RUMBoost: a Gradient Boosted Random Utility Model

Nicolas Salvade[a], Tim Hillel[a]

[a]*Behavioural and Infrastructure Group (BIG), Department of Civil, Environmental and Geomatic Engineering, University College London (UCL), Gower Street, London, WC1E 6BT, , United Kingdom*

Abstract

We modify the LightGBM algorithm with three constraints in order to obtain RUMBoost, a fully interpretable machine learning (ML) model based on gradient boosting decision trees (GBDT) and inspired by random utility models (RUM). In doing so, we combine the critical interpretability of RUM with the well-known predictive power of GBDT model on mode choice prediction tasks. To derive the model, we restrict alternative-specific features to influence only their corresponding predictive function, we prevent feature interaction and we apply monotonic constraint on key features to benefit from domain knowledge (by example travel time should impact negatively the utility function). These modifications allow to retrieve a fully interpretable, non-linear utility function. The methodology is applied on a mode choice dataset, and show a significantly higher predictive power than the benchmarks for traditional Multinomial Logit (MNL) models. We find that the utility function exhibits different behaviours for different transportation modes travel time, namely it is convex for walking and driving, linear for bus, concave for rail, and non-linear with a plateau for cycling. Furthermore, an increase in age shows a decline in the utility function for walking and cycling, an increase until the age of 20 and a fall from the age of 60 for public transport, and the opposite for driving.

Keywords: Gradient Boosting Decision Trees, Random Utility, Non-linear Utility Functions, Monotonicity constraint

1. Introduction

Discrete choice models (DCM) have been used extensively in choice modelling for several decades now (Ben-Akiva and Lerman, 1985; Train, 2009). Among DCM, one of the best-known and most widely used models is the multinomial logit model (MNL) which relies on the Independence of irrelevant alternatives (IIA) derived by McFadden et al. (1973). This model is simple to estimate and provides fully interpretable parameters, which is essential for informing intervention policies. However, the model relies on a rather inflexible linear-in-parameters utility function, making it difficult to capture complex phenomena. Furthermore, DCMs depends on utility specification, a burdensome task that could lead to misspecification.

These drawbacks, in addition to the uprising of data-driven models and big data have driven modellers to use machine learning (ML) or data-driven components in choice modelling, and especially in transportation. These models exhibit high prediction performances and, thanks to their data-driven predictive function, do not require any utility specification. Amid these ML models, Gradient Boosting Decision Trees (GBDT) models are powerful tools to forecast travel mode choice. More specifically, models built on libraries such as XGBoost (Chen and Guestrin, 2016) or LightGBM (Ke et al.) have been achieving state-of-the-art performances in machine learning competitions, and have been setting up benchmarks on several mode choice prediction datasets (Hillel et al., 2018; Martín-Baos et al., 2023). However, these models, like other ML models, lack tremendously of behaviour interpretability, and the underlying methodology is subject to pitfalls (Hillel et al., 2021). They are also highly sensitive to para

In order to address both of the limitations from DCM and ML models in mode choice prediction, and to take advantages of their strengths, we propose a novel approach, Gradient Boosting Random Utility (RUMBoost), that combines GBDT with Random Utility Models (RUM). To obtain the model, we apply three modification to traditional GBDT models:

- Alternative specific features

- Feature interaction constraints
- Monotonic constraints

The first one is used to specify which features should interact with which predictive function.. By example, we can restrict the walking travel time to impact only the walking alternative predictive function. This part can be seen as utility specification, but is much simpler than in DCM since only alternative-specific features have to be specified. The second constraint is crucial for interpretability, because restricting the interaction of features allows us to understand their direct impact on the predictive function. Finally, the monotonic constraints let us determine how the features should interact with the predictive function. A positive (resp. negative) monotonic constraint means that an increasing feature will impact positively (resp. negatively) the predictive function, i.e. the predictive function will rise (resp. drop). For example, we can constrain an increase in travel time to be negative monotonic, therefore to have a negative impact on the predictive function. These three modifications allow the predictive function to be fully interpretable, and therefore to be a *utility function*. Thus, our approach combines the predictive power of GBDT with the interpretability of RUM.

Furthermore, thanks to these constraints, we are able to estimate any MNL model with GBDT. We apply the methodology on a MNL model implemented with PandasBiogeme Bierlaire (2023) on the London Passenger Mode Choice (LPMC) dataset, a publicly available dataset composed of 81,096 observations over three years in Greater London (Hillel et al., 2018). We compare the predictive performance of the RUMBoost model with state-of-the-art ML classifiers, as well as with an MNL model, and analyse the non-linear impact of certain features on the utility function. The model is implemented with python and the code is freely available on github (https://github.com/NicoSlvd/RUMBoost).

The rest of the paper is organised as follows: Section 2 reviews the existing literature, Section 3 presents the methodology of this paper, the results of a case study are presented in Section 4 and Section 5 concludes and suggests areas of improvement for future work.

