Assisted Specification of Discrete Choice Models

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

July 8, 2020

Report TRANSP-OR 200708
Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne
transp-or.epfl.ch

[*] 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

Abstract

Determining appropriate utility specifications for discrete choice models is time-consuming and prone to errors. As the number of possible functions exponentially grows with the variables under consideration, analysts need to spend increasing amounts of time on searching for good specifications through trial-and-error and expert knowledge. This paper proposes a data-driven algorithm that aims at assisting modelers in their search. Our approach translates the task into a multi-objective combinatorial optimization problem and makes use of a variant of the variable neighborhood search algorithm to generate solutions. We apply the algorithm to a real mode choice dataset as a proof of concept. The results demonstrate its ability to generate high-quality specifications in reasonable amounts of time.

Keywords: discrete choice models, utility specification, multi-objective optimization, combinatorial optimization, metaheuristics.

1 Introduction

In the last 40 years, discrete choice models (DCMs) have been used to tackle a wide variety of demand modeling problems. This is due to their high *interpretability*, which allows researchers to verify their compliance with well-established behavioral theories (McFadden, 1974) and to provide support for policy and planning decisions founded on utility theory from microeconomics. 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 model, before testing incremental changes 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, *theory-driven* approaches of this kind can be time-consuming and prone to errors. Modelers tend to rely on common sense or intuition, which may lead to incorrectly specified models. The implications of misspecification include lower predictive power, biased parameter estimates and erroneous interpretations (Bentz and Merunka, 2000; Torres et al., 2011; Van Der Pol et al., 2014).

This issue, together with the advent of big data and the need to analyze ever-larger datasets, has induced an increasing focus on machine learning (ML) and other *data-driven* methods as a way of relieving the analyst of the burden of model specification. Unlike DCMs that require the form of their utility functions to be known *a priori*, ML allows for more flexible model structures to be directly learned from the data. In the past years, numerous studies have therefore investigated the usefulness of ML classifiers as an alternative to logit, mixed logit and nested logit models; these studies indicate that DCMs are generally outperformed in terms of prediction accuracy (Hagenauer and Helbich, 2017; Wang and Ross, 2018). However, most ML classifiers suffer from a severe limitation: they lack interpretability. The mathematical structure of DCMs enables the modeler to verify the behavioral realism of the relationships between explanatory variables and choice outcomes. Such a feature is crucial to support policy and planning decisions. Also, it allows the derivation of useful indicators that cannot be obtained from ML models, such as willingness to pay or consumer surplus. Whilst there is existing research focusing on extracting utility interpretations and economic indicators from certain ML classifiers (Zhao et al., 2018), obtaining a closed-form expression that maps inputs to outputs is still not possible.

In this paper, we postulate that appropriate use of data during the development of a DCM can mitigate the need for its utility specification to be known *a priori*, even without compromising the sought-after interpretability through the use of ML classifiers. To this end, the present study introduces a data-driven method for the specification of random utility models. Our approach involves a metaheuristic procedure that mimics the way an experienced modeler would develop a specification, while ensuring that the set of candidates is explored thoroughly, impartially, and efficiently. We define a set of operators that modify an existing model into another one that is not too different and make use of a multi-objective variable neighborhood search algorithm to organize the model development phase. Our algorithm is designed to provide relevant insights and assist analysts in the task of utility specification.

The remainder of this paper is organized as follows: Section 2 is a non-exhaustive overview of the recent attempts at "enhancing" discrete choice models with data-driven methods, Section 3 describes our proposed algorithm and Section 4 presents the results obtained through its application on a mode choice dataset. The last section summarizes the findings of the present study and identifies the future steps of this research.

2 Literature Review

2.1 Machine Learning Methods for Choice Prediction

The advancements in computational power and the availability of ever-larger datasets in the recent years have led to numerous breakthroughs in machine learning (ML) research. As a result, ML techniques have been applied to a wide horizon of research fields, including discrete choice analysis. Support vector machines (Hagenauer and Helbich, 2017; Paredes et al., 2017), neural networks (De Carvalho et al., 1998; Zhao et al., 2018; Alwosheel et al., 2019; Wong and Farooq, 2019), decision trees (Tang et al., 2015; Brathwaite et al., 2017) and ensemble learning (Hillel et al., 2018; Wang and Ross, 2018; Lhéritier et al., 2019) have been investigated as potential alternatives for DCMs in a variety of choice problems. Many of those empirical studies conclude that ML classifiers show superior predictive power, but there have been limited attempts to link these new methods with economic theory and human decision making. In discrete choice analysis, the behavioral interpretation of the estimated model is as important as prediction accuracy, because it provides relevant insights for policy and planning decisions. ML classifiers usually lack the coefficient readability of linear models and obtaining a closed-form expression that maps inputs to outputs is often impossible. Interpretable machine learning tools do exist (Zhao et al., 2018), but are rarely applied. As a matter of fact, out of all studies mentioned in this paragraph, only four go beyond computing variable importance. Brathwaite et al. (2017) provide a microeconomic framework for the interpretation of decision trees and combine those with DCMs to model semi-compensatory decision making; Zhao et al. (2018) use partial dependence plots, marginal effects and arc elasticities to extract behavioral findings; Alwosheel et al. (2019) use prototypical examples to examine the relationships learned by neural networks; and Wong and Farooq (2019) extract behavioral insights and econometric properties from the matrix parameters of a residual neural network.

Spurred by these observations, another stream of research has attempted combining DCMs and ML classifiers into a single framework, so as to mitigate their respective limitations. Sifringer et al. (2018), Pereira (2019) and Han et al. (2020) share the same idea of "enhancing" a standard logit model by means of a neural network (NN). The goal is to increase its overall predictive performance while keeping some key parameters interpretable. To this end, Sifringer et al. (2018) propose a neural embedded logit model. Its utility functions are divided into a manually specified, interpretable part and a nonlinear "representation" part that is learned by a NN. In other words, the alternative-specific constants (ASCs) are modeled as flexible functions of all variables that do not enter the manually specified part of the utility. Han et al. (2020) extend the work of Sifringer et al. (2018) by allowing all parameters — rather than the ASCs alone — to be learned as functions of the individuals' socioeconomic characteristics. In a similar line of thought, Pereira (2019) introduces a method for encoding categorical variables, based on natural language processing techniques; as a preliminary step to specifying a logit model, a NN is used to learn *embeddings* of those variables, allowing for richer nuances than the typical transformations. Through different uses of NNs, the three studies succeed in improving the predictive power of the standard logit while maintaining part of the utility specification interpretable. However, the three approaches suffer from a common drawback: the explanatory variables still need to be manually selected, with no better method than context knowledge and trial-and-error. While these enhanced models are proven more flexible and powerful than traditional DCMs, they require the same effort from the modeler, if not more.

2.2 Informing DCMs with Data-Driven Methods

The idea of combining DCMs with ML classifiers dates back to Bentz and Merunka (2000) and Hruschka et al. (2002). The main difference of these pioneering studies with the methodologies developed by Sifringer et al. (2018), Pereira (2019) and Han et al. (2020) lies in their sequential nature: an exploratory NN is used as a specification tool to inform a standard logit model, which *then* provides interpretable results and significance statistics. The preliminary step allows the identification of the key variables and the detection of nonlinear effects; the modeler can therefore rely on these insights to develop a suitable utility specification.

