Variable Neighborhood Search for Assisted Utility Specification in Discrete Choice Models

Nicola Ortelli [*] [†] Tim Hillel [†] Francisco Camara Pereira [‡]
Matthieu de Lapparent [*] Michel Bierlaire [†]

1 Introduction and Context

In the last 40 years, transportation demand modeling has almost exclusively been tackled using discrete choice models (DCMs). This is due to their high *interpretability*, which allows to verify their compliance with well-established behavioral theory. However, the development of DCMs through manual specification is laborious. The predominant approach for this task is to *a priori* include a certain number of variables that are regarded as essential in the specification of the model; incremental changes are then tested in order to improve its goodness of fit, while ensuring its behavioral realism (Koppelman and Bhat, 2006). Because the set of candidate specifications grows beyond manageable even with a moderate number of variables under consideration, this kind of theory-driven approaches appears to be time-consuming and prone to errors. Modelers tend to rely on common sense or intuition without further validation of the supposedly prevailing constructs they prioritize, while the implications of working with incorrectly specified models and possibly biased parameters are largely underestimated (Torres et al., 2011, Van Der Pol et al., 2014).

This issue, worsened by the advent of big data and the need to analyze ever-larger datasets, has driven an increasing focus on machine learning (ML) as a way of relieving the modeler of the burden of model specification. In the past years, numerous studies have investigated the usefulness of ML classifiers as an alternative to DCMs by comparing logit models with methods such as decision trees (Tang et al., 2015; Lhéritier et al., 2019), support vector machines (Zhang and Xie, 2008; Paredes et al., 2017) or neural networks (Zhao et al., 2018; Lee et al., 2018). The studies indicate that the latter are outperformed in terms of prediction accuracy (Hagenauer and Helbich, 2017; Wang and Ross, 2018); however, the former suffer from a crucial limitation: they lack interpretability. The goal of DCMs is to accurately predict the choices of a population in a particular context, but the estimated values of the parameters are equally important: DCMs have strong behavioral foundations that originate in random utility theory (McFadden, 1974) and their mathematical structure allows to *understand* the decision processes, in addition to predicting their outcome. DCMs may be worse at prediction than their ML counterparts, but the former provide valuable insights into the underlying process that individuals follow when making choices.

[*] School of Management and Engineering Vaud (HEIG-VD) HES-SO University of Applied Sciences and Arts Western Switzerland, {nicola.ortelli,matthieu.delapparent}@heig-vd.ch

[†] Transport and Mobility Laboratory (TRANSP-OR), École Polytechnique Fédérale de Lausanne (EPFL) Switzerland, {nicola.ortelli,tim.hillel,michel.bierlaire}@epfl.ch

[‡] Machine Learning for Smart Mobility Group (MLSM), Danmarks Tekniske Universitet (DTU) Denmark, {camara}@dtu.dk

To the best of our knowledge, few studies that combine DCMs with data-driven methods preserve the interpretable closed-form utility expressions of the former. These include several approaches: Sifringer et al. (2018), Pereira (2019) and Han et al. (2020) make use of neural networks to learn different representations to be included in standard logits; Brathwaite et al. (2017) provide a microeconomic framework for the interpretation of decision trees and combines those with DCMs to model semi-compensatory decision making; Hillel et al. (2019) use a gradient boosting decision trees ensemble to inform the utility specification of a DCM; Paz et al. (2019) use a simulated annealing algorithm to select the optimal set of variables and parameter random distributions of a mixed logit model.

In order to address the limitations of both ML classifiers and DCMs, in this paper we introduce a data-driven method for the specification of logit models. Our approach involves a metaheuristic procedure that mimics the way an experienced modeler would develop a specification, while ensuring the set of candidates is explored thoroughly, impartially, and efficiently. The approach combines three primary ingredients: (1) a set of operators that modify an existing model into another one that is not too different, (2) a measure of performance that allows to compare the quality of two specifications and (3) a variable neighborhood search heuristic that organizes the model development phase. We believe our algorithm can serve the scope of assisting inexperienced analysts in the task of model development, but also provide relevant insights to more accomplished modelers.