2. Literature Review

The literature review is divided into two sections. Section 2.1 gives an overview of the specification of hybrid utility functions and Section 2.2 presents the foundations of GBDT.

2.1. Hybrid Utility Specification

Several attempts have been made to incorporate data-driven components into DCM in order to combine the predictive power of ML with the interpretability of DCM. Wang et al. (2020) obtained economical indicators from a deep neural network (DNN). The author interprets the last layer of the DNN as a utility function, while all previous layers and activation functions are considered as utility specifications. This allows to get choice predictions, choice probabilities, market share, substitution patterns of alternatives, social welfare, probability derivatives, elasticity, marginal rates of substitution (MRS) and heterogeneous values of time (VOT). The disadvantages of this method include the fact that any feature can directly influence an alternative utility function, that the utility specification process is unknown since it is learned directly from the data set, which is therefore likely to reproduce any bias of the dataset and that the model is highly sensitive to hyperparameters. The first problem was addressed in a subsequent paper by the same authors (Wang et al., 2021), where the features specific to an alternative are not able to interact with the features of other alternatives. In a problem with J classes, this would be equivalent to having J independent DNNs. Socio-economic features are also treated initially in their own DNN and are added to all alternative specific DNN in a second phase. The two outputs are processed in a few more layers, before obtaining the final utility in the last layer. Following the idea that the last layer of a neural network can be interpreted as utility, Sifringer et al. (2020) used a convolutional neural network (CNN) to model transportation mode choice. The model can be divided in two parts: one accounting for linear-in-parameters utility, which could be seen as a generalisation of the MNL, and one accounting for non-linear specification. In the first part, features of an alternative are combined in a hidden layer using a convolution of stride 1, which is equivalent to the weighted sum operator in the MNL. In the second part, all the remaining unused features are combined in a DNN. The two outputs are added so that each utility function has a linear term and non-linear term, making it possible to interpret the linear-in-parameters part and to fight misspecifications with the latter.

Table 1: Summary of state-of-the-art practice in hybrid utility specification

Papers	Alternative specific features	Second order or higher feature interaction	Cross effect	Monotonicity constraint	Behavioural indicators	Observable non-linear utility
Wang et al. (2020)		X	X		X	
Wang et al. (2021)	X	X			X	
Sirfringer et al. (2020)	X	X			X	
Wong and Farooq (2021)	X	X	X		X	
Krueger and Daziano (2022)	X	X		X	X	
Ortelli et al. (2021)	X				X	
Hillel et al. (2019)	X	X			X	
Current research	X	X	(X)	X	X	X

Wong and Farooq (2021) used a residual neural network to learn the non-linear correction term added in the Mother Logit (McFadden et al., 1977) to account for alternative cross-effects in the error term. Each layer is composed of both the linear sum of the neurons from the previous layer (which are the deterministic utilities in the first layer) and a residual term representing a weighted sum of the neurons transformed by an activation function. This model can be used to calculate behavioural indicators such as point elasticities. To account for interaction between features in a stated preference discrete choice experiment, Krueger and Daziano (2022) used the Choquet integral to replace the standard weighted sum in DCM. This captures interactions between attributes that are subject to monotonicity constraints.

As model specification is one of the most burdensome and difficult task in DCM, Ortelli et al. (2021) and Hillel et al. (2019) have used data-driven techniques to assist the modeller in utility specification. In the first one, the authors developed an algorithm for evaluating an incalculable number of model specifications by drawing the pareto frontier of a multi-objective optimisation function. They use a metaheuristic with neighbour search to apply transformations to the model, with the aim of mimicking the human modeller. They start looking for a solution in a close neighbourhood and gradually move away from the initial solution. In the second one, the authors used gradient boosting decision trees (GBDT) to help the modeller specify utility functions. Potential non-linear interactions of features with respect to mode choice and interactions between socio-demographic features and alternative-specific features can be identified using the distribution of split points and corresponding gains in decision trees. The contribution of these studies and ours is summarised in Table 1

2.2. *Gradient Boosting Decision Trees*

The underlying theory behind GBDT has been derived by Friedman (2001). LightGBM (Ke et al.) and XGBoost (Chen and Guestrin, 2016) are two GBDT-based python libraries that have shown great predictive power but are severely lacking in behavioural interpretation (Martín-Baos et al., 2023). GBDT algorithms also suffer from over-specialisation, which makes a model difficult to use on out-of-sample data. To solve this problem, several regularisation parameters were implemented. These include feature interaction constraints and monotonicity constraints. While they would help solve the problem of over-specialisation in GBDT, they are also key components for a more interpretable ML model. Cano et al. (2019) in particular, outlines the potential of monotonic classification in multiclass classification.

3. Methodology

In this section, only key features of the literature related to GBDT are described, the complete derivation is available in Friedman (2001).

