Journal of Choice Modelling 46 (2023) 100393

Contents lists available at ScienceDirect

Journal of Choice Modelling

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

Modeling preference heterogeneity using model-based decision trees

Álvaro A. Gutiérrez-Vargas [*], Michel Meulders, Martina Vandebroek

Faculty of Economics and Business, KU Leuven, Leuven, Belgium

ARTICLE INFO

Keywords:
Discrete choice models
Mixed logit
Decision tree
Preference heterogeneity
Recursive partitioning
Machine learning
Simulated maximum likelihood
Latent class models
Allocation model

ABSTRACT

This article investigates the usage of a general model-based recursive partitioning algorithm to model preference heterogeneity. We use the algorithm to grow a decision tree based on statistical tests of the stability of individuals' preference parameters. In particular, we used a Mixed Logit (MIXL) model with alternative-specific attributes at the end leaves of the tree while using individual characteristics as partition variables. This configuration allows us to search for instabilities of the taste parameters across individuals' characteristics. We conduct a simulation study to investigate the algorithm's ability to recover different data generating processes with structural breaks in the taste parameters. The results show that the algorithm can correctly recover diverse tree-like data generating processes. Additionally, we applied the algorithm to stated choice data of the preferences for the environmental impact of (hypothetical) energy generation plans in Chile. The results show that the model-based decision tree fits the data better than MIXL in terms of information criteria. Moreover, we show that the derived tree structure depends on the assumptions on the parameters' distributions. Additionally, we compare the model-based decision tree model with Latent Class (LC) models with and without within-class heterogeneity. Finally, we show that the recursive partitioning algorithm can inform the selection of variables to be included in the LC allocation models.

1. Introduction

Since the introduction of the Multinomial Logit model (MNL) (McFadden, 1974), researchers have developed several model extensions to capture individuals' heterogeneous preferences. These efforts, mainly motivated by the bias and inconsistency generated when the assumption of homogeneous preferences across individuals does not hold (Chamberlain, 1980), have provided numerous extensions to the MNL model. The extensions are mainly triggered by the fact that including interaction terms between alternative-specific and individual-specific attributes is the only way to capture heterogeneity in MNL models. A popular extension is the Mixed Logit (MIXL) model (McFadden and Train, 2000), which captures the heterogeneity of preferences assuming a probability distribution on the model's parameters. The MIXL has been shown to be a powerful tool with substantial gains in terms of goodness of fit (Hensher and Greene, 2003). However, the interpretation of the heterogeneity is not immediately available to the researcher. This is mainly because allowing random parameters can show that the preferences are heterogeneous, but further steps, such as regression analysis on individual characteristics, are needed to gain insight into what might drive the heterogeneity captured by the model.

[*] Corresponding author.
E-mail addresses: alvaro.gutierrezvargas@kuleuven.be (Á.A. Gutiérrez-Vargas), michel.meulders@kuleuven.be (M. Meulders), martina.vandebroek@kuleuven.be (M. Vandebroek).

https://doi.org/10.1016/j.jocm.2022.100393
Received 15 October 2021; Received in revised form 17 September 2022; Accepted 14 November 2022
Available online 21 November 2022

Another major contribution to capture preference heterogeneity is the development of the Latent Class (LC) Model (Bhat, 1997; Train, 2008). Unlike the MIXL[1] model, it captures heterogeneity by assigning individuals into different classes. Each of the classes has different taste parameters, meaning that the model provides a discrete distribution of taste parameters across classes. The LC model uses an allocation model that can include characteristics of the individuals to compute the probability of belonging to a class. A common practice among practitioners is to label the resulting classes in terms of the allocation model, such as what observed characteristics would make an individual more likely to belong to a particular class. By doing so, researchers can characterize the taste parameters that an individual with given characteristics will most likely have.

Additionally, further attempts to combine the probabilistic classes created by LC models with the continuous random heterogeneity of MIXL has resulted in what we will refer to as the LC-MIXL model (Keane and Wasi, 2013). The LC-MIXL is, at its core, a LC model that allows random parameters for each of the classes. Keane and Wasi (2013), using ten different data sets, document that the LC-MIXL has superior model fit compared to LC and MIXL models. The authors attribute this to its capacity to capture a wide range of behavioral types present in the data, from lexicographic/non-compensatory choice behavior to *“random“* choice behavior, in the sense that the choices are little influenced by the observed attributes. Furthermore, the authors document the difficulty of using the LC-MIXL model in practice, arguing that not only the number of classes is unknown but also the distribution of the random parameters needs to be selected by the modeler, which leads to a large number of models that have to be fitted.

Following the same motivation as the above-mentioned models, this article proposes using the so-called MOdel Based Recursive Partitioning (MOB) algorithm (Zeileis et al., 2008) to capture individuals' preferences heterogeneity. The MOB algorithm generates partitions in the data based on structural tests of parameter stability. The basic idea of the parameter stability tests is to check whether the score functions of the model (i.e., the first derivative of the log-likelihood function) oscillate randomly around zero or if they exhibit systematic deviations generated by some variables, which are referred to as “partition variables”. That is to say, the stability tests analyze the influence of a given partition variable over the score functions of the model. Intuitively, if the scores at different values of the partition variable do not oscillate around zero (i.e. the theoretical mean value when evaluated at the maximum likelihood), the parameters' estimates are not stable across persons, and a data partition should be introduced. Originally, Zeileis et al. (2008) presented the algorithm with least-squares, logistic, and survival regression models, and later it was extended to models that incorporate random effects, for instance, the so-called generalized linear mixed-effects model tree (GLMM tree) algorithm (Fokkema et al., 2018) which uses the same battery of statistical tests proposed by Zeileis et al. (2008).

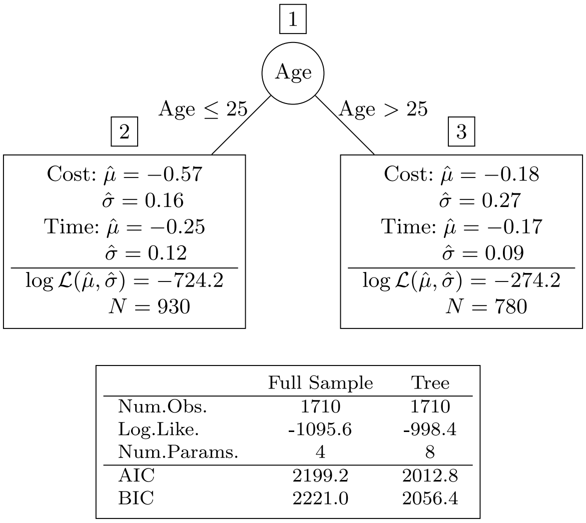
It is important to notice that the MOB algorithm, when used in a regression context, requires the user to specify which variables will be included in the parametric model at the end leaves and which variables will be used to partition the data. That being said, we will exploit the natural separation between alternative-specific attributes and individual-specific characteristics that arises in discrete choice applications. Concretely, we will use a MIXL model as the parametric model at the end leaves, including all the alternative-specific attributes, and we will use all the individual-specific characteristics to generate partitions on the data based on statistical tests of parameter stability. We will refer to the resulting model as the MOB-MIXL model. In this way, we expect our specification to benefit from the ability to automatically identify which individual-specific variables are relevant to create partitions on the data, which contrasts with the need to select the variables that will be included in latent class' allocation models. In the following, we refer to the partitions created by the MOB algorithm as *hard breaks* because they are deterministic. However, given their probabilistic nature, we refer to the segments created by LC models as *fuzzy breaks*. Additionally, from all the decision trees applied in the discrete choice literature (see Section 2.2), this is the first to allow for random coefficients in the parametric model fitted at the end leaves.

To briefly illustrate the potential of the MOB algorithm, Fig. 1 shows a hypothetical tree that has a MIXL model with two random coefficients (log-normally distributed). The Figure represents a hypothetical situation in which the algorithm used the individuals' age as a partition variable, dividing the sample between people older than 25 years old and younger than 25 years old. That is to say, the structural tests (see Section 4) rejected the null hypothesis of parameter stability and created a partition on the data based on the individual's age. Accordingly, the algorithm can improve the overall model fit (i.e., the Akaike Information Criterion (AIC) (Akaike, 1998) and the Bayesian Information Criteria (BIC) (Schwarz, 1978)) by estimating a separate model in each partition rather than having one global model fitted to the entire sample. Subsequently, the algorithm performed the same structural tests used in the global model (node $\boxed{1}$) at the end leaves (nodes $\boxed{2}$ and $\boxed{3}$), and it failed to reject the null hypothesis of parameter instability, hence the algorithm stopped. As illustrated, the algorithm can automatically detect different groups of individuals by partitioning the data based on their characteristics; that is, the algorithm does not require a pre-defined number of classes as LC and LC-MIXL models. Finally, since the algorithm is model agnostic, simpler models can be used as well in the end leaves, for instance, an MNL or a Nested Logit model. However, we use a MIXL model because of the considerable improvement in model fit derived from such models .

The contribution of our article is twofold. First, we carry out a simulation study to evaluate the MOB algorithm's performance when the parametric model at the end leaves consists of a MIXL model. We show the ability of the algorithm to recover several tree-like structures with hard breaks using simulated data. Second, we show a way to use the MOB algorithm as a diagnostic tool to select the variables to be included in the allocation model of LC models. Using simulated data, we show that the algorithm can discover the allocation model's variables associated with relevant variables in the true Data Generation Process (DGP).

This article is organized as follows. Section 2 presents a general literature review on decision trees, its applications in discrete choice modeling and some extensions to latent class models. Section 3 provides a brief description of the discrete choice models

[1] See Greene and Hensher (2003) for a detailed comparison of LC and MIXL models.


1
Age
Age ≤ 25
Age > 25
2
3
Cost: $\hat{\mu} = -0.57$
$\hat{\sigma} = 0.16$
Time: $\hat{\mu} = -0.25$
$\hat{\sigma} = 0.12$
$\log \mathcal{L}(\hat{\mu}, \hat{\sigma}) = -724.2$
$N = 930$
Cost: $\hat{\mu} = -0.18$
$\hat{\sigma} = 0.27$
Time: $\hat{\mu} = -0.17$
$\hat{\sigma} = 0.09$
$\log \mathcal{L}(\hat{\mu}, \hat{\sigma}) = -274.2$
$N = 780$

	Full Sample	Tree
Num.Obs.	1710	1710
Log.Like.	-1095.6	-998.4
Num.Params.	4	8
AIC	2199.2	2012.8
BIC	2221.0	2056.4

Fig. 1. Hypothetical decision tree resulting from using the MOB-MIXL algorithm and two log-normally distributed parameters.

we will use in this article. Section 4 introduces the MOB-MIXL algorithm. Section 5 presents two simulation studies. The first one shows that the MOB-MIXL can recover the true DGP from different tree-like with structures hard breaks. The second illustrates how a LC model will behave when in presence of data with hard breaks. Section 6 show a possible way to use the MOB algorithm as a variable selection step for latent classes' allocation models. Section 7 illustrates the usage of the algorithm on real data. Section 8 present a discussion that compares hard breaks with fuzzy breaks highlighting their advantages and disadvantages and presents the conclusions of the article.

2. Literature review

2.1. Decision trees: Overview

Decision Trees (DT) are classifiers that sequentially partition the data to generate a tree-like structure. There are substantial differences between different trees. For instance, algorithms like the Classification and Regression Trees (CART) (Breiman et al., 1984) and C4.5 (Quinlan, 1993), grow trees trying to create partitions that are as homogeneous as possible in terms of the dependent variable. In particular, they grow a large tree and then prune it using cross-validation to minimize the test error, that is to say, they try to maximize the accuracy of the predictions only. However, decision trees introduced later in the literature can include parametric models at the end leaves. For example, the Fast and Accurate Classification Tree (FACT) (Loh and Vanichsetakul, 1988), the Quick, Unbiased and Efficient Statistical Tree (QUEST) (Loh and Shih, 1997), the Generalized, Unbiased, Interaction Detection and Estimation (GUIDE) algorithm (Loh, 2002) and the Conditional Inference Trees algorithm (CTREE) (Hothorn et al., 2006) introduced statistical tests in the tree growing process. Additionally, the MOdel-Based (MOB) recursive partitioning algorithm (Zeileis et al., 2008), which is the one we use in this article, uses structural breaks of the fitted model's score functions. After the split variable is selected based on tests for parameter stability, the MOB algorithm selects the split point by maximizing the sum of log-likelihood functions in the emerging subgroups. However, unlike the GUIDE algorithm, the MOB algorithm requires to declare different variables to be used as splitting variables and as regressors (i.e., to be included in the parametric model defined at the end leaves). Here we exploit, in the context of discrete choice applications, the natural difference between individual characteristics and alternative specific variables using the former as partition variables and the latter as explanatory variables in the model. This is a reasonable assumption given that individual characteristics are very often used within class membership functions in LC models. Hence, we mimic the same behavior, yet creating hard breaks instead of probabilistic classes or fuzzy breaks.

2.2. Discrete choice applications of decision trees

There is an increasing interest in data-driven methods within the discrete choice community. A recent literature review by Hillel et al. (2021) reflects this, finding more than 70 articles that use Machine Learning techniques. For instance, the work of Karlaftis (2004) develops a multivariate recursive partitioning algorithm maximizing class purity using the Gini index (Breiman et al., 1984). The author shows that the predictive power of the proposed approach is higher than of conventional MNL models while obtaining a convenient series of "if-then" statements with the mode choice predictions. Similarly, Tang et al. (2015), and Liang et al. (2021) also use decision trees to model travel mode choice, showing that they outperform the predictive power of traditional logit models. However, they do not elaborate further on the interpretability of the models they present, reducing them to a purely predictive exercise. Using a more model-oriented approach, Arentze and Timmermans (2007) develop the so-called parametric action decision tree (PADT). This algorithm grows the tree in two steps, distinguishing between discrete and continuous attributes. PADT uses discrete attributes to produce the tree and continuous attributes to model utility at the end leaves for groups as homogeneous as possible in terms of the dependent variable. Another noteworthy attempt to bring together the predictive power of decision trees and discrete choice models is the work of Brathwaite et al. (2017). The authors use what they call a *"Bayesian model tree"* to model bicycle mode choice in the San Francisco Bay Area. They show that this tree can accommodate different types of non-compensatory behavior while automatically identifying the effect of bicycle infrastructure investment to be moderated by travel distance, topography, and individuals' socio-demographic characteristics.

