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:

- 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.
- Hajjem, A., Bellavance, F. and Larocque, D. (2014). Generalized Mixed Effects Regression Trees. Revise and resubmit at *Computational Statistics and Data Analysis*.

Outline I

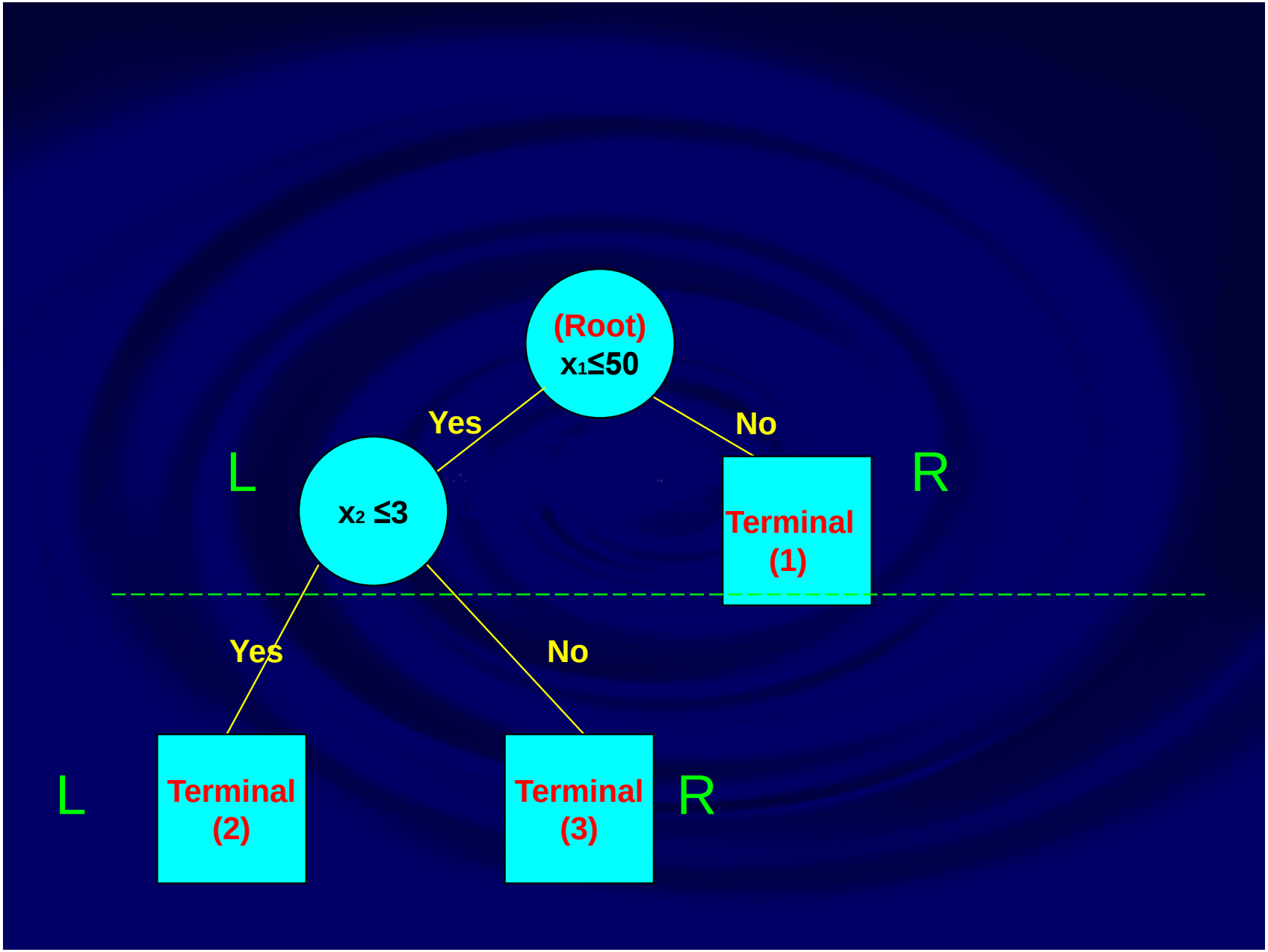
Outline II

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)
x1≤50
Yes
No
L
R
x2 ≤3
Terminal
(1)
Yes
No
L
Terminal
(2)
Terminal
(3)
R

- 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.

Recent surveys: Rokach (2008), Siroky (2009) and Verikas, Gelzinis, and Bacauskiene (2011).

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:

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

- $y_i = (y_{i1}, \ldots, y_{im_i})$ 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 $(q \times 1)$.
- β is the unknown parameter vector for the fixed effects $(p \times 1)$.
- ϵ_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

1. 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.

2. 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.
3. 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.

4. 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 Pearsons residual average.
5. 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.

6. 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.

7. 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.

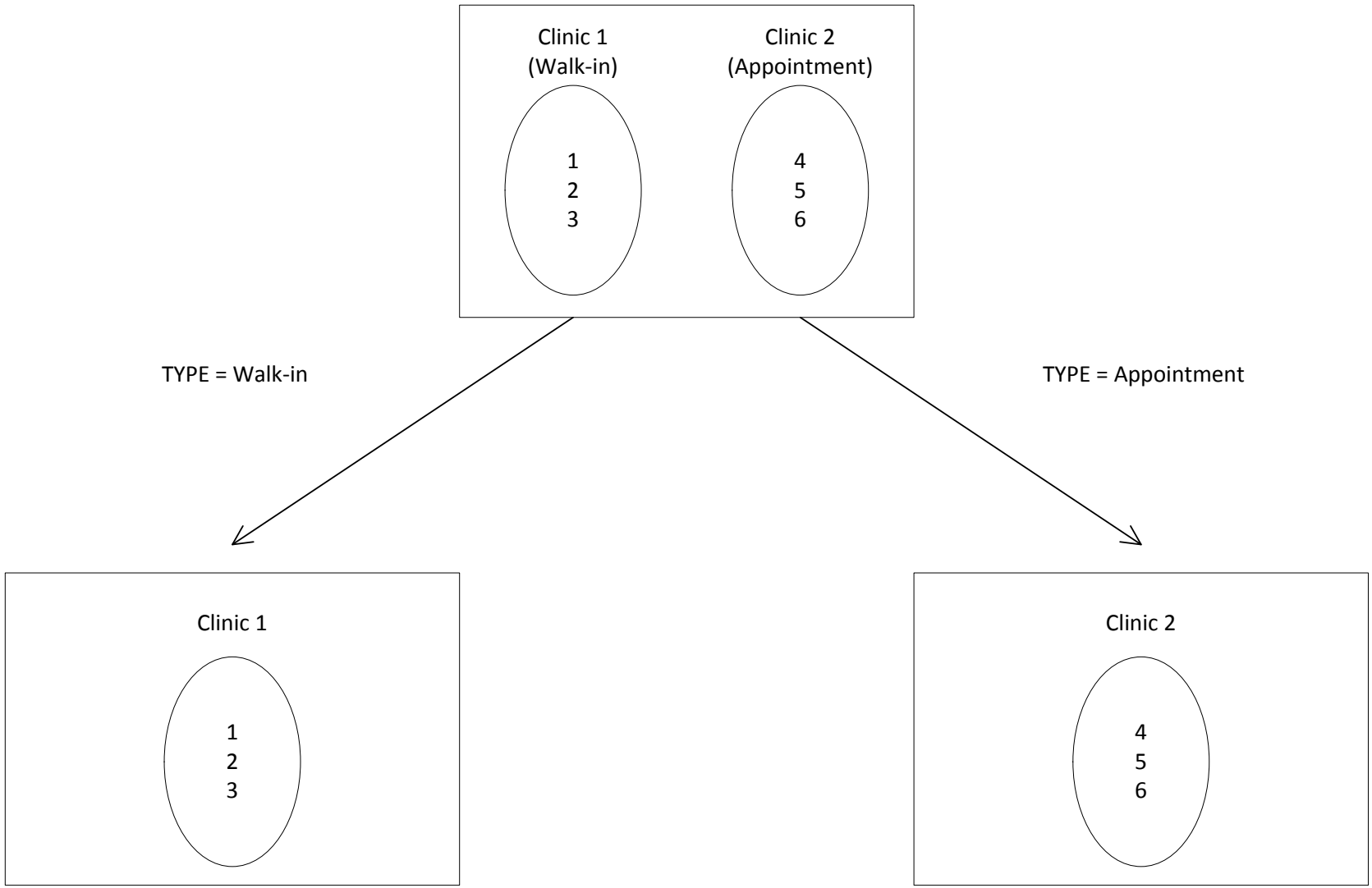
Our method has the following characteristics:

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).

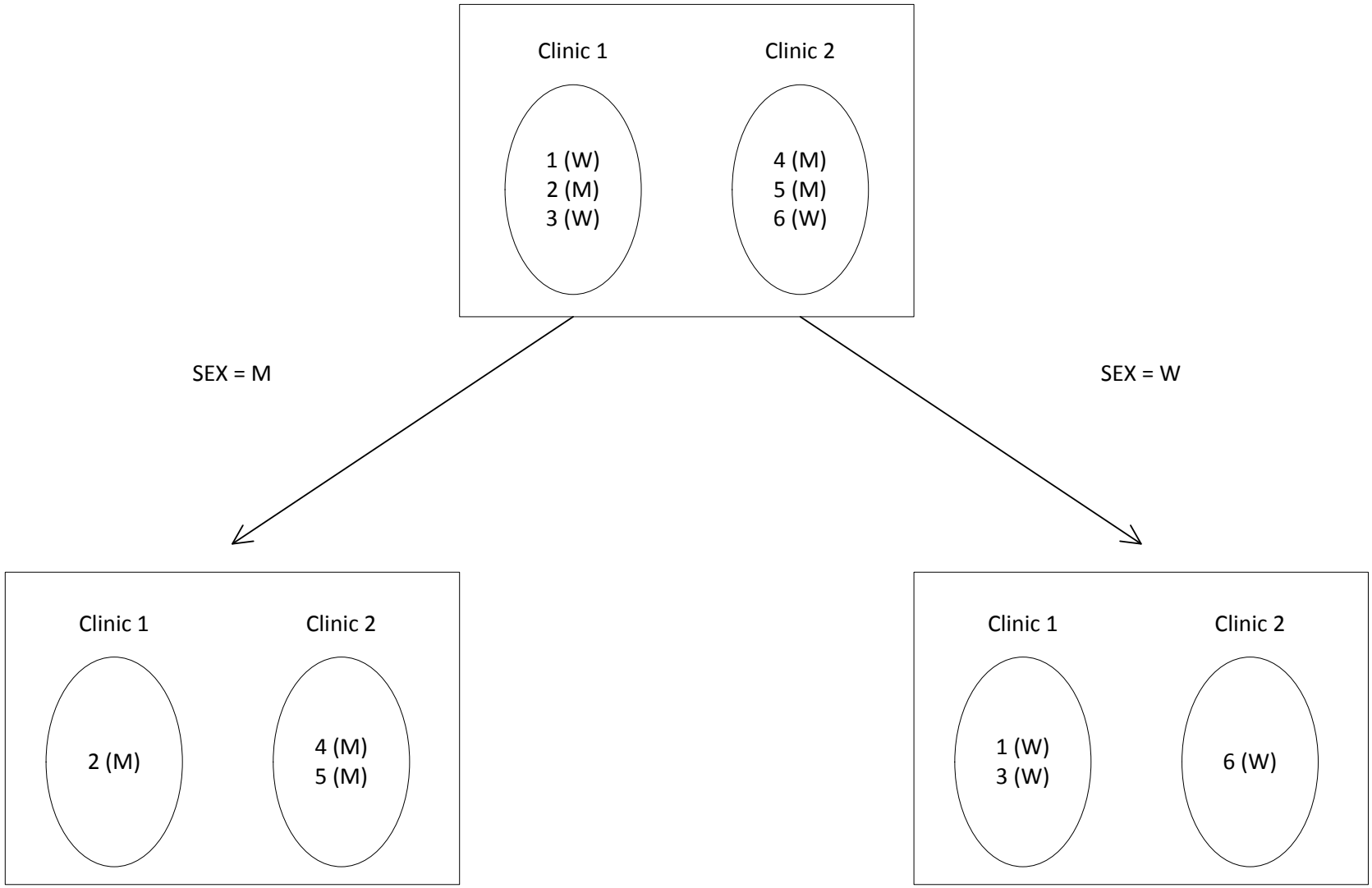
The following two examples illustrates some key points.