3.1. Multiclass classification

For a classification problem with K classes, assuming a loss function of the following form:

$$L = \sum_{n=1}^{N} \sum_{k=1}^{K} y_{kn} \cdot \log(p_k(x_n)) \tag{1}$$

where N is the number of individuals in the dataset, $y_{kn} = 1$ if the choice i_n of the individual n is k else 0 and $p_{kn}(x_n)$ is the prediction for class k and individual n with features x_n. The predictions of a class are obtain through the softmax function, namely:

$$p_k(x_n) = \frac{F_k(x_n)}{\sum_{l=1}^{K} F_l(x_n)} \tag{2}$$

where F_k is the additive predictive function of a class k. At each boosting iteration m, K decision trees are induced and each predictive function is updated with the residual γ. Assuming that we have J terminal nodes resulting in J region R_{jkm} for tree of class k at iteration m, and that we approximate the greedy function with a single Newton step, we obtain:

$$\gamma_{jkm} = \frac{K}{K-1} \frac{\sum_{x_n \in R_{jkm}} y_{kn} - p_k(x_n)}{\sum_{x_n \in R_{jkm}} p_k(x_n)(y_{kn} - p_k(x_n))} \tag{3}$$

At each iteration the class predictive functions are updated with the following equation:

$$F_{k,m}(x_n) = F_{k,m-1}(x_n) + \sum_{j=1}^{J} \gamma_{jkm} r_{jkm} \tag{4}$$

where $r_{jkm} = 1$ if $x_n \in R_{jkm}$, 0 otherwise. At each iteration, the goal is to find the optimal split points resulting in regions R_{jkm} so that the loss is the smallest. For convenience, the problem is turned into a maximisation problem by taking the negative of the loss, such that the optimal weights are:

$$\gamma^*_{jkm} = \underset{\gamma_{jkm}}{\arg\max} \sum_{n=1}^{N} \sum_{k=1}^{K} y_{kn} \cdot \log\left(\frac{F_{k,m-1}(x_n) + \sum_{j=1}^{J} \gamma_{jkm} r_{jkm}}{\sum_{l=1}^{K} F_{l,m-1}(x_n) + \sum_{j=1}^{J} \gamma_{jkm} r_{jkm}} \right) \tag{5}$$

The summary of the algorithmic procedure is described in Algorithm 1 Friedman (2001).

Algorithm 1 Multiclass classification for GBDT

```
$F_{k0}(\mathbf{x_k}) = 0, \quad k = 1, ..., K$
for $m = 1$ to $M$ do
    $p_k(x_n) = \frac{F_k(x_n)}{\sum_{l=1}^{K} F_l(x_n)}, \quad k = 1, ..., K$
    for $k = 1$ to $K$ do
        $\{R_{jkm}\}_{j=1}^{J}$ = J-terminal node $tree(y_{nk} - p_k(x_n), x_n)_{n=1}^{N}$
        $\gamma_{jkm} = \frac{K}{K-1} \frac{\sum_{x_n \in R_{jkm}} y_{kn} - p_k(x_n)}{\sum_{x_n \in R_{jkm}} p_k(x_n)(y_{kn} - p_k(x_n))}, \quad j = 1, ..., J$
        $F_{k,m}(x_n) = F_{k,m-1}(x_n) + \sum_{j=1}^{J} \gamma_{jkm} r_{jkm}, \quad r_{jkm} = 1$ if $x_n \in R_{jkm}$ else 0
    end for
end for
```

3.2. Alternative specific features

Since at each boosting round K decision trees are induces, it is possible to take advantage of that to restrict the potential splitting points of a tree belonging to class k to a subset of feature $x_k \subseteq x$. The predictive functions F_k are then computed on a different set of attributes for each class k. Without loss of generality, we define:

$$x_k = a_k \cup s \tag{6}$$

where a_k are alternative specific features, hence they can be part of only the corresponding subset of features, and s the socio-economic features, which are allowed to be in several subset of features. The implementation is summarised on Algorithm 2, which is a modified version of Alorithm 1.

Algorithm 2 Alternative specific features

```
$\mathbf{x_k} = \mathbf{a_k} + \mathbf{s}, \quad k = 1, ..., K$
$F_{k0}(\mathbf{x_k}) = 0, \quad k = 1, ..., K$
for $m = 1$ to $M$ do
    $p_k(x_{k_n}) = \frac{F_k(x_{k_n})}{\sum_{l=1}^{K} F_l(x_{k_n})}, \quad k = 1, ..., K$
    for $k = 1$ to $K$ do
        $\{R_{jkm}\}_{j=1}^{J}$ = J-terminal node $tree(y_{nk} - p_k(x_{k_n}), x_{k_n})_{n=1}^{N}$
        $\gamma_{jkm} = \frac{K}{K-1} \frac{\sum_{x_{k_n} \in R_{jkm}} y_{kn} - p_k(x_{k_n})}{\sum_{x_{k_n} \in R_{jkm}} p_k(x_{k_n})(y_{kn} - p_k(x_{k_n}))}, \quad j = 1, ..., J$
        $F_{k,m}(x_{k_n}) = F_{k,m-1}(x_{k_n}) + \sum_{j=1}^{J} \gamma_{jkm} r_{jkm}, \quad r_{jkm} = 1$ if $x_{k_n} \in R_{jkm}$ else 0
    end for
end for
```

3.3. Feature interaction constraint

Feature interactions happen when there are 2 or more features in one tree. For the sake of interpretability, feature interaction in GBDT must be restricted, which means that trees must contain split point on one feature maximum, which is equivalent to have a maximum depth of 1 (i.e. 2 terminal nodes). Its implementation is on algorithm 3.

