Statistics and Probability Letters 81 (2011) 451–459

Contents lists available at ScienceDirect

Statistics and Probability Letters

journal homepage: www.elsevier.com/locate/stapro

Mixed effects regression trees for clustered data

Ahlem Hajjem, François Bellavance, Denis Larocque *

Department of Management Sciences, HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal, QC, Canada H3T 2A7

ARTICLE INFO

Article history:
Received 29 April 2010
Received in revised form 18 October 2010
Accepted 6 December 2010
Available online 21 December 2010

Keywords:
Tree based methods
Clustered data
Mixed effects
Expectation-maximization (EM) algorithm

ABSTRACT

This paper presents an extension of the standard regression tree method to clustered data. Previous works extending tree methods to accommodate correlated data are mainly based on the multivariate repeated-measures approach. We propose a "mixed effects regression tree" method where the correlated observations are viewed as nested within clusters rather than as vectors of multivariate repeated responses. The proposed method can handle unbalanced clusters, allows observations within clusters to be split, and can incorporate random effects and observation-level covariates. We implemented the proposed method using a standard tree algorithm within the framework of the expectation-maximization (EM) algorithm. The simulation results show that the proposed regression tree method provides substantial improvements over standard trees when the random effects are non negligible. A real data example is used to illustrate the method.

1. Introduction

Clustered data, often obtained by multistage sampling with observations nested within higher-level units (clusters), is common throughout many areas of research (e.g. Raudenbush and Bryk, 2002; Goldstein, 2003; Fitzmaurice et al., 2004). The data structure consists of individuals nested within groups. Usually, observations that belong to the same cluster tend to be more similar to each other than observations from different clusters. The focus of this paper is to extend the standard regression tree methods to clustered data and therefore take into account the correlation between observations within a cluster.

Tree based methods became popular with the CART (classification and regression trees) algorithm (Breiman et al., 1984) and are now standard tools in statistical learning. Previous extensions of tree based methods to accommodate the correlation structure induced by clustered data were mainly developed for longitudinal settings. 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. Zhang (1998) treated the multivariate binary response case in a similar setting. Lee (2005) suggested a 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. By using a likelihood ratio test statistic from a mixed model as the splitting criterion, Abdollel et al. (2002) were able to lift the requirements that subjects have an equal number of repeated observations.

All these extensions of tree based methods to handle correlations induced by the data structure do not allow observation-level covariates (i.e. time-varying covariates in the context of longitudinal studies) to be candidates in the splitting process and, consequently, all repeated observations from a given subject remain together during the tree building process and can

* Corresponding author.
E-mail address: denis.larocque@hec.ca (D. Larocque).

0167-7152/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.spl.2010.12.003

not be split across different nodes. This is different from the method proposed in this paper which can split observations within clusters since observation-level covariates are candidates in the splitting process. Moreover, the proposed tree method can appropriately deal with the possible random effects of observation-level covariates. The proposed mixed effects regression tree method has the following characteristics: (1) it can handle clusters with different numbers of observations (unbalanced clusters); (2) it allows the inclusion of observation-level and cluster-level covariates in the splitting process, and consequently, observations from the same cluster can be separated into different nodes during the tree growing process; (3) it allows observation-level covariates to have random effects.

The remainder of this article is organized as follows: Section 2 describes the proposed mixed effects regression tree approach; Section 3 presents a simulation study to evaluate the performance of the method; Section 4 illustrates the application of the method with a real data set; and Section 5 provides a discussion about possible extensions and ends with concluding remarks.

2. Mixed effects regression tree (MERT) approach

A statistical model for clustered data typically includes two components: A fixed or population-averaged and a random or cluster-specific component. The basic idea behind the proposed mixed effects regression tree is to dissociate the fixed from the random effects. We use a standard regression tree to model the fixed effects and a node-invariant linear structure to model the random effects. The method is implemented using a standard tree algorithm within the framework of the expectation-maximization (EM) algorithm (McLachlan and Krishman, 1997). More precisely, the linear estimation of the fixed component in the linear mixed effects (LME) model (Fitzmaurice et al., 2004) is replaced by a standard regression tree algorithm. Let us first briefly review the LME model and the EM algorithm.