2 Methodology

Our algorithm makes use of a variable neighborhood search (VNS) procedure to sequentially apply small modifications on an initial utility specification, while assessing the induced improvement by means of a measure of performance. The current version of our algorithm is limited to logit models and linear-in-parameters utilities. Following Bierlaire (1998) we include non-linear transformations of variables by explicitly specifying Box-Cox transforms with prespecified parameters (Box and Cox, 1964) and segmentation of parameters using categorical variables. Notationally, this translates to writing the observed utility that individual n associates to alternative i from her choice set $\mathcal{C}_n$ as

$$V_{in} = \sum_{k=1}^{K_i} B_{ik}(\boldsymbol{c}_n, \boldsymbol{s}_{ik})\, x_{ink}^{(\lambda_{t_{ik}})} v_{ik}, \tag{1}$$

where $\boldsymbol{x}_{in} = [x_{in1} \cdots x_{inK_i}]$ is a *user-defined* vector of potential explanatory variables associated with alternative i and $\boldsymbol{v}_i = [v_{i1} \cdots v_{iK_i}]$ is a vector of indicators: each v_{ik} is equal to 1 if variable x_{ink} enters the model, and 0 otherwise. Furthermore, the notation $x^{(\lambda)}$ denotes a Box-Cox transformation of x, defined as

$$x^{(\lambda)} = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } x \neq 0, \\ \log(x) & \text{if } x = 0. \end{cases} \tag{2}$$

$\lambda_{t_{ik}}$ may only take value from the *user-defined* set $\{\lambda_1, \ldots, \lambda_L\}$; the indicator t_{ik} is therefore constrained to the values $\{1, \ldots, L\}$. Finally, we define $B_{ik}(\boldsymbol{c}_n, \boldsymbol{s}_{ik})$ as

$$B_{ik}(\boldsymbol{c}_n, \boldsymbol{s}_{ik}) = \sum_{d=1}^{D_{\boldsymbol{s}_{ik}}} \beta_{ikd}\delta_d(\boldsymbol{c}_n), \tag{3}$$

where, $\boldsymbol{c}_n$ is a *user-defined* vector of P categorical socioeconomic variables that may be considered for segmentation, $\boldsymbol{s}_{ik} = [s_{ik1} \cdots s_{ikP}]$ is a vector of indicators denoting the ones selected to interact with variable x_{ink} and $\delta_d(\boldsymbol{c}_n)$ is an indicator that equals 1 if individual n belongs to population segment d and 0 otherwise. We denote by $D_{\boldsymbol{s}_{ik}}$ the total number of population segments obtained through the division of the sample according to the selected socioeconomic variables. In other words, $B_{ik}(\boldsymbol{c}_n, \boldsymbol{s}_{ik})$ assigns a different parameter to each individual n depending on the population segment it belongs to.

Given the notation described in Equation 1, any model specification M generated by our algorithm may be unequivocally characterized by the three "controllers" $\boldsymbol{v}_i = [v_{i1} \cdots v_{iK_i}]$, $\boldsymbol{t}_i = [t_{i1} \cdots t_{iK_i}]$ and $\boldsymbol{S}_i = [\boldsymbol{s}_{i1} \cdots \boldsymbol{s}_{iK_i}]$:

$$M = \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i, \boldsymbol{S}_i\}. \tag{4}$$

We may now formulate the optimization problem our algorithm is designed to solve as

$$\begin{aligned} \min_{\boldsymbol{v}_i, \boldsymbol{t}_i, \boldsymbol{S}_i} \quad & f(M) \\ \text{subject to:} \quad & \boldsymbol{v}_i \in \{0,1\}^{K_i} && \forall i \in \mathcal{C}, \\ & \boldsymbol{t}_i \in \{1, \dots, L\}^{K_i} && \forall i \in \mathcal{C}, \\ & \boldsymbol{S}_i \in \{0,1\}^{K_i \times P} && \forall i \in \mathcal{C}. \end{aligned} \tag{5}$$

The remainder of this section is divided into three parts; each introduces one of the ingredients our algorithm is build on, namely: (1) the operators we use to bring small changes to an existing specification, (2) the measure of performance $f(\cdot)$ used to assess the quality of a model and (3) the metaheuristic procedure that organizes the specification development phase.