Algorithm 3 Feature interaction constraint

```
$\mathbf{x_k} = \mathbf{a_k} + \mathbf{s}, \quad k = 1, ..., K$
$F_{k0}(\mathbf{x_k}) = 0, \quad k = 1, ..., K$
for $m = 1$ to $M$ do
    $p_k(x_{k_n}) = \frac{F_k(x_{k_n})}{\sum_{l=1}^{K} F_l(x_{k_n})}, \quad k = 1, ..., K$
    for $k = 1$ to $K$ do
        $\{R_{jkm}\}_{j=1}^{2}$ = **2**-terminal node $tree(y_{nk} - p_k(x_{k_n}), x_{k_n})_{n=1}^{N}$
        $\gamma_{jkm} = \frac{K}{K-1} \frac{\sum_{x_{k_n} \in R_{jkm}} y_{kn} - p_k(x_{k_n})}{\sum_{x_{k_n} \in R_{jkm}} p_k(x_{k_n})(y_{kn} - p_k(x_{k_n}))}, \quad j = 1, 2$
        $F_{k,m}(x_{k_n}) = F_{k,m-1}(x_{k_n}) + \sum_{j=1}^{2} \gamma_{jkm} r_{jkm}, \quad r_{jkm} = 1$ if $x_{k_n} \in R_{jkm}$ else 0
    end for
end for
```

3.4. Monotonicity constraint

A positive monotonic constraint implies that an increasing feature value will also increase the predictive function, namely if a feature f is separated into two regions R_1 and R_2 with $f_1 \in R_1, f_2 \in R_2$ and $f_1 \cup f_2 = f$, such that we have $f_1 < f_2$ for any value in f_1, f_2 then:

$$f_1 < f_2 \Rightarrow \gamma_1 < \gamma_2 \tag{7}$$

Similarly, a negative monotonic constraint implies:

$$f_1 < f_2 \Rightarrow \gamma_1 > \gamma_2 \tag{8}$$

Without loss of generality, we will assume in this paper that a split point will always result on the left part of the split feature being smaller than the right part of the split feature. The implementation of this constraint in the GBDT algorithm is shown in Algorithm 4.

Algorithm 4 Final algorithm with Monotonicity constraint

```
$\mathbf{x_k} = \mathbf{a_k} + \mathbf{s}, \quad k = 1, ..., K$
Positive monotonic set of feature $\mathbf{pm_k} \subseteq \mathbf{x_k}$
Negative monotonic set of feature $\mathbf{nm_k} \subseteq \mathbf{x_k}$
$\mathbf{pm_k} \cap \mathbf{nm_k} = \varnothing$
$F_{k0}(\mathbf{x_k}) = 0, \quad k = 1, ..., K$
for $m = 1$ to $M$ do
    $p_k(x_{k_n}) = \frac{F_k(x_{k_n})}{\sum_{l=1}^{K} F_l(x_{k_n})}, \quad k = 1, ..., K$
    for $k = 1$ to $K$ do
        $\{R_{jkm}\}_{j=1}^{2}$ = 2-terminal node $tree(y_{nk} - p_k(x_{k_n}), x_{k_n})_{n=1}^{N}$
        if the regions $R_{jkm}$ are on a feature $f \in pm_k$ then
            $\gamma_{jkm} = \frac{K}{K-1} \frac{\sum_{x_{k_n} \in R_{jkm}} y_{kn} - p_k(x_{k_n})}{\sum_{x_{k_n} \in R_{jkm}} p_k(x_{k_n})(y_{kn} - p_k(x_{k_n}))}, \quad j = 1, 2$
            s. t. $\gamma_{1km} < \gamma_{2km}$
        else if the regions $R_{jkm}$ are on a feature $f \in nm_k$ then
            $\gamma_{jkm} = \frac{K}{K-1} \frac{\sum_{x_{k_n} \in R_{jkm}} y_{kn} - p_k(x_{k_n})}{\sum_{x_{k_n} \in R_{jkm}} p_k(x_{k_n})(y_{kn} - p_k(x_{k_n}))}, \quad j = 1, 2$
            s. t. $\gamma_{1km} > \gamma_{2km}$
        else
            $\gamma_{jkm} = \frac{K}{K-1} \frac{\sum_{x_{k_n} \in R_{jkm}} y_{kn} - p_k(x_{k_n})}{\sum_{x_{k_n} \in R_{jkm}} p_k(x_{k_n})(y_{kn} - p_k(x_{k_n}))}, \quad j = 1, 2$
        end if
        $F_{k,m}(x_{k_n}) = F_{k,m-1}(x_{k_n}) + \sum_{j=1}^{2} \gamma_{jkm} r_{jkm}, \quad r_{jkm} = 1$ if $x_{k_n} \in R_{jkm}$ else 0
    end for
end for
```