In the same avenue, the MOB algorithm we use in this article can grow a decision tree that contains a parametric model at the end leaves. One noticeable advantage of the MOB-MIXL algorithm over the previous decision trees applied so far in the discrete choice literature is its ability to include random coefficients in the model specification at the end leaves which can drastically increase the model fit. Additionally, the only application of the MOB algorithm in the context of choice modeling was described by Cockx and Canters (2020), who implement it using revealed preference data for residential location choice in Belgium. The authors show that the model can identify heterogeneous preferences in residential location, where the main drivers of the different groups are the individuals' education level, nationality and household type, and the tenure status of the house. However, the authors use an MNL model as the parametric model at the end leaves, instead of the MIXL model we use. Additionally, the authors do not compare the performance of the MOB algorithm with other commonly used discrete choice models nor do they perform any simulations to assess the algorithm's performance.

2.3. Beyond latent class models in discrete choice applications

Given that the MOB-MIXL algorithm yields a structure that is somewhat comparable with LC and LC-MIXL models, in the sense that both extract groups of individuals with different taste parameters in the data, we review the latest advances in those models for the sake of completeness. Most recent advances in the LC literature have been devoted to implementing more flexible allocation models. For instance, Han (2019) implements the so-called nonlinear-Latent Class Choice Model (nonlinear-LCCM), which combines the traditional LC models with a neural network in the allocation model. The model seeks to find nonlinear relationships at the level of the allocation model so it can better learn the mixing distribution that allocates individuals into classes. The authors train the nonlinear-LCCM and conventional LC models as a neural network using Stochastic Gradient Descent (SGD) methods which they found to perform better than using conventional Expectation–Maximization (EM) or Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithms in several aspects. First, it scales better for large data sets in terms of running time. Second, it produces more stable results under complex circumstances (large number of classes and parameters). Third, it performs better at identifying smaller classes present in the data.

Sfeir et al. (2021), in the same spirit of Han (2019), propose an LC model that replaces the conventional Logit specification used in the allocation model with a Gaussian–Bernoulli mixture model. The proposed model distinguishes between binary and continuous individual characteristics used in the allocation model. The authors show that their model outperforms traditional LC models in terms of model fit in two data sets without interpretability losses producing logical economic indicators (e.g., willingness to pay). Unfortunately, the authors do not investigate how the model would behave when introducing within-class heterogeneity, as in LC-MIXL models, and leave this open for future research.

To conclude, the significant advances in recent articles devoted to extensions of the LC model have been novel and have lead to more complex structures that replace the Logit model traditionally used as an allocation model. However, none of the articles have investigated within-class heterogeneity, which the MOB algorithm can do by using a MIXL as the parametric model at the end leaves. The algorithm used in this paper can be understood roughly as a LC-MIXL model with an allocation model that works in a deterministic way, creating non-overlapping data partitions with different taste parameters based on individual characteristics. Consequently, even though this way of proceeding is less flexible than a probabilistic allocation model, it is more straightforward to interpret than LC and LC-MIXL models. Additionally, we will also show how the MOB algorithm can be used as a variable selection procedure (see Section 6) when choosing the LC or LC-MIXL allocation models

3. Discrete choice models

This section describes the notation we will use for the discrete choice models that we have mentioned: the MNL, LC, MIXL and LC-MIXL models. We use this notation to introduce the MOB algorithm in Section 4.

3.1. The multinomial logit model

Consider a situation in which we have a sample of N decision-makers, and that respondent n can choose among J alternatives in each of t_n choice situations. Let the utility that this individual obtains from alternative i in choice set t be described by $U_{int} = \boldsymbol{\beta}'\boldsymbol{x}_{int} + \boldsymbol{z}_n'\boldsymbol{\alpha}\boldsymbol{x}_{int} + \varepsilon_{int}$, where $\boldsymbol{\beta}$ is a $K \times 1$ column vector of coefficients, $\boldsymbol{x}_{int}$ is a $K \times 1$ column vector characterizing the attribute levels of alternative i in choice set t for respondent n, $\boldsymbol{\alpha}$ is a $K_z \times K$ matrix of interaction coefficients, $\boldsymbol{z}_n$ is a $K_z \times 1$ column vector of individual characteristics of individual n, and ε_{int} is an error term that represents the unobserved component of the utility. The error term is assumed to be an independent and identically distributed type I extreme value. Additionally, we assume that individuals choose the alternative with maximum utility. Under these standard assumptions, the probability that individual n chooses alternative i in choice situation t can be expressed by a Multinomial Logit (MNL) formula (McFadden, 1974):

$$P_{int} = \frac{\exp\left(\boldsymbol{\beta}'\boldsymbol{x}_{int} + \boldsymbol{z}_n'\boldsymbol{\alpha}\boldsymbol{x}_{int}\right)}{\sum_{j=1}^{J}\exp\left(\boldsymbol{\beta}'\boldsymbol{x}_{jnt} + \boldsymbol{z}_n'\boldsymbol{\alpha}\boldsymbol{x}_{jnt}\right)}. \tag{1}$$

Using (1) we can define the sample log-likelihood function of the MNL model as

$$LL(\boldsymbol{\beta}) = \sum_{n=1}^{N}\sum_{t=1}^{t_n}\sum_{i=1}^{J} y_{int} \times \ln\left(P_{int}\right), \tag{2}$$

where y_{int} is the response variable that is one if individual n chooses alternative i in choice situation t and zero otherwise.

The MNL is the most simple and easy to interpret discrete choice model. However, it cannot exploit the panel structure produced by a series of choices made by the same individual (i.e., it treats choices made in different choice situations as independent) and the heterogeneity is limited to the inclusion of interaction terms with individual characteristic variables. The following Sections describe discrete choice models that extend the MNL to overcome these limitations.

3.2. The latent class model

The LC Model (Bhat, 1997; Train, 2008) assumes that there are C distinct classes of individuals with different taste parameters present in the data $\left(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_C\right)$. Accordingly, if individual n belongs to class c, the probability of observing the sequence of choices of individual n is a product of conditional logit formulas given by

$$P_n(\boldsymbol{\beta}_c) = \prod_{t=1}^{t_n}\prod_{i=1}^{J}\left\{\frac{\exp\left(\boldsymbol{\beta}_c'\,\boldsymbol{x}_{int}\right)}{\sum_{j=1}^{J}\exp\left(\boldsymbol{\beta}_c'\,\boldsymbol{x}_{jnt}\right)}\right\}^{y_{int}}, \tag{3}$$

with $\boldsymbol{\beta}_c$ being the vector of preference parameters in class c ($c = 1, \dots, C$). Given that the class membership is unknown, the econometrician needs to specify the unconditional probability, π_{cn}, of an individual n belonging to class c, which is typically described using a Logit function that can include individual characteristics ($\boldsymbol{z}_n$), such as:

$$\pi_{cn}(\boldsymbol{\gamma}_c, \lambda_c) = \frac{\exp\left(\lambda_c + \boldsymbol{\gamma}_c'\,\boldsymbol{z}_n\right)}{\sum_{l=1}^{C}\exp\left(\lambda_l + \boldsymbol{\gamma}_l'\,\boldsymbol{z}_n\right)}, \tag{4}$$

where $\boldsymbol{\gamma}_c$ is the parameter vector of the allocation model and λ_c is a constant term related to class c. The parameters $\boldsymbol{\gamma}_c$ capture the influence of the vector of individual characteristics, $\boldsymbol{z}_n$, on the class allocation probabilities. Additionally, if no information about individual characteristics is available, the allocation model can be a constant-only model, meaning only including λ_c parameters. For identification we set the parameters from one class (e.g.; $\boldsymbol{\gamma}_1$, λ_1) equal to zero.

Accordingly, we can define the sample log-likelihood function of the LC model in terms of Eqs. (3) and (4) as

$$LL(\boldsymbol{\beta}, \boldsymbol{\gamma}, \lambda) = \sum_{n=1}^{N}\ln\sum_{c=1}^{C}\pi_{cn}(\boldsymbol{\gamma}_c, \lambda_c) \times P_n\left(\boldsymbol{\beta}_c\right). \tag{5}$$

We can see that the LC model can specify heterogeneous preferences across classes by splitting individuals into probabilistic profiles based on their individual-specific characteristics.

3.3. The mixed logit model

The MIXL model (McFadden and Train, 2000), similarly to the models we have already presented, works with a sample of N individuals with t_n choice sets of J alternatives. The difference is to be found in the utility that individual n derives from choosing alternative i on choice situation t, which now is given by $U_{int} = \boldsymbol{\beta}_n'\boldsymbol{x}_{int} + \boldsymbol{z}_n'\boldsymbol{\alpha}\boldsymbol{x}_{int} + \varepsilon_{int}$, where $\boldsymbol{\beta}_n$ is a vector of individual-specific coefficients. Different from the MNL model, where the parameter $\boldsymbol{\beta}$ is assumed to be the same for all individuals, the MIXL assumes that $\boldsymbol{\beta}$ follows a density denoted as $f\left(\boldsymbol{\beta}|\boldsymbol{\varphi}\right)$, where $\boldsymbol{\varphi}$ are the parameters of the distribution, for example, if a normal distribution is assumed, its mean and variance. Accordingly, the full set of parameters of the model will be denoted by $\boldsymbol{\theta}$.

Conditional on knowing $\boldsymbol{\beta}_n$ for each individual, the probability that respondent n chooses alternative i on choice situation t is given by

$$P_{int}(\boldsymbol{\beta}_n, \boldsymbol{\alpha}) = \frac{\exp\left(\boldsymbol{\beta}_n'\boldsymbol{x}_{int} + \boldsymbol{z}_n'\boldsymbol{\alpha}\boldsymbol{x}_{int}\right)}{\sum_{j=1}^{J}\exp\left(\boldsymbol{\beta}_n'\boldsymbol{x}_{jnt} + \boldsymbol{z}_n'\boldsymbol{\alpha}\boldsymbol{x}_{jnt}\right)}, \tag{6}$$

which is almost identical to the formula of the MNL model (see Eq. (1)). The only difference is that now we have one vector of taste parameters $\boldsymbol{\beta}_n$ for each individual n. Additionally, the probability of the observed sequence of choices of individual n (conditional on knowing $\boldsymbol{\beta}_n$) is given by

$$P_n(\boldsymbol{\beta}_n) = \prod_{t=1}^{t_n}\prod_{i=1}^{J}\left\{P_{int}(\boldsymbol{\beta}_n)\right\}^{y_{int}}. \tag{7}$$

The unconditional probability of the observed sequence of choices is the conditional probability integrated over the entire domain of the distribution of $\boldsymbol{\beta}$. Accordingly, the sample log-likelihood function for the MIXL model is given by

$$LL(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln \left[\int_{\boldsymbol{\beta}} P_n(\boldsymbol{\beta}) f(\boldsymbol{\beta}|\boldsymbol{\varphi}) \, d\boldsymbol{\beta} \right]. \tag{8}$$

As the integral in Eq. (8) does not have a closed form, it is approximated using simulation (see Train (2009)). Accordingly, in what follows we estimate the model using Maximum Simulated Likelihood where we maximize the following simulated log-likelihood

$$SLL(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{R} \sum_{r=1}^{R} P_n(\boldsymbol{\beta}^r) \right\}, \tag{9}$$

where R is the number of replications and $\boldsymbol{\beta}^r$ is the rth drawn from $f(\boldsymbol{\beta}|\boldsymbol{\varphi})$.

3.4. Latent class model with random coefficients

The LC-MIXL model (Keane and Wasi, 2013) is at its core, an LC model that allows for within-class continuous heterogeneity, so we will have different classes denoted by c ($c = 1, \dots, C$) where inside of each class we have individual-level parameters specific for each class, $\boldsymbol{\beta}_{n|c}$. Hence, when individual n belongs to class c, the probability of observing its sequence of choices will be the product of conditional Logit formulas (conditional on knowing $\boldsymbol{\beta}_{n|c}$) given by:

$$P_n(\boldsymbol{\beta}_c) = \prod_{t=1}^{t_n} \prod_{i=1}^{J} \left\{ \frac{\exp\left(\boldsymbol{\beta}'_{n|c} \boldsymbol{x}_{int}\right)}{\sum_{j=1}^{J} \exp\left(\boldsymbol{\beta}'_{n|c} \boldsymbol{x}_{jnt}\right)} \right\}^{y_{int}}, \tag{10}$$

In the same fashion as in the LC model, we can use an allocation model that is a function of individual characteristics $\boldsymbol{z}_n$ (see Eq. (4)). However, given that we allow for random heterogeneity within classes, we need to integrate the conditional probability of the observed sequence of choices from individual n in class c over the entire domain of the distribution of $\boldsymbol{\beta}_{n|c}$ to obtain the unconditional choice probability. Accordingly, we can define the log-likelihood of the model using Eq. (10) and (4) as:

$$LL(\boldsymbol{\beta}, \boldsymbol{\gamma}, \lambda) = \sum_{n=1}^{N} \ln \sum_{c=1}^{C} \pi_{cn}(\boldsymbol{\gamma}_c, \lambda_c) \times \left[\int_{\boldsymbol{\beta}} P_n\left(\boldsymbol{\beta}_c\right) f\left(\boldsymbol{\beta}_c|\boldsymbol{\varphi}\right) d\boldsymbol{\beta}_c \right] \tag{11}$$

The log-likelihood function presented in Eq. (11) is also estimated using simulations, as for the case of the MIXL model, taking draws from the assumed distribution of the random coefficients. Accordingly, we will maximize the following simulated log-likelihood function

$$SLL(\boldsymbol{\beta}, \boldsymbol{\gamma}, \lambda) = \sum_{n=1}^{N} \ln \sum_{c=1}^{C} \pi_{cn}(\boldsymbol{\gamma}_c, \lambda_c) \times \left[\frac{1}{R} \sum_{r=1}^{R} P_n\left(\boldsymbol{\beta}_c^r\right) \right], \tag{12}$$

where R is the number of replications and $\boldsymbol{\beta}^r$ is the rth drawn from $f(\boldsymbol{\beta}_c|\boldsymbol{\varphi})$. The described model can create an arbitrary number of probabilistic profiles of individuals while being able to accommodate continuous random heterogeneity for the parameters inside of each of the classes while leads to the most flexible model of all presented in this article.

4. Parameter instability tests

4.1. The MOB-MIXL algorithm

The MOB algorithm (Zeileis et al., 2008) proposes the idea of recursive partitioning, which comes from the insight that, in some situations, it is unreasonable to assume that a single global model can fit all observations in the sample sufficiently well. Instead, it might be the case that partitions of the sample space with respect to a partition variable allow for a better local specification. The MOB algorithm generates partitions on the data based on a statistical test of parameter stability rather than using purity measures as, for example, the CART (Breiman et al., 1984) or the C4.5 (Quinlan, 1993) algorithms. Concretely, we use the MOB algorithm to divide the sample into subgroups and fit a separate MIXL model at these end leaves leading to the MOB-MIXL model. So we propose the MOB-MIXL algorithm which contains the following steps:

1. Fit a MIXL model once to all observations in the current node.
2. Assess whether the taste parameters are stable with respect to demographic variables. If there is parameter instability, select the demographic variable, Z_s, associated with the highest parameter instability; otherwise, stop.
3. Compute the split point that maximizes the sum of the simulated log-likelihood functions (see Eq. (9)) over the emerging subgroups.
4. Split the node into child nodes and repeat the procedure until some stopping criterion is met.