2.1. EM algorithm for the linear mixed effects model

The LME model is generally written in the following form:

$$\begin{aligned} &y_i = X_i\beta + Z_i b_i + \epsilon_i, \\ &b_i \sim N(0, D), \epsilon_i \sim N(0, R_i), \\ &i = 1, \ldots, n, \end{aligned} \tag{1}$$

where $y_i = [y_{i1}, \ldots, y_{in_i}]^T$ is the $n_i \times 1$ vector of responses for the n_i observations in cluster i, $X_i = [x_{i1}, \ldots, x_{in_i}]^T$ is the $n_i \times p$ matrix of fixed-effects covariates, $Z_i = [z_{i1}, \ldots, z_{in_i}]^T$ is the $n_i \times q$ matrix of random-effects covariates, $\epsilon_i = [\epsilon_{i1}, \ldots, \epsilon_{in_i}]^T$ is the $n_i \times 1$ vector of errors, $b_i = (b_{i1}, \ldots, b_{iq})^T$ is the $q \times 1$ unknown vector of random effects for cluster i, and β is the $p \times 1$ unknown vector of parameters for the fixed effects. The total number of observations is $N = \sum_{i=1}^{n} n_i$. The covariance matrix of b_i is D while R_i is the covariance matrix of ϵ_i. The usual LME model also assumes that b_i and ϵ_i are independent and normally distributed and that the between-clusters observations are independent. Hence, the covariance matrix of the vector of observations y_i in cluster i is $V_i = \mathrm{Cov}(y_i) = Z_i D Z_i^T + R_i$, and $V = \mathrm{Cov}(y) = \mathrm{diag}(V_1, \ldots, V_n)$, where $y = [y_1^T, \ldots, y_n^T]^T$. We will further assume that the correlation is induced solely via the between-clusters variation, that is, R_i is diagonal ($R_i = \sigma^2 I_{n_i}$, $i = 1, \ldots, n$). This assumption is suitable for a large class of clustered data problems (Raudenbush and Bryk, 2002, page 30). The parameters in LME models can be estimated by the method of maximum likelihood (ML) implemented with the EM algorithm.

The major cycle for the ML-based EM-algorithm, as described in Section 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 $\hat{b}_{i(r)}$

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

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

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

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

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

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

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.

2.2. EM algorithm for MERT

The proposed MERT model is:

$$\begin{aligned} &y_i = f(X_i) + Z_i b_i + \epsilon_i, \\ &b_i \sim N(0, D),\ \epsilon_i \sim N(0, R_i), \\ &i = 1, \ldots, n, \end{aligned} \tag{2}$$

where all quantities are defined as in Section 2.1 except that the linear fixed part $X_i\beta$ in (1) is replaced by the function $f(X_i)$ that will be estimated with a standard tree based model. The random part, $Z_i b_i$, is still assumed to be linear.

The MERT 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 standard tree structure. The algorithm is as follows:

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 $\hat{f}(X_i)_{(r)}$ an estimate of $f(X_i)$ obtained from a standard tree algorithm with $y^*_{i(r)}$ as responses and X_i, $i = 1, \ldots, n$, as covariates. Note that the tree is built as usual using all N individual observations as inputs along with their covariate vectors,

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

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

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

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

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

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

Step 3. Repeat steps 1 and 2 until convergence.

In words, the algorithm starts at step 0 with default values for $\hat{b}_i$, $\hat{\sigma}^2$, and $\hat{D}$. At step 1, it first calculates the fixed part of the response variable, y^*_i, i.e., the response variable from which we remove the current available value of the random part. Second, it estimates the fixed component $\hat{f}(X_i)$ using a standard tree algorithm with y^*_i as responses and X_i as covariates. Third, it updates $\hat{b}_i$. At step 2, it updates the variance components $\hat{\sigma}^2$ and $\hat{D}$ based on the residuals after the estimated fixed component $\hat{f}(X_i)$ is removed from the raw data y_i. It keeps iterating by repeating steps 1 and 2 until convergence which is monitored by computing, at each iteration, the following generalized log-likelihood (*GLL*) criterion:

$$GLL(f, b_i|y) = \sum_{i=1}^{n}\{[y_i - f(X_i) - Z_i b_i]^T R_i^{-1}[y_i - f(X_i) - Z_i b_i] + b_i^T D^{-1} b_i + \log|D| + \log|R_i|\}. \tag{3}$$

To predict the response for a new observation that belongs to a cluster among those used to fit the MERT model, we use both its corresponding population-averaged tree prediction and the predicted random part corresponding to its cluster. For a new observation that belongs to a cluster not included in the sample used to estimate the model parameters, we can only take the corresponding population-averaged tree prediction.

3. Simulation

In this section, we investigate the performance of MERT in comparison to standard trees. The proposed method was implemented in R (R Development Team, 2007) using the function *rpart* (Therneau and Atkinson, 1997). This function implements cost-complexity pruning based on cross-validation after an initial large tree is grown. The default settings of *rpart* are used; the largest tree is grown and pruned automatically using the 1-SE rule of Breiman et al. (1984).

