A Grammar-Based Framework for Utility Function Specification: Validation with Synthetic Data

Shadi Haj Yahia, Omar Mansour, Tomer Toledo

Faculty of Civil and Environmental Engineering,

Technion – Israel Institute of Technology, Haifa 32000, Israel

Emails:

shadi8@campus.technion.ac.il

omar@technion.ac.il

toledo@technion.ac.il

Abstract

This study introduces a grammar-based framework for utility function specification in discrete choice models (DCMs), combining interpretability and domain knowledge with data-driven automation. Using Grammatical Evolution (GE), the framework systematically explores variables, transformations, and interactions while enforcing theoretical constraints. Tested on synthetic data generated from a predefined "true" model, the approach demonstrates its ability to recover accurate utility specifications and prioritize key variables like travel time and cost. By balancing complexity and simplicity through parsimony penalties, this framework offers a robust, interpretable alternative to traditional trial-and-error methods, enhancing consistency and precision in DCM applications.

1. Introduction

Discrete choice models (DCMs) are widely used to understand and predict individual or group decision-making among alternatives in fields like transportation, marketing, and policy analysis. Traditionally built as random utility models (RUMs) (McFadden, 1974), they assume decision-makers maximize utility, with utility functions quantifying the influence of attributes and individual characteristics. These functions' analytical form enables interpretability, offering insights into choice behavior. However, utility specification often relies on a subjective, trial-and-error process, introducing inconsistencies, inefficiencies, and potential biases in model results. Manual specification is labor-intensive, prone to errors like omitting key variables (Torres et al., 2011) or using incorrect transformations, and lacks a systematic framework to incorporate domain knowledge (Kling, 1989).

Recent advances in machine learning (ML) provide tools to uncover complex relationships in data but face challenges of interpretability and incorporating theoretical constraints (Haj-Yahia et al., 2023). Hybrid approaches, such as combining ML with traditional DCMs (Sifringer et

al., 2020; Han et al., 2020), and optimization-based methods for automated utility specification (Ortelli et al., 2021) show promise but remain limited by computational demands, reliance on pre-specified variables, and inconsistent alignment with domain knowledge.

This study addresses these gaps by introducing a grammar-based approach to utility specification in DCMs. Leveraging grammatical evolution (GE), an evolutionary algorithm, this method automates the selection of variables, transformations, and interactions while adhering to theoretical constraints. By combining data-driven flexibility with interpretability, the framework offers a systematic alternative to manual specification methods, improving efficiency and robustness in model development.

2. Methods

2.1. Problem formulation

Specification of DCM utility functions requires formulating mathematical functions for each of the choice alternatives. The specification involves selection of variables to be included, their functional forms and interactions, and ensuring that the model predictions comply with expectations regarding behavior. At the same time, the specification should be parsimonious to reduce the risk of overfitting.

To facilitate these requirements, the utility specification task is formulated as an optimization problem with the objective is to maximize the model performance fitness subject to three types of constraints that guarantee identification of the generated model, its agreement with the modeling conventions and preferences set by the modeler and with domain knowledge:

$$\max \quad model\ performence\ measure\ (MPM) \tag{1}$$

$$\text{s.t.} \quad Model\ identification \tag{2}$$

$$Specification\ conventions \tag{3}$$

$$Domain\ knowledge\ adherence \tag{4}$$

The objective function evaluates candidate utility specifications based on two key criteria: how well the model fits the data and the specification simplicity, which makes its interpretation easier. The adopted objective function used to measure the model performance is therefore given by:

$$MPM = LL - \alpha \cdot k \tag{5}$$

where MPM is the model performance measure. LL is the model log-likelihood. k is the number of parameters in the specification, and α is a predefined penalty parameter. This function incorporates both AIC and BIC as special cases when $\alpha = 1$, and $\alpha = \frac{1}{2}\log(\mathrm{N})$ (where N is the sample size), respectively.