We briefly explain the details of the stability tests of step 2 in Section 4.2, but a detailed explanation of such tests can be found in Zeileis et al. (2008). Additionally, the algorithm allows the user to specify the minimum number of observations necessary to create a partition and the significance level of the test for parameter instabilities (the default being 0.05). That is to say that the algorithm can stop primarily for two reasons. First, the stability test (step 2) performed at a given node of the tree fails to reject the null hypothesis of parameter stability. Second, the partition to be created does not reach the required minimum number of observations specified by the user. Once the algorithm has stopped, the resulting tree can be pruned in order to maximize model fit based on either Akaike Information Criterion (AIC) (Akaike, 1998) or the Bayesian Information Criteria (BIC). Further details of the pruning process can be found in Hothorn and Zeileis (2015).

4.2. The empirical fluctuation process and the stability tests

In this Section we describe the *empirical fluctuation process*, which is used later in the stability tests of the MOB algorithm (step 2 in Section 4.1). For further details, and a general description of the empirical fluctuation process, we refer readers to Zeileis et al. (2008) and Merkle et al. (2014). In Zeileis et al. (2008) the authors stated that *"to assess parameter instabilities, a natural idea is to check whether the score functions, fluctuate randomly around their mean zero, or exhibit systematic deviation from zero over Z_p"* (Zeileis et al., 2008, p. 496), where Z_p is a possible partition variable. Accordingly, the stated deviations of the score functions can be captured by the so-called *empirical fluctuation process*.

In order to define the *empirical fluctuation process* we need to compute the score functions of the model. Given that, in the MIXL model, the analytical expression of the score functions depend on the parametric distribution of the random coefficients,[2] we refer to it, without loss of generality, as:

$$\frac{\partial \ln L_n(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \frac{1}{L_n(\boldsymbol{\theta})} \times \frac{\partial L_n(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \hat{\boldsymbol{\psi}}_n. \tag{13}$$

where $\hat{\boldsymbol{\psi}}_n$ is a $t_n \times K^*$ matrix of score functions related to the t_n choice situations answered by the individual n and K^* represents the length of the full vector of parameters, $\boldsymbol{\theta}$, included in the model specification. Additionally, we can define the matrix $\hat{\boldsymbol{\psi}}$ which stacks the score functions of the individuals into a $T \times K^*$ matrix, where $T = \sum_{n=1}^{N} t_n$, represents the total number of choice situations on the data. Besides, we refer to the choice situation s of the matrix $\hat{\boldsymbol{\psi}}$, independent of the individual who answered it, as $\hat{\boldsymbol{\psi}}_s$ which is a $1 \times K^*$ vector. Using $\hat{\boldsymbol{\psi}}_s$ we will construct the statistical tests of parameter stability as in Zeileis et al. (2008). To do so, we will define the *empirical fluctuation process* of partition variable Z_p as $\boldsymbol{W}^{(p)}(s^*)$ in Eq. (14) where $\hat{\boldsymbol{\psi}}_{s|z_p}$ represents a reordering of the rows of the matrix $\hat{\boldsymbol{\psi}}$ based on the ordering of variable $Z_p = \left(z_{p1}, \dots, z_{pT}\right)'$. Additionally, s^* ranges from 1 to T and represents the number of choice sets included in the cumulative sum, and $\hat{\boldsymbol{J}}$ is the estimated variance–covariance matrix.[3] This results in T row vectors of length K^* which are stacked together resulting in the $T \times K^*$ matrix $\boldsymbol{W}^{(p)}$ with elements $w_{sk}^{(p)}$:

$$\boldsymbol{W}^{(p)}(s^*) = \left(\sum_{s=1}^{s^*} \hat{\boldsymbol{\psi}}_{s|z_p}\right) T^{-1/2} \hat{\boldsymbol{J}}^{-1/2} \qquad s^* = 1, \dots, T \tag{14}$$

Using properties of empirical fluctuation processes and Brownian bridges, Zeileis and Hornik (2007) show that it is possible to do statistical inference about functions of $\boldsymbol{W}^{(p)}$ to check for parameters' instabilities. Appendix A illustrates the relation between the empirical fluctuation process and the score functions for a simple model with one quantitative attribute. In order to compute a formal test we need to apply a scalar function $\phi(\cdot)$ over the *empirical fluctuation process* that captures the instabilities over the partition variable Z_s. The applied scalar function differs based on the nature of the partition variable (i.e., continuous, categorical or ordinal). If the partition variable is continuous (e.g., age, income, etc.), Zeileis et al. (2008) propose that a natural test statistic is

$$\phi_{\sup LM}(\boldsymbol{W}^{(p)}) = \max_{s=\underline{s},\dots,\bar{s}} \left(\frac{s}{T} \times \frac{T-s}{T}\right)^{-1} \sum_{k=1}^{K^*} \left[w_{sk}^{(p)}\right]^2. \tag{15}$$

The $\phi_{\sup\ \mathrm{LM}}(\boldsymbol{W}^{(p)})$ is the $\sup LM$ statistic of Andrews (1993), which is asymptotically equivalent to the so-called Chow test (Chow, 1960). The test is specified in terms of the *empirical fluctuation process* and it is typically defined by requiring some minimal number of choice sets $\underline{s}$ (therefore $\bar{s} = T - \underline{s}$). The test computes the maximum sum of squared elements of the $\boldsymbol{W}^{(p)}$ matrix's rows scaled by its variance function.[4] The $\sup LM$ statistic has the advantage that it has to be fitted only once under the null hypothesis and not in the alternative of each possible breakpoint. Additionally, the limiting distribution is the supremum of a tied-down Bessel process, and the p-values can be computed as stated in Hansen (1997). Similar tests where proposed to deal with non-ordered categorical and ordered categorical variables which we present briefly in Appendix B. See Merkle et al. (2014) and Zeileis et al. (2008) for a full description of said statistical tests.

[2] For an exhaustive derivation of the analytical expressions of the score functions under normality assumptions we refer the readers to the Appendix A.1 in Zhang et al. (2017).

[3] In our case, when repeated choice situations are answered by the same individual, we use cluster-corrected variance–covariance matrices, $\hat{\boldsymbol{J}}$, to account for the dependence between choice sets answered by the same individual.

[4] This expression for the variance is taken from the asymptotic distribution of $\boldsymbol{W}^{(p)}$, which, under the null hypothesis of parameter stability, converges to Brownian bridge.

It is worth mentioning that in the presence of multiple choices ($t_n > 1$) per individual, if a partition is made, all choice sets of individual n will move as a "block" when growing the tree. This behavior occurs because, once we reject the null hypothesis of parameter stability for a given partition variable (step 2 in Section 4.1), the algorithm finds the split point by running a grid search across all possible values of the partition variable and by maximizing the sum of the (simulated) log-likelihood function over the emerging subgroups (see step 3 in Section 4.1). Accordingly, given that all the choice sets answered by the same individual have the same individual characteristics, they are assigned in "blocks" to the end leaves of the tree. Finally, the quality of the approximation of log-likelihood and the score functions, that is to say, the number of draws used in the estimation, will impact the structural tests used by the MOB algorithm, so the larger this number, the more powerful the test will be. We investigate this behavior in the empirical application finding that a different number of draws result in different tree structures.

4.3. Software details

We run all the computations using `R` (R Core Team, 2022) on a Windows 10 machine with an AMD EPYC 7552 48-Core Processor (2.20 GHz) with 256gb of RAM. Besides, we implement the MOB algorithm using the `R` package `partykit` (version 1.2.13) (Hothorn and Zeileis, 2015) together with `mlogit` (version 1.1.1) (Croissant, 2020) which is used to fit a MIXL model at the end leaves. The `mlogit` package also produces the score functions used by the structural tests performed by the MOB algorithm. Additionally, we used the `R` package `Apollo` (version 0.2.7) (Hess and Palma, 2019) to fit the LC and LC-MIXL models. Finally, all the figures were created using the `ggplot2` package (Wickham, 2016). Finally, the code that replicates the results presented in this article is available at https://github.com/alvarogutyerrez/mobmixl.

5. Simulation studies

5.1. The MOB-MIXL algorithm applied to data with hard breaks

To show the potential of the MOB-MIXL algorithm we present a large simulation study. In particular, we investigate how the algorithm's performance depends on the size of the data, the size of the parameter differences across the partitions, and how balanced the groups are at the end leaves. The DGP follows closely the scheme used by Schlosser et al. (2019) in the context of linear regression, but is adapted for discrete choice models. The discrete choice sets consist of three alternatives ($i = 1, 2, 3$) with two alternative-specific attributes ($\boldsymbol{x}_{int}$) that describe the observed utility (V_{in}) of each alternative. We simulated different tree-like structures with hard breaks on the taste parameters based on individual-specific characteristics (Z_1 to Z_5). The full description of the DGP and the assumed model for each scenario is available in Table 1.

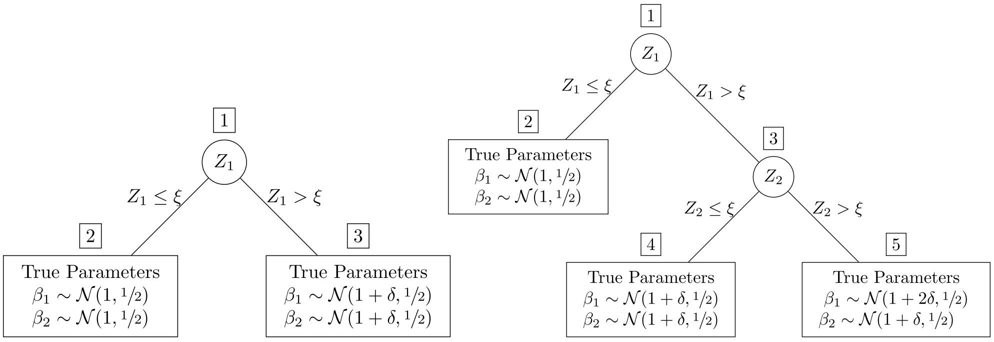
Table 1
Simulation setup.

Name	Notation	Specification
Variables:		
Random Utility	U_{int}	$= V_{int} + \varepsilon_{int}$
Deterministic Utility	V_{int}	$= \beta_1(Z_1, Z_2) \cdot \boldsymbol{x}_{1nt} + \beta_2(Z_1, Z_2) \cdot \boldsymbol{x}_{2nt}$
Alternative Specific Attribute	$\boldsymbol{x}_{1nt}, \boldsymbol{x}_{2nt}$	$\mathcal{U}([-2, 2])$
Error	ε_{int}	Gumbel distribution
Individual Characteristics with Split	Z_1 (or Z_2)	$\mathcal{U}([0, 1])$
Individual Characteristics without Split	Z_2 (or Z_3) - Z_5	$\mathcal{U}([0, 1])$
Parameters		
Number of individuals	N	$\in \{250, 500\}$
Number of choice sets per individual	t_n	$\in \{6, 12\}$
Taste Parameter 1	β_1	$\mathcal{N}(1, {}^1\!/_2)$ or $\mathcal{N}(1+\delta, {}^1\!/_2)$
Taste Parameter 2	β_2	$\mathcal{N}(1, {}^1\!/_2)$ or $\mathcal{N}(1+\delta, {}^1\!/_2)$ or $\mathcal{N}(1+2\delta, {}^1\!/_2)$
True split point	ξ	$\in \{0.5, 0.8\}$
Effect size	δ	$\in \{0.5, 1\}$

From Table 1, we can see that we are modeling the taste parameters (β_1 and β_2) as a function of individual characteristics (i.e., Z_1 or/and Z_2) of the individuals. Additionally, we vary the mean of the random parameter among groups with δ and the proportion of each group at the end leaves with ξ. Given that all the partition variables have a uniform distribution, ξ equal to 0.5 means that the groups are balanced. Conversely, if ξ is equal to 0.8, the groups are unbalanced at the end leaves.

The first scenario consists of the so-called *"stump"* scenario, where both taste parameters have only one split based on variable Z_1 as described in Fig. 2(a). The second scenario consists of a so-called *"tree"* scenario, where both taste parameters have one split based on the individual-specific variables illustrated in Fig. 2(b).

We evaluate the MOB-MIXL algorithm's performance using 50 different simulated data sets for each of the different combinations of the number of individuals (N), choice sets answered per individual (t_n), the differences of the mean of the random parameters (δ) and the proportions of each group (ξ), on two different scenarios, namely the *"stump"* and *"tree"*. We only include the alternative-specific attributes for the utility specification at the end leaves of the tree. We drop the interaction terms from Eq. (6) because we allow the individual-characteristics variables to act only as partition variables when growing the decision tree.


1
Z_1
$Z_1 \leq \xi$
$Z_1 > \xi$
2
True Parameters
$\beta_1 \sim \mathcal{N}(1, 1/2)$
$\beta_2 \sim \mathcal{N}(1, 1/2)$
3
True Parameters
$\beta_1 \sim \mathcal{N}(1+\delta, 1/2)$
$\beta_2 \sim \mathcal{N}(1+\delta, 1/2)$
1
Z_1
$Z_1 \leq \xi$
$Z_1 > \xi$
2
True Parameters
$\beta_1 \sim \mathcal{N}(1, 1/2)$
$\beta_2 \sim \mathcal{N}(1, 1/2)$
3
Z_2
$Z_2 \leq \xi$
$Z_2 > \xi$
4
True Parameters
$\beta_1 \sim \mathcal{N}(1+\delta, 1/2)$
$\beta_2 \sim \mathcal{N}(1+\delta, 1/2)$
5
True Parameters
$\beta_1 \sim \mathcal{N}(1+2\delta, 1/2)$
$\beta_2 \sim \mathcal{N}(1+\delta, 1/2)$

(a) "Stump" Scenario (b) "Tree" Scenario

Fig. 2. Graphical representation of the possible data generating process (DGP). Fig. 2(a) represents the *"stump"* scenario, where only one split is present. Fig. 2(b) represents the *"tree"* scenario where two splits are present.

Table 2
Illustration of partitions A and B over a set of N individuals.