To compare the performance of the standard and MERT methods, we evaluate both their ability to find the true tree structure used to generate the data, and their predictive accuracy measured by the predictive mean squared error (PMSE).

3.1. Simulation design

The first simulation design used has a hierarchical structure of 100 balanced clusters with 55 observations generated in each cluster. The first five observations in each cluster form the training sample, and the other 50 observations are left

Table 1
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

for the test sample. Consequently, the trees are built with 500 observations (100 clusters of 5 observations). Three random variables, X_1, X_2, and X_3, are first generated independently with a uniform distribution in the interval [0, 10]; they serve as predictors. The response variable y is generated based on the following fixed tree rules along with the random components:

Leaf 1. If $x_{1ij} \leq 5$ and $x_{2ij} \leq 5$ then $y_{ij} = \mu_1 + z_{ij}^T b_i + \epsilon_{ij}$,
Leaf 2. if $x_{1ij} \leq 5$ and $x_{2ij} > 5$ then $y_{ij} = \mu_2 + z_{ij}^T b_i + \epsilon_{ij}$,
Leaf 3. if $x_{1ij} > 5$ and $x_{3ij} \leq 5$ then $y_{ij} = \mu_3 + z_{ij}^T b_i + \epsilon_{ij}$,
Leaf 4. if $x_{1ij} > 5$ and $x_{3ij} > 5$ then $y_{ij} = \mu_4 + z_{ij}^T b_i + \epsilon_{ij}$,

where b_i and ϵ_i are generated according to $N(0, D)$ and $N(0, I)$ respectively, for $i = 1, \ldots, 100$ and $j = 1, \ldots, 55$. Each observation j in cluster i falls into only one of the four terminal nodes with mean response value equal to μ_1, μ_2, μ_3, or μ_4 respectively.

We consider 14 different data generating processes (DGP), summarized in Table 1. Two different scenarios are selected for the fixed components. In the first scenario, the means of the four terminal nodes are widely spread with $\mu_1 = -20$, $\mu_2 = -10$, $\mu_3 = 10$ and $\mu_4 = 20$, while in the second scenario, they are closer with $\mu_1 = 10$, $\mu_2 = 11$, $\mu_3 = 12$ and $\mu_4 = 13$. The random components are generated based on the following three different scenarios:

1. No random effects (NRE), i.e. $D = 0$.
2. Random intercept (RI), i.e. $z_{ij} = 1$ for $i = 1, \ldots, 100$, and $j = 1, \ldots, 55$, and $D = d_{11} > 0$.
3. Random intercept and covariate (RIC) which is a RI with a linear random effect for X_1. More precisely, $z_{ij} = [1, x_{1ij}]$ for $i = 1, \ldots, 100, j = 1, \ldots, 55$, and $D = \begin{pmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{pmatrix}$, $d_{11} > 0$ and $d_{22} > 0$.

In all cases, the within-cluster variance σ^2 is set to 1. We consider two levels for the between-clusters covariance matrix D. In the RI case, we use $D = d_{11} = 0.25$ and 0.5 which are equivalent to an intra-cluster correlation coefficient of 0.20 and 0.33 respectively. In the RIC case, we have two additional conditions based on the value of the correlation between the random components, $d_{12}/\sqrt{d_{11} + d_{22}} = 0$ and $d_{12}/\sqrt{d_{11} + d_{22}} = 0.5$; in the first correlation scenario, $d_{11} = d_{22} = 0.25$, and in the second $d_{11} = d_{22} = 0.5$.

We adjusted three models for each DGP scenario: (1) a standard (STD) tree model, (2) a random intercept (RI) tree model, and (3) a random intercept and covariate (RIC) tree model. The true model is the one corresponding to the DGP used to generate the data. Overall, we built 42 regression tree models (14 scenarios × 3 models). The simulation results are obtained by means of 100 runs.

Concerning the computational burden of MERT, the R code used in the simulations was not built to optimize the computation efficiency. Also, many intermediate elements were kept over all iterations in order to investigate the convergence of GLL and of other different elements of the model. With this in mind, we can report that the computing time of a single iteration of MERT is about 8 to 20 times greater than the time needed to build a single standard tree with rpart, depending on the DGP and the random effects structure in the fitted model (e.g., RI or RIC).

3.2. Simulation results

First, we evaluate the performance of the approaches in terms of recovering the right tree structure. The results are presented in Table 2. The left part shows the results for balanced clusters while the right one is for the unbalanced case (as described below). In all scenarios where the means of the terminal nodes are very different (i.e. large fixed effect: DGPs 1, 3,

Table 2
Results of the 100 simulation runs in terms of recovering the right tree structure and the predictive mean square error (PMSE) in the balanced and unbalanced scenarios.