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

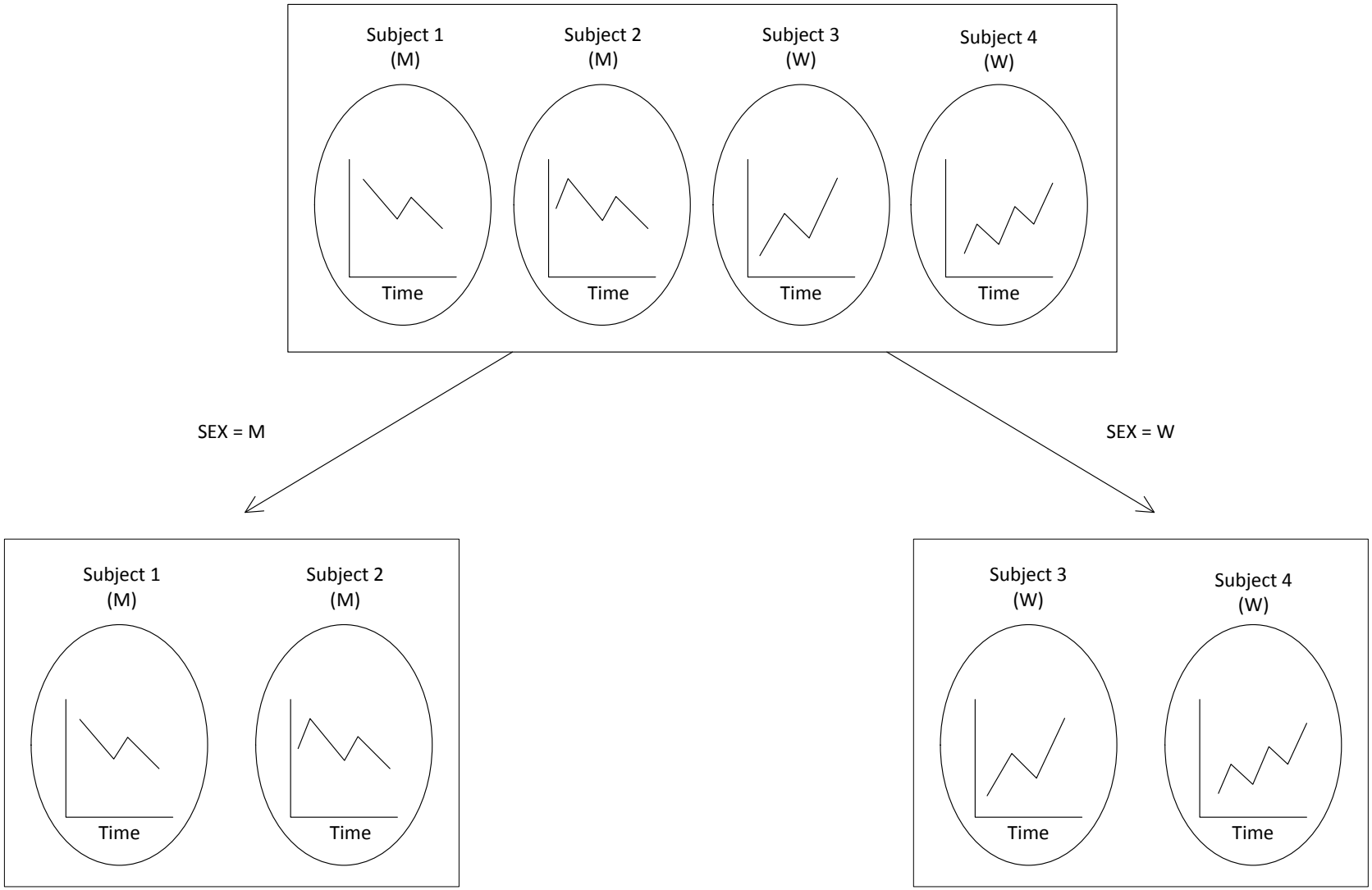
Clinic Level Covariate



PATIENT LEVEL COVARIATE



LONGITUDINAL DATA



Mixed Effects Regression Tree (MERT) and Forest (MERF)

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

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

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 $\hat{b}_{i(r)}$

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

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

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

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

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

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

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

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

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

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

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

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

where $\hat{\epsilon}_{i(r)} = y_i - \hat{f}(X_i)_{(r)} - Z_i \hat{b}_{i(r)}$

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.

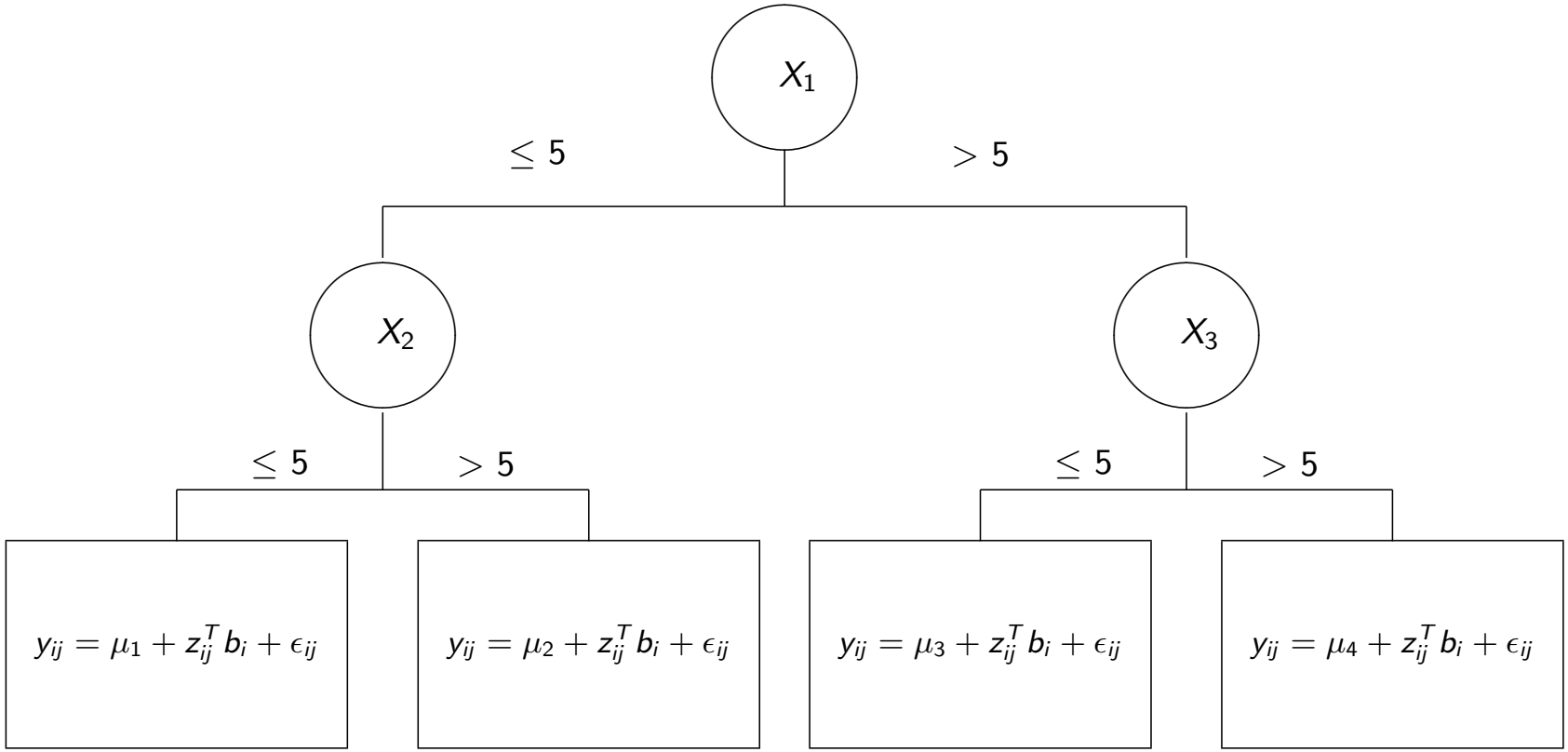


Figure: Mixed effects regression tree structure used for the simulation study.

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.