The constraints for utility function specification ensure consistency with fundamental principles and modeling preferences. In DCMs, choice probabilities depend on differences in utility between alternatives, not their absolute values. To maintain identification, normalization is required: for a model with J alternatives, only $J-1$ alternative-specific constants (ASCs) can be estimated, with the remaining ASC fixed to zero. Socio-demographic variables, such as income or age, that do not vary across alternatives can appear in at most $J-1$ utility functions. Categorical variables, such as gender or education, are converted into binary dummy indicators, with one category omitted as a base reference.

Modeling preferences allow for optional constraints reflecting the modeler's style and norms. When interaction terms between socio-demographic variables and alternative attributes are included in the utility function, the main effects of these attributes must also be included to ensure interpretability. To prevent overfitting and maintain simplicity, the level of interactions is limited, for example, by allowing only two-variable interactions and excluding higher-order terms. Utility functions may also be designed to share variables, interactions, or parameter structures, with flexibility for generic or alternative-specific parameters. Additionally, the modeler defines a set of allowable transformations, constraining the search space for optimization.

Finally, estimated model parameters must align with established behavioral theory. For instance, it is expected that increases in travel time or cost reduce an alternative's utility and choice probability, as reflected by negative parameter sensitivities and elasticities. These constraints ensure the resulting models remain theoretically sound and behaviorally meaningful.

2.2. Grammar

The grammar, shown in Figure 1 is tailored to generate mathematically valid utility functions that satisfy the constraints described above. Its starting symbol (S) is decomposed using Production rule (1) into J utility functions (U_j) – one for each alternative in the DCM. Production rule (2) constructs additive utility functions that include one or more expressions (E_j). The number of expressions in the utility is unbounded. Each expression can take two forms as specified in Production rule (3): It may be a single transformed variable (W_j) or an interaction between a transformed variable and an already existing expression. This version of the grammar does not bound the level of interaction. Modified versions can guarantee satisfying upper bounds on the interaction level. For example, the following production rule allows E_j to only take a single transformed variable or an interaction between two transformed variables, and so limits interactions to be between two variables at most:

$$< E_j > \rightarrow < W_j >< O >< W_j > \quad (6)$$
$$\mid < W_j >$$

Production rule (4) defines the mathematical operators that may be used. These include both multiplication and division. It is straightforward to include additional operators. Production rule (5) defines a transformed variable (W_j) results from applying a transformation function (F) to a variable (V_j). The available transformations, (e.g., linear, logarithmic, exponential, Box-Cox) are defined in Production rule (6). Finally, Production rules (7) list the variables that

are available for each of the utility functions U_j. These include alternative attributes and socio-demographic characteristics. Categorical variables with K levels are represented as a set of $K - 1$ dummy variables to maintain identification. For example, consider two variables for U_j: $cost_j$ and $income$. The cost variable is continuous, and income is a categorical variable with three levels. The Production rule expands V_j to either $cost_j$ or a set of two $income$ dummy variables, as follows:

$$< V_j > \rightarrow cost_j \qquad (7)$$
$$|(income_1 + income_2 + 1)$$

The categorical variables are one-hot dummy coded. The "+1" symbol signifies that the main effect is included in case an interaction of the categorical with an alternative attribute is included, which would satisfy the main effects constraints.

$\mathrm{N} = \{\mathrm{U_j},\ \mathrm{E_j},\ \mathrm{W_j},\ \mathrm{V_j},\ \mathrm{O},\ \mathrm{F}\}$

$\mathrm{T} = \{+,\ *,\ \text{available variables}\}$

$\mathrm{S} = \{[< \mathrm{U_1} >, \ldots, < \mathrm{U_j} >, \ldots, < \mathrm{U_J} >]\}$

P =

(1) $< \mathrm{S} > \rightarrow < \mathrm{U_1} >, \ldots, < \mathrm{U_j} >, \ldots, < \mathrm{U_J} > \quad (0)$

(2) $< \mathrm{U_j} > \rightarrow <\mathrm{U_j}>+<\mathrm{E_j}> \quad (0)$

$|<\mathrm{E_j}> \quad (1)$

(3) $< \mathrm{E_j} > \rightarrow <\mathrm{E_j}><\mathrm{O}><\mathrm{E_j}> \quad (0)$

