Beyond i.i.d: complex Random Utility Model specifications with gradient boosting

Nicolas Salvadé*[1] and Tim Hillel[2]

[1]PhD Student, Department of Civil, Environmental and Geomatic Engineering, University College London, United Kingdom
[2]Lecturer, Department of Civil, Environmental and Geomatic Engineering, University College London, United Kingdom

Short summary

This paper extends RUMBoost, a novel discrete choice modelling approach that combines the interpretability and behavioural robustness of Random Utility Models with the generalisation and predictive ability of deep learning methods, to complex RUM specifications. With RUMBoost, we obtain non-linear *pseudo-utilities* in the form of piece-wise constants by replacing each linear parameter in the utility functions of a RUM with an ensemble of gradient boosted regression trees. We further use an optimisation-based smoothing technique to identify non-linear utility functions with defined gradients from the piece-wise constants. This allows for the estimation of behavioural indicators such as the Value of Time (VoT) or the willingness to pay. Finally, we demonstrate how RUMBoost can mimic the estimation of complex model specifications with a case study on a mode choice dataset. This is achieved by adapting the probability function to account for alternative correlations in the error term.
Keywords: Discrete Choice, Ensemble Learning, Machine Learning, Mode Choice, Random Utility

1 Introduction and literature review

Discrete choice models (DCMs), based on Random Utility theory, have been used extensively to model choices over the last 50 years (Ben-Akiva & Lerman, 1985; Train, 2009). DCMs have many desirable qualities: most crucially, their parametric form is directly interpretable and allows for the integration of expert knowledge consistent with behavioural theory. In addition, they enable the derivation of key behavioural indicators, such as elasticities and Value of Time (VoT), used to inform transport policies. However, their linear-in-parameters utility functions are relatively inflexible and must be specified in advance by the modeller. As such, these models may fail to capture complex phenomena and non-linear effects in human behaviour.

There have been numerous attempts to apply machine learning (ML) probabilistic classification algorithms to investigate choice behaviour. These models exhibit high predictive performance and, thanks to their data-driven nature, do not require any utility functions to be specified in advance of model estimation. However, they lack an underlying behavioural model and so it is not possible to guarantee consistency of forecasts or derive behavioural indicators such as Value of Time (VoT) or willingness-to- pay. Initial approaches for analysing these models from a behavioural perspective rely on approximating the partial derivatives of the output probabilities of unconstrained ML classifiers in order to define elasticities for variables of interest (Wang et al., 2020; Martín-Baos et al., 2023). Unlike marginal utilities from a DCM, the probability derivatives of ML classifiers provide only a numeric estimate of the point elasticities at observed data points. Furthermore, as the underlying models are unconstrained, they exhibit several qualities that are inconsistent with random utility theory. As such, these techniques have seen limited real-world use and practitioners continue to rely predominantly on parametric DCMs. That being said, the ability of ML models to capture complex non-linear relationships as well as their improved predictive accuracy makes them an attractive proposition.

In response to these limitations, there has been an emergence of hybrid *data-driven utility models* in recent years, that attempt to combine the benefits of ML and DCMs. These can largely be grouped into two different approaches:

1. adding additional constraints to machine learning models (e.g. monotonicity, alternative specific attributes, etc) so that their output can mimic DCM utility values (Wang et al., 2021; Martín-Baos et al., 2021; Sifringer et al., 2020; Wong & Farooq, 2021; Kim & Bansal, 2023; Krueger & Daziano, 2022; Aboutaleb, 2022); and
2. using data-driven approaches to automate or assist with identifying suitable parametric utility functions (Han et al., 2022; Ortelli et al., 2021; Hillel et al., 2019).

While some studies incorporate key features of DCMs such as individual-specific parameters, monotonicity constraints and intrinsically interpretable utility functions, none of them are able to completely identify automatically non-linear utility functions.

Random Utility Models with Boosting (RUMBoost) (Salvadé & Hillel, 2024) combines GBDTs' predictive power with DCMs' interpretability and behavioural consistency. At a high level, RUMBoost replaces each parameter in the utility specifications of traditional DCMs with an ensemble of regression trees, allowing for non-linear parameters to be extracted directly from data. Algorithmically, RUMBoost consists of two parts: (i) *Gradient Boosted Utility Values* (GBUV), where ensembles of regression trees are used to impute piece-wise constant values for each parameter in each utility specification; and (ii) *Piece-wise Cubic Utility Functions* (PCUF), where monotonic piece-wise cubic splines are optimised to fit the GBUV outputs, to allow for a defined gradient for each parameter where elasticities are needed.

This paper focuses on adapting RUMBoost to complex RUM specifications[1]. RUMBoost is implemented in Python, with code available on GitHub (`https://github.com/NicoSlvd/RUMBoost`).

2 Methodology

We first explain here how we adapt the general GBDT model to output Gradient Boosted Utility Values (GBUV) to emulate parametric RUMs. We then present the Piecewise-Cubic Utility Functions (PCUF) algorithm, that outputs smoothed monotonic non-linear parameters.