Thus, in comparison to the hybrid models mentioned in the previous section, “feeding” a logit model with the findings of an exploratory data-driven method offers the additional advantage of assisting the analyst in the otherwise difficult task of model specification. This two-step approach proves itself worthy in several other studies: Hillel et al. (2019) use a gradient boosting decision trees ensemble to inform the utility specification of a DCM; Rodrigues et al. (2019) leverage the concept of automatic relevance determination to identify the most important features for explaining a dataset; and Paz et al. (2019) use a simulated annealing metaheuristic algorithm to select the optimal — according to the Bayesian information criterion (Schwarz, 1978) — set of variables and parameter distributions of a mixed logit model.

2.3 Utility Specification as a Combinatorial Problem

Translating the task of model specification into an optimization problem, as suggested by Paz et al. (2019), allows researchers to benefit from the vast combinatorial optimization literature and its variety of widely recognized metaheuristics. These methods are specifically designed to find “good” solutions for problems that cannot be solved through exhaustive testing; they hence seem particularly suited for the utility specification problem that Paz et al. (2019) address. An additional benefit of using metaheuristics in this context lies in their iterative nature. This allows any “post-estimation” measure of model quality to be seamlessly used as the objective function. We therefore believe that this direction of research is particularly promising. In this paper, we focus on the metaheuristic approach, in a similar way as Paz et al. (2019). The main contributions of this paper are the following:

- A multi-objective optimization algorithm that generates sets of promising models. Most studies mentioned in the current section propose a single model that is deemed as “best”. While a single best model may exist in experiments that use synthetic data — *i.e.*, the one model that created them — there is no such thing as an overall best model that explains all aspects of real data (Burnham and Anderson, 2002; Claeskens and Hjort, 2003; Burnham and Anderson, 2004; Claeskens and Hjort, 2008). Rather, we believe that the definition of a “suitable” model is context dependent and that different models should be developed according to the objective that is to be achieved. A number of studies try to justify the use of a single metric to identify the best model among a set of candidates; in our view, multi-objective optimization makes such approaches unnecessary. By defining appropriate objectives, sets of models can be generated that meet the plural needs of any predictive modeling problem.

- An embedded mechanism that verifies the behavioral soundness of the generated models. Inconsistency with economic theory is generally understood as the consequence of misspecification and therefore discredits any insight obtained from the estimated model. This mechanism is of crucial importance, as it confirms the actual relations between explanatory variables and choice outcomes are not inappropriately reflected.

3 Problem Formulation

The algorithm presented in this paper makes use of a metaheuristic procedure to mimic the way an experienced modeler would develop a discrete choice model (DCM). It proceeds by sequentially introducing small modifications to a set of "promising" utility specifications, with two goals: (i) improving the goodness of fit, which is measured as the maximized log likelihood (LL) yielded on the data of interest; while (ii) keeping the models as parsimonious as possible, *i.e.*, minimizing the number of estimated parameters. By definition, these two objectives are *conflicting*, in the sense that there exists no solution that simultaneously optimizes both; a nontrivial multi-objective optimization problem is hence defined.

Following Bierlaire (1998) we allow for nonlinear transformations of explanatory variables by means of power transforms, as well as segmentation of parameters using categorical variables. Hence, given (a) a vector $\boldsymbol{x}_{in} = [x_{in1} \cdots x_{inK_i}]$ of potential explanatory variables associated with each alternative i, (b) a vector of $\boldsymbol{\lambda} = [\lambda_1 \cdots \lambda_L]$ possible values for the parameters of the power transforms and (c) a vector $\boldsymbol{c}_n = [c_{n1} \cdots c_{nP}]$ of P categorical socioeconomic variables, solving the above-mentioned optimization problem consists in finding combinations of these features that offer good trade-offs between goodness of fit and parsimony.

In the next section, we show that any model specification that combines elements from $\boldsymbol{x}_{in}$, $\boldsymbol{\lambda}$ and $\boldsymbol{c}_n$ may be unequivocally characterized by means of three sets of *controllers*, namely $\boldsymbol{v}_i = [v_{i1} \cdots v_{iK_i}]$, $\boldsymbol{t}_i = [t_{i1} \cdots t_{iK_i}]$ and $\boldsymbol{S}_i = [\boldsymbol{s}_{i1} \cdots \boldsymbol{s}_{iK_i}]$. These controllers must be understood as the decision variables of the optimization problem under consideration. They respectively handle variable inclusion, nonlinear transformations and parameter segmentation, as illustrated in (1)–(3).

3.1 Utility Specification

In this framework, the most general formulation of the observed utility associated by individual n with alternative i from her choice set $\mathcal{C}_n$ may be written 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 K_i potential attributes, *i.e.*, explanatory variables associated with alternative i and $\boldsymbol{v}_i = [v_{i1} \cdots v_{iK_i}]$ is a vector of binary controllers: the effect of variable x_{ink} is considered in the model only if the corresponding controller v_{ik} is equal to 1. By construction, if v_{ik} is equal to 0 instead, then x_{ink} in not included in the corresponding utility function. Whilst the inclusion controllers v_{ik} could be defined as individual-specific and therefore depend on n for additional flexibility; we choose to keep them generic with respect to individuals for the sake of simplicity.

Turning to the second group of controllers of our approach, the notation $x^{(\lambda)}$ denotes a nonlinear transformation of x, defined as

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

As stated previously, $\lambda_{t_{ik}}$ may only take values from the user-defined vector $\boldsymbol{\lambda} = [\lambda_1 \cdots \lambda_L]$. The controllers t_{ik} are therefore constrained to the values $\{1, \ldots, L\}$. As an example, suppose $\boldsymbol{\lambda} = [1, \frac{1}{2}, 0]$. These values correspond to the identity, square-root and logarithmic transforms, respectively. Hence, if $t_{ik} = 1$ no transform is applied to variable x_{ink}. If $t_{ik} = 2$ the square root is used and if $t_{ik} = 3$ the logarithm is considered instead.

Finally, we define $B_{ik}(\boldsymbol{c}_n, \boldsymbol{s}_{ik})$ as

$$B_{ik}(\boldsymbol{c}_n, \boldsymbol{s}_{ik}) = \beta_{ik} + \sum_{p=1}^{P} \sum_{d=2}^{D_p} \beta_{ikpd} \delta_d(c_{np}) s_{ikp}, \tag{3}$$

where $\boldsymbol{c}_n$ is a *user-defined* vector of P categorical socioeconomic variables that may be considered for segmentation and $\boldsymbol{s}_{ik} = [s_{ik1} \cdots s_{ikP}]$ is a vector of binary controllers denoting the ones selected to interact with variable x_{ink}. We denote by D_p the number of possible discrete values of c_{np}, whereas $\delta_d(c_{np})$ is an indicator that equals 1 if the value of c_{np} belongs to category d and 0 otherwise. In other words, $B_{ik}(\boldsymbol{c}_n, \boldsymbol{s}_{ik})$ assigns additional taste parameters to each individual n depending on the population segments she belongs to. Parameter β_{ik} may therefore be seen as the coefficient associated with the reference socioeconomic category for variable x_{ink}. Alternative-specific constants (ASCs) are included in this formulation by adding variables that are constant throughout all observations of the dataset under consideration and setting all corresponding $\lambda_{t_{ik}}$ to the same arbitrary value. ASCs can therefore be segmented by socioeconomic variables just like any other parameter in the utility functions.