DGP	Data Structure								
	Fixed Component					Random Component			
	Effect	μ_1	μ_2	μ_3	μ_4	Structure	d_{11}	d_{22}	d_{12}
1	Large	-20	-10	10	20	No random effect	0.00	0.00	0.00
2	Small	10	11	12	13				
3	Large	-20	-10	10	20	Random intercept	0.25	0.00	0.00
4							0.50	0.00	0.00
5	Small	10	11	12	13		0.25	0.00	0.00
6							0.50	0.00	0.00
7	Large	-20	-10	10	20	Random intercept and covariate X_1 with 0 correlation	0.25	0.25	0.00
8							0.50	0.50	0.00
9	Small	10	11	12	13		0.25	0.25	0.00
10							0.50	0.50	0.00
11	Large	-20	-10	10	20	Random intercept and covariate X_1 with 0.5 correlation	0.25	0.25	0.125
12							0.50	0.50	0.25
13	Small	10	11	12	13		0.25	0.25	0.125
14							0.50	0.50	0.25

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

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

* 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.

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

* 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 $(X_1, ..., X_9) \sim N(0, \Sigma)$ with Σ 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:

$$y_{ij} = m \times g(x_{ij}) + b_i + \varepsilon_{ij}, \tag{1}$$
$$g(x_{ij}) = 2x_{1ij} + x_{2ij}^2 + 4(x_{3ij} > 0) + 2 \log |x_{1ij}| x_{3ij},$$
$$b_i \sim N(0, \sigma_b^2), \varepsilon_{ij} \sim N(0, \sigma_\varepsilon^2),$$
$$i = 1, ..., 100, j = 1, ..., m_i,$$

where $m \times g(x_{ij})$ represents the response fixed part with a variance $\sigma^2_{Fixed} = m^2 \sigma^2_g$. The parameter m serves as a tuning parameter to control the magnitude of σ^2_{Fixed}.

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.

DGP	ρ	PTEV*	PREV**	σ^2_{Fixed}	m	σ^2_b	ICC***
1	0.0	90	10	8.1	0.8	0.9	47.4
2			30	6.3	0.7	2.7	73.0
3			50	4.5	0.6	4.5	81.8
4		60	10	1.4	0.3	0.2	13.0
5			30	1.1	0.3	0.5	31.0
6			50	0.8	0.2	0.8	42.9
7	0.4	90	10	8.1	0.7	0.9	47.4
8			30	6.3	0.6	2.7	73.0
9			50	4.5	0.5	4.5	81.8
10		60	10	1.4	0.3	0.2	13.0
11			30	1.1	0.3	0.5	31.0
12			50	0.8	0.2	0.8	42.9

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

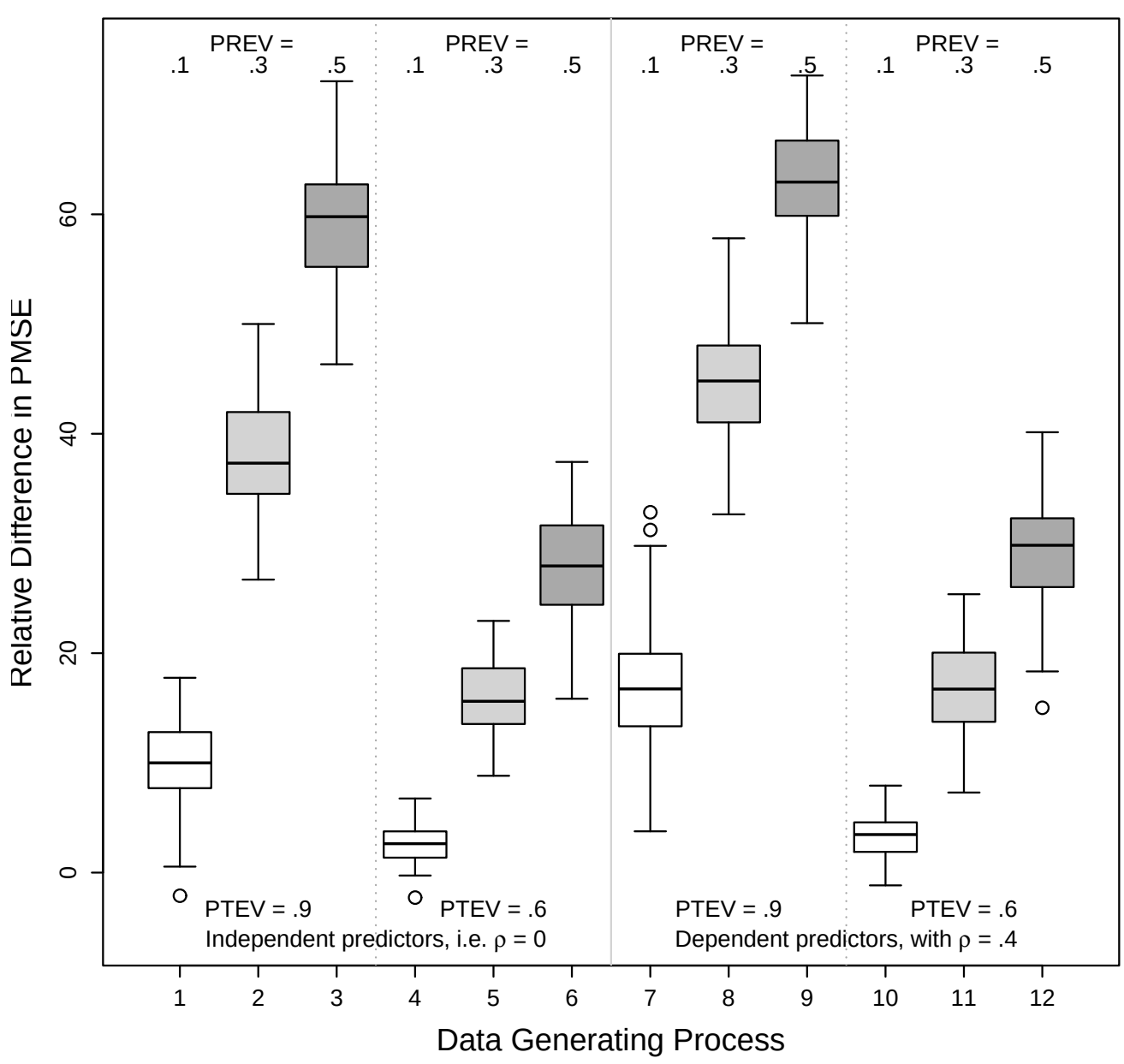
**Proportion of Random Effects Variability $= \frac{\sigma^2_b}{\sigma^2_{Fixed}+\sigma^2_b} \times 100$