2.1 Operators

The operators used by our algorithm arise from observing how modelers manually develop utility specifications; they correspond to the typical elementary modifications that are considered and tested during such process. Suppose an initial specification M, as the one shown in Equation (4). The operators used by our algorithm are defined as follows:

- Operator V-ADD adds a nonselected variable x_{jnk} to enter the utility of alternative j, which corresponds to switching the value of v_{jk} from 0 to 1:

  $$\text{V-ADD}(M, j, k) : \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i, \boldsymbol{S}_i\} \to \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i + \delta_{ij}\boldsymbol{e}_k, \boldsymbol{t}_i, \boldsymbol{S}_i\}, \tag{6}$$

  where δ_{ij} is the Kronecker delta and $\boldsymbol{e}_k$ is a vector of the natural basis.

- Operator V-REM is the reciprocal of V-ADD; it removes a variable x_{jnk} from the model, provided that $t_{jk} = 1$ and $\boldsymbol{s}_{jk} = \boldsymbol{0}$:

  $$\text{V-REM}(M, j, k) : \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i, \boldsymbol{S}_i\} \to \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i - \delta_{ij}\boldsymbol{e}_k, \boldsymbol{t}_i, \boldsymbol{S}_i\}. \tag{7}$$

- Operator T-ADD modifies the Box-Cox parameter of a given variable x_{jnk} from $\lambda_{t_{jk}}$ to $\lambda_{t_{jk}+1}$, provided that $t_{jk} < L$:

  $$\text{T-ADD}(M, j, k) : \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i, \boldsymbol{S}_i\} \to \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i + \delta_{ij}\boldsymbol{e}_k, \boldsymbol{S}_i\}. \tag{8}$$

- Operator T-REM is its reciprocal; it decrements the value of t_{jk} by 1 as long as $t_{jk} > 1$:

  $$\text{T-REM}(M, j, k) : \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i, \boldsymbol{S}_i\} \to \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i - \delta_{ij}\boldsymbol{e}_k, \boldsymbol{S}_i\}. \tag{9}$$

- Operator S-ADD interacts x_{jnk} with a socioeconomic variable c_{np}, which corresponds to switching the value of s_{jkp} from 0 to 1:

$$\text{S-ADD}(M, j, k, p) : \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i, \boldsymbol{S}_i\} \to \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i, \boldsymbol{S}_i + \delta_{ij} \boldsymbol{e}_{kp}\}, \tag{10}$$

  where $\boldsymbol{e}_{kp}$ is a matrix of the $K_i \times P$ natural basis.

- Operator S-REM is the reciprocal of S-ADD: it deactivates the interaction between x_{jnk} and c_{np} by switching the value of s_{jkp} from 1 to 0:

$$\text{S-REM}(M, j, k, p) : \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i, \boldsymbol{S}_i\} \to \bigcup_{i \in \mathcal{C}} \{\boldsymbol{v}_i, \boldsymbol{t}_i, \boldsymbol{S}_i - \delta_{ij} \boldsymbol{e}_{kp}\}. \tag{11}$$

2.2 Measure of Performance

The second ingredient of our algorithm is the measure of performance $f(M)$ that enables the comparison of the relative quality of two models. Traditionally, the superiority of a specification over another is evaluated by means of statistical tests. However, those are not appropriate in the context of an automated search (Thompson et al., 1991; Thompson, 1995; Whittingham et al., 2006; Harrell, 2015). As Smith (2018) points out, standard statistical tests assume a single verification of a prespecified model and are therefore inadequate when a sequence of iterations is employed to select explanatory variables. The consequence of repeated statistical testing for variable selection — or *data dredging* — is that it results in models with mediocre inferential properties and poorly estimated parameters, which must be avoided by any means (Lukacs et al., 2010).

We therefore diverge from this approach and, instead, use the Bayesian information criterion (BIC) to decide on the best direction to follow during the search. The BIC is defined by Schwarz (1978) as

$$f_{\text{BIC}}(M) = \log(N)K - 2\mathcal{L}(M), \tag{12}$$

where N is the number of observations in the considered dataset and K is the total number of estimated parameters and $\mathcal{L}(M)$ is the maximized log-likelihood of model M. The criterion is as a large-sample approximation of the log-Bayes factor, which, when equal priors are assumed on all candidate models, can be seen as a measure of the evidence in favor of a certain model to be the most probable (Burnham and Anderson, 2004).