$|<\mathrm{W_j}> \quad (1)$

(4) $< \mathrm{O} > \rightarrow * \quad (0)$

$|/ \quad (1)$

(5) $< \mathrm{W_j} > \rightarrow < \mathrm{F} > (< \mathrm{V_j} >) \quad (0)$

(6) $< \mathrm{F} > \rightarrow$ sef of allowed transformations

(7) $< \mathrm{V_j} > \rightarrow$ set of available variables for alternative j

Figure 1. Grammar encoding

2.3. Optimization

The grammar-based optimization framework begins by generating an initial population of chromosomes, which are decoded into utility function skeletons. Parameters are estimated using training data and maximum likelihood estimation. Candidate models are evaluated in two stages: first, adherence to theoretical constraints (e.g., negative sensitivities to travel time and

cost) is verified, and non-compliant models are assigned lowest fitness scores. Second, compliant models are assessed for fit using validation data to improve generalization and prevent overfitting.

The genetic algorithm iteratively refines solutions using selection, crossover, and mutation until termination criteria, such as performance convergence or a generation limit, are met. Selection is performed via a tournament, while crossover employs a k-point block operator to combine parent solutions without disrupting utility function integrity. Mutation introduces diversity by randomly altering codon values within a fixed range, ensuring broad exploration of the solution space.

3. Case study

The proposed framework is demonstrated with a synthetic dataset derived from the original Swissmetro observations (Bierlaire et al., 2001) using known choice model and utility functions. The data is used to evaluate the ability of the models developed using the grammar-based method to recover the "true" model, also depending on the value of the penalty parameter α.

The synthetic dataset was generated using a predefined "true" utility model with the following functional form:

$$V_j = \beta_{0_j} + \beta_{time_j} log\left(Travel\ time_j\right) + \beta_{headway_j} log\left(Headway_j\right) + \beta_{cost_j} log\left(Travel\ cost_j\right) + \sum_g \beta_{cost,g_j} log\left(Travel\ cost_j\right) \delta_{income,g} + \sum_h \beta_{age,h_j} \delta_{age,h} + \beta_{seats,SM} \delta_{airlineSeats} \delta_{SM} \quad (8)$$

Where,

- V_j is the systematic utility of alternative j.
- $Travel\ time_j$, $Headway_j$ and $Travel\ cost_j$ represent the alternative-specific attributes.
- $\delta_{income,g}$ and $\delta_{age,h}$ are indicators for income and age categories, respectively.
- $\delta_{airlineSeats}$ indicates the presence of airline seats, specific to the Swissmetro alternative (δ_{SM}).
- β terms are alternative-specific parameters.

This model uses non-linear transformations (logarithmic) for travel time, cost, and headway, as well as interactions between travel cost and income categories. It also accounts for alternative-specific effects, such as airline seats for Swissmetro. Parameter values, estimated using the original dataset with a multinomial logit model, were applied to compute mode choice probabilities. Synthetic mode choices were then drawn randomly based on these probabilities, preserving the original dataset's independent variable values. The synthetic dataset thus represents choices generated from a known "true" model, enabling validation of the proposed framework.

Consistency with behavioral expectations was imposed by constraining the sensitivities of utilities to changes in travel times and costs to be non-positive:

$$\frac{\partial V_j}{\partial Travel\ time_j} \leq 0, \quad \forall j \tag{9}$$

$$\frac{\partial V_j}{\partial Travel\ cost_j} \leq 0, \quad \forall j \tag{10}$$

Models that violate any of these constraints are assigned the worst possible performance measure values, so that they are not considered further in the evolutionary generation process.

Sensitivity analyses focused on parameter penalties (α), which balance model fit and simplicity. Penalty values ranged from 0 (no penalty, best fit) to 10 (maximum penalty, most parsimonious). Using $\alpha = 1$ aligns with AIC, while $\alpha \approx 3.7$ corresponds to BIC. The synthetic data, generated from a "true" model with generic utility functions and alternative-specific parameters, was used to develop grammar-based models following this structure.