Hence, given the user-defined vectors $\boldsymbol{x}_{in}$, $\boldsymbol{\lambda}$ and $\boldsymbol{c}_n$, any model specification M that contains nonlinear transformations and first-order interactions with categorical variables 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 $\boldsymbol{S}_i = [\boldsymbol{s}_{i1} \cdots \boldsymbol{s}_{iK_i}]$:

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

As an illustrative example, consider a choice scenario with two alternatives, denoted A and B, and the following input vectors:

$$\begin{aligned} \boldsymbol{x}_{An} &= [x_{An1}, x_{An2}, x_{An3}], \\ \boldsymbol{x}_{Bn} &= [x_{Bn1}, x_{Bn2}], \\ \boldsymbol{\lambda} &= [1, 0], \\ \boldsymbol{c}_n &= [c_{n1}, c_{n2}]. \end{aligned} \tag{5}$$

The vectors $\boldsymbol{x}_{An}$ and $\boldsymbol{x}_{Bn}$ gather five potential explanatory variables and $\boldsymbol{\lambda}$ includes the identity and logarithmic transformations. The two categorical variables c_{n1} and c_{n2} may take two and three distinct discrete values respectively, *i.e.*, $D_1 = 2$ and $D_2 = 3$. The ASCs of both alternatives are omitted for the sake of simplicity. Additionally, consider a model characterized as

$$M = \{\boldsymbol{v}_A, \boldsymbol{t}_A, \boldsymbol{S}_A\} \cup \{\boldsymbol{v}_B, \boldsymbol{t}_B, \boldsymbol{S}_B\}, \tag{6}$$

with

$$\boldsymbol{v}_A = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \boldsymbol{t}_A = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}, \boldsymbol{S}_A = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \boldsymbol{v}_B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \boldsymbol{t}_B = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \boldsymbol{S}_B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}. \tag{7}$$

The corresponding utility functions, with all superfluous terms removed, may be written as follows:

$$\begin{aligned} V_{An} &= \beta_{A1} \log(x_{An1}) + [\beta_{A3} + \beta_{A312}\delta_2(c_{n1}) + \beta_{A322}\delta_2(c_{n2}) + \beta_{A323}\delta_3(c_{n2})] x_{An3}, \\ V_{Bn} &= \beta_{B1} \log(x_{Bn1}) + [\beta_{B2} + \beta_{B212}\delta_2(c_{n1})] \log(x_{Bn2}). \end{aligned} \tag{8}$$

3.2 Formal Definition of the Problem

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

$$
\begin{aligned}
&\max_{\boldsymbol{v}_i,\boldsymbol{t}_i,\boldsymbol{S}_i} && \mathcal{L}(\boldsymbol{v}_i,\boldsymbol{t}_i,\boldsymbol{S}_i,\boldsymbol{x}_{in},\boldsymbol{\lambda},\boldsymbol{c}_n), \\
&\min_{\boldsymbol{v}_i,\boldsymbol{t}_i,\boldsymbol{S}_i} && \sum_{i\in\mathcal{C}}\sum_{k=1}^{K_i} v_{ik}(1+\sum_{p=1}^{P} s_{ikp}(D_p-1)), \\
&\text{subject to:} && t_{ik} \leq 1 + v_{ik}L \quad \forall i \in \mathcal{C}, k \in \{1,\dots,K_i\}, \\
& && s_{ikp} \leq v_{ik} \quad \forall i \in \mathcal{C}, k \in \{1,\dots,K_i\}, p \in \{1,\dots,P\}, \\
& && v_{ik} \in \{0,1\} \quad \forall i \in \mathcal{C}, k \in \{1,\dots,K_i\}, \\
& && t_{ik} \in \{1,\dots,L\} \quad \forall i \in \mathcal{C}, k \in \{1,\dots,K_i\}, \\
& && s_{ikp} \in \{0,1\} \quad \forall i \in \mathcal{C}, k \in \{1,\dots,K_i\}, p \in \{1,\dots,P\}, \\
& && M \in \mathcal{M},
\end{aligned}
\tag{9}
$$

where $\mathcal{L}(\cdot)$ is the maximum LL yielded by model M on the considered data and the expression of the second objective function equals the number of estimated parameters. The first two constraints in (9) prevent transformations and segmentation of non-chosen variables, while the last one ensures behavioral realism. Because inconsistency with economic theory is generally understood as evidence of misspecification and invalidates all insights extracted from the estimated model, behavioral soundness must always be verified. In practice, the user is given the option to indicate a collection of constraints that she deems as sufficient, if satisfied, to prove model validity; $\mathcal{M}$ is the set of candidate models that satisfy these constraints. For example, taste parameters related to the cost of an alternative are usually expected to be negative, so as to decrease the utility of the alternative if its price increases. If the user includes such a constraint in the validity verification process, any model with a positive-valued cost parameter will be regarded as invalid, regardless of its performance in terms of goodness of fit and parsimony.

The problem described in (9) is a bi-objective, nonlinear combinatorial optimization problem. It cannot be solved by exact methods. Additionally, given $\boldsymbol{x}_{in}$, $\boldsymbol{\lambda}$ and $\boldsymbol{c}_n$, the number of different utility specifications that can be generated may be computed as

$$(2^P)^{|\mathcal{C}|-1} \times (1 + L \times 2^P)^{\sum_{i\in\mathcal{C}} K_i}. \tag{10}$$

This number grows exponentially with the size of $\boldsymbol{x}_{in}$ and $\boldsymbol{c}_n$. Let us consider again the illustrative example described in the previous section: with $\sum_{i\in\mathcal{C}} K_i = 5$ explanatory variables, $L = 2$ possible power transforms and $P = 2$ variables for segmentation, the number of possible specifications nearly reaches 250 000. With a single additional explanatory variable, that number is over 2 millions. Hence, exhaustive search is computationally intractable in any but trivial instances. We therefore investigate a metaheuristic approach.

4 Proposed Algorithm

We propose a multi-objective adaption of the variable neighborhood search (VNS) metaheuristic from Mladenovic and Hansen (1997) to solve the optimization problem formulated in (9). Metaheursitics typically have three ingredients: (i) exploration; (ii) intensification; and (iii) diversification. The exploration of the solution space relies on operators and neighborhood structures. Inspired by what a modeler would do, they must be flexible enough to potentially allow any feasible solution to be reached. These operators are described in Section 4.1. Intensification typically consists in a local search. The neighborhoods are exploited until a solution that is better than all its neighbors is found. In a multi-objective context however, the concept of "being better" must be qualified. This is discussed in Section 4.2. Finally, diversification is the mechanism that allows escaping from local optima. To this end, the main idea behind the original VNS is to systematically alternate several neighborhood structures; our multi-objective VNS (MO-VNS) algorithm makes use of a similar technique; we describe it in Section 4.3.

4.1 Exploration: Operators and Neighborhood Structures

We begin by defining the operators used to generate the neighbors of a given solution. These operators correspond to elementary modifications that are typically tested by analysts during manual utility specification.