DGP	Fixed effect	Random effect	Fitted tree model[a]	Balanced						Unbalanced					
				% of trees with the right tree structure	PMSE					% of trees with the right tree structure	PMSE				
					Avg.	Med.	Min	Max	Std		Avg.	Med.	Min	Max	Std
1	Large	No random effect	STD	100	2.14	1.95	1.04	6.10	0.97	100	1.97	1.84	0.96	6.19	0.87
			RI	100	2.14	1.95	1.04	6.10	0.97	100	1.97	1.84	0.96	6.19	0.87
			RIC	100	2.15	1.96	1.04	6.10	0.97	100	1.97	1.84	0.96	6.20	0.87
2	Small		STD	95	1.04	1.03	0.96	1.21	0.04	94	0.94	0.93	0.88	1.06	0.03
			RI	97	1.04	1.03	0.96	1.21	0.04	95	0.94	0.93	0.88	1.05	0.03
			RIC	97	1.04	1.04	0.96	1.21	0.04	95	0.94	0.93	0.88	1.05	0.03
3	Large	Random intercept	STD	100	2.43	2.09	1.26	5.49	1.01	100	2.25	1.89	1.13	6.08	1.01
			RI	100	2.29	1.96	1.14	5.38	1.01	100	2.12	1.76	1.03	5.96	1.01
			RIC	100	2.29	1.96	1.14	5.38	1.01	100	2.12	1.76	1.03	5.97	1.01
4			STD	100	2.61	2.37	1.39	5.95	0.91	100	2.37	1.99	1.33	6.21	1.08
			RI	100	2.24	1.94	1.11	5.53	0.91	100	2.04	1.65	1.02	5.89	1.07
			RIC	100	2.25	1.94	1.11	5.53	0.91	100	2.04	1.65	1.03	5.89	1.07
5	Small		STD	77	1.31	1.30	1.18	1.52	0.07	81	1.18	1.18	1.05	1.48	0.07
			RI	91	1.16	1.15	1.07	1.33	0.05	91	1.04	1.03	0.96	1.21	0.05
			RIC	91	1.17	1.16	1.08	1.33	0.05	93	1.04	1.04	0.97	1.18	0.04
6			STD	60	1.58	1.59	1.35	1.82	0.10	61	1.43	1.44	1.20	2.02	0.12
			RI	86	1.20	1.18	1.08	1.37	0.06	85	1.06	1.05	0.96	1.27	0.06
			RIC	88	1.20	1.19	1.08	1.37	0.06	84	1.07	1.06	0.98	1.27	0.06
7	Large	Random intercept and covariate with 0 correlation	STD	100	10.95	10.99	7.62	14.96	1.62	100	9.71	9.53	6.10	18.59	1.80
			RI	100	4.90	4.70	3.25	7.91	1.02	100	4.43	4.13	2.59	12.14	1.22
			RIC	100	2.48	2.22	1.30	5.68	0.94	100	2.41	2.19	1.29	9.24	1.12
8			STD	100	19.49	19.13	13.15	28.68	2.69	100	16.95	16.94	11.38	25.26	2.89
			RI	100	7.44	7.08	5.00	13.98	1.42	100	6.41	6.21	4.36	14.66	1.36
			RIC	100	2.69	2.41	1.32	8.17	1.25	100	2.34	2.09	1.25	9.67	1.12
9	Small		STD	0	10.28	9.95	7.10	14.58	1.45	0	9.18	9.06	6.70	13.53	1.40
			RI	6	3.93	3.91	3.05	4.96	0.38	4	3.52	3.50	2.79	4.77	0.41
			RIC	64	1.41	1.40	1.23	1.61	0.10	71	1.30	1.27	1.07	1.89	0.11
10			STD	0	18.90	18.65	14.30	26.44	2.59	0	17.06	16.90	12.55	24.92	2.59
			RI	0	6.46	6.31	4.90	9.99	0.91	1	5.77	5.67	4.22	8.63	0.76
			RIC	60	1.46	1.46	1.25	1.82	0.11	68	1.39	1.36	1.16	1.80	0.13
11	Large	Random intercept and covariate with 0.5 correlation	STD	100	12.25	11.85	8.65	18.47	2.15	100	10.86	10.83	7.77	15.36	1.65
			RI	100	4.96	4.59	3.40	10.10	1.45	100	4.34	4.10	2.78	8.30	1.02
			RIC	100	2.57	2.11	1.30	7.28	1.33	100	2.26	1.85	1.18	6.19	0.99
12			STD	100	21.52	21.19	15.45	30.98	2.85	100	20.01	19.88	12.23	27.95	3.15
			RI	100	7.10	6.91	5.21	12.22	1.11	100	6.65	6.36	3.86	13.47	1.37
			RIC	100	2.34	2.06	1.27	6.76	0.90	100	2.31	1.95	1.28	9.64	1.14
13	Small		STD	0	11.75	11.47	9.04	17.92	1.70	0	10.46	10.47	6.36	15.78	1.66
			RI	5	4.01	4.00	2.82	5.50	0.41	3	3.56	3.52	2.69	5.05	0.41
			RIC	68	1.39	1.38	1.21	1.72	0.10	75	1.27	1.25	1.10	1.63	0.10
14			STD	0	21.60	21.51	15.80	28.85	2.91	0	20.24	20.06	12.50	28.36	3.02
			RI	1	6.45	6.40	4.96	8.59	0.79	0	6.00	5.93	4.21	8.77	0.82
			RIC	67	1.42	1.41	1.20	1.77	0.11	71	1.36	1.36	1.12	1.75	0.12