The grammar-based models were evaluated on their fit to testing data using metrics like average log-likelihood (ALL), accuracy, and root mean square error (RMSE) of predicted market shares. For the synthetic data, model similarity to the "true" model was assessed using two metrics:

Brier Score (BS): This measures the difference between predicted and true choice probabilities, with lower scores indicating higher accuracy.

$$BS = \frac{1}{N}\sum_{n=1}^{N}\sum_{j=1}^{J}\left(\tilde{P}(Y_{nj}) - \hat{P}(Y_{nj})\right)^2 \tag{11}$$

where, $\tilde{P}(Y_{nj})$ are the choice probability calculated by the "true" model.

Specification Similarity Indices (SSI): Based on the Jaccard index, these evaluate overlap between variables in the estimated and "true" models:

$$SSI_i = J(A_i, B_i) = \frac{|A_i \cap B_i|}{|A_i \cup B_i|} \tag{12}$$

where, A_i and B_i are the sets of variables that are present in the utility functions of the estimated model and the "true" model, respectively. The index i signifies the level of overlap between the variables in utility functions that is considered similar:

SSI_1: Measures variable overlap

SSI_2: Accounts for transformations (e.g., log(time) vs. time).

SSI_3: Considers interaction terms holistically (e.g., Income·cost²).

These indices range from 0 to 1, where 0 indicates no shared variables, and 1 indicates an identical set of variables.

4. Results

4.1. Synthetic data

Figure 2 shows the results with respect to the number of parameters in the final model and the goodness of fit measures: the average log-likelihood, accuracy, and RMSE as a function of the penalty parameter α. For each level of α, the figure shows the mean and 95% confidence intervals for each metric were calculated based on the best 20 models.

Figure 2(a) shows that as α increases the number of parameters estimated in the model decreases and stabilizes at about 7 parameters for $\alpha \geq 4$. The narrow confidence intervals indicate high consistency in the complexity of the estimated models. The average log-likelihood (ALL) is shown in Figure 2(b) for the training, validation, and testing sets. As expected ALL is best for the training set and worst for the testing set that was not used to develop the models. The mean ALL values decrease as α increases, particularly between up to $\alpha = 2$. With higher values of α, confidence intervals are wider, which suggest lower robustness of the model fit when models are overly parsimonious and may omit important variables. Figure 2(c) that shows the prediction accuracy exhibits a similar trend. Accuracy decreases as α increases. The confidence intervals for of the training set are wider compared validation and testing, suggesting that models with similar accuracy on training may have greater differences on unseen data. The RMSE of market shares are shown in Figure 2(d). RMSE is exactly zero for the training set, which is a mathematical property of the multinomial logit model (Ben-Akiva and Lerman, 1985). For the validation and testing sets, RMSE increases with α values up to $\alpha = 3$, after which it stabilizes. The confidence intervals wider for the testing set, especially with larger α values. This again indicates that the prediction performance of models that are more constrained by the penalty term tends to be less reliable.

Figure 3 presents the evaluation of the similarity of the estimated models to the "true" model that was used to generate the data as a function of α. Figure 3(a) shows the structural similarity indices: SSI_1 , SSI_2, and SSI_3. As expected from their definitions the values of SSI_1 are highest and of SSI_3 are the lowest. The values of all three similarity indices decrease with an increase in α values and stabilize for larger values. With the larger penalties, the models become more selective, prioritizing less developed utility functions and so tend to miss some of the lesser variables in the utility functions and exhibit lower similarity to the "true" model.

Figure 3 (b) presents the Brier Scores, which measures the difference in predicted choice probabilities between the estimated and "true" models, for the training, validation, and testing sets. BS increase with an increase in the value of α. The confidence intervals for these metrics are also wider for larger values of α, which suggests less robust performance on different samples.