Suppose that an arbitrary model M is characterized by the three controllers $\boldsymbol{v}_i$, $\boldsymbol{t}_i$ and $\boldsymbol{S}_i$, as in (4). The three operators considered by our algorithm are the following:

- Operator V-MOD either includes variable x_{ink} in M or excludes it. This is equivalent to switching the value of the corresponding controller v_{ik}:
$$\text{V-MOD}(M, i, k) : v_{ik} \leftarrow 1 - v_{ik}. \tag{11}$$
- Operator T-MOD substitutes the nonlinear transformation applied to variable x_{ink} by modifying its power parameter $\lambda_{t_{jk}}$:
$$\text{T-MOD}(M, i, k) : t_{ik} \leftarrow t_{ik} \pm 1. \tag{12}$$
- Operator S-MOD adds or removes an interaction between variable x_{ink} and a socioeconomic variable c_{np}. This corresponds to switching the value of s_{ikp}:
$$\text{S-MOD}(M, i, k, p) : s_{ikp} \leftarrow 1 - s_{ikp}. \tag{13}$$

All candidates generated from an arbitrary model M by these operators are grouped into three distinct neighborhoods of M. In other words, each of the neighborhoods $\mathcal{N}_{\text{V}}(M)$, $\mathcal{N}_{\text{T}}(M)$ and $\mathcal{N}_{\text{S}}(M)$ gathers all possible applications of one operator on M:

$$\begin{aligned} \mathcal{N}_{\text{V}}(M) &= \{\text{V-MOD}(M, i, k)\ \forall i, k\}, \\ \mathcal{N}_{\text{T}}(M) &= \{\text{T-MOD}(M, i, k)\ \forall i, k \mid v_{ik} = 1,\ 1 < t_{ik} < L\}, \\ \mathcal{N}_{\text{S}}(M) &= \{\text{S-MOD}(M, i, k, p)\ \forall i, k, p \mid v_{ik} = 1\}. \end{aligned} \tag{14}$$

We refer to $\mathcal{N}_{\text{V}}$, $\mathcal{N}_{\text{T}}$ and $\mathcal{N}_{\text{S}}$ as neighborhood structures. These may include models that do not verify the last constraint of (9); as a matter of fact, consistency with behavioral theory can only be verified after the model is estimated. This is discussed in the next section.

4.2 Intensification: Multi-Objective Local Search

Solving a multi-objective optimization problem is usually understood as obtaining a representative set of *non-dominated* solutions, called the *Pareto front*. Suppose two feasible solutions $M_1, M_2 \in \mathcal{M}$. M_1 is said to *dominate* M_2 if two conditions are fulfilled: (i) M_1 is no worse than M_2 in any objective $f_z, \forall z \in Z$; and (ii) M_1 is strictly better than M_2 in at least one. Notationally, this is written as

$$M_1 \prec M_2 \Leftrightarrow \begin{cases} f_z(M_1) \leq f_z(M_2), \forall z \in Z, \\ \exists z \in Z : f_z(M_1) < f_z(M_2). \end{cases} \tag{15}$$

By extension, a solution is *Pareto-optimal* if it is not dominated by any other feasible solution — *i.e.*, if none of the objectives can be improved without degrading another. The Pareto front $\mathcal{F}^*$ of the optimization problem in (9) may then be defined as

$$\mathcal{F}^* = \{M^* \in \mathcal{M} \mid \nexists M \in \mathcal{M} : M \prec M^*\}. \tag{16}$$

One should note that the true Pareto front $\mathcal{F}^*$ cannot be computed in any but trivial problem instances, as it requires all feasible solutions to be tested. As a consequence, the purpose of our MO-VNS algorithm is to generate good approximations of $\mathcal{F}^*$ in reasonable amounts of time.

In practice, local search (Algorithm 1, Lines 5–11) is used to iteratively improve an initial approximation of the Pareto front of the optimization problem under consideration. We denote such an approximation of $\mathcal{F}^*$ by $\mathcal{F}$. The MO-VNS starts by generating $\mathcal{D}$, a set of candidate solutions that contains all neighbors of all models in $\mathcal{F}$. A single neighborhood structure $\mathcal{N}_h$ is used to generate those; Section 4.3 explains how $\mathcal{N}_h$ is selected among $\{\mathcal{N}_\text{V}, \mathcal{N}_\text{T}, \mathcal{N}_\text{S}\}$. Then, a new candidate $M' \in \mathcal{D}$ is randomly selected (Algorithm 1, Line 5). After it is estimated through maximum likelihood estimation (MLE) and its behavioral validity is ascertained, M' is either added to $\mathcal{F}$ or rejected, depending on its objective values. Specifically, M' enters the front only if it is not dominated by any model $M \in \mathcal{F}$, *i.e.*, if no model M in $\mathcal{F}$ outperforms M' in one of the objective values while being at least as good in the other. If such condition is observed, the Pareto front is updated by removing all models dominated by M' and a new set $\mathcal{D}$ of candidates is generated (Algorithm 1, Lines 8–9). The local search algorithm stops after all candidate models in $\mathcal{D}$ have been tested without any of them improving the Pareto front.

Algorithm 1: MO-VNS for Assisted MNL

```
inputs: initial Pareto front $\mathcal{F}$,
        neighborhood structures $\mathcal{N}_1, \dots, \mathcal{N}_H$
1  $h \leftarrow 1$;
2  $improved \leftarrow$ False;
3  $\mathcal{D} \leftarrow \bigcup_{M \in \mathcal{F}} \mathcal{N}_h(M)$;
4  while $h \leq H$ do
5      $M' \leftarrow$ pop($\mathcal{D}$);
6      MLE($M'$);
7      if $M' \in \mathcal{M}$ and $\{M \in \mathcal{F} \mid M' \succ M\} = \emptyset$ then
8          $\mathcal{F} \leftarrow \mathcal{F} \cup \{M'\} \setminus \{M \in \mathcal{F} \mid M' \prec M\}$;
9          $\mathcal{D} \leftarrow \bigcup_{M \in \mathcal{F}} \mathcal{N}_h(M)$;
10         $improved \leftarrow$ True;
11     end
12     if $\mathcal{D} = \emptyset$ then
13         if $improved$ then
14             $improved \leftarrow$ False;
15             $h \leftarrow 1$;
16         else
17             $h \leftarrow h + 1$;
18         end
19         $\mathcal{D} \leftarrow \bigcup_{M \in \mathcal{F}} \mathcal{N}_h(M)$;
20     end
21 end
```

4.3 Diversification: Neighborhood Changes

As mentioned above, the main idea behind the VNS algorithm is to consider systematic changes of neighborhood structure during a local search subroutine so as to avoid getting stuck in local optima. Specifically, Mladenovic and Hansen (1997) suggest moving to a new structure each time a local optimum is found. Our MO-VNS makes use of a similar mechanism (Algorithm 1, Lines 12–20) to "diversify" the neighborhood structure $\mathcal{N}_h$ that is used to generate $\mathcal{D}$. Each time the multi-objective local search stops — *i.e.*, each time the Pareto front $\mathcal{F}$ cannot be improved by any of its neighbors in $\mathcal{D}$ — a new local search subroutine is started in a new neighborhood structure. The choice of the new structure depends on whether $\mathcal{N}_h$ has effectively improved the Pareto front: if $\mathcal{F}$ was updated at least once while searching in $\mathcal{N}_h$, the algorithm moves back to the first structure $\mathcal{N}_1$; otherwise, structure $\mathcal{N}_{h+1}$ is considered instead. The MO-VNS algorithm stops when none of the structures can improve the Pareto front any further.

4.4 Post-Processing Validation

By construction, our algorithm solves the multi-objective utility specification problem by approximating the Pareto front of all feasible model specifications. Because the two objective functions of the problem — the maximized log likelihood (LL) and the number of estimated parameters — are conflicting, the non-dominated solutions obtained by the MO-VNS cannot be ranked. The choice of a single best model is left to the user; it should be motivated by the purpose that the selected model will serve. Nonetheless, we believe that some of the predominantly used metrics in model selection deserve to be examined as guidelines.

We first consider out-of-sample validation for assessing how accurately the models in the Pareto front generalize to new data. Testing the model on unseen data is the only way to effectively reduce the possibility of erroneous correlation between inputs and choice outcomes that may have been captured during the training. The method consists in excluding part of the data from the model identification process. All models tested by the MO-VNS are therefore estimated only on part of the full dataset. Then, after the algorithm stops, the accuracy of the trained models is measured on the hold-out data; the LLs yielded "in-sample" and "out-of-sample" are compared to detect underfitting or overfitting issues.