2.3 Metaheuristic Procedure

Finally, we describe the metaheuristic procedure used by our algorithm. It consists of a variation of the basic VNS (Hansen and Mladenović, 2002) that utilizes a first-improvement local search (FILS) subroutine and a maximum-iteration stopping condition, as illustrated in Algorithm 1.

The VNS algorithm is originally proposed by Mladenovic and Hansen (1997). It relies on local search, but tackles its main limitation by considering several neighborhood structures rather than a single one. Those are systematically alternated during the search, which effectively prevents getting stuck in local minima. The change of neighborhood structure is shown in Lines 10 – 15 of Algorithm 1; the outcome depends on whether the specification obtained through the FILS subroutine outperforms the best current model or not. The shaking step described in Algorithm 2 is an additional mechanism that serves the same scope of avoiding local minima: whenever a neighborhood change is performed, the starting point of the next FILS subroutine is randomly drawn from the neighbors of the best solution encountered so far, rather than the best solution itself.

Algorithm 1: VNS for assistedMNL

```
inputs : initial specification $M$,
         neighborhood structures $\mathcal{N}_1, \ldots, \mathcal{N}_H$,
         maximum number of iterations $i_{max}$
output: best encountered specification $M$

$i \leftarrow 0$;
while $i < i_{max}$ do
    $h \leftarrow 1$;
    repeat
        $M' \leftarrow$ Shake($M$,$\mathcal{N}_h$);
        $i \leftarrow i + 1$;
        $M', j \leftarrow$ FILS($M'$,$\mathcal{N}_h$);
        $i \leftarrow i + j$;
        if $f_{\text{BIC}}(M') < f_{\text{BIC}}(M)$ then
            $M \leftarrow M'$;
            $h \leftarrow 1$;
        else
            $h \leftarrow h + 1$;
        end
    until $h = H$;
end
return $M^*$
```

Algorithm 2: `Shake` function

```
inputs : specification $M$,
         neighborhood structure $\mathcal{N}$
output: randomly selected neighbor $M'$

$\{M_1, \ldots, M_W\} \leftarrow$ Shuffle($\mathcal{N}(M)$);
$M' \leftarrow M_1$;
return $M'$
```

Algorithm 3: `FILS` function

```
inputs  : initial specification $M$,
          neighborhood structure $\mathcal{N}$
outputs: locally optimal specification $M$,
          number of iterations $i$

$i \leftarrow 0$;
repeat
    $\{M_1, \ldots, M_W\} \leftarrow$ Shuffle($\mathcal{N}(M)$);
    $w \leftarrow 0$;
    while $w < W$ do
        $w \leftarrow w + 1$;
        if $f_{\text{BIC}}(M_w) < f_{\text{BIC}}(M)$ then
            $M \leftarrow M_w$;
            $i \leftarrow i + 1$;
            break
        end
    end
until $w = W$;
return $M, i$
```

The current version of our algorithm cycles through three different neighborhood structures: each gathers the specifications obtained through all possible applications of one operator from Section 2.1 or its reciprocal on the current solution M:

$$\begin{aligned}
\mathcal{N}_{\mathrm{V}}(M) = &\{\text{V-ADD}(M,i,k) \mid v_{ik} = 0\} \\
&\cup \{\text{V-REM}(M,i,k) \mid v_{ik} = 1, t_{ik} = 1, \boldsymbol{s}_{ik} = \mathbf{0}\}, \\
\mathcal{N}_{\mathrm{T}}(M) = &\{\text{T-ADD}(M,i,k) \mid t_{ik} < L\} \\
&\cup \{\text{T-REM}(M,i,k) \mid t_{ik} > 1\}, \\
\mathcal{N}_{\mathrm{S}}(M) = &\{\text{S-ADD}(M,i,k,p) \mid s_{ikp} = 0\} \\
&\cup \{\text{S-REM}(M,i,k,p) \mid s_{ikp} = 1\}.
\end{aligned} \tag{13}$$

3 Experiments

We test the algorithm described in the previous section on the Swissmetro dataset (Bierlaire et al., 2001), which consists of survey data collected in Switzerland in 1998 to analyze the potential impact of the Swissmetro, an innovative mode of transportation. Respondents were asked to state their favorite transportation mode among three alternatives — train, Swissmetro and car — in nine different hypothetical situations. $10'395$ observations remain after removing incomplete data; 20% of these are set aside for out-of-sample validation.