A	B				Sums
	B_1	B_1	$\cdots$	B_q	
A_1	n_{11}	n_{12}	$\cdots$	n_{1q}	a_1
A_2	n_{21}	n_{22}	$\cdots$	n_{2q}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
A_r	n_{r1}	n_{r2}	$\cdots$	n_{rq}	a_r
Sums	b_1	b_2	$\cdots$	b_q	N

5.1.1. Evaluation criteria

We measure the performance of the algorithm in terms of two different criteria. First, its ability to correctly recover the underlying data structure, that is to say, that the estimated tree corresponds to the true DGP process. Second, its ability to correctly recover the true values of the parameters for each of the partitions.

Data Structure Recovery: In terms of the model's ability to recover the underlying data structure, we use different metrics depending on the underlying data-generating process. When recovering the *"stump"* scenario, following Schlosser et al. (2019), we use as a performance metric the proportion of the 50 different simulated data sets for which the selected splitting variable was the correct one (i.e., Z_1) among all five possible splitting variables. This proportion is further denoted as the *"selection probability"*.

On the other hand, to assess the performance for the *"tree"* scenario, we use the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) to evaluate the degree of similarity between the real tree structure and the estimated tree. We will define the ARI in terms of a given set $\boldsymbol{G}$ of N elements (individuals), and two partitions, namely $A = \{A_1, A_2, \dots, A_r\}$ and $B = \{B_1, B_2, \dots, B_q\}$. The overlap between partitions A and B can be summarized in a contingency table, where each entry n_{ij} denotes $n_{ij} = \left|A_i \cap B_j\right|$:

In our case, we have N individuals, A indicates the partitions produced by the true DGP, and B the ones produced by the estimated tree. Consequently, the ARI is defined in terms of the contingency table in Table 2 as

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{N}{2}}. \tag{16}$$

Parameter Recovery: To assess the ability to recover the true parameters, we compute the Mean Absolute Error (MAE) across all individuals, defined as

$$\text{MAE}(\hat{\beta}_k) = \frac{1}{N}\sum_{n=1}^{N}\left|\beta_{kn} - \hat{\beta}_{kn}\right|, \tag{17}$$

where the β_{kn} represents the true coefficient of attribute k for individual n and $\hat{\beta}_{kn}$ the estimated value of attribute k for individual n.

5.1.2. Results

We start by looking at the model's ability to recover the data structure produced by the *"stump"* scenario. Fig. 3 presents the *"selection probability"* of the correct splitting variable on the first node, namely Z_1. As expected, the metric increases with the number of individuals and the number of choice situations. In addition, we can see that when groups are balanced ($\xi = 0.5$), and the size of the parameter differences is large ($\delta = 1$), regardless of the number of choice situations per individual, the selection probability is virtually equal to 1. On the other hand, when groups are unbalanced ($\xi = 0.8$), and the parameter differences are smaller ($\delta = 0.5$), the mean selection probability goes from 60% to more than 80% when increasing the number of individuals from 250 to 500, which are moderately small sample sizes. This is evidence that the algorithm can correctly identify the true partition variable Z_1 among the other possible partition variables ($Z_2 - Z_5$).

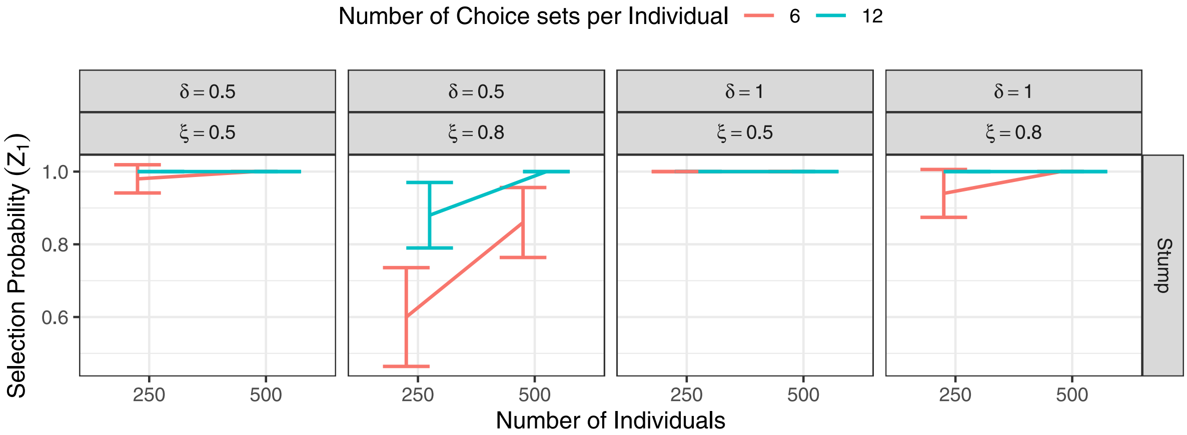


Fig. 3. "Selection probability" of the correct splitting variable (Z_1) resulting from 50 runs of the MOB-MIXL algorithm for different values of δ, ξ, N and t_n. Presented 95% confidence intervals were computed using a population proportion formula, namely $p = \bar{p} \pm z_{2.5\%}\sqrt{(\bar{p}(1-\bar{p}))/N}$.

In the so-called *"tree"* scenario, we used the ARI to assess the model's ability to capture the underlying data structure. Fig. 4 presents box plots of the ARI resulting from the simulations. The results are very similar to the ones obtained in the *"stump"* scenario. In particular, we can see that for large parameter differences ($\delta = 1$) and balanced groups ($\xi = 0.5$), the ARI goes quickly towards one even when only six choice situations were answered per individual in a reduced sample size of 250 individuals. On the contrary, for unbalanced groups ($\xi = 0.8$) and smaller parameter differences ($\delta = 0.5$), we see a slower convergence towards one, observing that only for 500 individuals answering 12 choice situations the ARI values are very close to one.

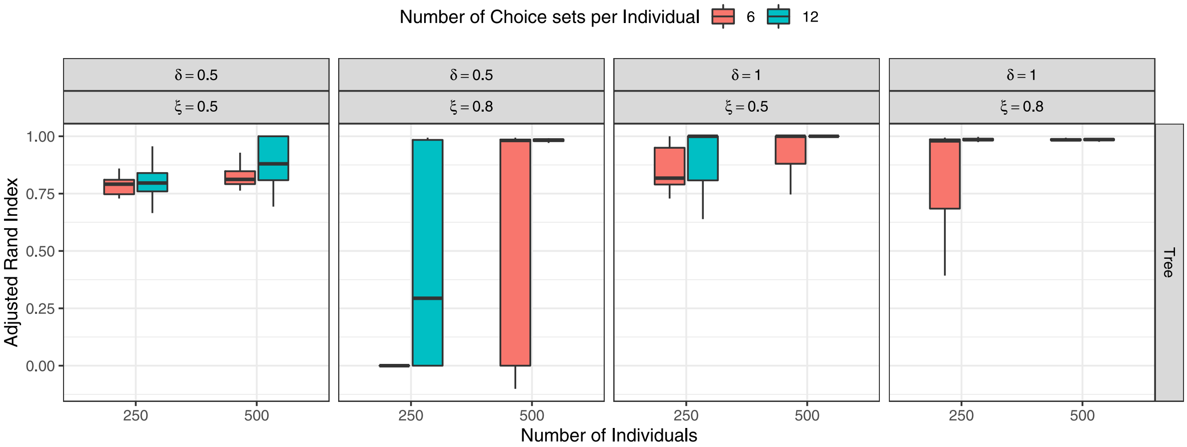


Fig. 4. Box-plot of the Adjusted Rand Index (ARI) resulting from 500 runs of the MOB-MIXL algorithm at different number of individuals (N), choice situations (t_n), size of parameter differences (δ) and proportion of each group which is controlled by the value of ξ. The solid horizontal line connects the ARI's mean for each configuration.

We present the box-plots of the parameter estimates' MAE for $\hat{\beta}_1$, in Fig. 5. Results for the estimates of $\hat{\beta}_2$ are omitted as they are almost identical to $\hat{\beta}_1$. The results are in line with the aforementioned capacity to recover the underlying tree structure, meaning that the MAE values go towards zero when increasing the number of individuals and choice sets.

Finally, we can see from this simulation that, based on our setup, the algorithm is suitable primarily for experimental designs with several choice situations per individual. In particular, we observe that even for moderately reduced sample sizes the algorithm is able to recover the true DGP of two different tree-like structures with hard breaks.

5.2. The LC-MIXL model fitted to data with hard breaks

This section illustrates how an LC-MIXL model will behave when the assumption of probabilistic profiles (or fuzzy breaks) does not hold and, instead, the data presents hard breaks. To do so, we simulate two data sets using the *"tree"* scenario described in

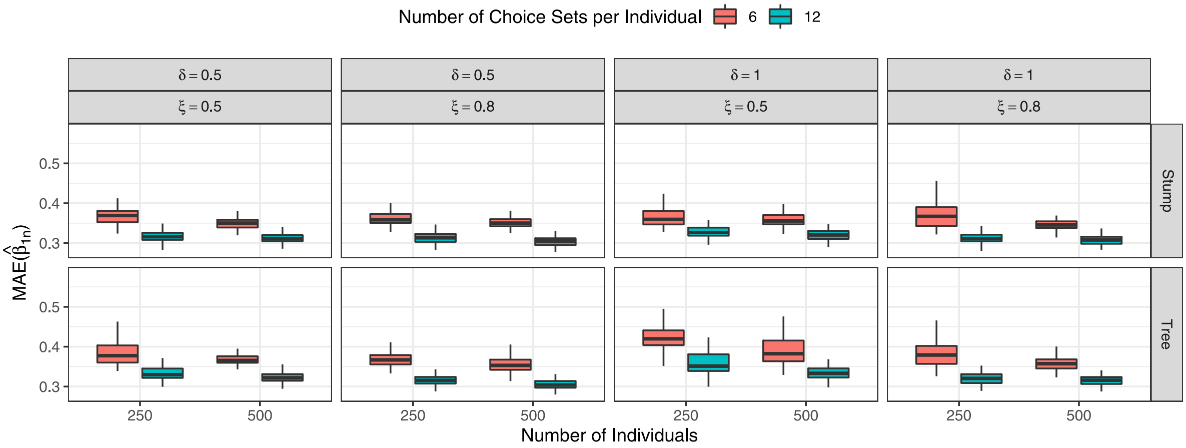


Fig. 5. Box-plot of the Mean Absolute Error (MAE) for $\hat{\beta}_{1n}$ resulting from 50 runs of the MOB algorithm at different number of individuals (N), choice sets per individual (T), size of parameter differences (δ) and proportion of each groups (ξ).

Section 5.1. Given that only two of the simulated individual characteristics are the partitions' drivers, we expect only the parameters associated with those variables to be statistically significant from zero in the allocation model.

The first data set is simulated using $\delta = 0.5$, representing a DGP where there is some overlap among the distributions of the parameters of each group. That is to say, the different groups present in the data are relatively similar. On the other hand, the second data set is simulated using $\delta = 1$, creating less overlap among the parameters of each group and, consequently, the groups have rather different taste parameters. Both simulated samples use values of $\xi = 0.5$, indicating that the hard breaks divide the groups at the end leaves, roughly, in half (see Table 1). We simulate a data set that consists of 1500 individuals answering 10 choice sets each. We fitted a three-class LC-MIXL model with an allocation model that includes all the individual-specific characteristics (Z_1 to Z_5) and normally distributed parameters for each attribute to this data. We estimated the models 30 times for each data set using random starting values,[5] and 5000 Halton draws to approximate the integral of Eq. (11).

Table 3 presents the models with the best log-likelihood value, among the 30 different starting values, for each data set. We observe that regardless of the value of δ, the true values (see Fig. 2(b)) of the parameters' distributions are recovered fairly accurately. Additionally, the size of the classes ($\bar{\pi}_c$ with $c \in (1, 2, 3)$) is also very close to the true proportion of 50% for the larger class, and 25% for each of the smaller ones. However, we observe that parameters from the allocation model are exploding, with large point estimate values and standard errors. Surprisingly, almost all the parameters in the allocation model seem statistically significant for at least one of the classes, which goes against the DGP. However, we observe that the larger point estimates correspond to the variables that generated the hard breaks (Z_1 and Z_2). In summary, the only clear sign that the DGP might contain sharp breaks is the explosive behavior in the allocation model. Accordingly, from an applied perspective, it might be worth investigating the MOB algorithm as an alternative to LC or LC-MIXL models when in the presence of said behavior.

6. The MOB algorithm as a variable selection for the LC allocation model

In this section, we will do the reverse of Section 5.2, in the sense that we will apply the MOB algorithm to data that contains fuzzy breaks or probabilistic profiles. When applying the MOB algorithm to this kind of data, we noticed that it created many partitions using the variables included in the LC allocation model of the true DGP. This behavior renders the algorithm useful for identifying relevant variables to be included in the LC class allocation model. That being said, in this section, we show one way to use the MOB algorithm as a variable selection procedure that can identify the relevant variables of the LC allocation model. To do so, we simulate one data set that follows a LC-MIXL model (see Section 3.4) and, via bootstrapping this data set, we grow one hundred decision trees using the MOB algorithm. We use bootstrapped versions of the data following the Bagging principle (Breiman, 1996) from Ensemble Learning. However, instead of being focused on the predictions of the model, we compute metrics about the relevance of the partition variables to be included in the LC allocation model. Furthermore, to speed up the exercise, we only used a simple MNL model as the parametric model (MOB-MNL hereafter) and we observe that the algorithm is able to retrieve the important variables from the allocation model. We assume that the use of MIXL models can improve the results, however further assumptions are necessary (i.e., the distribution of the random parameters) and the computation time will most likely increase drastically. In

[5] The different starting values for the classes of the LC and LC-MIXL models are sampled from a uniform distribution that ranges ±1.5 units around the parameter estimates of an MNL model. In particular, for the LC-MIXL models, we used those parameters to initialize the mean and mode of the normal and triangular distributions, respectively. Finally, all the starting values of the extra parameters of the LC and LC-MIXL that do not appear in a simple MNL model (i.e., the allocation model and variance parameters) are sampled from the same distribution but centered at zero.

Table 3
LC-MIXL models fitted to data with hard breaks.