[a] STD: Standard tree model; RI: Random intercept tree model; RIC: Random intercept and covariate tree model.

4, 7, 8, 11, and 12), both the proposed approach (RI and RIC tree) and the standard tree algorithm succeed in finding the right tree structure. However, when the difference between the means of the terminal nodes is small, the higher the intra-cluster correlation is the harder it is for all methods to find the right tree structure (see DGPs 5, 6, 9, 10, 13 and 14). In all of these cases however, RIC tree results are closer to the true data partition compared to partitions obtained from the RI tree or the standard tree. For DGPs 9, 10, 13 and 14, the standard tree has never identified the right tree structure, while the RIC tree approach does best with recovery rates of 64%, 60%, 68%, and 67%, respectively.

The performance of the methods is also judged based on their predictive accuracy measured by the predictive mean squared error: PMSE $= (5000)^{-1} \sum_{i=1}^{100} \sum_{j=1}^{50} (y_{ij} - \hat{y}_{ij})^2$, where $\hat{y}_{ij}$ is the predicted response for observation j in cluster i in the test set. Recall that the trees are built with 100 clusters of 5 observations each but the PMSE is computed on a much larger number of observations, i.e. 5000 observations in the test set (50 observations in each cluster), in order to obtain a more accurate estimate of the performance of the methods. The average, median, minimum, maximum and standard deviation of PMSE over the 100 runs were calculated, and the results are presented in the left side of Table 2.

All three methods have exactly the same average performance when the data are uncorrelated (DGPs 1 and 2). But in all cases with a random component (DGPs 3 to 14), the proposed MERT approach does better than the standard tree algorithm even with the wrong specification of the random component part. Note that, as for DGPs 1 and 2, over-specifying the random component part performs as well as the true specification. Again, the higher the intra-cluster correlation the more difficult it is for the standard tree to predict accurately the response variable, but not for the mixed effects approach which handles appropriately this correlation. The improvement of the new approach over the standard tree algorithm is often large, especially when a random covariate effect is present (DGPs 7 to 14). For example, in DGP 14, the RIC tree has an average PMSE of 1.42 compared to 21.6 for the standard tree.

We repeated the above simulation using unbalanced clusters. More precisely, we used a hierarchical structure of 100 unbalanced clusters and 5000 observations: 20 clusters with 10 observations, 20 with 30 observations, 20 with 50 observations, 20 with 70 observations, and 20 with 90 observations. The first 10% of the generated observations in each cluster form the training sample, and the other 90% are kept for the test sample. Consequently, the trees are built with 500 observations nested within 100 unbalanced clusters having 1, 3, 5, 7, or 9 observations. The remaining 4500 observations form the test set. The right side of Table 2 shows the results for the 14 DGPs obtained when using unbalanced clusters. These results are very similar to those obtained in the balanced case and indicate that MERT performs also well with unbalanced clusters.

4. Data example

In this section, we illustrate the proposed tree method using a real data set on first-week box office revenues of movies presented in the province of Quebec in Canada from 2001 to 2008. The unit of analysis is a screen showing the new movie during its first week of release. The importance of the first-week revenues is well-known in the industry. Typically, it represents about 25% of the total box office of a general public film (Simonoff and Sparrow, 2000). The total number of observations (screens) is 60175. This data includes information on 2656 movies and each movie is treated as a cluster. These clusters are highly unbalanced with an average size of 22.7 screens per movie (minimum = 1; first quartile = 1; median = 8; third quartile = 47; maximum = 93).

4.1. Description of observation and cluster level covariates

We have three covariates at the screen-level (observation-level) and eight at the movie-level (cluster-level). The three screen-level covariates are: (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), and (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).