We allow the algorithm to consider 8 potential explanatory variables, 3 different values for the Box-Cox parameters — 1, $\frac{1}{2}$, 0 — and 5 categorical variables for segmentation.[1] For the sake of simplicity, we limit the number of simultaneous segmentating variables to two for each parameter of alternative-specific constant; still, the number of possible specifications is over 10^{15}. Table 1 gathers the results of four runs of the algorithm in such configuration, together with the maximized log-likelihood of the obtained specifications both on the training and validation sets of observations. We compare those with the benchmark logit model presented in Bierlaire et al. (2001).

Table 1: Comparison of the models obtained from the four runs with the benchmark model.

	Run 1	Run 2	Run 3	Run 4	Benchmark
BIC	**11947.5**	11981.9	12010.9	**11947.5**	13211.3
Number of estimated models	1174	942	1034	1041	–
Running time [h]	3.7	3.4	3.7	3.6	–
In-sample log-likelihood	−5851.9	−5819.5	−5874.6	−5851.9	−6565.0
Out-of-sample log-likelihood	−1550.0	−1615.5	−1548.9	−1550.0	−1633.5
Estimated parameters	27	38	29	27	9
Considered variables	11	12	12	11	7

As expected, the four specifications obtained after 150 iterations vary substantially in terms of BIC, despite all runs having the same constants-only initial specification. This is due to the stochastic nature of the shaking step and FILS subroutine. Interestingly, Run 1 and Run 4 reach the same specification, which is also the best of the four in terms of BIC. As regards the log-likelihood yielded on the validation data, the model reached by Run 3 performs better. Figure 1 illustrates the evolution of the BIC during each of the runs.

[1] A description of the considered variables is provided in the Appendix. We refer the reader to Antonini et al. (2007) for a complete description of the dataset.

Finally, Table 2 gathers the estimation results of the model reached by Run 1 and Run 4. All parameters appear to be significant and have the expected sign; additionally, the parameters of the Box-Cox transformations seem to be behaviorally realistic.