Variables	LC-MIXL			
	($\delta = 0.5$)		($\delta = 1.0$)	
	True[a]	Estimates	True[a]	Estimates
$x_1(\mu_1)$	1	0.982(0.038)***	2	1.900(0.065)***
$x_1(\mu_2)$	2	2.106(0.095)***	1	0.987(0.029)***
$x_1(\mu_3)$	1.5	1.500(0.052)***	3	3.116(0.119)***
$x_1(\sigma_1)$	0.5	−0.475(0.032)***	0.5	−0.412(0.090)***
$x_1(\sigma_2)$	0.5	−0.500(0.209)*	0.5	−0.476(0.031)***
$x_1(\sigma_3)$	0.5	0.451(0.079)***	0.5	−0.362(0.164)*
$x_2(\mu_1)$	1	1.004(0.036)***	2	1.977(0.067)***
$x_2(\mu_2)$	1.5	1.509(0.105)***	1	1.006(0.029)***
$x_2(\mu_3)$	1.5	1.575(0.052)***	2	2.153(0.089)***
$x_2(\sigma_1)$	0.5	0.440(0.040)***	0.5	0.525(0.074)***
$x_2(\sigma_2)$	0.5	−0.598(0.069)***	0.5	0.444(0.035)***
$x_2(\sigma_3)$	0.5	−0.491(0.078)***	0.5	0.610(0.079)***
λ_2		−1256.46(6.94)***		713.93(0.11)***
$Z_1(\gamma_2)$		1393.02(10.20)***		−1363.37(0.17)***
$Z_2(\gamma_2)$		548.09(49.47)***		90.99(1.33)***
$Z_3(\gamma_2)$		−0.07(2.69)		−39.02(3.23)***
$Z_4(\gamma_2)$		96.52(71.41)		−39.03(2.01)***
$Z_5(\gamma_2)$		−89.31(12.44)***		65.44(3.03)***
λ_3		−439.59(20.32)***		−156.33(1.88)***
$Z_1(\gamma_3)$		751.54(25.62)***		42.17(1.94)***
$Z_2(\gamma_3)$		−47.45(17.10)**		219.52(1.13)***
$Z_3(\gamma_3)$		−10.52(27.24)		15.54(3.23)***
$Z_4(\gamma_3)$		77.41(14.78)***		1.73(2.88)
$Z_5(\gamma_3)$		21.73(7.27)**		−7.38(2.50)**
N		15 000		15 000
LL		−10 337.07		−9443.21
Num.Params		24		24
AIC		20 772.1		18 934.4
BIC		20 904.9		19 117.2
$\bar{\pi}_1$	50%	54.17%	25%	24.17%
$\bar{\pi}_2$	25%	19.05%	50%	54.27%
$\bar{\pi}_3$	25%	26.78%	25%	21.57%

Clustered standard errors in parenthesis.
* p < 0.1, ** p < 0.05, *** p < 0.01.
[a] Given the nature of the true GDP (see Fig. 2(b)), the only parameters we know are the taste parameters withing each class. Hence, this is why there are no "True" values for the parameters of the allocation model.

Section 6.1 we describe the true DGP and the configuration of the MOB-MNL we fit to the bootstrapped versions of the simulated data. Section 6.2 describes the three metrics we used to shed some light on the relevance of each variable as a candidate for the allocation model. Finally, 6.3 shows the results of this simulation study.

6.1. LC-MIXL data description and the MOB-MNL algorithm

The three-class LC-MIXL model we used to simulate the artificial data set has an allocation model that contains five different individuals' characteristics (Z_1 to Z_5), which follow a uniform distribution from −1 to 1. The true parameters of the allocation model are described in columns $\lambda_c, \gamma_{1,c}, \gamma_{2,c}, \gamma_{3,c}, \gamma_{4,c}$ and $\gamma_{5,c}$ in Table 4 where only tree variables have a non-zero coefficient (i.e., Z_1, Z_2 and Z_3). Inside each class, the utility was based on two alternative specific attributes, ($X_{1,c}$ and $X_{2,c}$ with $c \in \{1, 2, 3\}$) with coefficients that follow a standard normal distribution. Additionally, the class-specific distributions of the individual level parameters of the alternative-specific attributes are described in columns $\beta_{1n|c}$ and $\beta_{2n|c}$ in Table 4. We simulated the data using a sample size of 1500 individuals which answered 10 choice situations each. Finally, we ran the MOB-MNL algorithm 100 times over 100 different bootstrapped versions of the simulated data set. We used all the five simulated individual characteristics as candidates for partition variables and modeled the MNL model at the end leaves using the two alternative-specific attributes (X_1 and X_2).

6.2. Metrics of the partition variables' relevance

From the 100 resulting decision trees, we retrieve three metrics of variable importance that show the relevance of each partition variable as a candidate to be included in the allocation model of an LC or LC-MIXL model. First, we compute the proportion of times that each partition variable created a root split; that is to say, it was the first variable selected as a partition variable. Second,

Table 4
True Parameters of LC-MIXL model used to simulate data.

Class (c)/Param.	$\beta_{1n\|c}$	$\beta_{2n\|c}$	λ_c	$\gamma_{1,c}$	$\gamma_{2,c}$	$\gamma_{3,c}$	$\gamma_{4,c}$	$\gamma_{5,c}$
1	$\mathcal{N}(1, 0.25)$	$\mathcal{N}(-1, 0.25)$	0	0	0	0	0	0
2	$\mathcal{N}(2, 0.25)$	$\mathcal{N}(-2, 0.25)$	1	2	2	1	0	0
3	$\mathcal{N}(3, 0.25)$	$\mathcal{N}(-3, 0.25)$	−1	−2	−2	−1	0	0

we compute the average number of splits created by each partition variable over the 100 trees. Finally, we pruned the 100 decision trees maximizing the overall BIC of the tree, and we recomputed the average number of splits per variable. These three metrics should indicate how good the method is at identifying the correct variables to be included in the allocation model.

6.3. Results of the MOB algorithm as variable selection step

Table 5 displays the results of the metrics described in Section 6.2. As expected, we observe that the variables that created the larger number of root splits were the ones with the larger coefficients in the allocation model (Z_1 and Z_2). Additionally, the average number of splits, before and after pruning the decision trees, was at least three times larger for the variables with non-zero coefficients in the allocation model. These results show the possible usage of the MOB algorithm as a variable selection procedure for the allocation model when using LC or LC-MIXL models. Accordingly, the same procedure might be used by applied researchers as a guide to select the most relevant variables to be included in the allocation model of LC or LC-MIXL models.

Table 5
Variable importance based on the MOB-MNL algorithm as diagnostic tool.

	Z_1	Z_2	Z_3	Z_4	Z_5
Root node split (%)	34	66	0	0	0
Average number of splits	2.41	1.93	0.58	0.22	0.23
Average number of splits after pruning	2.18	1.57	0.33	0.11	0.10

7. Real data application

7.1. Data description

We use data from De La Maza et al. (2021) to compare the performance of the MOB-MIXL algorithm with standard discrete choice models, namely MNL, MIXL, LC, and LC-MIXL models. The data[6] consists of stated choice data of preferences for environmental impact of two (hypothetical) energy generation plans in Chile (alternatives A and B). Respondents are required to trade-off increases in their future energy costs with the environmental impact generated by each plan. Additionally, respondents also have a "status quo" option, representing the cost and ecological impact of the current energy generation plan. This scheme leads to a three-alternative setup ($J = 3$), where each individual answers twelve choice situations ($t_n = 12$).[7] The attributes that describe each of the survey's plans are hectares of native forest destroyed (*Forest*), number of emergency room visits for respiratory or cardiovascular diseases (*Morbidity*), hectares of land used (*Land*), a dummy variable indicating if the used area will be pristine or not (*Location*) and the resulting increase in the electric bill from each of the energy plans (*Cost*). For a complete description of the choice experiment, see Appendices 5.1 and 5.2 in De La Maza et al. (2021).

The survey also captures information about the individuals, such as *Gender*, *Age*, *Income*, average amount paid for electricity (*Electricity Bill*), whether the individual belongs to a specific ethnicity (*Ethnicity*), membership in an environmental Non-Governmental Organization (*NGO*), having children (*Children*), whether participants visited any potentially affected area (*Visit*), or whether participants had family in those areas (*Family*). Finally, participants were randomly selected to sign an oath to give truthful answers (*Signed Oath*). Attitudinal variables such as trust in the government, pro-social behavior, and a sense of individual responsibility are also present in the sample. However, we exclude those variables from our analysis primarily because they are hard to interpret as partition variables given their subjective nature. For a detailed description of the survey and descriptive statistics, we refer readers to De La Maza et al. (2021).

7.2. Model specification

7.2.1. The MOB-MIXL model

We implement the MOB-MIXL algorithm imposing a minimum of 360 choice sets (equivalent to 30 individuals) at each end leaf and generating breaks when the stability test rejects the null hypothesis of parameter stability at 5% significance. We selected a minimum of 30 individuals so the algorithm does not fail to estimate a MIXL model at the end leaves because the sample size is too small. Regarding the MIXL model, we assumed the *Land* coefficient to be fixed given that in preliminary investigations, the standard

[6] The data set from De La Maza et al. (2021) is available at osf.io/uqtjb.

[7] In the data, nine individuals answered only eleven out of the twelve choice sets. They were not dropped from the sample.

deviation parameter (when assumed normally distributed) was very close to zero. Additionally, the dummy variables *Location* and the alternative specific constants (ASC) were also assumed to have fixed parameters. On the other hand, the *Cost* parameter was assumed to follow a restricted triangular distribution, where the mode and the spread parameters of a triangular distribution are restricted to have the same value. We selected this distribution mainly because more complex non-positive distributions suitable for a money metric (e.g., Log-normal, truncated normal distributions) produced unreasonably long tails.

Additionally, we implement the MOB-MIXL algorithm using two different distributions for the *Forest* and *Morbidity* coefficients to see how sensitive the resulting tree was to the selected parametric distribution of the random coefficients. We used a Normal distribution in one of the specifications, estimating its mean (μ) and variance (σ), and in the other, we used a triangular distribution, where we estimate its mode (b) and spread (v) parameters. Finally, to see how the number of draws affected the decision tree, we tried 1000, 5000, and 10.000 Halton draws to see how sensitive the tree structure was to the quality of the approximation of the log-likelihood and score functions.

7.2.2. The LC and LC-MIXL models

We also estimated LC and LC-MIXL models to the same data, where we estimated two specifications for each model which differ in term of their allocation model. The first one only includes a constant (λ_s), and the second one includes an allocation model that includes sociodemographic variables. We selected those variables using the MOB algorithm as a diagnostic tool (See Section 6). We display the results we obtained, using the data from De La Maza et al. (2021), in Table 6. We observe that out of the 10 possible partition variables, only 5 created partitions in the sample and that 53% of the time no split was found. Additionally, the largest percentage of root splits was found with the variable *Visit* which was used 25% of the time. The rest of the variables that were selected as partitions were *Age*, *Electric Bill*, *Signed Oath* and *Income*. We estimated different versions of the LC and LC-MIXL models sequentially adding variables from the most relevant to the least relevant based on the relevance metric obtained from the MOB algorithm. We estimated each of those models using two and three classes each. Additionally, for the LC-MIXL model, we assume the same distributions as for the MIXL model used in the MOB algorithm, namely *Cost* follows a restricted triangular distribution and *Forest* and *Morbidity* follow Normal distributions. For the LC-MIXL, we used 2500 Halton draws for the simulated maximum likelihood estimation. Finally, we estimated LC-MIXL and LC models, 50 and 100 times[8], respectively, using different starting points (see Footnote 5) to avoid local optima and selecting the model that attained the highest log-likelihood value function.

Table 6
Summary of the MOB-MNL algorithm over bootstrapped De La Maza et al.'s data.

	None	Visit	Age	Electric bill	Signed oath	Income
Root node split (%)	53	25	9	5	5	3
Average number of splits	–	0.29	0.16	0.12	0.09	0.05
Average number of splits after pruning	–	0.29	0.16	0.12	0.08	0.05

7.3. Results

7.3.1. The MOB-MIXL: Coefficients with a normal distribution

Fig. 6 presents the resulting tree assuming Normal distributions for the coefficients of the attributes *Forest* and *Morbidity* and using 10,000 Halton draws to estimate the MIXL models. The p-value of the stability test for each partition variable is displayed below its name in each node. We present the number of individuals (N) and the number of choice situations (T) present in each data segment at the end leaves. From Fig. 6, we observe that the first partition created by the tree is based on the variable *Visit*. Subsequently, another partition is created for the segment of the sample that has not visited the potentially affected areas based on individuals' *Age*. Additionally, for respondents older than 32 years old, three subsequent partitions are created based on how much they paid in their current *Electricity Bill*. Here it is important to mention that, when using 1000 draws, the tree grew the partition for *Visit* (Node $\boxed{1}$) and *Age* (Node $\boxed{2}$) only. However, when increasing the number of draws to 5000, the tree created the first partition in terms of *Electric Bill*, and only when using 10,000 draws the algorithm finds the tree presented in Fig. 6. Accordingly, the tree depends on the approximation of the integral in Eq. (8), hence large number of draws is necessary for a proper approximation of the score functions we use to construct the stability test. In terms of computation time, it took 35 min, 8.85 h, and 1.23 days respectively, to grow the tree using 1000, 5000, and 10,000 draws. Finally, we perform a post pruning procedure that searches to minimize the global tree's BIC, and it kept only the first partition. The pruned tree is presented in Fig. 7 and the parameter estimates of the each of the branches (nodes $\boxed{2}$ and $\boxed{3}$) and the model fitted to the entire sample (node $\boxed{1}$) are presented in Table 7. Additionally, a graphical comparisson of the parameter estimates of the different nodes is presented in Fig. 8.

In terms of goodness-of-fit, we can see that the MOB-MIXL model's fit is better in terms of AIC (84 points less) and BIC (26 points less) compared to the MIXL model fitted to the entire sample. Additionally, in Table 7 we can see the parameters' differences between the end leaves. From Table 7, we observe that the differences in the parameter estimates across the end leaves are not very pronounced. The only exception is that the ASC related to the *Status Quo* option has a larger negative effect for those people that have visited potentially affected areas.

[8] We only estimated 50 different starting values for LC-MIXL models because of how computationally demanding those models are. In our case, it took around 3 days to complete the 50 runs using different starting values using 5000 draws for each model.

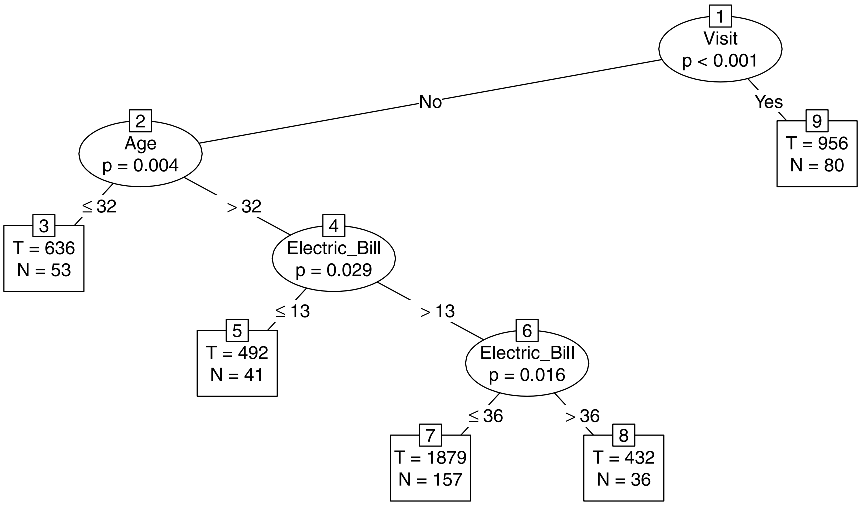


Fig. 6. Pre-pruning tree using normal distributions.