Gradient Boosted Utility Values (GBUV)

In RUMBoost-GBUV, we replace each parameter in the utility functions of a RUM with an ensemble of regression trees, where the leaves in the regression trees represent the partial utility contribution for the corresponding value of that variable. These can then be added over each tree in the ensemble to find the contribution of each variable to the utility. The overall utility for each alternative can therefore be found by summing the ensembles for each variable over all variables in the utility function. For K parameters applied to K variables, we have:

$$V_{in} = \mathrm{ASC}_i + \sum_{k}^{K_i} \sum_{m}^{M_{ik}} f_{imk}(x_{ink}) \tag{1}$$

where ASC_i is an Alternative-Specific Constant for alternative i and M_{ik} is the number of regression trees in the ensemble for parameter k for alternative i. Probabilities for each alternative can then be calculated with the appropriate transformation (e.g. softmax for the MNL). In a Nested Logit (NL) model, the probability of choosing alternative i is:

$$P(i) = P(i|m)P(m) \tag{2}$$

where the probability of choosing i knowing the nest m is:

$$P(i|m) = \frac{e^{\mu_m V_i}}{\sum_{j \in m} e^{\mu_m V_j}}, \tag{3}$$

while the probability to choose the nest m is:

$$P(m) = \frac{e^{\tilde{V}_m}}{\sum_{p=1}^{M} e^{\tilde{V}_p}}, \tag{4}$$

where:

[1] Note to reviewers: GBUV methodology is presented at IATBR; this paper details PCUF and complex model specifications. For a more detailed GBUV methodology, refer to Salvadé & Hillel (2024)

- $\tilde{V}_m = \frac{1}{\mu_m} \ln \left(\sum_{i \in m} e^{\mu_m V_i} \right)$
- M the number of nest
- μ_m the scaling parameter of nest m

In a Cross-Nested Logit (CNL) model with M nests, the probability of choosing alternative i is:

$$P(i) = \sum_{m=1}^{M} P(i|m)P(m) \tag{5}$$

where the probability of choosing i knowing the nest m is:

$$P(i|m) = \frac{\alpha_{im}^{\mu_m} e^{\mu_m V_i}}{\sum_{j \in m} \alpha_{jm}^{\mu_m} e^{\mu_m V_j}}, \tag{6}$$

while the probability to choose the nest m is:

$$P(m) = \frac{\left(\sum_{j \in m} \alpha_{jm}^{\mu_m} e^{\mu_m V_j} \right)^{\frac{1}{\mu_m}}}{\sum_{n=1}^{M} \left(\sum_{j \in n} \alpha_{jn}^{\mu_n} e^{\mu_n V_j} \right)^{\frac{1}{\mu_n}}}, \tag{7}$$

Note that in all these equations, we omitted the scaling parameter of the error term μ (normalised to 1). Finally, these transformations can be used within the cross-entropy loss to form a basis (with the first and second derivatives of the loss) for boosting trees at each iteration.

Piece-wise Cubic Utility Function (PCUF)

The GBUV ensembles for each parameter in Section 2 are non-continuous, and so have a gradient of either zero or infinite at any point. However, many behavioural indicators require the utility function to have defined gradient to be computed. Therefore, we interpolate the utility values into a smooth function using piece-wise cubic Hermite splines. Using the approach introduced by Fritsch & Carlson (1980), it is possible to guarantee monotonic splines, as required. The interpolation must satisfy two conflicting objectives: (i) fitting the data as well as possible to maintain good predictive power on out-of-sample data; and (ii) being as smooth as possible to obtain relevant behavioural indicators.

The first objective favours a higher number of knots, while the second aims for a lower number so that the derivative is well defined. A natural objective function to capture the trade-off of both these objectives is the Bayesian Information Criterion (BIC), which takes the following form:

$$BIC = -2N \cdot L + df \cdot \ln(N) \tag{8}$$

where L is the loss function, df is the degree of freedom of the model, and N is the number of observations.

RUMBoost-PCUF, therefore, has two parameters to tune: (i) the number of knots; and (ii) their positions. Given a sequence of $Q+1$ knots $a_k = t_{0,k} < t_{1,k} < ... < t_{Q,k} = b_k$ for an attribute k where a_k and b_k are the domain where that attribute is defined, the optimal positions and numbers of knots are determined by the following optimisation problem:

$$\begin{aligned} \min_{t_{q,k}} \quad & -2N \cdot L + df \cdot ln(N) \\ \text{s.t.} \quad & t_{q+1,k} - t_{q,k} > 0 && \forall q = 0, ..., Q-1, \forall k \\ & t_{0,k} = a_k && \forall k \\ & t_{Q,k} = b_k && \forall k \end{aligned} \tag{9}$$

Given the number of knots, there is an optimal position of knots that minimises the loss function. Therefore, the two hyperparameters can be tuned sequentially: the number of knots is selected first and their optimal positions are found with a constrained optimisation solver afterwards.

Code and implementation

We implement the model in Python, making use of the library LightGBM for the utility regression ensembles (Ke et al., 2017). We have implemented an interface which allows the modeller to specify:

- which attributes should be included in each utility function
- control attribute interactions
- specify which attributes should have monotonic marginal utilities.