3.5. Interpretable non-linear utility function

With the three modifications applied in Sec. 3.2, 3.3 and 3.4, the predictive function is now a *interpretable utility* function. By summing all of the terminal nodes values for a feature f, it is possible to fully retrieve its impact on the utility function. Furthermore, this utility function is not bound to be *linear-in-parameters*, accounting for potential non-linearity in features impact on the mode choice.

4. Results

4.1. Dataset and underlying MNL model

The dataset used to train the RUMBoost model is the London Passenger Mode Choice (LPMC), a publicly available dataset containing more than 80000 observations. The dataset is an augmented version of the London Travel Demand Survey (LTDS) trip diary dataset, to include travel times and the cost of alternatives. The dataset contains observations from 17615 households over over a three-year period, and there are four possible alternatives: walking, driving, public transport and driving. The underlying MNL model used to create the RUMbooster is a 62-parameter model with alternative specific constants (ASC). Note that the walking ASC has been normalised to 0. As the utility function is no longer *linear-in-parameters* in RUMBoost, it is not necessary to normalise socio-economic features to 0 for one alternative. The model specification is summarised in Table 2.

This MNL model is used to specify the constraints of the RUMBoost model. The alternative-specific feature constraint is directly satisfied by the MNL utility specification. Interactions between features are restricted, so that trees with a maximum depth of 1 are induced. Finally, monotonicity constraints are obtained directly from the bounds that would be applied on the MNL beta parameters (see Table 3).

One big advantage of the RUMboost is that, due to its constraints, the model is not subject to overfitting. Therefore, no regularisation parameter is needed and hyperparameter search is not necessary. The model is trained using cross-entropy loss and the softmax function, which are common practices for multiclass classification. Lastly, the model is trained on the first two years of the dataset (i.e. around 2/3 of the observations) and is tested on the trips in the dataset performed in the third year.

Table 2: Attributes used in the underlying MNL model

	Walking	**Cycling**	**Public Transport**	**Driving**
alterative-specific attributes				
travel time	✓	✓	✓	✓
access time			✓	
transfer time			✓	
waiting time			✓	
# of PT changes			✓	
cost			✓	✓
generic attributes				
constant		✓	✓	✓
straight line distance	✓	✓	✓	✓
starting time	✓	✓	✓	✓
day of the week	✓	✓	✓	✓
gender	✓	✓	✓	✓
age	✓	✓	✓	✓
driving license	✓	✓	✓	✓
car ownership	✓	✓	✓	✓
purpose home-based work	✓	✓	✓	✓
purpose home-based education	✓	✓	✓	✓
purpose home-based other	✓	✓	✓	✓
purpose employers business	✓	✓	✓	✓
purpose non-home-based other	✓	✓	✓	✓
fueltype diesel	✓	✓	✓	✓
fueltype hybrid	✓	✓	✓	✓
fueltype petrol	✓	✓	✓	✓
fueltype average	✓	✓	✓	✓

Table 3: Features constrained to monotonicity

Monotonic negative	Travel time, cost
Monotonic positive	Car ownership*, driving license*

*Only for the driving alternative

4.2. *results with other ML*

The results of the RUMboost model are compared against 5 other state-of-the art ML models and one MNL models. Theses 6 models are directly taken from Martín-Baos et al. (2023) w where more details are given on the search for hyperparameters and their specifications. The models are compared with their cross entropy loss (CLE) on the test set. Results are shown in Table 4.

Table 4: Benchmark of classification on LPMC dataset (negative cross-entropy)

	MNL	NN	DNN	RF	SVM	XGBoost	**RUMBoost**
Martin-Baos et al. (2023)	0.7164	0.6728	0.6702	0.6900	0.6755	0.6568	0.6791

The model with the best CEL on this dataset is the GBDT-based XGBoost (0.6568), followed by the deep neural network (0.6702), the neural network (0.6728) and the support vector machine (0.6755). Our model then came 5th with 0.6791, ahead of the random forest (0.6900) and the MNL (0.7164). The model does not perform as well as the other ML models because of its relative lack of flexibility (i.e. no interaction between features, monotonicity constraints). However, the model still performs significantly better than MNL and RF.