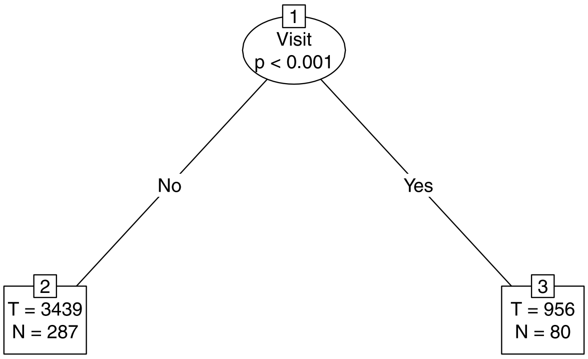


Fig. 7. Post-pruning tree using normal distributions.

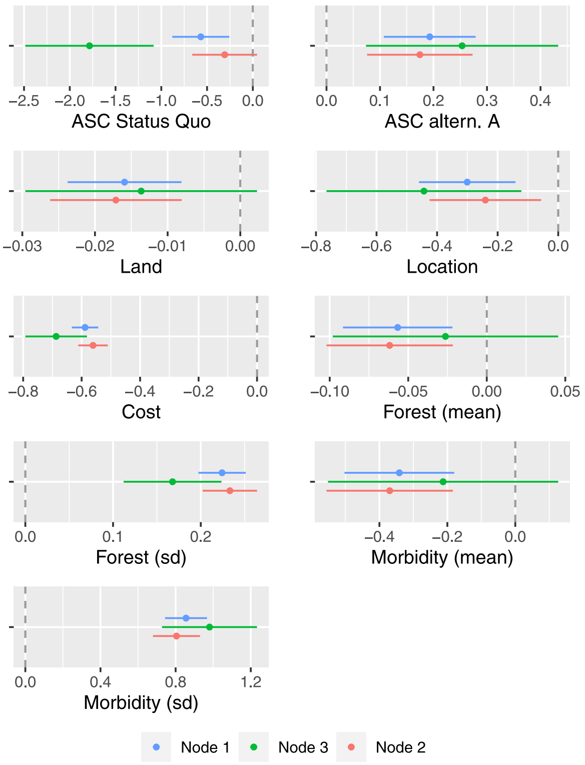


Fig. 8. Point estimates and 95% confidence interval of the end leaves of tree in Figure Fig. 7.

Table 7
Parameter estimates of the end leaves of tree in Fig. 7.

	2	3	1
cte_{sq}	−0.31	−1.78***	−0.57***
	(0.19)	(0.41)	(0.17)
cte_{A}	0.17***	0.25**	0.19***
	(0.06)	(0.12)	(0.05)
Location	−0.24**	−0.44**	−0.30***
	(0.10)	(0.19)	(0.09)
Land	−0.02***	−0.01	−0.02***
	(0.00)	(0.01)	(0.00)
Forest (μ)	−0.06***	−0.03	−0.06***
	(0.02)	(0.03)	(0.02)
Forest (σ)	0.23***	0.17***	0.22***
	(0.01)	(0.03)	(0.01)
Morbidity (μ)	−0.37***	−0.21	−0.34***
	(0.09)	(0.15)	(0.08)
Morbidity (σ)	0.80***	0.98***	0.86***
	(0.07)	(0.14)	(0.06)
Cost ($b = v$)	−0.56***	−0.69***	−0.59***
	(0.03)	(0.08)	(0.03)
Num.Obs.	3439	956	4395
Log.Lik.	−3217.08	−827.6	−4095.87
Num.Param.	18		9
AIC	8125.36		8209.74
BIC	8240.34		8267.26

Clustered standard errors in parenthesis.
* p < 0.1, ** p < 0.05, *** p < 0.01.

7.3.2. The MOB-MIXL: Coefficients with a triangular distribution

We repeat the exercise using triangular distributions for the *Forest* and *Morbidity* attributes where we estimate their mode (b) and spread (v) parameters, while keeping all the other specification, of the models unaltered. Fig. 9 displays the resulting tree when using 10,000 draws, and we see that the change in the distribution of those two attributes drastically changed the resulting tree. For instance, the first data partition was created using the variable *Income* dividing the sample between the people with the lowest income in the sample and the other individuals. Additionally, it created a data partition using *Visit* and then two subsequent partitions using *Electric Bill* and *Age*. Here it is worth mentioning that, differently from the previous case using Normal distributions, the tree estimated with a different number of draws remains the same. Additionally, in terms of running time, it took 1.38 h, 17.46 h, and 1.06 days respectively, to grow the tree using 1000, 5000, and 10,000 draws. Finally, after pruning the final tree minimizing the BIC, only the partitions using the variables *Income* and *Visit* remain. Table 8 show the parameter estimates of the pruned tree for the different partitions (nodes $\boxed{2}$, $\boxed{4}$ and $\boxed{5}$) and the model fitted to the entire sample (node $\boxed{1}$). Additionally, a graphical comparisson of the parameter estimates of the different nodes is presented in Fig. 11.

In this case, using triangular distributions, we see a larger improvement in model fit from the model fitted to the entire sample, both in AIC (272 units less) and BIC (157 units less) compared to the case using normal distributions in Section 7.3.1. Additionally, when we compare the parameter estimates among the partitions, we see, for example, that individuals in the lowest income segment (node $\boxed{2}$) present a larger cost sensitivity than those with higher income that have not visited the potentially affected areas (node $\boxed{4}$), but smaller than those who visited such places (node $\boxed{5}$). Also, we observe that people with higher income who have visited the sites have a larger parameter for the ASC related to the Status Quo situation. On the other hand, we observe that people from the lower-income segment have larger parameter estimates for the ASC related to alternative A. Finally, we notice that some parameters that were statistically significant when the model was fitted to the entire sample, are not significant for some of the segments, as it happens, for example, with *Location* and the mode parameter of the *Forest* attribute in node $\boxed{2}$.

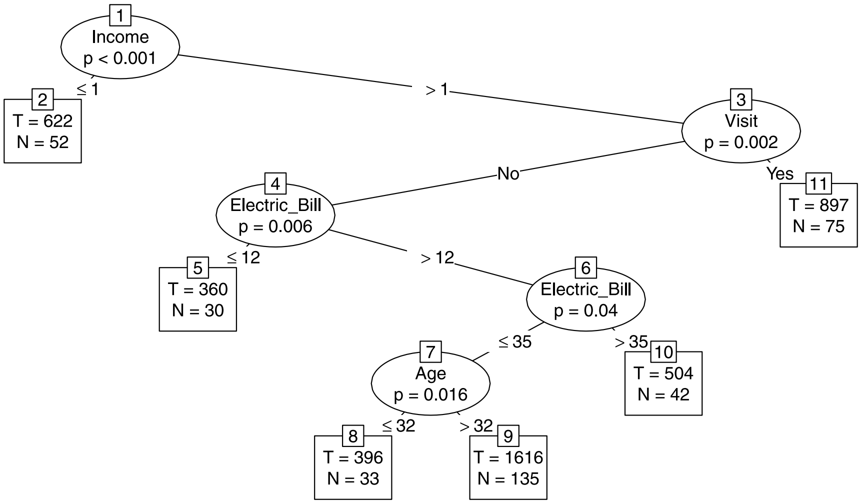


Fig. 9. Pre-pruning tree using triangular distributions.

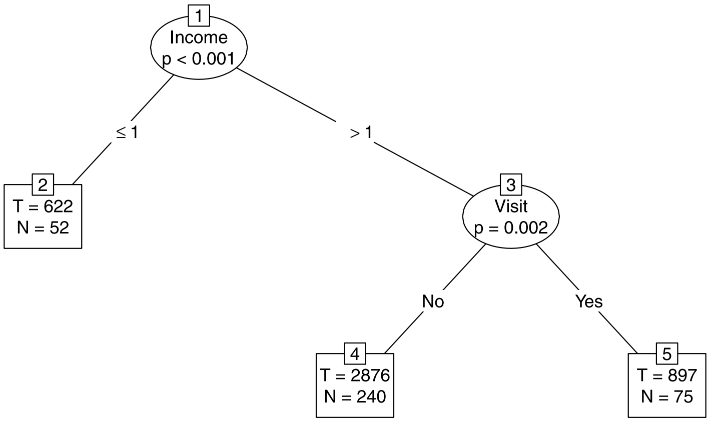


Fig. 10. Post-pruning tree using triangular distributions.

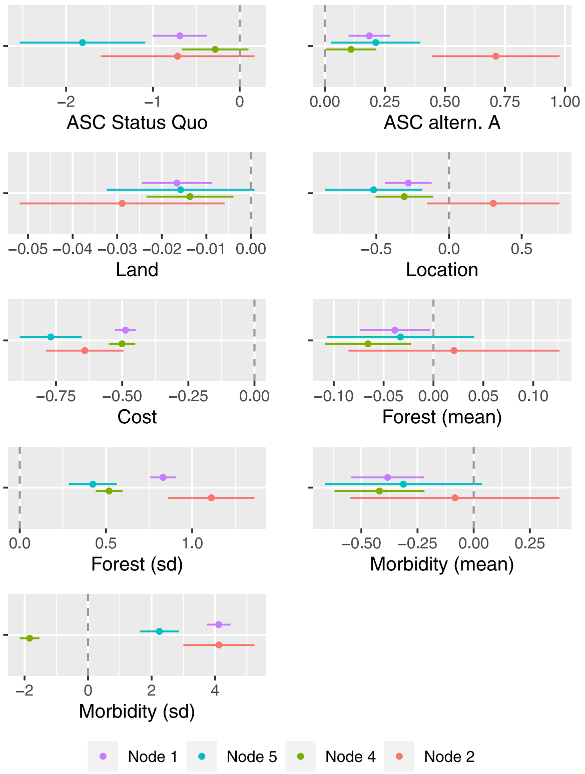


Fig. 11. Point estimates and 95% confidence interval of the nodes end of tree in Fig. 10.

Table 8
Parameter estimates of the nodes end of tree in Fig. 10.

	2	4	5	1
cte$_{sq}$	−0.72	−0.28	−1.81***	−0.69***
	(0.50)	(0.21)	(0.42)	(0.17)
cte$_{A}$	0.71***	0.11*	0.21*	0.19***
	(0.14)	(0.07)	(0.12)	(0.05)
Location	0.30	−0.31***	−0.52***	−0.28***
	(0.25)	(0.11)	(0.19)	(0.09)
Land	−0.03**	−0.01***	−0.02*	−0.02***
	(0.01)	(0.00)	(0.01)	(0.00)
Forest (b)	0.02	−0.07***	−0.03	−0.04**
	(0.05)	(0.02)	(0.03)	(0.02)
Forest (v)	1.11***	0.52***	0.42***	0.83***
	(0.09)	(0.04)	(0.09)	(0.03)
Morbidity (b)	−0.08	−0.42***	−0.31**	−0.38***
	(0.24)	(0.09)	(0.16)	(0.08)
Morbidity (v)	4.12***	−1.84***	2.25***	4.11***
	(0.48)	(0.17)	(0.38)	(0.15)
Cost ($b = v$)	−0.64***	−0.50***	−0.77***	−0.49***
	(0.10)	(0.03)	(0.09)	(0.03)
Num.Obs.	622	2876	897	4395
Log.Lik.	−526.42	−2744.01	−765.01	−4189.81
Num.Param.		27		9
AIC		8124.90		8397.63
BIC		8297.38		8455.12

Clustered standard errors in parenthesis.
* p < 0.1, ** p < 0.05, *** p < 0.01.

7.3.3. Results of the latent class models

To avoid ending up in a local optimum, we estimated each LC model 100 times using different starting values (see Section 7.2.2). We reported the model with the highest likelihood function for each model specification. We estimated LC models that include two and three classes. Additionally, we sequentially included the variables suggested by the MOB algorithm (see Table 6) from the most relevant to the least relevant ones based on the average number of splits created per variable.

We present the model fit of all these models in Fig. 12 where we compared it with the constant-only allocation model, which has zero variables included in the allocation model. Fig. 12 shows that for the two-class model (LC-2), the best model in terms of AIC is the one that includes the first four most relevant variables. However, in terms of BIC, a model including the suggested variables does not outperform the constant-only model. On the other hand, for the three-class model (LC-3), the best model in terms of BIC is the one that includes only one variable, and the best model in terms of AIC is the one that includes four variables in the allocation model. Additionally, we present parameter estimates of the best-performing models together with the constant-only model in Table 9. From Table 9, we can see that all the variables included in the models were statistically different from zero for at least one of the classes. This result is promising because it shows that, using the guidance from the MOB algorithm when selecting variables for the allocation model, we were able to outperform the constant-only latent class model with three classes.

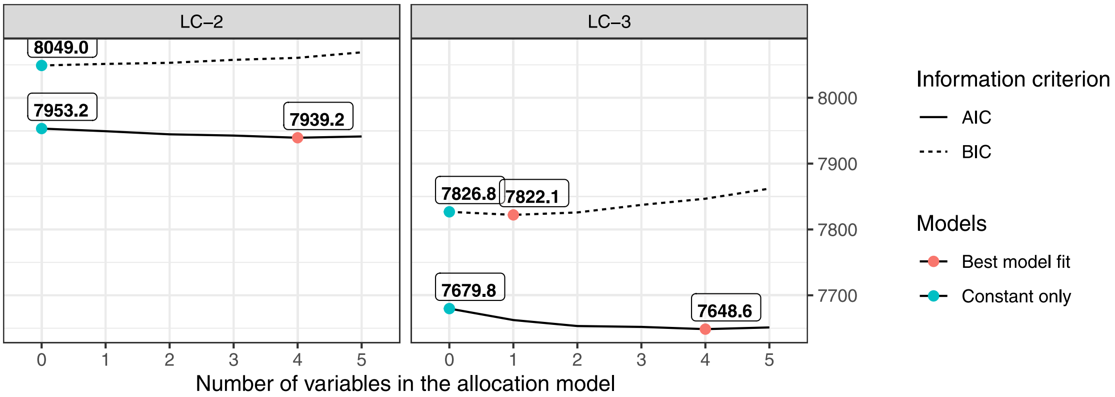


Fig. 12. Information criterion for LC models with different allocation model.