The code is freely available on Github (`https://github.com/NicoSlvd/RUMBoost`)

3 Results and discussion

We apply our methodology to a case study, where we benchmark RUMBoost against three DCMs (MNL, NL, and CNL). These models are re-implemented from Martín-Baos et al. (2023) (The code is freely available at `https://github.com/JoseAngelMartinB/prediction-behavioural-analysis-ml-travel-mode-choice`) and have respectively 62, 63 and 65 parameters (see Appendix C). When estimating the DCMs, we normalise the ASC, the generic attributes and the socio-economic characteristics of the walking alternative to zero. For the NL and CNL, we use the same utility specification as in the MNL, but with nests arbitrarily defined as motorised alternatives (PT and Driving) for NL and CNL and flexible alternatives (Walking, Cycling and Driving) for CNL only.
We use the London Passenger Mode Choice (LPMC) (Hillel et al., 2018) dataset for our case study, a publicly available dataset providing details of more than 80000 trips in London, alongside their associated mode choice decisions. The dataset contains observations from 17615 households over a three-year period, and there are four possible alternatives: walking, cycling, public transport and driving. We train/estimate the models on the first two years of observations and test them on the third year of observations.

RUMBoost model specification

The DCMs are directly used to specify the constraints of the RUMBoost models. We use the same variables and the alternative-specific attributes constraint is directly satisfied by their utility specifications. Interactions between attributes are restricted, such that each tree corresponds to a single parameter. Finally, monotonicity constraints are applied negatively on travel time, headway, cost, congestion rate (only for driving) and distance, and positively on car ownership and driving license (only for driving).
For RUMBoost-PCUF, we apply the smoothing process on all monotonic alternative-specific attributes. We make use of the SciPy (Virtanen et al., 2020) implementation of monotonic cubic splines (Fritsch & Carlson, 1980) to smooth the GBUV outputs to produce piece-wise cubic utility functions. We make use of the Hyperopt (Bergstra et al., 2013) Python library to identify the optimal number of knots. Each search in Hyperopt involves selecting a different number of knots, constrained to be an integer value between a minimum of 3 and up to 8. In total, 25 searches are conducted (i.e. 25 different combinations of numbers of knots for each variable). The inner optimisation loop then identifies optimal knot locations, given a fixed number of knots for each variable, using the SLSQP (Sequential Least Squares Quadratic Programming) algorithm, implemented in SciPy. We constrain the first and last knots to be at the location of the first and last observations for each variable. The optimised number of knots for each variable are shown in Table 1.
We also include RUMBoost-NL and RUMBoost-CNL which are RUMBoost-GBUV with NL and CNL probability functions. For both models, we treat μ (the scaling parameter of the nest) and α (the nest membership parameter for a CNL model) as hyperparameters (see Appendix B). Note that we did not apply the smoothing process PCUF on the two models, but, thanks to the modularity of our approach, they are completely suitable for PCUF.

Comparison with other ML models and DCMs

We compare the RUMBoost models and the DCMs with their cross-entropy loss (CLE) on the test set (lower is better) and their computational time per cross-validation iteration. The results are shown in Table 2.
Overall, all RUMBoost models outperform their respective DCMs on both training and testing validations, whilst still ensuring a directly interpretable functional form. Interestingly, the loss of information due to smoothing is minimal, and the CE loss even improves on the LPMC dataset, even with an objective function penalising complex models. Therefore, we deduce that the piece-wise splines act as further regularisation of the RUMBoost-GBUV model. We also observe that

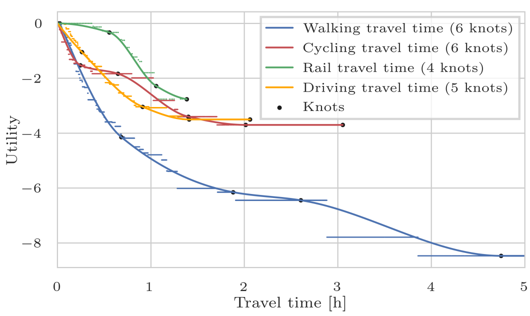
Table 1: The optimal number of knots for PCUF. The number of knots is chosen with a hyperparameter search of 25 iterations. The straight-line distance is not included for public transport as there were no regression trees in the parameter ensemble.

Attributes	Number of knots
Walking	
Travel time	6
Distance	6
Cycling	
Travel time	6
Distance	6
Public transport	
Rail travel time	3
Bus travel time	4
Access travel time	4
Interchange waiting time	8
Interchange walking time	7
Cost	3
Driving	
Travel time	5
Distance	4
Congestion rate	8
Cost	3

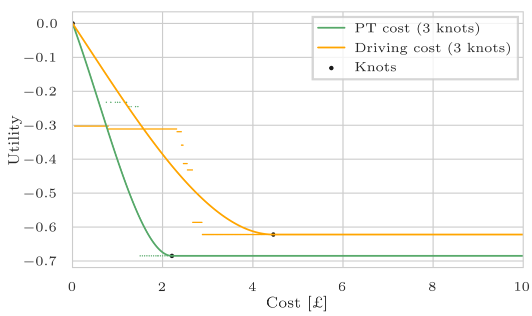
both RUMBoost-NL and RUMBoost-CNL improve the prediction compared to RUMBoost with a softmax predictive function. However, this is at the cost of higher computational time. These models still exhibit significantly better scalability than the DCMs.

Interpretability

The primary advantage of using RUMBoost over other unconstrained ML algorithms is that we have full interpretability of the model. We present here the piece-wise cubic utility functions obtained from RUMBoost-PCUF for the travel time and cost parameters in Figure 1. The derivatives from PCUF can be used to compute the VoT for the PT and driving alternative from the attributes space. This is shown in Figure 2. For comparison, we also compute the VoT for each individual in the population in Figure 3.