4.3. Non-linear utility

The great advantage of using RUMBoost is that non-linear utility functions can be observed. These functions will be presented for travel time and age. Figure 1 shows the impact of travel times on utility functions. It should be noted that the public transport alternative includes both bus and train travel times. First of all, the curvature of the walking alternative (Figure 1a) and the driving alternative (Figure 1e) has a convex shape, which means that the increase in a short trip has a greater influence on the utility function than a longer trip. This contrasts with the two utility functions linked to public transport, where bus travel time has an approximately linear curve, and train travel time impact results in a concave utility function. For rail, this means that trips from 0 to 0.4 hour have a similar influence on the utility function, whereas trips of 0.6 to 1 hour have a very different impact on the utility function. Finally, the utility function of cycling is initially convex, then reaches a plateau between around 0.5 and 1 hour of travel time, and decreases rapidly after 1 hour.

The impact of the age on each alternative utility function are presented in Figure 2. This time, there are no monotonicity constraints on these features. We observe that age reduces the utility of walking and driving, with an essentially convex shape, which is in line with what might be expected. For public transport, it is at its lowest at younger and older ages, and peak around an age of 20. Finally, for the driving alternative, the utility is higher when the age is young and when it is old, meaning that kids and old people are more likely to take a car (driving alternative includes passenger). However, the lowest point is at the age of 20, which is consistent with the usual observations, given that this is an age mainly associated with students.