The eight movie-level covariates are: (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 and 227 min, (3) Movie *genre* (1-Comedy; 2-Drama; 3-Thriller; 4-Action/Adventure; 5-Science fiction; 6-Cartoons; 7-Others), (4) *Visa*, the assigned movie classification (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), and (8) *Size*, total number of screens for a movie in its first-week, commonly used as a proxy for the marketing effort.

Using a learning sub-sample of 30 018 screens within the 2656 movies, we fitted the following three models: (1) a standard regression tree (STD) model, (2) a random intercept regression tree (RI) model, and (3) a random intercept linear regression (LME) model. As is commonly done in box office prediction studies, we model the log transform of the first-week box office revenues since it has a distribution highly skewed to the right. We also took the logarithm of the covariate *Size* to lessen its asymmetry and improve the fit of the LME model. Note that the latter asymmetry has no effect for the STD and RI models but affects the linear mixed effects model.

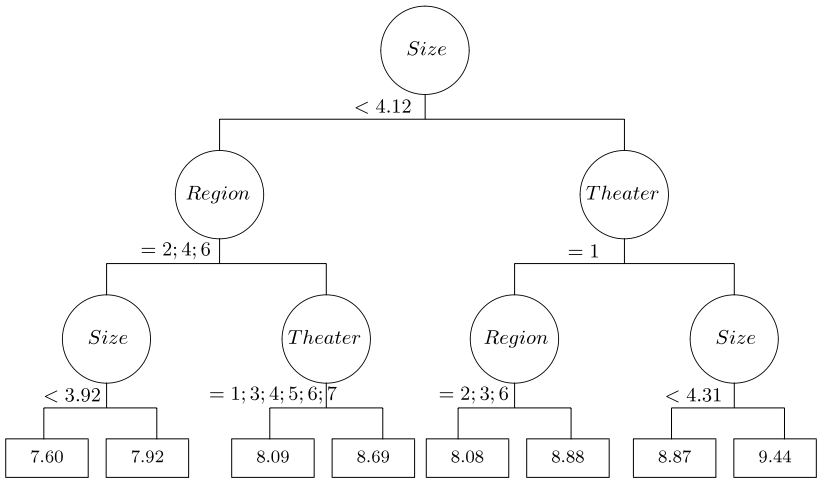

Fig. 1. The first three levels of the standard regression tree for the data example on first-week box office revenues (on the log scale). When the condition below a node is true then go to the left node, otherwise go to the right node. The complete tree has 44 leaves.

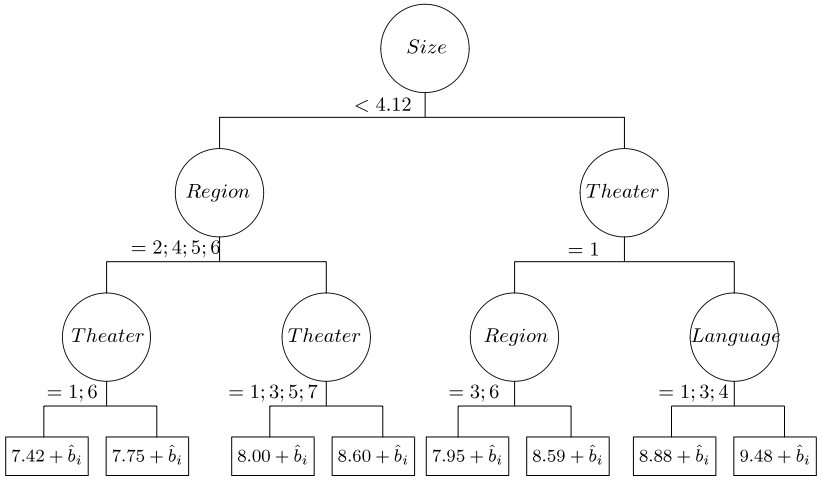

Fig. 2. The first three levels of the random intercept regression tree for the data example on first-week box office revenues (on the log scale). When the condition below a node is true then go to the left node, otherwise go to the right node. The complete tree has 28 leaves.

4.2. Results

All covariates are statistically significant in the LME model (results not shown), but only eight covariates (*Size, Region, Theater, Language, length, Month*, *rating*) are retained in the STD model and only four (*Size, Region, Theater, Language*) are retained by the algorithm in the RI model. The STD structure is larger than the RI structure, i.e., the standard regression tree has 44 leaves while the random intercept regression tree has 28 leaves. However, the RI is not a subtree of the STD; the first splits of the two trees are identical, but their second splits use different partitions based on the same movie-level covariate *Region* (i.e. Region $= 2; 4; 5; 6$ vs. Region $= 2; 4; 6$, respectively). Figs. 1 and 2 show the first three levels of the fitted STD and RI, respectively.

