}$</th>
<th>$d_{22}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2</td>
<td>Large</td>
<td>$0.10$ $0.20$ $0.80$ $0.20$ $0.80$ $0.90$</td>
<td>No random effect</td>
<td>-</td>
<td>$0.00$</td>
<td>$0.00$</td>
</tr>
<tr>
<td>3, 4, 5, 6</td>
<td>Large</td>
<td>$0.10$ $0.20$ $0.80$ $0.20$ $0.80$ $0.90$</td>
<td>Random intercept</td>
<td>Small</td>
<td>$4.00$</td>
<td>$0.00$</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>$0.20$ $0.40$ $0.70$ $0.30$ $0.60$ $0.80$</td>
<td></td>
<td>Large</td>
<td>$10.00$</td>
<td>$0.00$</td>
</tr>
<tr>
<td>7, 8, 9, 10</td>
<td>Large</td>
<td>$0.10$ $0.20$ $0.80$ $0.20$ $0.80$ $0.90$</td>
<td>Random intercept and covariate</td>
<td>Small</td>
<td>$2.00$</td>
<td>$0.05$</td>
</tr>
<tr>
<td></td>
<td>Small</td>
<td>$0.20$ $0.40$ $0.70$ $0.30$ $0.60$ $0.80$</td>
<td></td>
<td>Large</td>
<td>$5.00$</td>
<td>$0.25$</td>
</tr>
</tbody>
</table>
Figure: Generalized mixed effects tree structure used for the simulation study, with \(g(.)\) being the logit link function and \(g(.)^{-1}\) the inverse-logit or logistic function in binary responses scenarios, and respectively, the log link function and the inverse-log or power function in Poisson responses scenarios.
### Table: Results of the 100 simulation runs in terms of the predictive probability mean absolute deviation (PMAD) and the predictive misclassification rate (PMCR), for binary responses generated according to a tree structure.

<table>
<thead>
<tr>
<th>DGP</th>
<th>Fixed effect</th>
<th>Random effect</th>
<th>Fitted model*</th>
<th>PMAD (%)</th>
<th>PMCR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Large</td>
<td>No random effect</td>
<td>STD</td>
<td>3.09</td>
<td>15.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>3.86</td>
<td>16.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>4.17</td>
<td>16.85</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLMM</td>
<td>21.44</td>
<td>29.76</td>
</tr>
<tr>
<td>2</td>
<td>Small</td>
<td></td>
<td>STD</td>
<td>4.97</td>
<td>29.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>6.35</td>
<td>31.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>6.32</td>
<td>31.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLMM</td>
<td>15.12</td>
<td>36.87</td>
</tr>
</tbody>
</table>

* STD: Standard tree; RI: Random intercept tree; RIC: Random intercept and covariate tree; GLMM: Naive mixed effect logistic
### GMERT vs Standard Tree

Table: Results of the 100 simulation runs in terms of the predictive probability mean absolute deviation (PMAD) and the predictive misclassification rate (PMCR), for binary responses generated according to a tree structure.

<table>
<thead>
<tr>
<th>DGP</th>
<th>Fixed effect</th>
<th>Random effect</th>
<th>Fitted model*</th>
<th>PMAD (%)</th>
<th>PMCR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>STD</td>
<td>21.70</td>
<td>26.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>9.20</td>
<td>19.82</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>9.69</td>
<td>20.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLMM</td>
<td>18.73</td>
<td>26.53</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Large</td>
<td>STD</td>
<td>30.24</td>
<td>33.65</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>8.59</td>
<td>16.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>9.37</td>
<td>16.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLMM</td>
<td>15.14</td>
<td>20.73</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Random intercept</td>
<td>STD</td>
<td>12.56</td>
<td>31.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>10.71</td>
<td>31.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>10.79</td>
<td>31.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLMM</td>
<td>16.17</td>
<td>36.14</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>Small</td>
<td>STD</td>
<td>26.77</td>
<td>39.32</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>11.20</td>
<td>24.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>11.40</td>
<td>24.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLMM</td>
<td>13.83</td>
<td>25.54</td>
</tr>
</tbody>
</table>

* STD: Standard tree; RI: Random intercept tree; RIC: Random intercept and covariate tree; GLMM: Naive mixed effect logistic
Introduction
Mixed Effects Regression Tree (MERT) and Forest (MERF)
Simulation Study: Part 1
Data Example 1
Generalized Mixed Effects Regression Tree (GMERT)
Simulation Study: Part 2
Data Example 2

**GMERT vs Standard Tree**

Table: Results of the 100 simulation runs in terms of the predictive probability mean absolute deviation (PMAD) and the predictive misclassification rate (PMCR), for binary responses generated according to a tree structure.

<table>
<thead>
<tr>
<th>DGP</th>
<th>Fixed effect</th>
<th>Random effect</th>
<th>Fitted model*</th>
<th>PMAD (%)</th>
<th>PMCR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Large</td>
<td></td>
<td>STD</td>
<td>20.37</td>
<td>25.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>10.86</td>
<td>20.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>10.58</td>
<td>20.83</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLMM</td>
<td>20.42</td>
<td>29.00</td>
</tr>
<tr>
<td></td>
<td>Random intercept and covariate</td>
<td>STD</td>
<td>30.90</td>
<td>34.34</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>12.37</td>
<td>18.15</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>RIC</td>
<td>10.67</td>
<td>17.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLMM</td>
<td>15.61</td>
<td>20.68</td>
</tr>
<tr>
<td>9</td>
<td>Small</td>
<td></td>
<td>STD</td>
<td>12.86</td>
<td>31.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>11.12</td>
<td>31.36</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>11.04</td>
<td>31.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLMM</td>
<td>16.51</td>
<td>36.14</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>STD</td>
<td>25.42</td>
<td>39.02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RI</td>
<td>13.11</td>
<td>25.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RIC</td>
<td>12.54</td>
<td>25.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>GLMM</td>
<td>15.27</td>
<td>27.53</td>
</tr>
</tbody>
</table>

* STD: Standard tree; RI: Random intercept tree; RIC: Random intercept and covariate tree; GLMM: Naive mixed effect logistic
Similar results for the case of a Poisson outcome with a tree DGP.