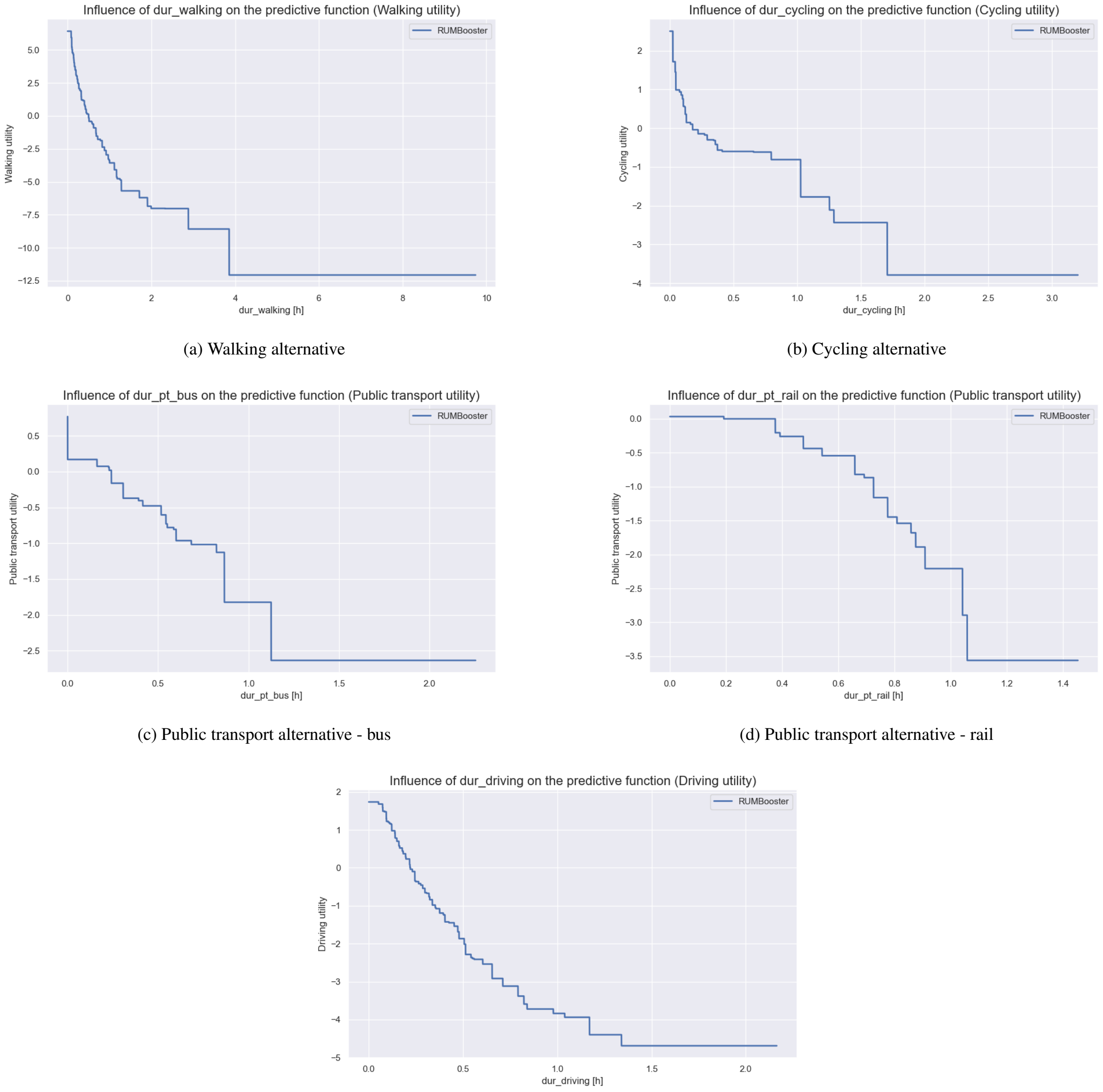
5. Conclusion and further work

The methodology presented in this paper allows for a fully interpretable ML model (RUMBoost), based on GBDT and inspired by random utility models. The three modifications made to the GBDT method consist of providing specific features for the alternatives, restricting the interactions between features in the decision trees and applying a monotonic constraint on the key features, based on domain knowledge. These modifications make it possible to considerably improve the predictions of a traditional MNL model and to derive non-linear utility functions. The utility function has a convex shape for walking and cycling travel time, a linear and concave shape for buses and trains travel time, and a non-linear shape with a plateau for cycling travel time. The impact of age is mainly reflected in a convex decrease in the utility function for walking and cycling, lower utility at younger and older ages for public transport, and an opposite behaviour to public transport for driving. These figures show how differently the travel time is perceived and how age influence the utility when travelling with a different mode.

It would be interesting to apply the RUMBoost model to other datasets to determine whether the shape of these utility functions is the same. In this way, we could assess whether these utility functions are just a bias of this dataset or whether the shapes really represent how humans perceive travel time in these different modes, and how age affects mode choice. The predictive performance is not as good as the GBDT benchmarks on this dataset. One solution to this problem would be to allow the interaction of two features per alternative. In this way, their impact on the utility function could always be observed on a 2D plot, which would give the model more flexibility, and therefore greater predictive potential. Another idea would be to combine categorical variables with continuous variables as latent variables. For example, if we intersect a binary variable such as whether the day of the trip is a weekday or not with the travel time, we could have two utility functions for the travel time, one for weekdays and one for weekends.

In this form, the model does not take account of the cross-effect of features on alternative utility functions. We could imagine adding an additional tree at each iteration that would capture this cross-effect by adding all the features that are not present in the corresponding alternative utility. In this way, we could also consider the interaction between features.

The non-linear utility functions obtained are not differentiable, since they are staircase functions. An approximation is required to obtain the elasticities, and the calculation fails on a fine scale. One way of relaxing this would be to transform these functions into piece-wise linear functions making them partially differentiable. Or the staircase functions



(a) Walking alternative

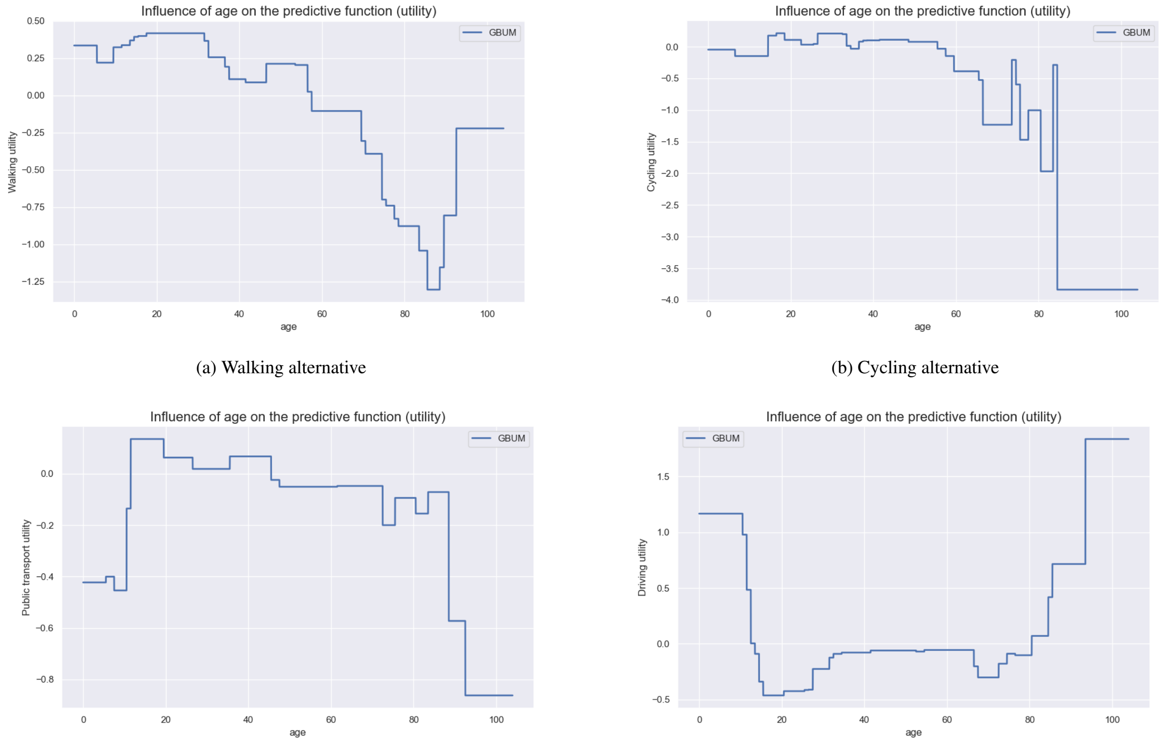
(b) Cycling alternative

(c) Public transport alternative - bus

(d) Public transport alternative - rail

(e) Driving alternative

Figure 1: Impact of travel time on the walking, cycling, public transport and driving utility function