In addition to out-of-sample validation, we also identify the optimal models in terms of the Akaike and Bayesian information criteria (Akaike, 1974; Schwarz, 1978). These criteria work by "balancing" the LL yielded by a model on a given dataset with a penalty that is proportional to the number of estimated parameters; in other words, they quantify the trade-off between the two objectives of the problem defined in (9). The Akaike information criterion (AIC) and the Bayesian information criterion (BIC) are defined as

$$\begin{aligned} \text{AIC} &= -2\mathcal{L} + 2K, \\ \text{BIC} &= -2\mathcal{L} + K\log(N), \end{aligned} \tag{17}$$

where, for brevity, $\mathcal{L}$ is the in-sample maximized LL, K is the number of estimated parameters and N is the sample size. The penalty considered by the BIC is more severe than the one of the AIC as soon as $N \geq 8$; we therefore expect BIC-optimal models to be more parsimonious — and worse in terms of LL — than AIC-optimal ones.

4.5 Implementation Notes

For the sake of clarity, some details related to the implementation of the MO-VNS are omitted from the previous sections. In particular, a number of measures are undertaken to reduce the computational time of its runs. We discuss these measures here.

All MLEs are performed using the Biogeme package for Python (Bierlaire, 2018; Bierlaire, 2020) and its implementation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. Substantial amounts of time are gained by limiting the estimation outputs provided by Biogeme to what is strictly necessary: as the variance-covariance matrix of the parameter estimates is expensive and of no use in our context, we simply avoid computing it.

Additionally, because the estimation of a model through MLE may take up to several minutes depending on its number of parameters, a list of estimation results for all tested models is updated throughout the whole run of the algorithm. Those are retrieved whenever appropriate, so as to avoid estimating the same model twice.

Finally, whenever possible, the estimation outputs of previously tested models are used in order to reduce the number of iterations that the BFGS algorithm requires to converge. As every model arises from the application of a single operator on a previously estimated model, the parameter estimates of the former are good approximations of the values that should be reached by the latter; they are therefore used as initial values whenever appropriate. In practice, no parameter that has already been estimated in a previous model is ever estimated from scratch again, greatly reducing the overall computational cost of our algorithm.

5 Case Study: Swissmetro

5.1 Dataset

The Swissmetro dataset (Bierlaire et al., 2001) consists of survey data collected in Switzerland in 1998, initially used to analyze the potential impact of the Swissmetro, the Swiss precursor of Hyperloop. The 1 192 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. Table 1 gives a brief description of the subset of variables used in the context of this study; a more detailed and complete description of the dataset and its collection procedure may be found in Antonini et al. (2007). The Swissmetro dataset is publicly available online.

The 8 continuous variables of Table 1 are considered as potential explanatory variables. It is common practice to use logarithmic transforms of variables related to time or cost (Ben-Akiva and Lerman, 1985). We therefore define $\boldsymbol{\lambda} = \{1, \frac{1}{2}, 0\}$, where $\lambda = \frac{1}{2}$ corresponds to a square-root transformation, which we see as a middle ground between the identity and logarithmic transforms. Additionally, the 5 categorical variables of Table 1 are considered for segmentation. GA, MALE and FIRST are binary, whereas LUGGAGE and WHO each have three categories. According to (10), there are over 8×10^{18} possible specifications in this setting. Finally, due to the nature of the potential explanatory variables, all $B_{ik}(\boldsymbol{c}_n, \boldsymbol{s}_{ik})$ are imposed to be negative for the sake of consistency with behavioral interpretation.

5.2 Results

We apply the MO-VNS on the Swissmetro dataset with the configuration described in the previous section. The initial Pareto front contains a single specification that only includes the alternative-specific constants (ASCs). The ASC of the train alternative is normalized to zero. 11 522 models are tested in little more than 20 hours. Figure 1 illustrates their performance in terms of final log likelihood (LL) and number of parameters. We use the logit model presented in Bierlaire et al. (2001) as a benchmark for comparison; its estimation results are reported in Table 3 in the Appendix.

Table 1: Swissmetro dataset. Definition of the considered variables and statistics.

Variable	Min.	Max.	Mean	Std.
$\mathrm{TT}_{\mathrm{TRAIN}}$ *Train travel time [min]. Based on the car distance.*	31	1 049	166.63	77.35
$\mathrm{CO}_{\mathrm{TRAIN}}$ *Train cost [CHF]. If the traveler owns a GA, equal to its price.*	4	5 040	514.34	1 088.93
$\mathrm{HE}_{\mathrm{TRAIN}}$ *Train headway [min].*	30	120	70.10	37.43
$\mathrm{TT}_{\mathrm{SM}}$ *Swissmetro travel time [min]. A speed of 500 km/h is considered.*	8	796	87.47	53.55
$\mathrm{CO}_{\mathrm{SM}}$ *Swissmetro cost [CHF]. Proportional to the rail fare.*	6	6 720	670.34	1 441.59
$\mathrm{HE}_{\mathrm{SM}}$ *Swissmetro headway [min].*	10	30	20.02	8.16
$\mathrm{TT}_{\mathrm{CAR}}$ *Car travel time [min].*	0	1 560	123.80	88.71
$\mathrm{CO}_{\mathrm{CAR}}$ *Car cost [CHF]. A fixed average cost per kilometer is considered.*	0	520	78.74	55.26
FIRST *1 if first-class traveler, 0 otherwise.*	0	1	0.47	0.50
GA *Travel card ownership. 1 if the traveler owns one, 0 otherwise.*	0	1	0.14	0.35
LUGGAGE *0 if none, 1 if one piece, 3 if several pieces.*	0	3	0.68	0.60
MALE *Traveler's gender. 0 if female, 1 if male.*	0	1	0.75	0.43
WHO *Who pays for the trip. 1 if self, 2 if employer, 3 if half-half.*	1	3	1.49	0.71

As depicted in Figure 1, the models in the Pareto front approximately follow the shape of a hyperbola branch. They have between 2 and 51 estimated parameters and range from $-7\,358.2$ to $-5\,819.9$ in terms of final LL. The leftmost non-dominated model corresponds to the ASCs-only model in the initial Pareto front. It should be noted that the front does not contain any model with 28 parameters; the reason behind this discrepancy is that all candidates with 28 parameters that outperform the BIC-optimal solution in terms of goodness of fit are behaviorally invalid. The validity verification mechanism proves itself useful again when the number of estimated parameters is larger than 30. Several models outperform the Pareto front in terms of LL, but they are all rejected due to their invalidity. The same mechanism also limits the number of estimated parameters to 51, despite the theoretical maximum being 80 given the considered configuration. As reported in Table 2, the BIC-optimal and AIC-optimal solutions yield a LL of $-5\,873.3$ and $-5\,823.0$, respectively. The AIC-optimal model includes 18 more parameters than the BIC-optimal one for an improvement of 50.3 in the log-likelihood. The estimation results of these two models are shown in Table 5 and Table 6 in the Appendix, respectively.