Furthermore, the best performing model in terms of BIC (column LC-3-A-1 in Table 9) model has similarities with the results obtained from the MOB-MIXL algorithm where, for instance, different cost sensitivities are observed across classes. However, the differences are much more noticeable than those captured by the MOB-MIXL algorithm. For example, class number one has a negative cost coefficient at least three times larger than what we observe in the decision tree, while class three has a cost coefficient virtually equal to zero. Regarding the ASC variables, we also see drastically different preferences across classes. For example, the Status Quo constant of the first class is positive and statistically significant; however, it is negative for the second class. More generally, we observe that class number three presents larger values for the estimated parameters. In contrast, class number three is almost indifferent to the observed attributes, which differs from what we see in the tree estimates, which are relatively similar, with only substantial differences in two or three coefficients, like *Cost*, *Location* and the ASC parameters.

Table 9
LC models estimates using data from De La Maza et al. (2021).

Variables	LC-2-A-0	LC-2-A-4	LC-3-A-0	LC-3-A-2	LC-3-A-4
cte$_{sq,1}$	−1.01(0.17)***	−1.01(0.17)***	2.24(0.66)***	−2.03(0.32)***	−2.00(0.31)***
cte$_{A,1}$	0.12(0.04)**	0.12(0.04)**	0.14(0.16)	0.02(0.06)	0.02(0.06)
cte$_{sq,2}$	1.06(0.44)*	1.04(0.43)*	−1.94(0.39)***	0.18(0.31)	0.20(0.33)
cte$_{A,2}$	0.21(0.14)	0.22(0.14)	0.02(0.07)	0.37(0.09)***	0.38(0.08)***
cte$_{sq,3}$			0.20(0.32)	2.23(0.64)***	2.20(0.63)***
cte$_{A,3}$			0.40(0.09)***	0.15(0.16)	0.16(0.16)
Forest 1	−0.07(0.02)***	−0.07(0.02)***	−0.47(0.10)***	−0.09(0.02)***	−0.09(0.02)***
Forest 2	−0.18(0.06)**	−0.18(0.06)**	−0.09(0.02)***	−0.01(0.04)	−0.01(0.03)
Forest 3			0.00(0.04)	−0.46(0.09)***	−0.46(0.09)***
Morbidity 1	−0.33(0.08)***	−0.33(0.08)***	−2.04(0.44)***	−0.45(0.10)***	−0.45(0.10)***
Morbidity 2	−0.67(0.29)*	−0.65(0.28)*	−0.45(0.10)***	0.03(0.16)	0.04(0.16)
Morbidity 3			0.05(0.17)	−2.04(0.41)***	−2.00(0.41)***
Cost 1	−0.18(0.02)***	−0.18(0.02)***	−2.43(0.49)***	−0.25(0.03)***	−0.25(0.03)***
Cost 2	−0.92(0.15)***	−0.91(0.15)***	−0.25(0.04)***	−0.04(0.05)	−0.03(0.05)
Cost 3			−0.03(0.06)	−2.43(0.42)***	−2.37(0.45)***
Land 1	−0.01(0.00)	−0.01(0.00)	0.02(0.02)	−0.01(0.00)*	−0.01(0.00)*
Land 2	−0.01(0.01)	−0.01(0.01)	−0.01(0.00)*	−0.01(0.01)	−0.01(0.01)
Land 3			0.00(0.01)	0.02(0.02)	0.02(0.02)
Location 1	−0.27(0.08)***	−0.27(0.08)***	−0.87(0.37)*	−0.44(0.12)***	−0.44(0.12)***
Location 2	0.14(0.24)	0.15(0.24)	−0.43(0.12)***	0.26(0.16)	0.27(0.17)
Location 3			0.29(0.18)	−0.86(0.35)*	−0.83(0.35)*
λ_2	−0.73(0.12)***	−1.11(0.48)*	0.64(0.20)**	−1.27(0.44)**	−0.73(0.58)
Visit γ_2		−0.87(0.33)**		−1.81(0.46)***	−1.75(0.46)***
Age γ_2		0.01(0.01)		0.03(0.01)*	0.02(0.01)*
Electric bill γ_2		−0.02(0.01)			−0.02(0.01)
Signed oath γ_2		0.56(0.25)*			−0.11(0.37)
λ_3			0.14(0.24)	−1.81(0.48)***	−1.30(0.54)*
Visit γ_3				−1.11(0.38)**	−1.19(0.37)**
Age γ_3				0.03(0.01)**	0.02(0.01)*
Electric bill γ_3					−0.02(0.01)*
Signed oath γ_3					0.66(0.31)*
N	4395	4395	4395	4395	4395
LL	−3961.59	−3950.62	−3816.91	−3799.67	**−3793.32**
Num.Params	15	19	23	27	31
AIC	7953.19	7939.25	7679.82	7653.34	**7648.63**
BIC	8049.01	8060.62	7826.75	**7825.83**	7846.67
$\bar{\pi}_1$	%67.38	%67.23	%24.68	%45.52	%45.98
$\bar{\pi}_2$	%32.62	%32.77	%46.86	%29.81	%29.1
$\bar{\pi}_3$			%28.46	%24.67	%24.92

Clustered standard errors in parenthesis.
* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

7.3.4. Results of the latent class model with random coefficients

As we did for the LC models, to avoid local optima, we estimated each model specification 50 times using different starting values and reported the model with the highest likelihood value. In addition, we estimated LC-MIXL models with two and three classes, and we sequentially added the variables suggested by the MOB-MNL algorithm to the allocation model from the most relevant to the least relevant based on the average number of splits per variable. We present the model fit of the different models in Fig. 13. Similarly to the LC models, we observe that the inclusion of demographic variables for the two-class model (LC-MIXL-2) did not improve the model performance in terms of BIC with respect to the constant-only model. However, in terms of AIC, the best model was the one that included the four most relevant variables suggested by the MOB algorithm. On the other hand, for the three-class model (LC-MIXL-3), the best models included two and four variables when selecting the best model based on the BIC and AIC, respectively. Additionally, we present the best performing models and the constant-only LC-MIXL models in Table 10. From Table 10 we can see that for the three-class models only one of the included variables in the allocation model (*Electric bill*) was not statistically different from zero. Again, this result suggests that we could improve the constant-only model performance by using the MOB algorithm as an intermediate step for variable selection for the allocation model.

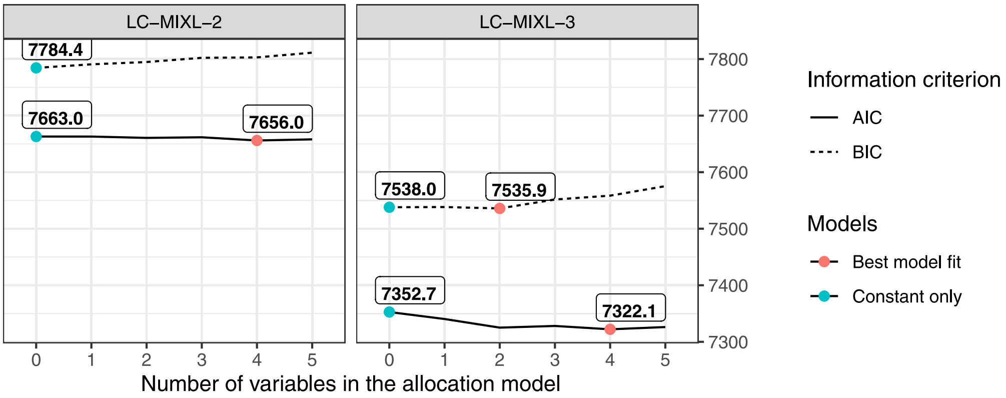


Fig. 13. Information criterion for LC-MIXL models with different allocation model.

Finally, the best model performance in terms of BIC is the three-class model that includes two variables in the allocation model (column LC-MIXL-3-A-2 in Table 10). In general, we observe the same patterns for the LC models, where the differences among the classes are much more pronounced than those obtained by the MOB-MIXL algorithm. We also observe that the ASC variable have very different signs and magnitudes across classes. For instance, the Status Quo variable has a positive value for the third class while being negative for the second class. We also observe the same behavior for the cost sensitivity, where the third class is much more cost sensitive than the second and the first. For the rest of the parameters, the parameter estimates are relatively similar as in the LC model.

Table 10
LC-MIXL models estimates using data from De La Maza et al. (2021).

Variables	LC-MIXL-2-A-0	LC-MIXL-2-A-4	LC-MIXL-3-A-0	LC-MIXL-3-A-2	LC-MIXL-3-A-4
$\text{cte}_{sq,1}$	1.85(0.86)*	1.77(1.02)	0.03(0.25)	0.08(0.25)	0.05(0.25)
$\text{cte}_{A,1}$	0.02(0.22)	0.02(0.20)	0.38(0.07)***	0.37(0.07)***	0.37(0.07)***
$\text{cte}_{sq,2}$	−0.77(0.20)***	−0.78(0.20)***	−3.35(0.50)***	−3.46(0.51)***	−3.47(0.53)***
$\text{cte}_{A,2}$	0.16(0.05)**	0.15(0.05)**	−0.11(0.11)	−0.07(0.11)	−0.08(0.11)
$\text{cte}_{sq,3}$			1.94(0.73)**	1.89(0.74)*	1.90(0.74)*
$\text{cte}_{A,3}$			−0.07(0.22)	−0.08(0.22)	−0.06(0.23)
Forest (μ_1)	−0.45(0.14)**	−0.42(0.16)**	−0.04(0.03)	−0.04(0.03)	−0.04(0.03)
Forest (μ_2)	−0.05(0.02)*	−0.05(0.02)*	−0.07(0.03)*	−0.07(0.03)*	−0.07(0.03)*
Forest (μ_3)			−0.49(0.09)***	−0.48(0.10)***	−0.48(0.10)***
Forest (σ_1)	0.51(0.13)***	0.47(0.11)***	−0.15(0.03)***	0.14(0.02)***	−0.14(0.02)***
Forest (σ_2)	−0.16(0.02)***	−0.16(0.02)***	−0.11(0.05)*	−0.12(0.03)***	−0.12(0.04)***
Forest (σ_3)			0.55(0.08)***	0.54(0.08)***	0.53(0.09)***
Morbidity (μ_1)	−2.88(0.94)**	−2.78(1.13)*	−0.10(0.14)	−0.10(0.13)	−0.09(0.13)
Morbidity (μ_2)	−0.24(0.09)**	−0.25(0.09)**	−0.43(0.16)**	−0.43(0.14)**	−0.44(0.14)**
Morbidity (μ_3)			−3.27(0.50)***	−3.27(0.50)***	−3.24(0.53)***
Morbidity (σ_1)	1.48(0.43)***	−1.44(0.48)**	−0.61(0.16)***	0.54(0.09)***	−0.55(0.10)***
Morbidity (σ_2)	−0.66(0.07)***	−0.66(0.07)***	0.73(0.21)***	0.77(0.15)***	−0.78(0.16)***
Morbidity (σ_3)			−1.57(0.27)***	−1.60(0.28)***	−1.58(0.28)***
Cost ($b_1 = \upsilon_1$)	−2.02(0.71)**	−2.00(0.84)*	−0.03(0.02)	−0.03(0.02)	−0.03(0.02)
Cost ($b_1 = \upsilon_2$)	−0.11(0.02)***	−0.11(0.02)***	−0.23(0.04)***	−0.22(0.04)***	−0.23(0.04)***
Cost ($b_1 = \upsilon_3$)			−2.40(0.42)***	−2.47(0.44)***	−2.42(0.50)***
Land 1	−0.01(0.02)	−0.01(0.03)	−0.01(0.01)	−0.01(0.01)	−0.01(0.01)
Land 2	−0.01(0.00)*	−0.01(0.00)*	−0.02(0.01)*	−0.02(0.01)*	−0.02(0.01)*
Land 3			0.00(0.02)	0.00(0.02)	0.00(0.02)
Location 1	−1.19(0.59)*	−1.11(0.72)	0.20(0.14)	0.18(0.14)	0.19(0.14)
Location 2	−0.26(0.10)**	−0.26(0.10)**	−0.84(0.17)***	−0.75(0.17)***	−0.77(0.17)***
Location 3			−1.38(0.36)***	−1.38(0.37)***	−1.36(0.37)***
λ_2	0.81(0.17)***	1.47(0.52)**	−0.41(0.19)*	1.06(0.47)*	1.04(0.61)
Visit γ_2		0.56(0.36)		1.56(0.44)***	1.44(0.46)**
Age γ_2		−0.01(0.01)		−0.04(0.01)***	−0.04(0.01)***
Electric bill γ_2		0.01(0.01)			−0.01(0.01)
Signed oath γ_2		−0.68(0.25)**			0.58(0.33)
λ_3			−0.38(0.15)*	−0.50(0.45)	−0.48(0.53)
Visit γ_3				0.33(0.41)	0.20(0.42)
Age γ_3				0.00(0.01)	0.00(0.01)
Electric bill γ_3					−0.01(0.01)
Signed oath γ_3					0.83(0.28)**
N	4395	4395	4395	4395	4395
LL	−3812.50	−3805.01	−3647.36	−3629.54	**−3624.06**
Num.Params	19	23	29	33	37
AIC	7663.01	7656.02	7352.71	7325.08	**7322.12**
BIC	7784.38	7802.95	7537.97	**7535.89**	7558.49
$\bar{\pi}_1$	%30.75	%30.44	%42.5	%41.4	%41.69
$\bar{\pi}_2$	%69.25	%69.56	%28.31	%29.73	%29.34
$\bar{\pi}_3$			%29.2	%28.87	%28.97

Clustered standard errors in parenthesis.
* $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

8. Discussion and conclusions

An important point worth mentioning is the similarity of the models resulting from the MOB-MIXL algorithm and from the LC and LC-MIXL models. The MOB-MIXL algorithm resembles what those latter models can produce by generating different taste parameters for different groups of individuals. However, the LC and LC-MIXL models provide the econometrician with *probabilistic profiles* or *fuzzy breaks* of individuals, meaning that given individual-specific characteristics, we can compute the probability of an individual to belong to a given class. On the other hand, the MOB-MIXL algorithm captures the heterogeneity using *hard breaks* that work as a deterministic function of individual-specific variables; that is to say, it allocates people to different tree leaves based on their individuals' characteristics not allowing for uncertainty.

Although the assumption of deterministic or *hard breaks* in the taste parameters based on individual characteristics might seem too "crude" at first, it is based on statistical arguments of the stability of the parameter estimates. Additionally, it provides some advantages compared to latent class models. First, given that the algorithm splits the data set, the groups are easily identifiable in terms of the characteristics of the individuals. Hence it could be very interesting for segmentation policies highly used in, for example, marketing contexts. Second, there is no need to select the variables to be included in the allocation model beforehand. Instead, the algorithm automatically identifies which variables are relevant to create a partition and grows a decision tree accordingly. Finally, LC and LC-MIXL models can also benefit from the MOB algorithm used as a diagnostic tool to identify the most relevant variables to be included in the allocation model, as illustrated in Section 6.