(a) Travel time



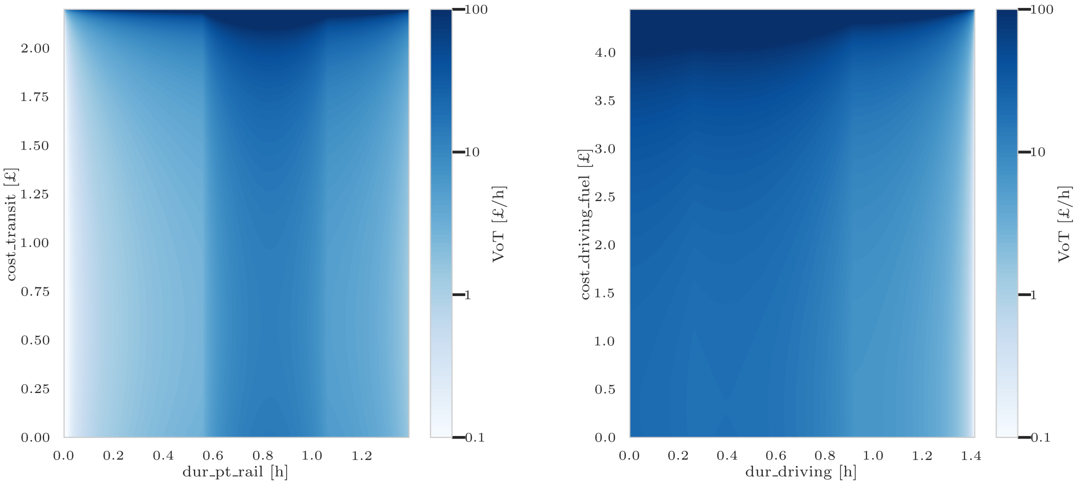
(b) Cost

Figure 1: Piece-wise monotonic cubic spline interpolation of a) travel time and b) cost on the LPMC dataset. The knots are drawn in black and the first and last knots are omitted for clarity. The GBUV used for interpolation are plotted as a scatter plot.

Table 2: Benchmark of Classification on LPMC Dataset. The models are compared with their CEL (negative Cross-Entropy Loss, lower the better) on the test set and their computational time for one CV iteration. The training results of RUMBoost-PCUF are the results of the optimisation problem described in Section 2.

Models	**Metrics**	**LPMC** 5 fold CV	Holdout test set
MNL	CEL	0.6913	0.7085
	Comp. Time [s]	242.14	-
NL	CEL	0.6921	0.7091
	Comp. Time [s]	1067.04	-
CNL	CEL	0.6908	0.7070
	Comp. Time [s]	5120.01	-
RUMBoost-GBUV	CEL	0.6570	0.6737
	Comp. Time [s]	6.48	-
RUMBoost-PCUF	CEL	0.6479*	0.6730
	Comp. Time [s]	712.48*	-
RUMBoost-NL	CEL	0.6568	0.6731
	Comp. Time [s]	48.53	-
RUMBoost-CNL	CEL	0.6546	**0.6716**
	Comp. Time [s]	183.91	-

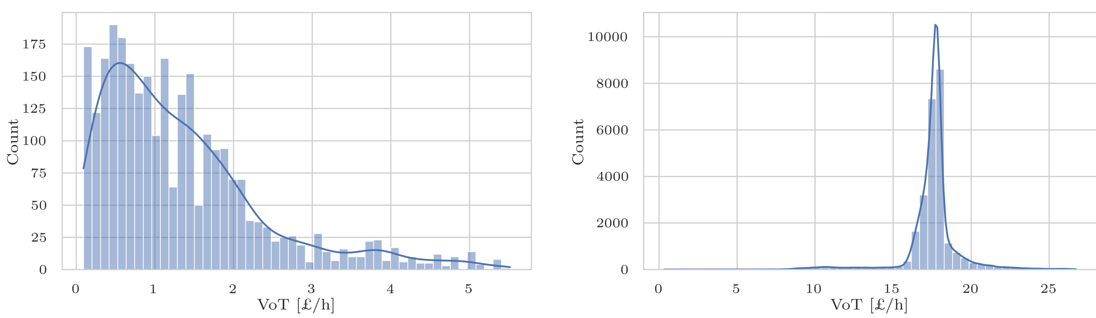
*Not with CV



(a) Rail (LPMC dataset)

(b) Driving alternative (LPMC dataset)

Figure 2: Value of Time (VoT) for a) rail, b) driving. The VoT is capped at 100£/h, and displayed only where the utility functions derivatives are non zero.



(a) PT alternative

(b) Driving alternative

Figure 3: Histogram of the population Value of Time (VoT) for a) rail, b) driving. The observations with zero travel times, as well as the highest 0.1% VoT values are excluded. The solid line represents the kernel density estimates.

4 Conclusion and further work

We showed in this paper that the modularity of RUMBoost allows for the estimation of complex model specifications such as an error term accounting for correlation within alternatives. In addition, our approach offers to observe the full functional form of the utility function with a defined gradient, just like in DCMs. The key difference is that the utility function is directly learnt from the data. In addition, the smoothing algorithm enables the calculation of behavioural indicators such as the VoT. As the traditional parameters of DCMs are replaced with functions depending on variable values, the VoT is represented as a contour plot. This enables us to observe the VoT with more nuances with respect to the interaction of travel time and cost.
Whilst applied here to choice models, this methodology could be used in place of any linear-in-parameters models, for regression, classification, or any task for which the gradient and hessian of the cost function are well defined. Further work includes applying the model to various problems to demonstrate this statement. The PCUF algorithm could be improved by applying B-splines, which would provide a C^2 monotonic interpolation of the data, where shape constraint could be incorporated. Finally, the GBUV could be computed directly with linear trees, quadratic trees or splines, to obtain directly piece-wise utility functions with defined gradient.