The RI model has the smallest in-sample MSE (0.44). The MSE of the STD model and of the LME model are 0.86 and 0.54, respectively. Thus, in-sample, the RI reduces the MSE of the STD model by 48.93% and reduces the MSE of the LME model by 18.30%. Using the test sub-sample of 30 157 screens within 1920 movies, the RI model also has the best predictive performance; its PMSE is 0.53 while the PMSE of the STD and LME models are 0.90 and 0.62, respectively. Thus, the RI reduces the PMSE of the STD model by 41.63% and reduces the PMSE of the LME model by 14.94%.

In this example, compared to a standard tree, the MERT approach reduced both the PMSE and the tree size and thus improved interpretability.

5. Extensions and concluding remarks

Two directions for possible extensions are discussed in this section followed by a brief conclusion.

Table 3
Results of the 100 simulation runs in terms of recovering the right tree structure and the predictive mean square error (PMSE) when the generated random effects are node specific.

DGP	Fixed effect	Random effect	Fitted tree model[a]	Learning sample size = 500 observations						Learning sample size = 2000 observations					
				% of trees with the right tree structure	PMSE					% of trees with the right tree structure	PMSE				
					Avg.	Med.	Min	Max	Std		Avg.	Med.	Min	Max	Std
4	Large	Node Specific Random intercept	STD	100	2.49	2.27	1.49	7.17	0.84	100	1.77	1.66	1.45	3.03	0.30
			RI	100	2.46	2.23	1.45	7.10	0.84	100	1.69	1.58	1.37	2.94	0.30
			NSRI	100	2.37	2.14	1.35	6.91	0.83	100	1.43	1.33	1.13	2.67	0.30
6	Small		STD	51	1.59	1.58	1.41	1.76	0.08	100	1.51	1.51	1.41	1.68	0.04
			RI	61	1.55	1.53	1.40	1.72	0.08	100	1.43	1.43	1.32	1.53	0.04
			NSRI	59	1.50	1.49	1.33	1.69	0.09	100	1.17	1.18	1.11	1.24	0.03

[a] STD: Standard tree model; RI: Random intercept tree model; NSRI: Node specific random intercept tree model.

5.1. Node specific random effects

As for the LME model, the MERT model (2) separates the fixed effects from the random effects. In fact, the random effects must be specified in advance with the design matrices Z_i. In the spirit of a tree-based model, which let to some extent the data find adaptively a good model, it could be interesting to also let the data find the random effects structure. However, there is no straightforward or unique way to achieve this. Here, we discuss only one possibility that would warrant future research. The idea is to use the tree built for the fixed effects to define a node-specific random effects structure. To achieve this, we allow the matrices Z_i of the MERT algorithm in Section 2.2 to vary from one iteration to the next. However, when the cluster sizes are small, we will quickly run out of observations to estimate random effects separately in each node as the tree grows larger. The idea is then to only define the random effects structures with the first few splits of the tree. Hence the tree defining the random effects structure would be, most of the time, a subtree of the full fixed effects tree. To investigate this approach, we performed a small simulation study using DGPs 4 and 6 of Section 3 with balanced clusters. We generated data as in the Section 3.1 except that the b_i were node-specific. Namely, the response y was generated based on the following model:

Leaf 1. If $x_{1ij} \leq 5$ and $x_{2ij} \leq 5$ then $y_{ij} = \mu_1 + z_{ij}^T b_{1i} + \epsilon_{ij}$,
Leaf 2. if $x_{1ij} \leq 5$ and $x_{2ij} > 5$ then $y_{ij} = \mu_2 + z_{ij}^T b_{2i} + \epsilon_{ij}$,
Leaf 3. if $x_{1ij} > 5$ and $x_{3ij} \leq 5$ then $y_{ij} = \mu_3 + z_{ij}^T b_{3i} + \epsilon_{ij}$,
Leaf 4. if $x_{1ij} > 5$ and $x_{3ij} > 5$ then $y_{ij} = \mu_4 + z_{ij}^T b_{4i} + \epsilon_{ij}$,