To summarize, this article illustrated the use of the MOB algorithm (Zeileis et al., 2008) in a discrete choice context. The algorithm allows the modeler to grow a decision tree that divides the sample based on individual characteristics. To the best of the authors' knowledge, it is also the first decision tree used in the discrete choice literature that allows for the inclusion of random coefficients in the models used at the end leaves. To illustrate the usage of the proposed algorithm, we presented three simulation studies. The first showed that the algorithm could correctly recover different tree-like data generation processes when these are present in the data. In the second one we showed how a latent class model would behave when *hard breaks* are present on the data. This simulation study showed that exploding parameters in the allocation model are caused by having *hard breaks* in the taste parameters. The third simulation study illustrated the use of the MOB algorithm as a variable selection step for the latent classes' allocation model via bootstrapping.

Additionally, we illustrated how the MOB-MIXL algorithm performs on real data using stated choice data of the preferences for the environmental impact of (hypothetical) energy generation plans in Chile. The results showed that the model obtained outperforms the MIXL model fitted to the whole sample in terms of model fit. Besides, we observed that the resulting tree is sensitive to the distributions of the random coefficients and needs enough draws to obtain a stable tree structure. Furthermore, we also compared the MOB-MIXL results with other models conventionally used in discrete choice applications, such as LC and LC-MIXL models. For those models, we used the MOB algorithm as a variable selection step when selecting the variables to be included in the allocation model, and by doing so, we were able to outperform the latent classes with constant-only allocation models in terms of information criterion. Furthermore, we observe that the LC and LC-MIXL models produced parameter estimates that are much more different across the classes than those produced by the MOB algorithm.

To conclude, we claim that the MOB algorithm is not only a data-driven method in itself, which grows a fully interpretable decision tree based on statistical tests, but it can also be helpful as a heuristic to perform variable selection for the allocation models of LC and LC-MIXL models. Finally, future research using the MOB algorithm in combination with ensemble methods, such as bagging or boosting, might be worthwhile if the purpose is to attain a model with high predictive power.

CRediT authorship contribution statement

Álvaro A. Gutiérrez-Vargas: Conceptualization, Methodology, Software, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Michel Meulders:** Conceptualization, Writing and reviewing of the manuscript, Supervision. **Martina Vandebroek:** Conceptualization, Writing and reviewing of the manuscript, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank to Cristóbal De La Maza for giving us permission to use his data in our empirical application. Also the first author thanks to Tomás Rossetti for his help regarding some of the Apollo functionalities. Additionally, this work was produced while Álvaro A. Gutiérrez-Vargas was a Ph.D. student at the Research Centre for Operations Research and Statistics (ORSTAT) at KU Leuven funded by Bijzonder Onderzoeksfonds, Belgium KU Leuven (Special Research Fund KU Leuven).

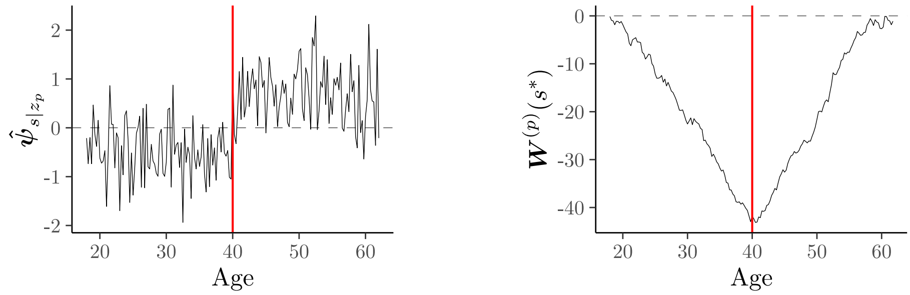


Fig. A.1. *Left panel*: Estimated score functions (Eq. (13)) sorted by variable *Age*. *Right panel*: Empirical fluctuation process (Eq. (14)) of variable *Age*.

Appendix A. Illustration of the empirical fluctuation process

For the sake of illustration of the *empirical fluctuation process* and its relationship with the score functions, suppose we estimate a hypothetical model with a single attribute ($k = 1$) in the utility specification. Additionally, let us assume that his parameter is the "*total cost*" of a given route and that we estimate the parameter without using random coefficients. That is to say, only one parameter is estimated. Besides, assume we are interested in analyzing whether the estimated parameter is stable across the age of the individuals in our sample. In Fig. A.1 we plotted the score function (Eq. (13)) sorted by individuals' age (*left panel*), and the *empirical fluctuation process* (Eq. (14)) of the variable age (*right panel*). The figure shows that the score function does not fluctuate around zero, which serves as a diagnosis of the instability of the estimated parameter. In particular, we can see that it systematically fluctuates below zero for individuals younger than 40 years and above zero for individuals older than 40 years. This fact implies that a model ignoring this parameter instability in our hypothetical situation would overestimate the cost parameter of younger individuals and underestimate it for older ones. Therefore, using this fact, we could get better local models (e.g., with score functions fluctuating randomly around zero) if we fit two separate models; one for people younger than 40 years and one for people older than 40 years.

It is important to notice that the visualization described in Fig. A.1 is only possible because our hypothetical model just has one parameter ($k = 1$), and therefore also, one single score function. In general, we will have as many score functions as estimated parameters. The formal statistical tests presented in Section 4.2 and in Appendix B, also work in multivariate parametric spaces.

Appendix B. Stability tests for categorical variables

In this Appendix we briefly present the test for non-ordered categorical and ordered categorical variables. In the first case, if the partition variable Z_p is a non-ordered categorical variable, having C different categories (e.g., gender, geographical areas, marital status, etc.), the following test statistic was initially proposed by (Zeileis et al., 2008)

$$\phi_{\chi^2}(\boldsymbol{W}^{(p)}) = \sum_{c=1}^{C}\left(\frac{|I_c|}{T}\right)^{-1}\sum_{k=1}^{K^*}\left[w_{ck}^{(p)}\right]^2, \tag{B.1}$$

where $|I_c|$ represents the number of observations in class c. The test captures the increments of the empirical fluctuation process over the observations in category $c \in \{1,\dots,C\}$. In other words, it computes the square of the elements of the $\boldsymbol{W}^{(p)}$ matrix's rows scaled by the inverse of the percentage of participation of each category ($|I_c|/T$) and then it sum them up across all the categories. The test is invariant to the ordering of the categories (e.g., it is insensitive to the ordering of the C labels). The asymptotic distribution is a χ^2 distribution with $K^*\times(C-1)$ degrees of freedom where K^* is the number of estimated parameters. The corresponding p-values can be computed as in Hjort and Koning (2002).

A modified test that does consider the ordering of the different classes (e.g., educational level, income range, etc.) is proposed by Merkle et al. (2014), the so-called ordinal maximum Lagrange multiplier statistic

$$\phi_{\max\ \mathrm{LM}}(\boldsymbol{W}^{(p)}) = \max_{s=s_1,\dots,s_m}\left(\frac{s}{T}\times\frac{T-s}{T}\right)^{-1}\sum_{k=1}^{K^*}\left[w_{sk}^{(p)}\right]^2. \tag{B.2}$$

The test is similar to the one proposed in Eq. (15), but it considers bins of individuals at each level of the given ordinal variable. That is to say, instead of aggregating $s = 1,\dots,T$ choice situations, it first computes the empirical fluctuation process of the m levels (s_m) of the considered partition variable, with m the number of levels of the ordinal categorical variable. In other words, instead of computing the maximum over the total number of choice situations, it calculates the maximum value over the total number of possible levels of the first $m-1$ levels in the partition variable. The asymptotic distribution for the statistic is derived in Merkle et al. (2014), yet no closed-form solution is available. Instead, critical values and corresponding p-values can be obtained repeatedly simulating Brownian bridges, which can be computed, for example, by using the `strucchange` R package (Zeileis, 2006).

References

Akaike, H., 1998. In: Parzen, E., Tanabe, K., Kitagawa, G. (Eds.), Information Theory and an Extension of the Maximum Likelihood Principle. Springer New York, New York, NY, pp. 199–213.

Andrews, D.W.K., 1993. Tests for parameter instability and structural change with unknown change point. Econometrica 61 (4), 821–856.

Arentze, T., Timmermans, H., 2007. Parametric action decision trees: Incorporating continuous attribute variables into rule-based models of discrete choice. Transp. Res. B 41 (7), 772–783.

Bhat, C.R., 1997. An endogenous segmentation mode choice model with an application to intercity travel. Transp. Sci. 31 (1), 34–48.

Brathwaite, T., Vij, A., Walker, J.L., 2017. Machine learning meets microeconomics: The case of decision trees and discrete choice. https://arxiv.org/abs/1711.04826.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140.

Breiman, L., Friedman, J., Olshen, R., Stone, C., 1984. Classification and Regression Trees. Wadsworth & Brooks/Cole Advanced Books & Software, Pacific California.

Chamberlain, G., 1980. Analysis of Covariance with Qualitative Data. Rev. Econom. Stud. 47 (1), 225–238.

Chow, G.C., 1960. Tests of equality between sets of coefficients in two linear regressions. Econometrica 28 (3), 591–605.

Cockx, K., Canters, F., 2020. Determining heterogeneity of residential location preferences of households in Belgium. Appl. Geogr. 124, 102271.

Croissant, Y., 2020. Estimation of random utility models in R: The mlogit package. J. Stat. Softw. 95 (1), 1–41.

De La Maza, C., Davis, A., Azevedo, I., 2021. Welfare analysis of the ecological impacts of electricity production in Chile using the sparse multinomial logit model. Ecol. Econom. 184, 107010.

Fokkema, M., Smits, N., Zeileis, A., Hothorn, T., Kelderman, H., 2018. Detecting treatment-subgroup interactions in clustered data with generalized linear mixed-effects model trees. Behav. Res. Methods 50 (5), 2016–2034.

Greene, W., Hensher, D., 2003. A latent class model for discrete choice analysis: contrasts with mixed logit. Transp. Res. B 37 (8), 681–698.

Han, Y., 2019. Neural-Embedded Discrete Choice Models (Ph.D. thesis). Massachusetts Institute of Technology.

Hansen, B.E., 1997. Approximate asymptotic p values for structuras-change tests. J. Bus. Econom. Statist. 15 (1), 60–67.

Hensher, D., Greene, W., 2003. The mixed logit model: The state of practice. Transportation 30 (2), 133–176.

Hess, S., Palma, D., 2019. Apollo: a flexible, powerful and customisable freeware package for choice model estimation and application. J. Choice Model. 32, 100170.

Hillel, T., Bierlaire, M., Elshafie, M.Z., Jin, Y., 2021. A systematic review of machine learning classification methodologies for modelling passenger mode choice. J. Choice Model. 38, 100221.

Hjort, N.L., Koning, A., 2002. Tests for constancy of model parameters over time. J. Nonparametr. Stat. 14 (1–2), 113–132.

Hothorn, T., Hornik, K., Zeileis, A., 2006. Unbiased recursive partitioning: A conditional inference framework. J. Comput. Graph. Statist. 15 (3), 651–674.

Hothorn, T., Zeileis, A., 2015. Partykit: A modular toolkit for recursive partytioning in R. J. Mach. Learn. Res. 16 (1), 3905–3909.

Hubert, L., Arabie, P., 1985. Comparing partitions. J. Classification 2 (1), 193–218.

Karlaftis, M.G., 2004. Predicting mode choice through multivariate recursive partitioning. J. Transp. Eng. 130 (2), 245–250.

Keane, M., Wasi, N., 2013. Comparing alternative models of heterogeneity in consumer choice behavior. J. Appl. Econometrics 28 (6), 1018–1045.

Liang, L., Xu, M., Grant-Muller, S., Mussone, L., 2021. Household travel mode choice estimation with large-scale data—an empirical analysis based on mobility data in milan. Int. J. Sustain. Transp. 15 (1), 70–85.

Loh, W.-Y., 2002. Regression tress with unbiased variable selection and interaction detection. Statist. Sinica 12 (2), 361–386.

Loh, W.-Y., Shih, Y.-S., 1997. Split selection methods for classification trees. Statist. Sinica 7 (4), 815–840.

Loh, W.-Y., Vanichsetakul, N., 1988. Tree-structured classification via generalized discriminant analysis. J. Amer. Statist. Assoc. 83 (403), 715–725.

McFadden, D., 1974. Conditional logit analysis of qualitative choice behavior'. In: Zarembka, P. (Ed.), Frontiers in Econometrics. Academic Press, New York.

McFadden, D., Train, K., 2000. Mixed MNL models for discrete response. J. Appl. Econometrics 15 (5), 447–470.

Merkle, E.C., Fan, J., Zeileis, A., 2014. Testing for measurement invariance with respect to an ordinal variable. Psychometrika 79 (4), 569–584.

Quinlan, J.R., 1993. C4. 5: Programs for Machine Learning. Morgan Kaufmann.

R Core Team, 2022. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL https://www.R-project.org/.

Schlosser, L., Hothorn, T., Zeileis, A., 2019. The power of unbiased recursive partitioning: A unifying view of ctree, MOB, and GUIDE. arXiv:1906.10179.

Schwarz, G., 1978. Estimating the Dimension of a Model. Ann. Statist. 6 (2), 461–464.

Sfeir, G., Abou-Zeid, M., Rodrigues, F., Pereira, F.C., Kaysi, I., 2021. Latent class choice model with a flexible class membership component: A mixture model approach. J. Choice Model. 41, 100320.

Tang, L., Xiong, C., Zhang, L., 2015. Decision tree method for modeling travel mode switching in a dynamic behavioral process. Transp. Plan. Technol. 38 (8), 833–850.

Train, K.E., 2008. EM algorithms for nonparametric estimation of mixing distributions. J. Choice Model. 1 (1), 40–69.

Train, K.E., 2009. Discrete Choice Methods with Simulation. Cambridge University Press.

Wickham, H., 2016. Ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag, New York.

Zeileis, A., 2006. Implementing a class of structural change tests: An econometric computing approach. Comput. Statist. Data Anal. 50 (11), 2987–3008.

Zeileis, A., Hornik, K., 2007. Generalized M-fluctuation tests for parameter instability. Stat. Neerl. 61 (4), 488–508.

Zeileis, A., Hothorn, T., Hornik, K., 2008. Model-based recursive partitioning. J. Comput. Graph. Statist. 17 (2), 492–514.

Zhang, W., Mandal, A., Stufken, J., 2017. Approximations of the information matrix for a panel mixed logit model. J. Stat. Theory Pract. 11 (2), 269–295.