Acknowledgements

We sincerely thank Prof. Michel Bierlaire for his guidance on the smoothing process. We also would like to express our sincere gratitude to Dr. José Ángel Martín-Baos for his assistance with the benchmarks on the LPMC.

References

Aboutaleb, Y. M. (2022). *Theory-constrained data-driven model selection, specification, and estimation: Applications in discrete choice models* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Ben-Akiva, M. E., & Lerman, S. R. (1985). *Discrete choice analysis: theory and application to travel demand*. Cambridge, Mass: MIT Press.

Bergstra, J., Yamins, D., & Cox, D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning* (pp. 115–123).

Fritsch, F. N., & Carlson, R. E. (1980). Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, *17*(2), 238–246.

Han, Y., Pereira, F. C., Ben-Akiva, M., & Zegras, C. (2022). A neural-embedded discrete choice model: Learning taste representation with strengthened interpretability. *Transportation Research Part B: Methodological*, *163*, 166–186.

Hillel, T., Bierlaire, M., Elshafie, M., & Jin, Y. (2019). Weak teachers: Assisted specification of discrete choice models using ensemble learning. In *8th symposium of the european association for research in transportation, budapest.*

Hillel, T., Elshafie, M. Z. E. B., & Jin, Y. (2018, March). 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*(1), 29–42. Retrieved 2023-06-19, from 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., … Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*.

Kim, E.-J., & Bansal, P. (2023). A new flexible and partially monotonic discrete choice model. *Available at SSRN 4448172*.

Krueger, R., & Daziano, R. A. (2022, December). Stated choice analysis of preferences for COVID-19 vaccines using the Choquet integral. *Journal of Choice Modelling*, *45*, 100385. Retrieved 2023-06-19, from https://linkinghub.elsevier.com/retrieve/pii/S1755534522000422 doi: 10.1016/j.jocm.2022.100385

Martín-Baos, J. Á., Garcia-Rodenas, R., & Rodriguez-Benitez, L. (2021). Revisiting kernel logistic regression under the random utility models perspective. an interpretable machine-learning approach. *Transportation Letters*, *13*(3), 151–162.

Martín-Baos, J. Á., López-Gómez, J. A., Rodriguez-Benitez, L., Hillel, T., & García-Ródenas, R. (2023, November). A prediction and behavioural analysis of machine learning methods for modelling travel mode choice. *Transp. Res. Part C Emerg. Technol.*, *156*(104318), 104318.

Ortelli, N., Hillel, T., Pereira, F. C., de Lapparent, M., & Bierlaire, M. (2021, June). Assisted specification of discrete choice models. *Journal of Choice Modelling*, *39*, 100285. Retrieved 2023-06-19, from https://linkinghub.elsevier.com/retrieve/pii/S175553452100018X doi: 10.1016/j.jocm.2021.100285

Salvadé, N., & Hillel, T. (2024). Rumboost: Gradient boosted random utility models. *arXiv preprint arXiv:2401.11954*.

Sifringer, B., Lurkin, V., & Alahi, A. (2020, October). Enhancing discrete choice models with representation learning. *Transportation Research Part B: Methodological*, *140*, 236–261. Retrieved 2023-06-19, from 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.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. doi: 10.1038/s41592-019-0686-2

Wang, S., Mo, B., & Zhao, J. (2021, April). Deep Neural Networks for Choice Analysis: Architectural Design with Alternative-Specific Utility Functions. Retrieved 2023-06-19, from http://arxiv.org/abs/1909.07481 (arXiv:1909.07481 [cs, econ, q-fin, stat])

Wang, S., Wang, Q., & Zhao, J. (2020, September). Deep neural networks for choice analysis: Extracting complete economic information for interpretation. *Transportation Research Part C: Emerging Technologies*, *118*, 102701. Retrieved 2023-06-19, from https://linkinghub.elsevier.com/retrieve/pii/S0968090X20306161 doi: 10.1016/j.trc.2020.102701

Wong, M., & Farooq, B. (2021, May). ResLogit: A residual neural network logit model for data-driven choice modelling. *Transportation Research Part C: Emerging Technologies*, *126*, 103050. Retrieved 2023-06-19, from https://linkinghub.elsevier.com/retrieve/pii/S0968090X21000802 doi: 10.1016/j.trc.2021.103050

A DCMs utility specification

Table 3: Variables used in the LPMC RUMBoost and DCMs. For the DCMs estimation, the socio-economic characteristics and generic attributes are normalised to 0 for the walking alternative. Purpose and Fuel type are dummy variables where one category is normalised to 0. The constants are not included in the RUMBoost training, but are reconstructed afterwards.