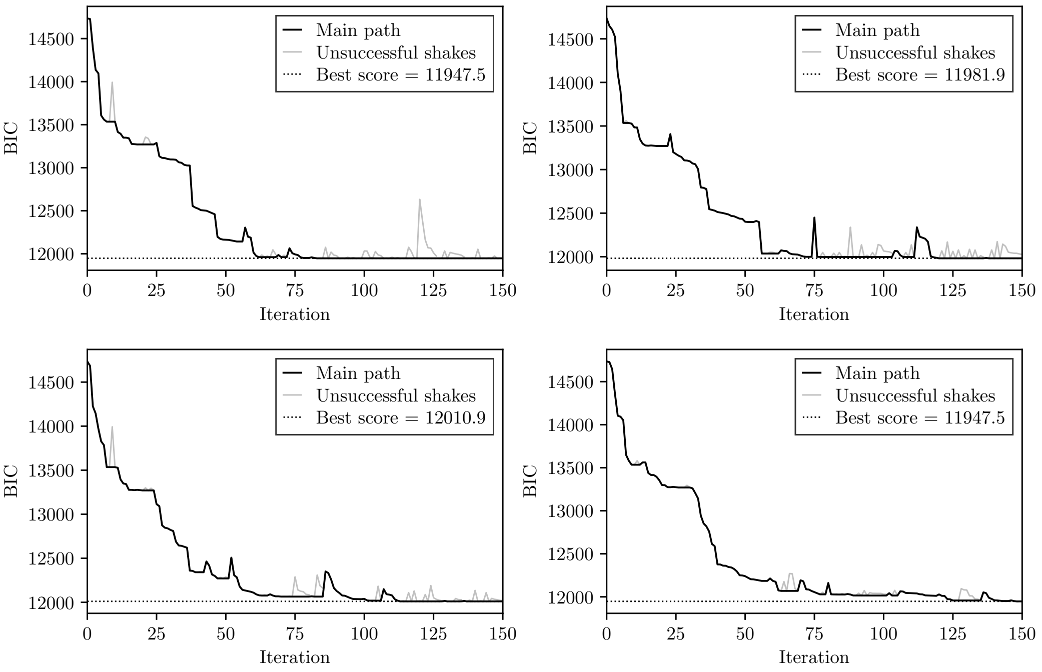

Figure 1: BIC vs iterations for all runs.

4 Conclusion

In this study, we introduce a new approach for assisted specification of DCMs that uses a metaheuristic procedure to generate good models. The validity of the proposed algorithm is empirically demonstrated using choice data. Out-of-sample validation shows that our algorithm reaches high-quality specifications while maintaining an interpretable model structure. The parameter values are consistent with behavioral theory and statistically different from zero, but this should be taken with a grain of salt: in stated-preference data the essential variables are known from the design of the survey.

Intended future work includes the development of a mechanism that systematically rejects behaviorally inconsistent specifications during the search procedure, so as to obtain equally good results when applying our algorithm on revealed-preference datasets. Other relevant directions of search include (1) further testing and additional case studies to prove the validity of our approach, (2) extending the framework to advanced DCM structures such as nesting, mixtures and more complex interactions between variables, and (3) rigorous investigation to find ways of bringing the VNS to convergence faster.

Table 2: Estimation results of the best model reached by Run 1 and Run 4.

Parameter	Value	Rob. Std err	Rob. t-test
$\text{B\_TRAIN\_TT}^{(0)}{}_{\text{GA}=0,\text{MALE}=0}$	−2.94	0.126	−23.4
$\text{B\_TRAIN\_TT}^{(0)}{}_{\text{GA}=0,\text{MALE}=1}$	−3.52	0.12	−29.3
$\text{B\_TRAIN\_TT}^{(0)}{}_{\text{GA}=1,\text{MALE}=0}$	−0.391	0.194	−2.02
$\text{B\_TRAIN\_TT}^{(0)}{}_{\text{GA}=1,\text{MALE}=1}$	−0.521	0.189	−2.75
$\text{B\_TRAIN\_CO}^{(\frac{1}{2})}{}_{\text{FIRST}=0,\text{MALE}=0}$	−0.124	0.00881	−14.1
$\text{B\_TRAIN\_CO}^{(\frac{1}{2})}{}_{\text{FIRST}=0,\text{MALE}=1}$	−0.114	0.00811	−14.0
$\text{B\_TRAIN\_CO}^{(\frac{1}{2})}{}_{\text{FIRST}=1,\text{MALE}=0}$	−0.0934	0.00837	−11.2
$\text{B\_TRAIN\_CO}^{(\frac{1}{2})}{}_{\text{FIRST}=1,\text{MALE}=1}$	−0.1	0.0068	−14.8
$\text{B\_TRAIN\_HE}^{(0)}{}_{\text{FIRST}=0}$	−0.463	0.068	−6.81
$\text{B\_TRAIN\_HE}^{(0)}{}_{\text{FIRST}=1}$	−0.632	0.0712	−8.88
$\text{ASC\_SM}_{\text{GA}=0}$	−5.54	0.551	−10.1
$\text{ASC\_SM}_{\text{GA}=1}$	25.7	3.05	8.42
$\text{B\_SM\_TT}^{(0)}{}_{\text{WHO}=1}$	−1.78	0.0755	−23.5
$\text{B\_SM\_TT}^{(0)}{}_{\text{WHO}=2}$	−1.54	0.0762	−20.2
$\text{B\_SM\_TT}^{(0)}{}_{\text{WHO}=3}$	−1.59	0.0851	−18.7
$\text{B\_SM\_CO}^{(0)}{}_{\text{GA}=0,\text{MALE}=0}$	−0.851	0.0709	−12.0
$\text{B\_SM\_CO}^{(0)}{}_{\text{GA}=0,\text{MALE}=1}$	−1.45	0.0712	−20.3
$\text{B\_SM\_CO}^{(0)}{}_{\text{GA}=1,\text{MALE}=0}$	−4.28	0.384	−11.2
$\text{B\_SM\_CO}^{(0)}{}_{\text{GA}=1,\text{MALE}=1}$	−4.15	0.36	−11.5
ASC_CAR	−15.2	0.623	−24.5
$\text{B\_CAR\_TT}^{(\frac{1}{2})}{}_{\text{FIRST}=0,\text{MALE}=0}$	−0.0723	0.0163	−4.45
$\text{B\_CAR\_TT}^{(\frac{1}{2})}{}_{\text{FIRST}=0,\text{MALE}=1}$	−0.173	0.0102	−16.9
$\text{B\_CAR\_TT}^{(\frac{1}{2})}{}_{\text{FIRST}=1,\text{MALE}=0}$	−0.0718	0.0161	−4.44
$\text{B\_CAR\_TT}^{(\frac{1}{2})}{}_{\text{FIRST}=1,\text{MALE}=1}$	−0.195	0.0106	−18.4
$\text{B\_CAR\_CO}^{(1)}{}_{\text{WHO}=1}$	−0.0113	0.00113	−9.98
$\text{B\_CAR\_CO}^{(1)}{}_{\text{WHO}=2}$	−0.00533	0.00113	−4.72
$\text{B\_CAR\_CO}^{(1)}{}_{\text{WHO}=3}$	−0.00478	0.00183	−2.61
Sample size:			8316
Init log likelihood:			−8603.3
Final log likelihood:			−5851.9