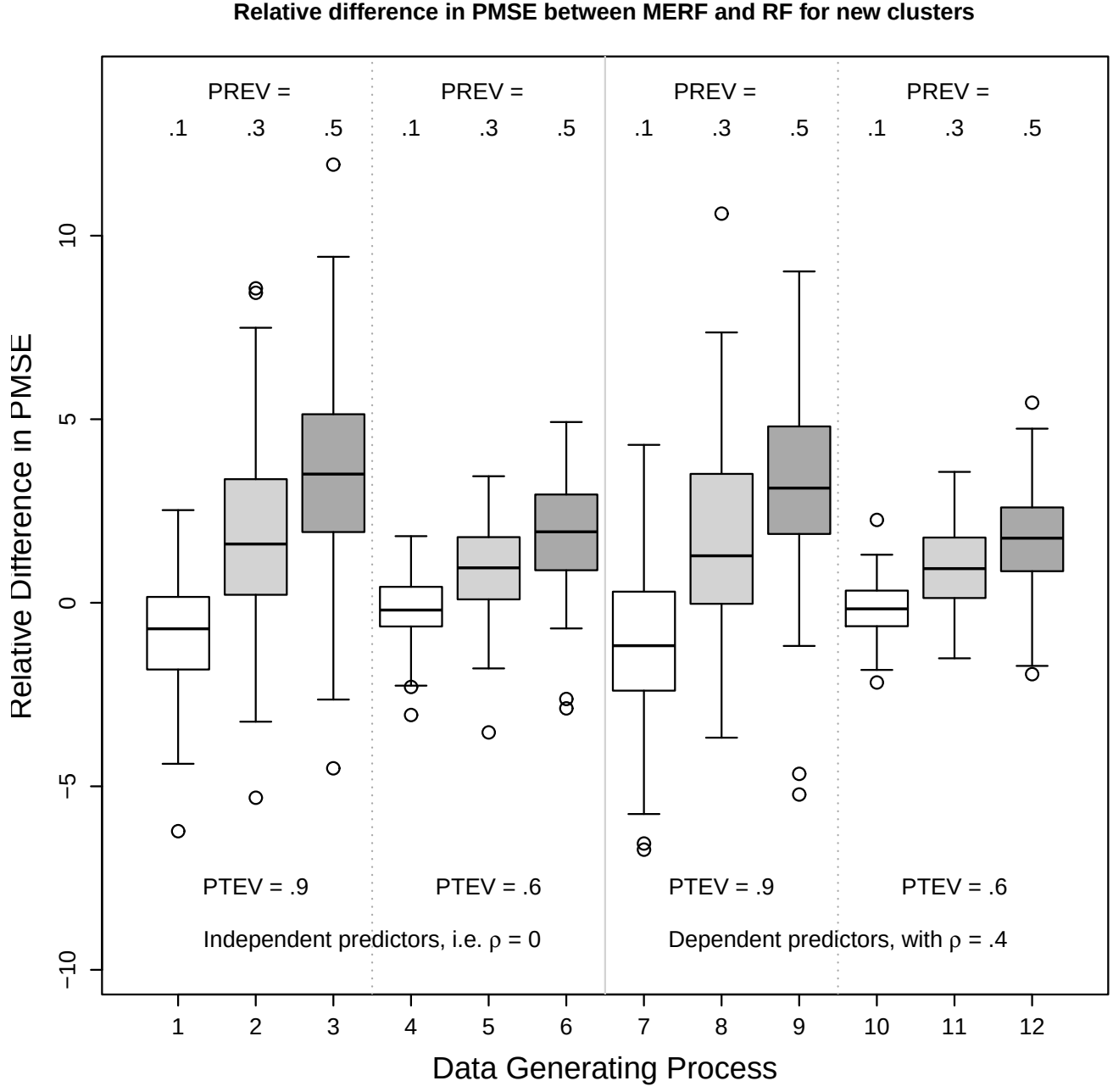
***Intra Cluster Correlation $= \frac{\sigma^2_b}{\sigma^2_b+\sigma^2_\varepsilon} \times 100$

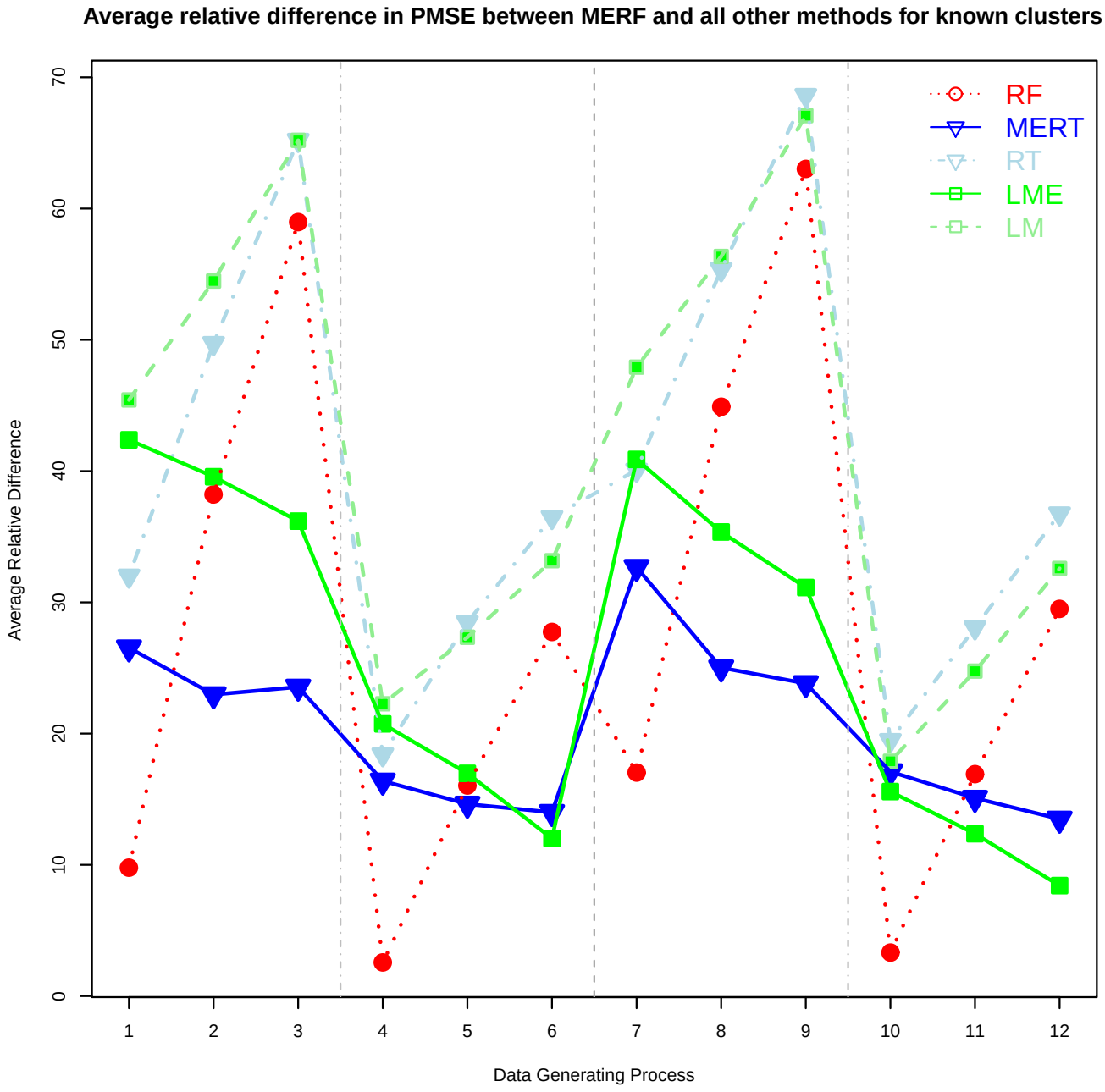
Results:

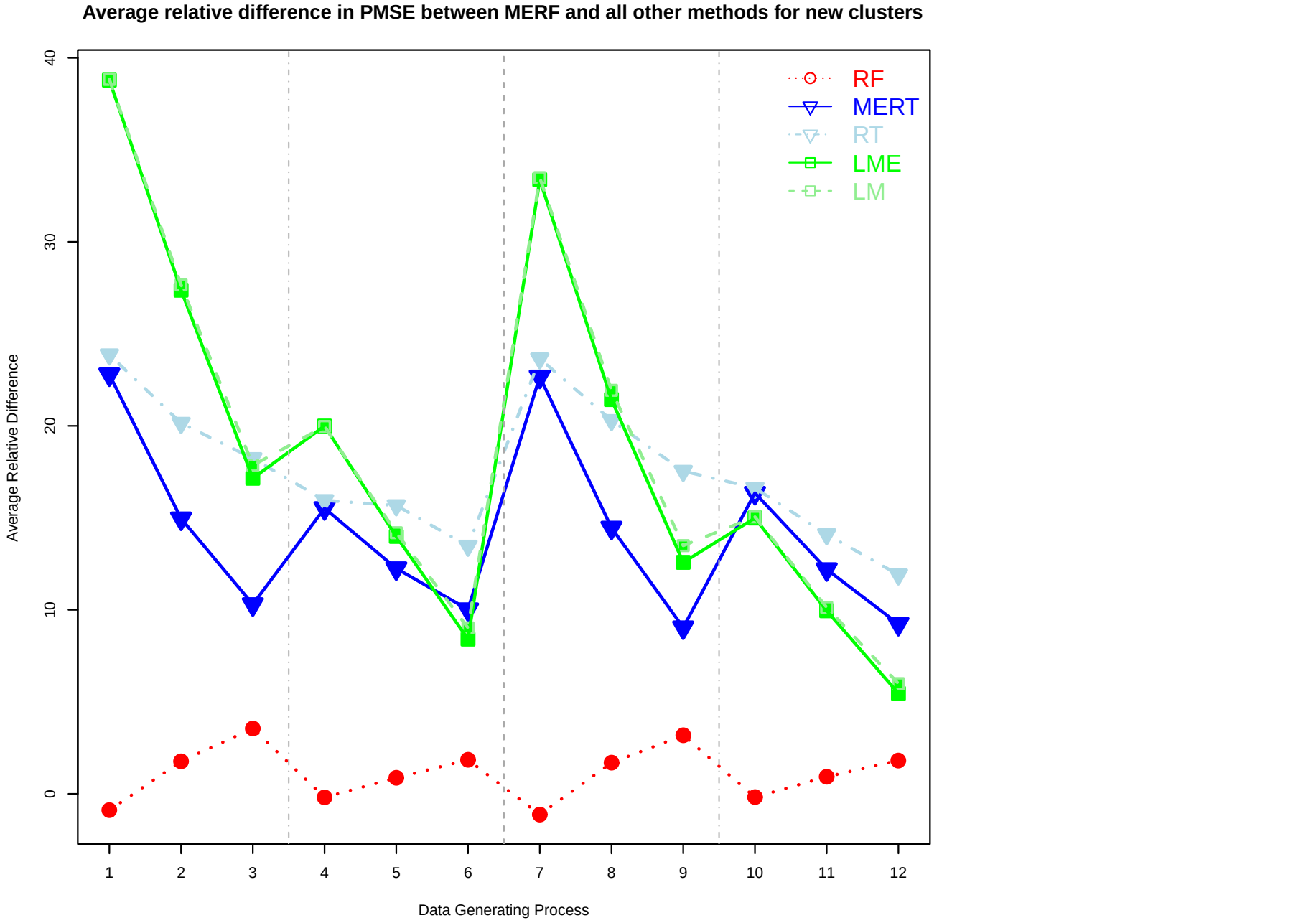
Relative difference (RD) in PMSE between MERF and RF.

$$RD = \frac{PMSE_{RF} - PMSE_{MERF}}{PMSE_{RF}} \times 100.$$


Relative difference in PMSE between MERF and RF for known clusters
PREV = .1 .3 .5
PREV = .1 .3 .5
PREV = .1 .3 .5
PREV = .1 .3 .5
Relative Difference in PMSE
0
20
40
60
PTEV = .9
PTEV = .6
Independent predictors, i.e. ρ = 0
PTEV = .9
PTEV = .6
Dependent predictors, with ρ = .4
1 2 3 4 5 6 7 8 9 10 11 12
Data Generating Process


Relative difference in PMSE between MERF and RF for new clusters
PREV =
.1
.3
.5
PTEV = .9
PTEV = .6
Independent predictors, i.e. ρ = 0
Dependent predictors, with ρ = .4
Relative Difference in PMSE
10
5
0
−5
−10
1
2
3
4
5
6
7
8
9
10
11
12
Data Generating Process


Average relative difference in PMSE between MERF and all other methods for known clusters
RF
MERT
RT
LME
LM
Average Relative Difference
0
10
20
30
40
50
60
70
1
2
3
4
5
6
7
8
9
10
11
12
Data Generating Process


Average relative difference in PMSE between MERF and all other methods for new clusters
Average Relative Difference
0
10
20
30
40
1
2
3
4
5
6
7
8
9
10
11
12
Data Generating Process
RF
MERT
RT
LME
LM

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 Québec 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.

6. Movie *distributer* (1-Vivafilm; 2-Sony; 3-Warner; 4-Fox; 5-Universal; 6-Paramount; 7-Disney; 8-Christal Films; 9-Films Séville; 10-DreamWorks; 11-MGM; 12-TVA Films; 13-Equinoxe; 14-Others).
7. *Country* of origin (1-USA; 2-Québec; 3-France; 4-Rest of Canada; 5-Other countries).
8. *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).
    3. Standard regression tree (RT).
    4. Random intercept tree (MERT).
    5. Linear model (LM).
    6. Random intercept linear mixed model (LMM).

Table: Results (PMSE) for the first–week box office revenues example.