	Walking	**Cycling**	**Public Transport**	**Driving**
Alternative-specific attributes				
Constant		✓	✓	✓
Travel time	✓	✓	✓	✓
Access time			✓	
Transfer time			✓	
Waiting time			✓	
Num. of PT changes			✓	
Cost			✓	✓
Congestion rate				✓
Congestion charge				✓
Socio-economic characteristics and generic attributes				
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	✓	✓	✓	✓
Fuel type: diesel	✓	✓	✓	✓
Fuel type: hybrid	✓	✓	✓	✓
Fuel type: petrol	✓	✓	✓	✓
Fuel type: average	✓	✓	✓	✓

B Hyperparameter search

Table 4: Hyperparameter search and optimal value for RUMBoost-GBUV, RUMBoost-Nested, RUMBoost-CN and RUMBoost-FE on the LPMC dataset

			RUMBoost			
			GBUV	**Nested**	**CN**	**Fe**
Number of searches			1	25	50	100
Time [s]			44.5	5378	51643	5366
Hyperparameter	*Distribution*	*Search space*				
Mean of CV num_iterations	early stopping	-	1300	1256	2024	1099
bagging_fraction	uniform	[0.5, 1]	-	-	-	0.700
bagging_freq	choice	(0, 1, 5, 10)	-	-	-	10
feature_fraction	uniform	[0.5, 10]	-	-	-	0.867
lambda_l1	log uniform	[0.0001, 10]	-	-	-	6.592
lambda_l2	log uniform	[0.0001, 10]	-	-	-	1.028
learning_rate	fixed	0.1	-	-	-	0.1
max_bin	uniform	[100, 500]	-	-	-	131
min_data_in_leaf	uniform	[1, 200]	-	-	-	27
min_gain_to_split	log uniform	[0.0001, 5]	-	-	-	0.800
min_sum_hessian_in_leaf	log uniform	[1,100]	-	-	-	1.783
num_leaves	uniform	[2, 100]	-	-	-	74
$mu_{motorised}$	uniform	[1, 2]	-	1.167	1.821	-
$mu_{existing}$	uniform	[1, 2]	-	-	1.000	-
$alpha_{driving}$*	uniform	[1, 2]	-	-	0.364	-

$alpha_{driving}$ is the degree of membership of driving to the motorised nest

C Estimation of the DCMs

Table 5: Parameter estimates of the LPMC MNL. Out of the 62 parameters, 9 are not significant at a 95% confidence interval.

LPMC - MNL			
	Value	Active bound	Rob. p-value
ASC_Bike	-3.380	0.000	0.000
ASC_Car	-2.592	0.000	0.000
ASC_Public_Transport	-1.908	0.000	0.000
B_age_Bike	-0.004	0.000	0.032
B_age_Car	0.005	0.000	0.000
B_age_Public_Transport	0.011	0.000	0.000
B_car_ownership_Bike	0.036	0.000	0.590
B_car_ownership_Car	0.694	0.000	0.000
B_car_ownership_Public_Transport	-0.213	0.000	0.000
B_con_charge_Car	-1.147	0.000	0.000
B_cost_driving_fuel_Car	0.000	1.000	1.000
B_cost_transit_Public_Transport	-0.115	0.000	0.000
B_day_of_week_Bike	-0.020	0.000	0.201
B_day_of_week_Car	0.030	0.000	0.000
B_day_of_week_Public_Transport	-0.044	0.000	0.000
B_distance_Bike	-0.232	0.000	0.040
B_distance_Car	0.000	1.000	1.000
B_distance_Public_Transport	0.000	1.000	1.000
B_driving_license_Bike	0.678	0.000	0.000
B_driving_license_Car	0.663	0.000	0.000
B_driving_license_Public_Transport	-0.526	0.000	0.000
B_dur_cycling_Bike	-2.670	0.000	0.000
B_dur_driving_Car	-4.777	0.000	0.000
B_dur_pt_access_Public_Transport	-4.608	0.000	0.000
B_dur_pt_bus_Public_Transport	-1.911	0.000	0.000
B_dur_pt_int_waiting_Public_Transport	-4.284	0.000	0.000
B_dur_pt_int_walking_Public_Transport	-2.335	0.000	0.027
B_dur_pt_rail_Public_Transport	-1.338	0.000	0.000

Table 5: Parameter estimates of the LPMC MNL. Out of the 62 parameters, 9 are not significant at a 95% confidence interval.

LPMC - MNL			
	Value	Active bound	Rob. p-value
B_dur_walking_Walk	-8.596	0.000	0.000
B_female_Bike	-0.834	0.000	0.000
B_female_Car	0.100	0.000	0.002
B_female_Public_Transport	0.160	0.000	0.000
B_fueltype_Avrg_Bike	-0.691	0.000	0.000
B_fueltype_Avrg_Car	-1.400	0.000	0.000
B_fueltype_Avrg_Public_Transport	-0.221	0.000	0.000
B_fueltype_Diesel_Bike	-0.822	0.000	0.000
B_fueltype_Diesel_Car	-0.228	0.000	0.000
B_fueltype_Diesel_Public_Transport	-0.419	0.000	0.000
B_fueltype_Hybrid_Bike	-1.000	0.000	0.000
B_fueltype_Hybrid_Car	-0.721	0.000	0.000
B_fueltype_Hybrid_Public_Transport	-0.945	0.000	0.000
B_fueltype_Petrol_Bike	-0.867	0.000	0.000
B_fueltype_Petrol_Car	-0.242	0.000	0.000
B_fueltype_Petrol_Public_Transport	-0.323	0.000	0.000
B_pt_n_interchanges_Public_Transport	-0.101	0.000	0.154
B_purpose_B_Bike	-0.029	0.000	0.775
B_purpose_B_Car	-0.043	0.000	0.543
B_purpose_B_Public_Transport	-0.012	0.000	0.874
B_purpose_HBE_Bike	-1.054	0.000	0.000
B_purpose_HBE_Car	-0.756	0.000	0.000
B_purpose_HBE_Public_Transport	-0.237	0.000	0.000
B_purpose_HBO_Bike	-0.773	0.000	0.000
B_purpose_HBO_Car	-0.352	0.000	0.000
B_purpose_HBO_Public_Transport	-0.442	0.000	0.000
B_purpose_HBW_Bike	-0.291	0.000	0.000
B_purpose_HBW_Car	-1.062	0.000	0.000
B_purpose_HBW_Public_Transport	-0.502	0.000	0.000
B_purpose_NHBO_Bike	-1.233	0.000	0.000
B_purpose_NHBO_Car	-0.379	0.000	0.000
B_purpose_NHBO_Public_Transport	-0.715	0.000	0.000
B_start_time_linear_Bike	0.017	0.000	0.015
B_start_time_linear_Car	0.027	0.000	0.000
B_start_time_linear_Public_Transport	0.010	0.000	0.016
B_traffic_perc_Car	-2.404	0.000	0.000