Finally, Figure 2 serves the purpose of comparing the in-sample and out-of-sample LLs yielded by all models in the Pareto front. These values are normalized with respect to the number of observations in the training and validation data, so as to compare them despite the difference in size between the two sets. We observe that the ratio between in-sample and

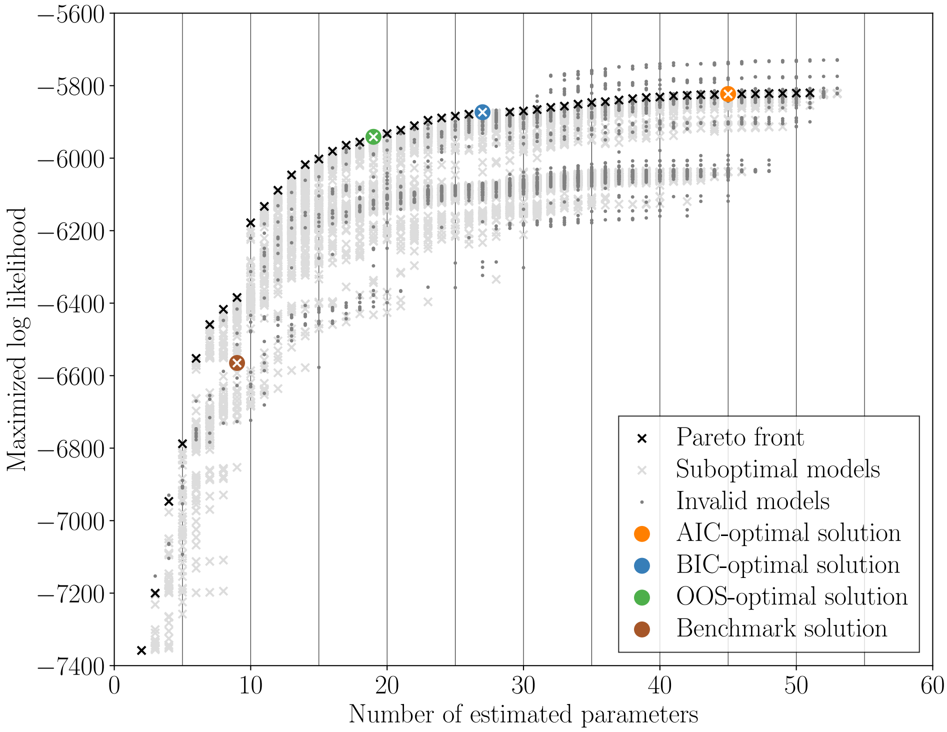

Figure 1: MO-VNS results. Pareto front visualization.

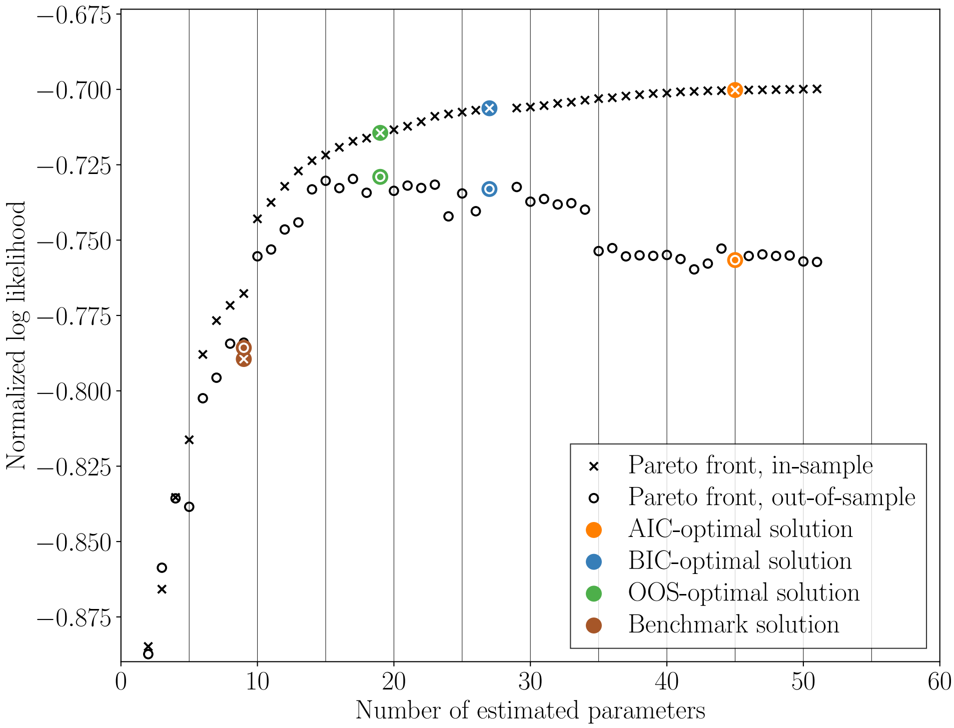

Figure 2: MO-VNS results. In-sample and out-of-sample log likelihood comparison.

out-of-sample LLs is approximately constant for models that include up to 15 parameters. Then, as the number of parameters grows, the accuracy on the validation data decreases until it reaches a plateau, while the fit on the training data keeps increasing. The plateau suggests that models with 35 parameters or more — including the AIC-optimal model — may be overfitting. On the contrary, the limited explanatory power of all utility specifications with less than 10 parameters is a clear sign of underfitting. The largest out-of-sample LL is obtained by the 19-parameter model; we denote it as the OOS-optimal model. Its estimation results are given in Table 4 in the Appendix. As shown in Table 2, the OOS-optimal model is even more parsimonious than the BIC-optimal one; its in-sample LL is slightly worse as a result.

Table 2: Performance of the benchmark, OOS-, BIC- and AIC-optimal models.

	Benchmark	OOS-optimal	BIC-optimal	AIC-optimal
Est. parameters	9	19	27	45
In-sample LL	−6 565.0	−5 941.1	−5 873.2	−5 823.0
Out-of-sample LL	−1633.5	−1515.6	−1524.0	−1573.0
AIC	13 148.0	11 920.3	11 800.5	11 735.9
BIC	13 211.3	12 053.8	11 990.2	12 052.0

6 Conclusion

In this paper, we propose a new methodology for assisted specification of discrete choice models. We define a multi-objective combinatorial optimization problem with two conflicting objectives and make use of a variable neighborhood metaheuristic to generate solutions in a way that mimics human modelers. The validity of the proposed algorithm is empirically demonstrated using a publicly available mode choice dataset. Out-of-sample validation shows that our algorithm generates sets of high-quality specifications in reasonable amounts of time. We believe our approach can assist analysts in the task of model development and provide relevant insights. Model interpretability and behavioral realism are ensured in all generated models by means of a mechanism that systematically verifies economic soundness throughout the search and rejects all specifications that are behaviorally inconsistent.

Intended future work includes the development of a system that allows “fixing” part of the utility specification while letting the algorithm “enrich” it freely. It is known from empirical studies that some key variables of well-studied choice problems should be considered in a particular manner; compliance with the existing literature and improvements in terms of computational effort could therefore be obtained by introducing appropriate restrictions to the search space. Another relevant direction of search consists in extending the framework to advanced model structures. Our methodology is already applicable to nested and cross-nested logits, as it deals with the specification of utility functions. An extension would consist in providing algorithmic assistance to the definition of nests in nested logit models, or to the specification of error components in mixed logits. In such cases, computational cost and overall runtime could be reduced by using heuristics based on partial estimation, where the maximum likelihood algorithm is prematurely interrupted if the model under consideration is not promising. Finally, additional algorithmic improvements that could be brought to the MO-VNS include (a) developing new operators and neighborhood structures, (b) replacing the local search subroutine with simulated annealing and (c) using distinct starting points so as to improve diversification.

References

Akaike, H. (1974). A new look at the statistical model identification, *IEEE transactions on automatic control* **19**(6): 716–723.

Alwosheel, A., van Cranenburgh, S. and Chorus, C. G. (2019). 'computer says no'is not enough: Using prototypical examples to diagnose artificial neural networks for discrete choice analysis, *Journal of choice modelling* **33**: 100186.