(a) Walking alternative

(b) Cycling alternative

(c) Public transport alternative

(d) Driving alternative

Figure 2: Impact of age on the walking, cycling, public transport and driving utility function

could be used to help specify of piece-wise linear utility of RUM. Finally, the limitations of MNL models are well known and several improvements, such as the nested or crossed logit model, have been proposed. Since in this paper we compare results only with the MNL model, it would be interesting to extend the comparison to any Generalised Extreme Value (GEV) model.

References

Ben-Akiva, M.E., Lerman, S.R., 1985. Discrete choice analysis: theory and application to travel demand. MIT Press series in transportation studies, MIT Press, Cambridge, Mass.

Bierlaire, M., 2023. A short introduction to biogeme .

Cano, J.R., Gutiérrez, P.A., Krawczyk, B., Woźniak, M., García, S., 2019. Monotonic classification: An overview on algorithms, performance measures and data sets. Neurocomputing 341, 168–182. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0925231219302383`, doi:10.1016/j.neucom.2019.02.024.

Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. URL: `http://arxiv.org/abs/1603.02754`, doi:10.1145/2939672.2939785. arXiv:1603.02754 [cs].

Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. The Annals of Statistics 29. doi:10.1214/aos/1013203451.

Hillel, T., Bierlaire, M., Elshafie, M.Z., Jin, Y., 2021. A systematic review of machine learning classification methodologies for modelling passenger mode choice. Journal of Choice Modelling 38, 100221. URL: `https://linkinghub.elsevier.com/retrieve/pii/S1755534520300208`, doi:10.1016/j.jocm.2020.100221.

Hillel, T., Bierlaire, M., Elshafie, M., Jin, Y., 2019. Weak teachers: Assisted specification of discrete choice models using ensemble learning .

Hillel, T., Elshafie, M.Z.E.B., Jin, Y., 2018. Recreating passenger mode choice-sets for transport simulation: A case study of London, UK. Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction 171, 29–42. URL: `https://www.icevirtuallibrary.com/doi/10.1680/jsmic.17.00018`, doi:10.1680/jsmic.17.00018.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y., . LightGBM: A Highly Efficient Gradient Boosting Decision Tree .

Krueger, R., Daziano, R.A., 2022. Stated choice analysis of preferences for COVID-19 vaccines using the Choquet integral. Journal of Choice Modelling 45, 100385. URL: `https://linkinghub.elsevier.com/retrieve/pii/S1755534522000422`, doi:10.1016/j.jocm.2022.100385.

Martín-Baos, J., López-Gómez, J.A., Rodriguez-Benitez, L., Hillel, T., García-Ródenas, R., 2023. A prediction and behavioural analysis of machine learning methods for modelling travel mode choice. URL: `http://arxiv.org/abs/2301.04404`. arXiv:2301.04404 [cs].

McFadden, D., Tye, W.B., Train, K., 1977. An application of diagnostic tests for the independence from irrelevant alternatives property of the multinomial logit model. Institute of Transportation Studies, University of California Berkeley.

McFadden, D., et al., 1973. Conditional logit analysis of qualitative choice behavior .

Ortelli, N., Hillel, T., Pereira, F.C., de Lapparent, M., Bierlaire, M., 2021. Assisted specification of discrete choice models. Journal of Choice Modelling 39, 100285. URL: `https://linkinghub.elsevier.com/retrieve/pii/S175553452100018X`, doi:10.1016/j.jocm.2021.100285.

Sifringer, B., Lurkin, V., Alahi, A., 2020. Enhancing discrete choice models with representation learning. Transportation Research Part B: Methodological 140, 236–261. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0191261520303830`, doi:10.1016/j.trb.2020.08.006.

Train, K.E., 2009. Discrete choice methods with simulation. Cambridge university press.

Wang, S., Mo, B., Zhao, J., 2021. Deep Neural Networks for Choice Analysis: Architectural Design with Alternative-Specific Utility Functions. URL: `http://arxiv.org/abs/1909.07481`. arXiv:1909.07481 [cs, econ, q-fin, stat].

Wang, S., Wang, Q., Zhao, J., 2020. Deep neural networks for choice analysis: Extracting complete economic information for interpretation. Transportation Research Part C: Emerging Technologies 118, 102701. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0968090X20306161`, doi:10.1016/j.trc.2020.102701.

Wong, M., Farooq, B., 2021. ResLogit: A residual neural network logit model for data-driven choice modelling. Transportation Research Part C: Emerging Technologies 126, 103050. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0968090X21000802`, doi:10.1016/j.trc.2021.103050.