Table 6: Parameter estimates of the LPMC NL. Out of the 63 parameters, 10 are not significant at a 95% confidence interval.

LPMC - NL			
	Value	Active bound	Rob. p-value
ASC_Bike	-3.346	0.000	0.000
ASC_Car	-2.439	0.000	0.000
ASC_Public_Transport	-1.969	0.000	0.000
B_age_Bike	-0.004	0.000	0.026
B_age_Car	0.007	0.000	0.000
B_age_Public_Transport	0.011	0.000	0.000
B_car_ownership_Bike	0.062	0.000	0.351
B_car_ownership_Car	0.628	0.000	0.000
B_car_ownership_Public_Transport	-0.037	0.000	0.369
B_con_charge_Car	-0.816	0.000	0.000

Table 6: Parameter estimates of the LPMC NL. Out of the 63 parameters, 10 are not significant at a 95% confidence interval.

LPMC - NL			
	Value	Active bound	Rob. p-value
B_cost_driving_fuel_Car	0.000	1.000	1.000
B_cost_transit_Public_Transport	-0.077	0.000	0.000
B_day_of_week_Bike	-0.023	0.000	0.155
B_day_of_week_Car	0.022	0.000	0.007
B_day_of_week_Public_Transport	-0.033	0.000	0.000
B_distance_Bike	-0.225	0.000	0.042
B_distance_Car	-0.005	0.000	0.960
B_distance_Public_Transport	0.000	1.000	1.000
B_driving_license_Bike	0.705	0.000	0.000
B_driving_license_Car	0.484	0.000	0.000
B_driving_license_Public_Transport	-0.396	0.000	0.000
B_dur_cycling_Bike	-1.839	0.000	0.003
B_dur_driving_Car	-3.409	0.000	0.000
B_dur_pt_access_Public_Transport	-3.410	0.000	0.000
B_dur_pt_bus_Public_Transport	-1.445	0.000	0.000
B_dur_pt_int_waiting_Public_Transport	-3.036	0.000	0.000
B_dur_pt_int_walking_Public_Transport	-1.876	0.000	0.017
B_dur_pt_rail_Public_Transport	-1.085	0.000	0.000
B_dur_walking_Walk	-8.171	0.000	0.000
B_female_Bike	-0.831	0.000	0.000
B_female_Car	0.112	0.000	0.000
B_female_Public_Transport	0.156	0.000	0.000
B_fueltype_Avrg_Bike	-0.680	0.000	0.000
B_fueltype_Avrg_Car	-1.175	0.000	0.000
B_fueltype_Avrg_Public_Transport	-0.322	0.000	0.000
B_fueltype_Diesel_Bike	-0.817	0.000	0.000
B_fueltype_Diesel_Car	-0.251	0.000	0.000
B_fueltype_Diesel_Public_Transport	-0.391	0.000	0.000
B_fueltype_Hybrid_Bike	-0.985	0.000	0.000
B_fueltype_Hybrid_Car	-0.755	0.000	0.000
B_fueltype_Hybrid_Public_Transport	-0.941	0.000	0.000
B_fueltype_Petrol_Bike	-0.864	0.000	0.000
B_fueltype_Petrol_Car	-0.259	0.000	0.000
B_fueltype_Petrol_Public_Transport	-0.315	0.000	0.000
B_pt_n_interchanges_Public_Transport	-0.088	0.000	0.093
B_purpose_B_Bike	-0.032	0.000	0.748
B_purpose_B_Car	-0.066	0.000	0.337
B_purpose_B_Public_Transport	-0.053	0.000	0.445
B_purpose_HBE_Bike	-1.036	0.000	0.000
B_purpose_HBE_Car	-0.665	0.000	0.000
B_purpose_HBE_Public_Transport	-0.271	0.000	0.000
B_purpose_HBO_Bike	-0.771	0.000	0.000
B_purpose_HBO_Car	-0.317	0.000	0.000
B_purpose_HBO_Public_Transport	-0.397	0.000	0.000
B_purpose_HBW_Bike	-0.272	0.000	0.000
B_purpose_HBW_Car	-0.976	0.000	0.000
B_purpose_HBW_Public_Transport	-0.570	0.000	0.000
B_purpose_NHBO_Bike	-1.235	0.000	0.000
B_purpose_NHBO_Car	-0.415	0.000	0.000
B_purpose_NHBO_Public_Transport	-0.678	0.000	0.000
B_start_time_linear_Bike	0.016	0.000	0.016
B_start_time_linear_Car	0.024	0.000	0.000
B_start_time_linear_Public_Transport	0.012	0.000	0.002
B_traffic_perc_Car	-1.947	0.000	0.000
MU_m	1.391	0.000	0.000

Table 7: Parameter estimates of the LPMC CNL. Out of the 65 parameters, 11 are not significant at a 95% confidence interval.