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

Ben-Akiva, M. E. and Lerman, S. R. (1985). *Discrete choice analysis: theory and application to travel demand*, Vol. 9, MIT press.

Bentz, Y. and Merunka, D. (2000). Neural networks and the multinomial logit for brand choice modelling: a hybrid approach, *Journal of Forecasting* **19**(3): 177–200.

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

Bierlaire, M. (2018). Pandasbiogeme: a short introduction, *Technical report*, TRANSP-OR 181219. Transport and Mobility Laboratory, ENAC, EPFL.

Bierlaire, M. (2020). A short introduction to pandasbiogeme, *Technical report*, TRANSP-OR 200605. Transport and Mobility Laboratory, ENAC, EPFL.

Bierlaire, M., Axhausen, K. and Abay, G. (2001). The acceptance of modal innovation: The case of swissmetro, *Proceedings of the 1st Swiss Transportation Research Conference*.

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. (2002). A practical information-theoretic approach, *Model selection and multimodel inference, 2nd ed. Springer, New York* .

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

Claeskens, G. and Hjort, N. L. (2003). The focused information criterion, *Journal of the American Statistical Association* **98**(464): 900–916.

Claeskens, G. and Hjort, N. L. (2008). *Model Selection and Model Averaging*, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press.

De Carvalho, M., Dougherty, M., Fowkes, A. and Wardman, M. (1998). Forecasting travel demand: a comparison of logit and artificial neural network methods, *Journal of the Operational Research Society* **49**(7): 717–722.

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* .

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

Hillel, T., Elshafie, M. Z. and 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**(1): 29–42.

Hruschka, H., Fettes, W., Probst, M. and Mies, C. (2002). A flexible brand choice model based on neural net methodology a comparison to the linear utility multinomial logit model and its latent class extension, *OR spectrum* **24**(2): 127–143.

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

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.

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* .

Rodrigues, F., Ortelli, N., Bierlaire, M. and Pereira, F. (2019). Bayesian automatic relevance determination for utility function specification in discrete choice models, *arXiv preprint arXiv:1906.03855* .

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

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

Tang, L., Xiong, C. and Zhang, L. (2015). Decision tree method for modeling travel mode switching in a dynamic behavioral process, *Transport. Plan. Techn.* **38**(8).

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.

Wong, M. and Farooq, B. (2019). Reslogit: A residual neural network logit model, *arXiv preprint arXiv:1912.10058* .

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* .

Appendix

Table 3: Benchmark model. Estimation results. Variable SEATS is binary and describes configuration of seats in the Swissmetro.

Parameter	Value	Rob. std err	Rob. t-test
ASC_{CAR}	0.0620	0.0585	1.06
$\text{ASC}_{\text{TRAIN}}$	−1.16	0.127	−9.14
β_AGE	0.190	0.0352	5.39
β_CO	−0.00123	0.0000777	−15.8
β_GA	7.49	0.444	16.9
β_HE	−0.00718	0.000921	−7.80
β_LUGGAGE	−0.123	0.0484	−2.55
β_SEATS	0.162	0.0868	1.87
β_TIME	−0.0113	0.000755	−15.0
Initial log likelihood			−8603.2
Final log likelihood			−6565.0

Table 4: OOS-optimal model. Estimation results. The numbers in parentheses refer to the nonlinear transform of the variable associated with each reported parameter.

Parameter	Value	Rob. Std err	Rob. t-test
β_$\text{TT}_{\text{TRAIN}}{}^{(0)}$	−2.87	0.114	−25.2
β_$\text{TT}_{\text{TRAIN}}{}^{(0)}{}_{\text{GA}=1}$	1.90	0.0753	25.2
β_$\text{CO}_{\text{TRAIN}}{}^{(0)}$	−1.48	0.0769	−19.2
β_$\text{HE}_{\text{TRAIN}}{}^{(\frac{1}{2})}$	−0.0618	0.00822	−7.53
$\text{ASC}_{\text{SM}}{}^{(-)}$	−2.81	0.229	−12.3
$\text{ASC}_{\text{SM}}{}^{(-)}{}_{\text{GA}=1}$	3.81	0.172	22.2
β_$\text{CO}_{\text{SM}}{}^{(0)}$	−1.50	0.0700	−21.5
β_$\text{CO}_{\text{SM}}{}^{(0)}{}_{\text{MALE}=1}$	0.0832	0.0130	6.38
β_$\text{TT}_{\text{SM}}{}^{(0)}$	−1.79	0.0758	−23.6
β_$\text{TT}_{\text{SM}}{}^{(0)}{}_{\text{WHO}=2}$	0.242	0.0199	12.1
β_$\text{TT}_{\text{SM}}{}^{(0)}{}_{\text{WHO}=3}$	0.193	0.0359	5.37
$\text{ASC}_{\text{CAR}}{}^{(-)}$	−8.99	0.329	−27.3
$\text{ASC}_{\text{CAR}}{}^{(-)}{}_{\text{MALE}=1}$	1.32	0.186	7.10
β_$\text{TT}_{\text{CAR}}{}^{(\frac{1}{2})}$	−0.0885	0.0191	−4.63
β_$\text{TT}_{\text{CAR}}{}^{(\frac{1}{2})}{}_{\text{MALE}=1}$	−0.0903	0.0171	−5.27
β_$\text{CO}_{\text{CAR}}{}^{(1)}$	−0.00962	0.00124	−7.79
β_$\text{CO}_{\text{CAR}}{}^{(1)}{}_{\text{FIRST}=1}$	−0.00425	0.000655	−6.48
β_$\text{CO}_{\text{CAR}}{}^{(1)}{}_{\text{WHO}=2}$	0.00640	0.000968	6.61
β_$\text{CO}_{\text{CAR}}{}^{(1)}{}_{\text{WHO}=3}$	0.00680	0.00166	4.10
Initial log likelihood			−8603.2
Final log likelihood			−5941.1

Table 5: BIC-optimal model. Estimation results. The numbers in parentheses refer to the nonlinear transform of the variable associated with each reported parameter.