where b_{1i}, b_{2i}, b_{3i}, b_{4i} are generated according to $N(0, D)$. Thus, each node has its own random intercept. The remaining parameters are set as previously. We ran 100 runs of simulations with a training set of 500 observations (as in the simulations of Section 3) and then with a training set of 2000 observations. The size of the test set is still 5000 observations. Three methods were compared: (1) a standard tree (STD), (2) a random intercept (RI) tree model, and (3) a node-specific random intercept (NSRI) tree model. Model (2) is the basic MERT model and model (3) is the extension described above. For the NSRI model, we allowed the random effects structure to be defined by at most two splits (four nodes). The results are given in Table 3. Both RI and NSRI have lower PMSE compared to STD in all cases. With a training sample size of 500, the average PMSE improvement of NSRI over RI in DGP 4 and 6 is about 3.7% and 3.2%, respectively. With a training sample size of 2000, the average PMSE improvements are about 15.4% and 18.2%. Hence, the benefit of using NSRI is more striking with a larger sample size. One way to explain this is to realize that with 500 observations and in the best case scenario, i.e. when the tree structure is well-estimated, 125 observations will end up in each leaf and these are used to predict 100 random effects (recall that we have 100 clusters). Thus, most clusters will be represented by a single observation. However, with 2000 observations, we will have about 500 of them in each leaf (about five per cluster) and then the node-specific model can achieve even a better performance. This is only one possibility to extend the MERT approach but this small simulation shows that future work in that direction is worthwhile.

5.2. Inference with MERT models

Traditionally, tree-based models are appreciated for their interpretability. Since there are no obvious parameters, statistical inference has not been developed. Partial linear trees (Chen et al., 2007; Yu et al., 2010) is one promising way to blend traditional statistical analysis with trees. It would then be interesting to investigate "partial linear mixed effects regression trees" that could be defined by

$$y_i = f(X_{1i}) + X_{2i}\beta + Z_i b_i + \epsilon_i$$

by separating the covariates X_i into two parts. The second part, X_{2i} would be the covariates of interest for which inference is needed while the first part, X_{1i}, could include control variables. The literature for these models is well developed when the

nonparametric part f is estimated with a smoothing method but not with trees. Thus, future work in that direction could be of interest. In particular, the bootstrap could be used to make an inference about β.

5.3. Concluding remarks

Statistical models for clustered data typically include two components: A fixed or population-averaged and a random or cluster-specific component. If these two components have an underlying linear and additive structure, and if the normality assumption is reasonable, the LME models are appropriate. If the linear assumption is too restrictive, other structures may be more suitable to represent the true underlying relationship between the covariates and the response variable. The proposed MERT method relaxes the linearity assumption of the fixed component of LME models. In the light of the simulation results, the proposed MERT approach seems to be more appropriate for clustered data than standard tree procedures. Future work could extend the methodology to other types of outcomes, e.g., binary. An R program implementing the MERT procedure is available from the first author.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by Le Fonds québécois de la recherche sur la nature et les technologies (FQRNT). The authors would like to thank the Associate editor and a referee for constructive comments. In particular, they encouraged us to explore the extensions discussed in Sections 5.1 and 5.2. They want to thank the Carmelle and Rémi Marcoux Chair in Arts Management for providing the movie box-office data used in the example, Renaud Legoux for interesting discussions and Mohamed Jendoubi for preparing the data set.

References

Abdollel, M., LeBlanc, M., Stephens, D., Harrison, R.V., 2002. Binary partitioning for continuous longitudinal data: categorizing a prognostic variable. Statistics in Medicine 21, 3395–3409.
Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Wadsworth International Group, Belmont, California.
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, 238–251.
Fitzmaurice, G.M., Laird, N.M., Ware, J.H., 2004. Applied Longitudinal Analysis. Wiley, New York.
Goldstein, H., 2003. Multilevel Statistical Models, 3rd ed. Arnold, London.
Lee, S.K., 2005. On Generalized multivariate decision tree by using GEE. Computational Statistics & Data Analysis 49, 1105–1119.
McLachlan, G.J., Krishman, T., 1997. The EM Algorithm and Extensions. Wiley, New York.
R Development Team. 2007. R: a Language and environment for statistical computing. R Foundation for Statistical Computing: www.R-project.org.
Raudenbush, S.W., Bryk, A.S., 2002. Hierarchical Linear Models: Applications and Data Analysis Method, 2nd ed. Sage, Newbury Park, CA.
Segal, M.R., 1992. Tree-structured methods for longitudinal data. Journal of the American Statistical Association 87, 407–418.
Simonoff, J.S., Sparrow, I.R., 2000. Predicting movie grosses: winners and losers, blockbusters and sleepers. Chance 13 (3), 15–24.
Therneau, T.M., Atkinson, E.J., 1997. An introduction to recursive partitioning using the rpart routines. Technical Report 61, Department of Health Science Research, Mayo Clinic, Rochester.
Wu, H., 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, 89–96.
Zhang, H., 1998. Classification trees for multiple binary responses. Journal of the American Statistical Association 93, 180–193.