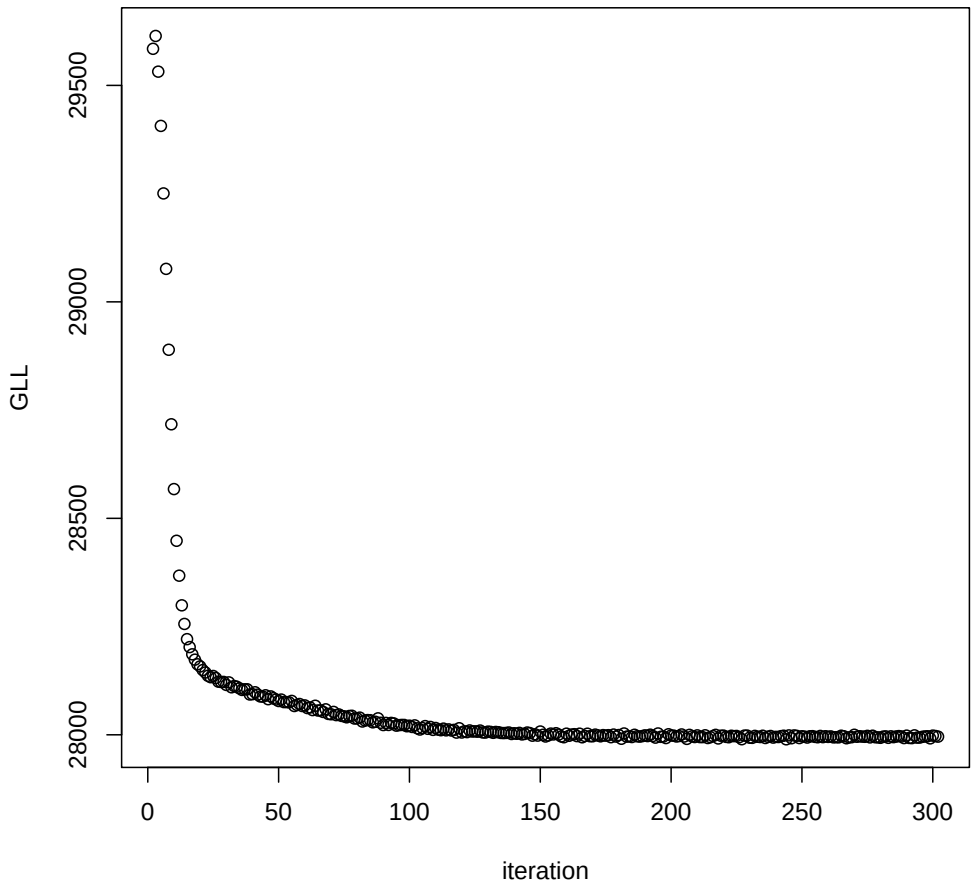
		PMSE	Estimated ICC
Random Forest	MERF	0.47	0.54
	RF	0.60	–
Single Tree	MERT	0.53	0.51
	RT	0.90	–
Linear Model	LMM	0.62	0.42
	LM	1.00	–

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:

$$GLL(f, b_i|y) = \sum_{i=1}^{n} \{[y_i - f(X_i) - Z_i b_i]^T R_i^{-1} [y_i - f(X_i) - Z_i b_i] + b_i^T D^{-1} b_i + \log|D| + \log|R_i|\}. \quad (2)$$

For the movie data, here is the graph of GLL by iterations for MERF.


GLL
28000
28500
29000
29500
0
50
100
150
200
250
300
iteration

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. $V[Y_{ij}|b_i] = \upsilon(E[Y_{ij}|b_i])\phi$, where υ 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) $\upsilon(u) = 1$, $\phi = \sigma^2$ 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:

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

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 β, D and predictions of the b_i's. The model is fitted using weights that are inversely proportional to the variance of $\upsilon^{-1}(\hat{\mu_{ij}^b})\epsilon_{ij}$.
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:
$g(E[Y_{ij}|b_i]) = \eta_{ij} = f(X_{ij}) + Z'_{ij}b_i$.

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 $\upsilon^{-1}(\hat{\mu_{ij}^b})\epsilon_{ij}$.
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.

DGP	Data Structure										
	Fixed Component							Random Component			
	Effect	φ^1	φ^2	φ^3	φ^4	φ^5	φ^6	Structure	Effect	d_{11}	d_{22}
1	Large	0.10	0.20	0.80	0.20	0.80	0.90	No random effect	-	0.00	0.00
2	Small	0.20	0.40	0.70	0.30	0.60	0.80				
3	Large	0.10	0.20	0.80	0.20	0.80	0.90	Random intercept	Small	4.00	0.00
4									Large	10.00	0.00
5	Small	0.20	0.40	0.70	0.30	0.60	0.80		Small	0.50	0.00
6									Large	4.00	0.00
7	Large	0.10	0.20	0.80	0.20	0.80	0.90	Random intercept and covariate	Small	2.00	0.05
8									Large	5.00	0.25
9	Small	0.20	0.40	0.70	0.30	0.60	0.80		Small	0.25	0.01
10									Large	2.00	0.05