Parameter	Value	Rob. Std err	Rob. t-test
$\beta\_\mathrm{TT}_{\mathrm{TRAIN}}{}^{(0)}$	−2.71	0.137	−19.8
$\beta\_\mathrm{TT}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{FIRST}=1}$	−0.860	0.134	−6.44
$\beta\_\mathrm{TT}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{GA}=1}$	2.61	0.200	13.1
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}$	−1.36	0.0843	−16.1
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{FIRST}=1}$	0.297	0.0446	6.66
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{GA}=1}$	−3.46	0.374	−9.25
$\beta\_\mathrm{HE}_{\mathrm{TRAIN}}{}^{(\frac{1}{2})}$	−0.0816	0.0101	−8.06
$\beta\_\mathrm{HE}_{\mathrm{TRAIN}}{}^{(\frac{1}{2})}{}_{\mathrm{LUGGAGE}=1}$	0.0245	0.00661	3.70
$\beta\_\mathrm{HE}_{\mathrm{TRAIN}}{}^{(\frac{1}{2})}{}_{\mathrm{LUGGAGE}=3}$	−0.0190	0.0152	−1.25
$\mathrm{ASC}_{\mathrm{SM}}{}^{(-)}$	−3.01	0.263	−11.4
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}$	−1.50	0.0993	−15.1
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{FIRST}=1}$	−0.639	0.136	−4.69
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{WHO}=2}$	0.256	0.0212	12.1
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{WHO}=3}$	0.179	0.0367	4.86
$\beta\_\mathrm{CO}_{\mathrm{SM}}{}^{(0)}$	−1.44	0.0712	−20.1
$\beta\_\mathrm{CO}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{GA}=1}$	−1.93	0.353	−5.46
$\beta\_\mathrm{CO}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{MALE}=1}$	0.0866	0.0137	6.31
$\mathrm{ASC}_{\mathrm{CAR}}{}^{(-)}$	−8.46	0.360	−23.5
$\mathrm{ASC}_{\mathrm{CAR}}{}^{(-)}{}_{\mathrm{FIRST}=1}$	−1.23	0.282	−4.35
$\mathrm{ASC}_{\mathrm{CAR}}{}^{(-)}{}_{\mathrm{GA}=1}$	−11.7	1.51	−7.76
$\mathrm{ASC}_{\mathrm{CAR}}{}^{(-)}{}_{\mathrm{MALE}=1}$	1.31	0.187	7.02
$\beta\_\mathrm{TT}_{\mathrm{CAR}}{}^{(\frac{1}{2})}$	−0.0898	0.0193	−4.66
$\beta\_\mathrm{TT}_{\mathrm{CAR}}{}^{(\frac{1}{2})}{}_{\mathrm{MALE}=1}$	−0.0894	0.0173	−5.18
$\beta\_\mathrm{CO}_{\mathrm{CAR}}{}^{(1)}$	−0.00728	0.0015	−4.85
$\beta\_\mathrm{CO}_{\mathrm{CAR}}{}^{(1)}{}_{\mathrm{FIRST}=1}$	−0.00832	0.00159	−5.24
$\beta\_\mathrm{CO}_{\mathrm{CAR}}{}^{(1)}{}_{\mathrm{WHO}=2}$	0.00689	0.00101	6.84
$\beta\_\mathrm{CO}_{\mathrm{CAR}}{}^{(1)}{}_{\mathrm{WHO}=3}$	0.00645	0.00165	3.90
Initial log likelihood			−8603.2
Final log likelihood			−5873.2

Table 6: AIC-optimal model. Estimation results. The numbers in parentheses refer to the nonlinear transform of the variable associated with each reported parameter.

Parameter	Value	Rob. std err	Rob. t-test
$\beta\_\mathrm{TT}_{\mathrm{TRAIN}}{}^{(0)}$	−2.85	0.141	−20.3
$\beta\_\mathrm{TT}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{FIRST}=1}$	−0.923	0.145	−6.38
$\beta\_\mathrm{TT}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{GA}=1}$	2.61	0.199	13.1
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}$	−1.96	0.138	−14.2
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{FIRST}=1}$	0.266	0.0455	5.85
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{GA}=1}$	−1.46	0.131	−11.2
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{LUGGAGE}=1}$	0.499	0.105	4.75
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{LUGGAGE}=3}$	−0.0825	0.260	−0.317
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{MALE}=1}$	0.180	0.0999	1.80
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{WHO}=2}$	0.0980	0.0386	2.54
$\beta\_\mathrm{CO}_{\mathrm{TRAIN}}{}^{(0)}{}_{\mathrm{WHO}=3}$	0.178	0.0619	2.88
$\beta\_\mathrm{HE}_{\mathrm{TRAIN}}{}^{(\frac{1}{2})}$	−0.0627	0.0172	−3.64
$\beta\_\mathrm{HE}_{\mathrm{TRAIN}}{}^{(\frac{1}{2})}{}_{\mathrm{LUGGAGE}=1}$	0.00428	0.0189	0.227
$\beta\_\mathrm{HE}_{\mathrm{TRAIN}}{}^{(\frac{1}{2})}{}_{\mathrm{LUGGAGE}=3}$	−0.0682	0.0415	−1.64
$\mathrm{ASC}_{\mathrm{SM}}{}^{(-)}$	−2.10	0.35	−6.01
$\mathrm{ASC}_{\mathrm{SM}}{}^{(-)}{}_{\mathrm{MALE}=1}$	−1.06	0.326	−3.27
$\mathrm{ASC}_{\mathrm{SM}}{}^{(-)}{}_{\mathrm{WHO}=2}$	0.315	0.324	0.970
$\mathrm{ASC}_{\mathrm{SM}}{}^{(-)}{}_{\mathrm{WHO}=3}$	−2.33	0.423	−5.50
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}$	−1.76	0.154	−11.5
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{FIRST}=1}$	−0.727	0.15	−4.83
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{LUGGAGE}=1}$	−0.204	0.0749	−2.72
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{LUGGAGE}=3}$	−0.337	0.211	−1.60
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{MALE}=1}$	0.404	0.141	2.87
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{WHO}=2}$	0.113	0.14	0.811
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{WHO}=3}$	0.915	0.186	4.92
$\beta\_\mathrm{CO}_{\mathrm{SM}}{}^{(0)}$	−2.17	0.125	−17.4
$\beta\_\mathrm{CO}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{LUGGAGE}=1}$	0.586	0.0928	6.32
$\beta\_\mathrm{CO}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{LUGGAGE}=3}$	0.0427	0.24	0.177
$\beta\_\mathrm{CO}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{MALE}=1}$	0.322	0.0952	3.38
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{WHO}=2}$	0.0660	0.06	1.10
$\beta\_\mathrm{TT}_{\mathrm{SM}}{}^{(0)}{}_{\mathrm{WHO}=3}$	0.423	0.102	4.17
$\beta\_\mathrm{HE}_{\mathrm{SM}}{}^{(\frac{1}{2})}$	−0.0319	0.0132	−2.42
$\mathrm{ASC}_{\mathrm{CAR}}{}^{(-)}$	−9.81	0.442	−22.2
$\mathrm{ASC}_{\mathrm{CAR}}{}^{(-)}{}_{\mathrm{FIRST}=1}$	−0.932	0.281	−3.31
$\mathrm{ASC}_{\mathrm{CAR}}{}^{(-)}{}_{\mathrm{GA}=1}$	−3.24	0.236	−13.8
$\mathrm{ASC}_{\mathrm{CAR}}{}^{(-)}{}_{\mathrm{LUGGAGE}=1}$	0.881	0.259	3.40
$\mathrm{ASC}_{\mathrm{CAR}}{}^{(-)}{}_{\mathrm{LUGGAGE}=3}$	−0.896	0.766	−1.17
$\mathrm{ASC}_{\mathrm{CAR}}{}^{(-)}{}_{\mathrm{MALE}=1}$	1.49	0.283	5.26
$\beta\_\mathrm{TT}_{\mathrm{CAR}}{}^{(\frac{1}{2})}$	−0.0746	0.0225	−3.31
$\beta\_\mathrm{TT}_{\mathrm{CAR}}{}^{(\frac{1}{2})}{}_{\mathrm{FIRST}=1}$	−0.0824	0.0152	−5.43
$\beta\_\mathrm{TT}_{\mathrm{CAR}}{}^{(\frac{1}{2})}{}_{\mathrm{MALE}=1}$	−0.0680	0.0195	−3.48
$\beta\_\mathrm{CO}_{\mathrm{CAR}}{}^{(1)}$	−0.0141	0.00141	−9.95
$\beta\_\mathrm{CO}_{\mathrm{CAR}}{}^{(1)}{}_{\mathrm{GA}=1}$	−0.0150	0.00698	−2.15
$\beta\_\mathrm{CO}_{\mathrm{CAR}}{}^{(1)}{}_{\mathrm{WHO}=2}$	0.00940	0.00169	5.56
$\beta\_\mathrm{CO}_{\mathrm{CAR}}{}^{(1)}{}_{\mathrm{WHO}=3}$	0.0132	0.00276	4.77
Initial log likelihood			−8603.2
Final log likelihood			−5823.0