References

Antonini, G., Gioia, C. and Frejinger, E. (2007). Swissmetro: description of the data.

Bierlaire, M. (1998). Discrete choice models, *Operations research and decision aid methodologies in traffic and transportation management*, Springer, pp. 203–227.

Bierlaire, M., Axhausen, K. and Abay, G. (2001). The acceptance of modal innovation: The case of swissmetro, *Swiss Transport Research Conference*, number CONF.

Box, G. E. and Cox, D. R. (1964). An analysis of transformations, *Journal of the Royal Statistical Society: Series B (Methodological)* **26**(2): 211–243.

Brathwaite, T., Vij, A. and Walker, J. L. (2017). Machine learning meets microeconomics: The case of decision trees and discrete choice, *arXiv preprint arXiv:1711.04826* .

Burnham, K. P. and Anderson, D. R. (2004). Multimodel inference: understanding aic and bic in model selection, *Sociological methods & research* **33**(2): 261–304.

Hagenauer, J. and Helbich, M. (2017). A comparative study of machine learning classifiers for modeling travel mode choice, *Expert Systems with Applications* **78**: 273–282.

Han, Y., Zegras, C., Pereira, F. C. and Ben-Akiva, M. (2020). A neural-embedded choice model: Tastenet-mnl modeling taste heterogeneity with flexibility and interpretability, *arXiv preprint arXiv:2002.00922* .

Hansen, P. and Mladenović, N. (2002). Developments of variable neighborhood search, *Essays and surveys in metaheuristics*, Springer, pp. 415–439.

Harrell, Jr., F. E. (2015). *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*, Springer.

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

Koppelman, F. S. and Bhat, C. (2006). A self instructing course in mode choice modeling: multinomial and nested logit models.

Lee, D., Derrible, S. and Pereira, F. C. (2018). Comparison of four types of artificial neural network and a multinomial logit model for travel mode choice modeling, *Transportation Research Record* **2672**(49): 101–112.

Lhéritier, A., Bocamazo, M., Delahaye, T. and Acuna-Agost, R. (2019). Airline itinerary choice modeling using machine learning, *Journal of choice modelling* **31**: 198–209.

Lukacs, P. M., Burnham, K. P. and Anderson, D. R. (2010). Model selection bias and freedman's paradox, *Annals of the Institute of Statistical Mathematics* **62**(1): 117.

McFadden, D. (1974). The measurement of urban travel demand, *Journal of Public Economics* **3**(4): 303 – 328.