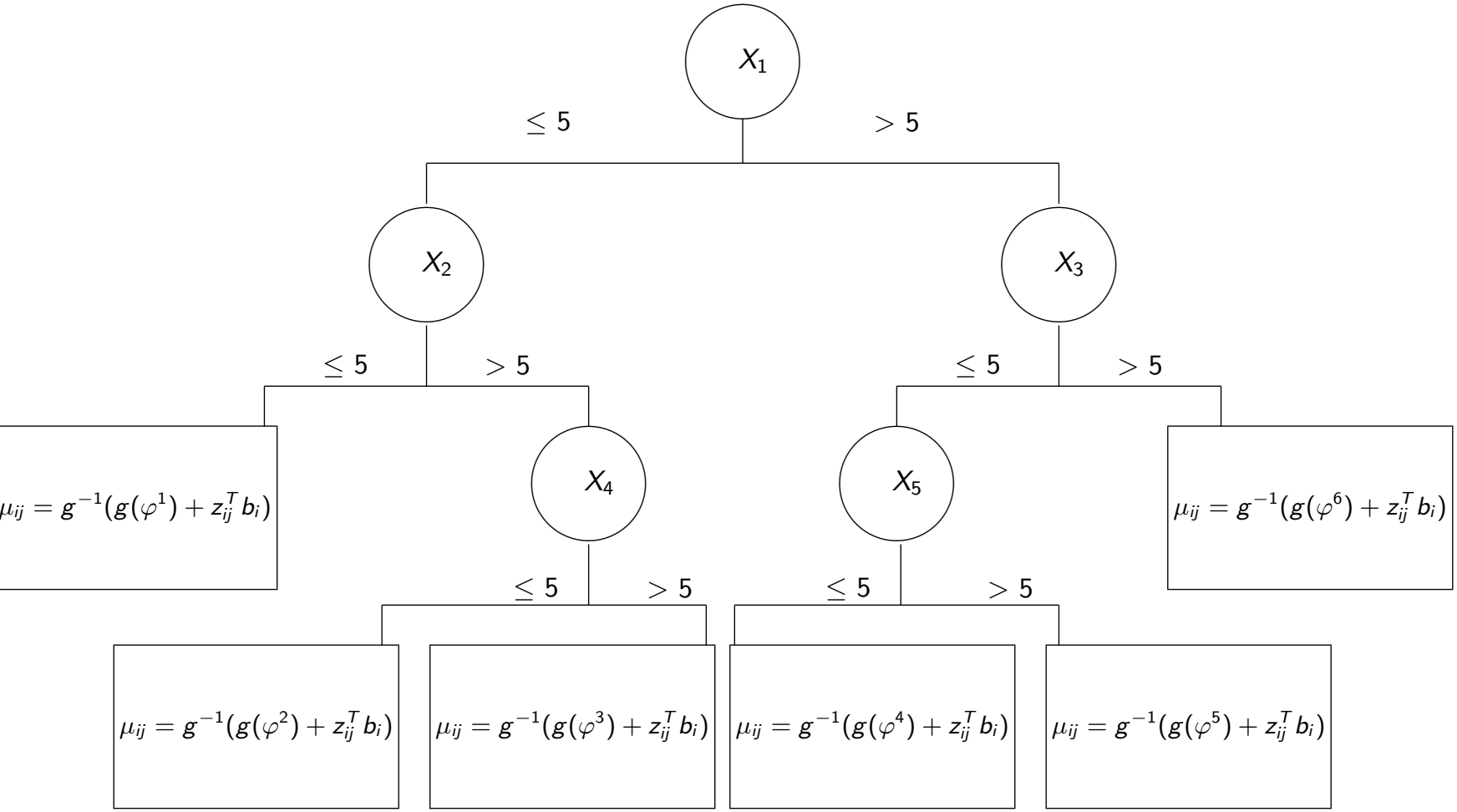


Figure: Generalized mixed effects tree structure used for the simulation study, with $g(.)$ being the logit link function and $g(.)^{-1}$ the inverse-logit or logistic function in binary responses scenarios, and respectively, the log link function and the inverse-log or power function in Poisson responses 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.

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

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

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.

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

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

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.

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

* 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).

5. `PronomOfTheme`: Pronominality of theme (phrases headed by pronouns (personal, demonstrative, and indefinite) vs. Those headed by nonpronouns such as nouns and gerunds).
6. `DefinOfRec`: Definiteness of recipient.
7. `DefinOfTheme`: Definiteness of theme.
8. `AnimacyOfRec`: Animacy of recipient.
9. `AnimacyOfTheme`: Animacy of theme.
10. `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.

STD	10.7%
GLMM	8,0%
GMERT	6.6%

References

- Abdollel, M., LeBlanc, M., Stephens, D. and Harrison, R. V. (2002). Binary partitioning for continuous longitudinal data: Categorizing a prognostic variable. *Statistics in Medicine* **21**, 3395-3409.
- Adler W., Potapov S. and Lausen B. (2011). Classification of repeated measurements data using tree-based ensemble methods. *Computational Statistics* **26**, 355-369
- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees.* Wadsworth International Group. Belmont, California.
- Breiman, L. (2001). Random forests. *Machine Learning* **45**, 5-32.

- Bresnan, J., Cueni, A., Nikitina, T. and Baayen, R. H. (2005). Predicting the Dative Alternation. Knaw Academy Colloquium: Cognitive Foundations of Interpretation. October 2728, 2004, Amsterdam.
- Chen, J., Yu, K., Hsing, A., Therneau, T.M. (2007). A partially linear tree-based regression model for assessing complex joint gene-gene and gene-environment effects. *Genetic Epidemiology* **32**, 238251.
- Ciampi, A., R. du Berger, H. G. Taylor and J. Thiffault. (1991). RECPAM: A computer program for recursive partition and amalgamation for survival data and other situations frequently occurring in biostatistics. iii. Classiffication according to a multivariate construct. Application to Data on

Haemophilus Influenzae Type b Meningitis. *Computer Methods and Programs in Biomedicine* **36**, 51-61.

- Eo, S.-H. and Cho, H. (2013). Tree-structured Mixed-effects Regression Modeling for Longitudinal Data. *Journal of Computational and Graphical Statistics*. In press.
- Field, C. A. and Welsh, A. H. (2007). Bootstrapping clustered data. *Journal of the Royal Statistical Society, Series B* **69**, 369-390
- Fitzmaurice, G.M,, Laird, N.M. and Ware, J.H. (2011). *Applied Longitudinal Analysis*. 2nd edition. Wiley.
- Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **1**, 14-23.

- Loh, W. and Zheng, W. (2013). Regression Trees for Longitudinal and Multiresponse Data. *Annals of Applied Statistics* **7**, 495-522.
- 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.
- Hajjem, A., Bellavance, F. and Larocque, D. (2014). Generalized Mixed Effects Regression Trees. Revise and resubmit at *Computational Statistics and Data Analysis*.

- Karpievitch, Y. V., Hill, E. G., Leclerc, A. P., Dabney, A. R. and Almeida, J. S. (2009). An introspective comparison of random forest-based classifiers for the analysis of cluster-correlated data by way of RF++. *PLoS ONE* 4(9):e7087.
- Rokach, L. (2009). Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis* **53**, 4046-4072
- Segal, M. R. (1992). Tree-structured methods for longitudinal data. *Journal of the American Statistical Association,* **87**, 407-418.

- 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.
- Siroky, D. S. (2009). Navigating random forests and related advances in algorithmic modeling. *Statistics Surveys* **3**, 147-163.
- Verikas, A., Gelzinis, A. and Bacauskiene, M. (2011). Mining data with random forests: A survey and results of new tests. *Pattern Recognition* **44**, 330-349.
- Wu, H. and Zhang, J. T. (2006). *Nonparametric regression methods for longitidunal data analysis: Mixed-effects modeling approaches.* Wiley. New York.

- Yu, K., Wheeler, W., Li, Q., Bergen, A.W., Caporaso, N., Chatterjee, N., Chen, J. (2010). A partially linear tree-based regression model for multivariate outcomes. *Biometrics* **66**, 8996.
- Zhang, H. (1998). Classification trees for multiple binary responses. *Journal of the American Statistical Association* **93**, 180-193.

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 $W_i = diag(w_{ij})$, with $i = 1, ..., n$ and $j = 1, ..., m_i$.

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 $\tilde{y}^*_{i_t j_t}$, $i_t = 1, ..., n_t$ and $j_t = 1, ..., m_{i_t}$. Then, given the weights $w_{i_t j_t}$ 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 $\bar{\tilde{y}}^*_t = \frac{\sum_{i_t=1}^{n_t} \sum_{j_t=1}^{m_{i_t}} w_{i_t j_t} \tilde{y}^*_{i_t j_t}}{\sum_{i_t=1}^{n_t} \sum_{j_t=1}^{n_{i_t}} w_{i_t j_t}}$. This corresponds to the fitted value $\hat{\tilde{y}}^*_t = \hat{f}(X_{i_t})$ when t is a terminal node.
- The error of node t equals its weighted sums of squares or corrected deviance DEV_t, with $DEV_t = \sum_{i_t=1}^{n_t} \sum_{j_t=1}^{m_{i_t}} w_{i_t j_t} (\tilde{y}^*_{i_t j_t} - \bar{\tilde{y}}^*_{lt})^2$.
- 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., $Improve = 1 - \frac{(DEV_{t_l} + DEV_{t_r})}{DEV_t}$.

- The cross-validated relative error corresponding to a given complexity parameter value for the tree T is defined as follows: $xerror = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m_i} w_{ij}(\tilde{y}^*_{ij} - \hat{\tilde{y}}^*_{(-ij)})^2}{DEV_{root}}$, with $\hat{\tilde{y}}^*_{(-ij)}$ being the predicted value for observation j in cluster i, from the standard regression tree model that is fitted without this observation.