LPMC - CNL			
	Value	Active bound	Rob. p-value
ASC_Bike	-3.338	0.000	0.000
ASC_Car	-2.380	0.000	0.000
ASC_Public_Transport	-2.061	0.000	0.000
B_age_Bike	-0.004	0.000	0.023
B_age_Car	0.006	0.000	0.000
B_age_Public_Transport	0.012	0.000	0.000
B_car_ownership_Bike	0.072	0.000	0.278
B_car_ownership_Car	0.603	0.000	0.000
B_car_ownership_Public_Transport	0.053	0.000	0.186
B_con_charge_Car	-0.841	0.000	0.000
B_cost_driving_fuel_Car	0.000	1.000	1.000
B_cost_transit_Public_Transport	-0.070	0.000	0.000
B_day_of_week_Bike	-0.023	0.000	0.148
B_day_of_week_Car	0.019	0.000	0.019
B_day_of_week_Public_Transport	-0.029	0.000	0.000
B_distance_Bike	-0.225	0.000	0.060
B_distance_Car	-0.007	0.000	0.948
B_distance_Public_Transport	0.000	1.000	1.000
B_driving_license_Bike	0.711	0.000	0.000
B_driving_license_Car	0.460	0.000	0.000
B_driving_license_Public_Transport	-0.370	0.000	0.000
B_dur_cycling_Bike	-1.643	0.000	0.008
B_dur_driving_Car	-3.193	0.000	0.000
B_dur_pt_access_Public_Transport	-3.032	0.000	0.000
B_dur_pt_bus_Public_Transport	-1.341	0.000	0.000
B_dur_pt_int_waiting_Public_Transport	-2.730	0.000	0.000
B_dur_pt_int_walking_Public_Transport	-2.123	0.000	0.002
B_dur_pt_rail_Public_Transport	-1.015	0.000	0.000
B_dur_walking_Walk	-8.071	0.000	0.000
B_female_Bike	-0.832	0.000	0.000
B_female_Car	0.117	0.000	0.000
B_female_Public_Transport	0.150	0.000	0.000
B_fueltype_Avrg_Bike	-0.679	0.000	0.000
B_fueltype_Avrg_Car	-1.155	0.000	0.000
B_fueltype_Avrg_Public_Transport	-0.330	0.000	0.000
B_fueltype_Diesel_Bike	-0.818	0.000	0.000
B_fueltype_Diesel_Car	-0.237	0.000	0.000
B_fueltype_Diesel_Public_Transport	-0.416	0.000	0.000
B_fueltype_Hybrid_Bike	-0.976	0.000	0.000
B_fueltype_Hybrid_Car	-0.747	0.000	0.000
B_fueltype_Hybrid_Public_Transport	-0.968	0.000	0.000
B_fueltype_Petrol_Bike	-0.866	0.000	0.000
B_fueltype_Petrol_Car	-0.242	0.000	0.000
B_fueltype_Petrol_Public_Transport	-0.348	0.000	0.000
B_pt_n_interchanges_Public_Transport	-0.072	0.000	0.129
B_purpose_B_Bike	-0.032	0.000	0.749
B_purpose_B_Car	-0.080	0.000	0.257
B_purpose_B_Public_Transport	-0.069	0.000	0.322
B_purpose_HBE_Bike	-1.031	0.000	0.000
B_purpose_HBE_Car	-0.623	0.000	0.000
B_purpose_HBE_Public_Transport	-0.308	0.000	0.000
B_purpose_HBO_Bike	-0.771	0.000	0.000
B_purpose_HBO_Car	-0.311	0.000	0.000

Table 7: Parameter estimates of the LPMC CNL. Out of the 65 parameters, 11 are not significant at a 95% confidence interval.

LPMC - CNL			
	Value	Active bound	Rob. p-value
B_purpose_HBO_Public_Transport	-0.394	0.000	0.000
B_purpose_HBW_Bike	-0.268	0.000	0.000
B_purpose_HBW_Car	-0.938	0.000	0.000
B_purpose_HBW_Public_Transport	-0.613	0.000	0.000
B_purpose_NHBO_Bike	-1.236	0.000	0.000
B_purpose_NHBO_Car	-0.429	0.000	0.000
B_purpose_NHBO_Public_Transport	-0.678	0.000	0.000
B_start_time_linear_Bike	0.016	0.000	0.015
B_start_time_linear_Car	0.023	0.000	0.000
B_start_time_linear_Public_Transport	0.013	0.000	0.001
B_traffic_perc_Car	-1.700	0.000	0.000
MU_f	1.000	1.000	0.000
MU_m	2.025	0.000	0.000
alpha_f	0.467	0.000	0.000