Mladenovic, N. and Hansen, P. (1997). Variable neighborhood search, *Computers & Operations Research* **24**(11): 1097 – 1100.

Paredes, M., Hemberg, E., O'Reilly, U.-M. and Zegras, C. (2017). Machine learning or discrete choice models for car ownership demand estimation and prediction?, *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, IEEE, pp. 780–785.

Paz, A., Arteaga, C. and Cobos, C. (2019). Specification of mixed logit models assisted by an optimization framework, *Journal of choice modelling* **30**: 50–60.

Pereira, F. C. (2019). Rethinking travel behavior modeling representations through embeddings, *arXiv preprint arXiv:1909.00154* .

Schwarz, G. (1978). Estimating the dimension of a model, *The annals of statistics* **6**(2): 461–464.

Sifringer, B., Lurkin, V. and Alahi, A. (2018). Let me not lie: Learning multinomial logit, *arXiv preprint arXiv:1812.09747* .

Smith, G. (2018). Step away from stepwise, *Journal of Big Data* **5**(1): 32.

Tang, L., Xiong, C. and Zhang, L. (2015). Decision tree method for modeling travel mode switching in a dynamic behavioral process, *Transportation Planning and Technology* **38**(8): 833–850.

Thompson, B. (1995). Stepwise regression and stepwise discriminant analysis need not apply here: A guidelines editorial.

Thompson, B., Smith, Q., Miller, L. and Thomson, W. (1991). Stepwise methods lead to bad interpretations: Better alternatives.

Torres, C., Hanley, N. and Riera, A. (2011). How wrong can you be? implications of incorrect utility function specification for welfare measurement in choice experiments, *Journal of Environmental Economics and Management* **62**(1): 111–121.

Van Der Pol, M., Currie, G., Kromm, S. and Ryan, M. (2014). Specification of the utility function in discrete choice experiments, *Value in Health* **17**(2): 297–301.

Wang, F. and Ross, C. L. (2018). Machine learning travel mode choices: Comparing the performance of an extreme gradient boosting model with a multinomial logit model, *Transportation Research Record* **2672**(47): 35–45.

Whittingham, M. J., Stephens, P. A., Bradbury, R. B. and Freckleton, R. P. (2006). Why do we still use stepwise modelling in ecology and behaviour?, *Journal of animal ecology* **75**(5): 1182–1189.

Zhang, Y. and Xie, Y. (2008). Travel mode choice modeling with support vector machines, *Transportation Research Record* **2076**(1): 141–150.

Zhao, X., Yan, X., Yu, A. and Van Hentenryck, P. (2018). Modeling stated preference for mobility-on-demand transit: a comparison of machine learning and logit models, *arXiv preprint arXiv:1811.01315* .

5 Appendix

Table 3: Swissmetro dataset. Definition of the considered variables and statistics.

Variable	min	max	mean	std.
TRAIN_TT *Train travel time [min]. Based on the car distance.*	31	1049	166.63	77.35
TRAIN_CO *Train cost [CHF]. If the traveler owns a GA, equal to its price.*	4	5040	514.34	1088.93
TRAIN_HE *Train headway [min].*	30	120	70.10	37.43
SM_TT *Swissmetro travel time [min]. A speed of 500 km/h is considered.*	8	796	87.47	53.55
SM_CO *Swissmetro cost [CHF]. Proportional to the rail fare.*	6	6720	670.34	1441.59
SM_HE *Train headway [min].*	10	30	20.02	8.16
CAR_TT *Car travel time [min].*	0	1560	123.80	88.71
CAR_CO *Car cost [CHF]. A fixed average cost per kilometer is considered.*	0	520	78.74	55.26
GA *Travel card ownership. 1 if the traveler owns one, 0 otherwise.*	0	1	0.14	0.35
MALE *Traveler's gender. 0 if female, 1 if male.*	0	1	0.75	0.43
FIRST *1 if first-class traveler, 0 otherwise.*	0	1	0.47	0.50
LUGGAGE *0 if none, 1 if one piece, 3 if several pieces.*	0	3	0.68	0.60
WHO *Who pays for the trip. 1 if self, 2 if employer, 3 if half-half.*	1	3	1.49	0.71