Simulations with other DGPs (including GLMMs DGPs) are currently running for both the binary and Poisson cases.
Dative data: Bresnan et al. (2005)

- Data available in the R package `languageR`.
- Dative observations (i.e. instances of dative constructions) from the three-million-word Switchboard collection of recorded telephone conversations.
- 2360 observations nested within the 38 verbs (clusters in this example).
- The objective is to predict the dative alternation represented by the binary outcome variable `RealizationOfRecipient` that may be a double object structure (NP) or a prepositional dative structure (PP).
For example:

Suppose we want to say that Susan gave toys to some children. After the expression “Susan gave...” has already been constructed, two constructions are possible. If toys is inserted first, a prepositional dative structure is built: “Susan gave toys to the children”. If children is inserted first, a double object structure is built: “Susan gave the children toys”.

10 covariates:

1. **SemanticClass**: Semantic class: abstract (abbreviated ‘a’) as in give it some thought; transfer of possession (‘t’) as in give an armband, send; future transfer of possession (‘f’), exemplified by owe, promise; prevention of possession (‘p’), exemplified by cost, deny); and communication (‘c’) as in tell, give me your name, said on a telephone.

2. **AccessOfRec**: Discourse accessibility of recipient.

3. **AccessOfTheme**: Discourse accessibility of theme.

4. **PronomOfRec**: Pronominality of recipient (phrases headed by pronouns (personal, demonstrative, and indefinite) vs. Those headed by nonpronouns such as nouns and gerunds).
PronomOfTheme: Pronominality of theme (phrases headed by pronouns (personal, demonstrative, and indefinite) vs. Those headed by nonpronouns such as nouns and gerunds).

DefinOfRec: Definiteness of recipient.

DefinOfTheme: Definiteness of theme.

AnimacyOfRec: Animacy of recipient.

AnimacyOfTheme: Animacy of theme.

BresnanLength: Length difference: a sign-preserving log transform of the absolute value of the difference in number of graphemic words between the theme and recipient to measure their relative weight.
Results

Data splitted into a training sample of size 1162 and a test set of size 1198. Three predictive models are used:

1. Standard classification tree (STD)
2. Parametric random intercept logistic regression model (GLMM)
3. Random intercept GMERT.

Table: Predictive misclassification rate (PMCR) for the dative data.

<table>
<thead>
<tr>
<th></th>
<th>PMCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>STD</td>
<td>10.7%</td>
</tr>
<tr>
<td>GLMM</td>
<td>8.0%</td>
</tr>
<tr>
<td>GMERT</td>
<td>6.6%</td>
</tr>
</tbody>
</table>
References


Appendix
Here we clarify how the weights intervene in the standard regression tree fitted at each micro iteration within the GMERT algorithm. At any micro iteration within a given macro iteration, the standard regression tree uses the corresponding 
\( \tilde{y}_i^* = \tilde{y}_i^{(M)} - Z_i \hat{b}_i \) as the dependent variable and \( X_i \) as the covariates, along with the weights \( W_i = \text{diag}(w_{ij}) \), with \( i = 1, \ldots, n \) and \( j = 1, \ldots, m_i \).
Let \( T \) be the fitted standard regression tree, and let \( t \) be one of its nodes. Node \( t \) contains a subset of \( N_t < N \) observations that belong to a subset of \( n_t \leq n \) clusters with pseudo-responses \( \tilde{y}_{it,j_t}^* \), \( i_t = 1, \ldots, n_t \) and \( j_t = 1, \ldots, m_{it} \). Then, given the weights \( w_{it,j_t} \) of observation \( j_t \) in cluster \( i_t \) in node \( t \), we have: 
The summary statistic to be attached to node $t$ corresponds to its weighted response average $\tilde{y}_t^* = \frac{\sum_{i_t=1}^{n_t} \sum_{j_t=1}^{m_{it}} w_{itjt} \tilde{y}_{itjt}}{\sum_{i_t=1}^{n_t} \sum_{j_t=1}^{m_{it}} w_{itjt}}$. This corresponds to the fitted value $\hat{y}_t^* = \hat{f}(X_{it})$ when $t$ is a terminal node.

The error of node $t$ equals its weighted sums of squares or corrected deviance $DEV_t$, with

$$DEV_t = \sum_{i_t=1}^{n_t} \sum_{j_t=1}^{m_{it}} w_{itjt} (\tilde{y}_{itjt}^* - \bar{y}_{it})^2.$$ 

The splitting criterion is the improvement or the percent change in the weighted sums of squares for a given split of node $t$ into two nodes $t_l$ and $t_r$, i.e.,

$$\text{Improve} = 1 - \frac{(DEV_{tl} + DEV_{tr})}{DEV_t}.$$
The cross-validated relative error corresponding to a given complexity parameter value for the tree $T$ is defined as follows: 

$$xerror = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m_i} w_{ij}(\hat{y}_{ij}^* - \hat{y}_{(-ij)}^*)^2}{\text{DEV}_{\text{root}}}$$

with $\hat{y}_{(-ij)}^*$ being the predicted value for observation $j$ in cluster $i$, from the standard regression tree model that is fitted without this observation.