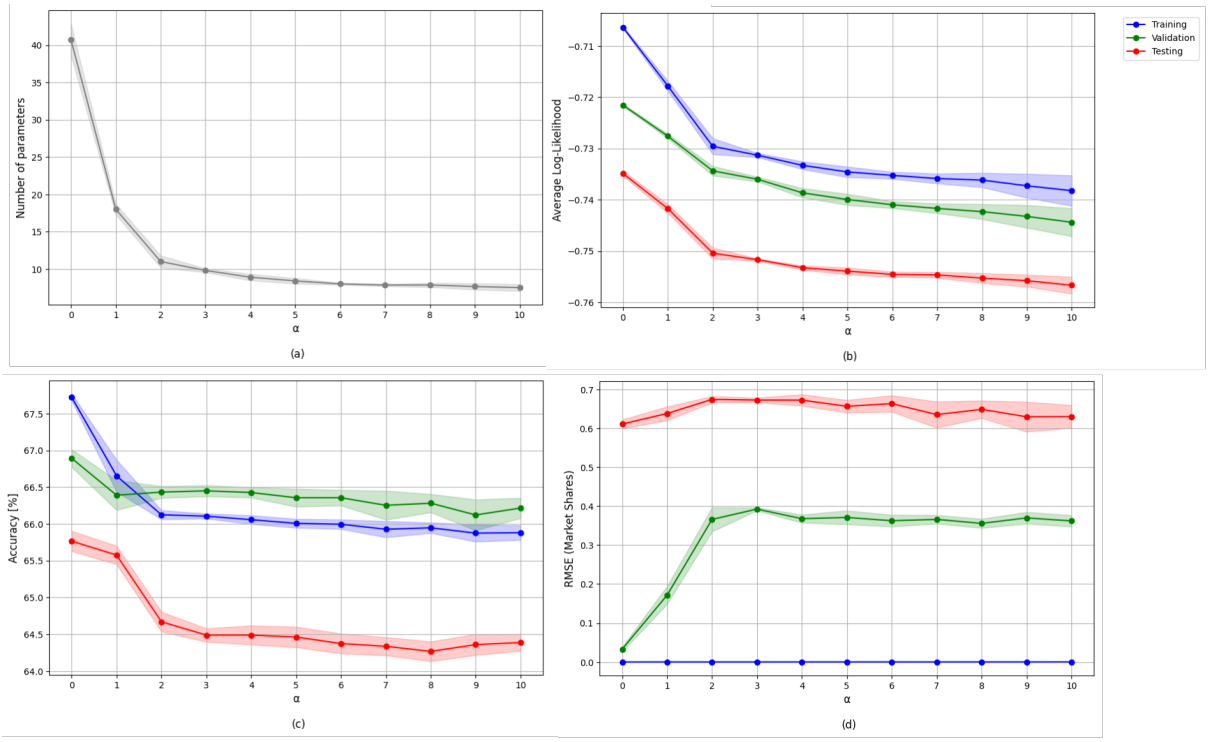


Figure 2. Goodness of fit measures for the synthetic data models

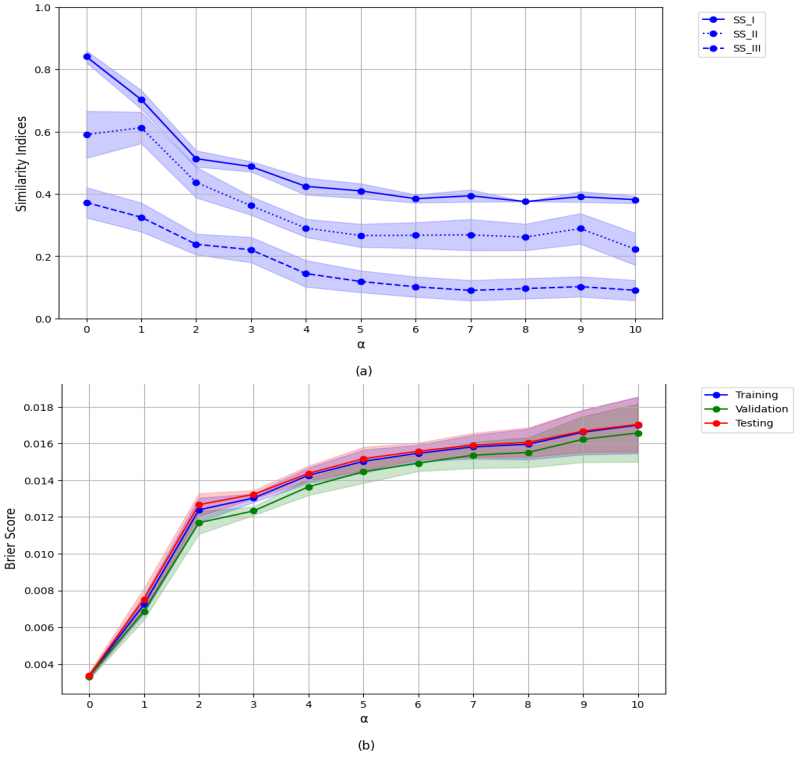


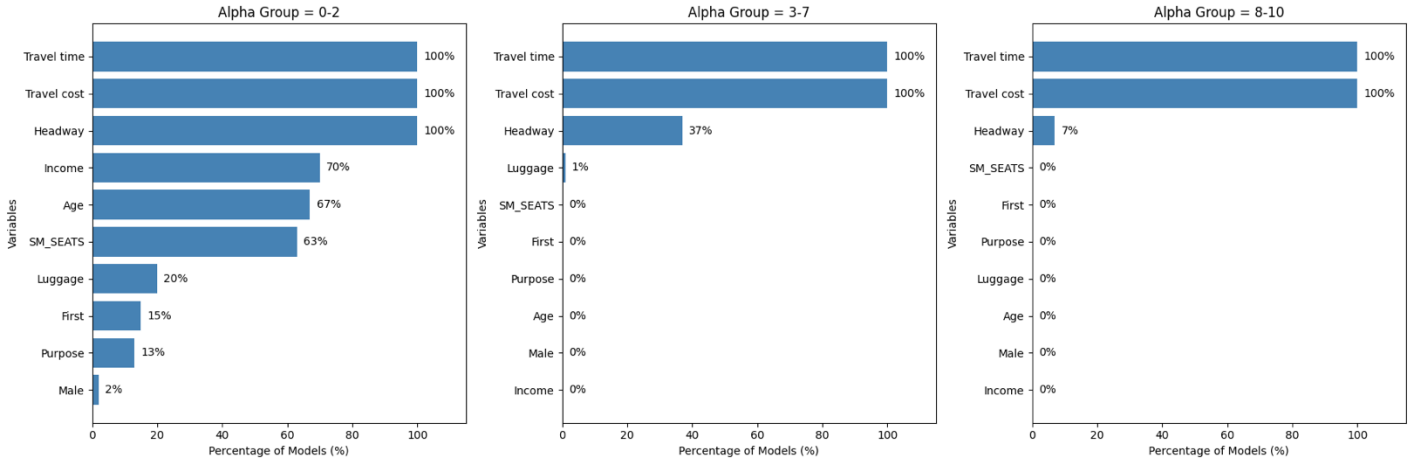
Figure 3. Similarity measures for the synthetic data models

The variable inclusion results show a clear pattern of prioritization and simplification as α increases. At low penalty levels ($\alpha = 0 - 2$), travel time, travel cost, and headway are universally retained (100%), aligning well with the "true" model. Income (70%), age (67%), and SM_SEATS (63%) are included moderately, while variables like luggage (20%) and first-class seating (15%) appear less frequently, indicating secondary importance.

At moderate penalty levels ($\alpha = 3 - 7$), only travel time and travel cost remain universally included (100%). Headway drops sharply to 37%, diverging from the "true" model, where it is a key variable. Socio-demographic variables, such as income and age, and all contextual variables are excluded, reflecting a shift toward parsimony.

At high penalty levels ($\alpha = 8 - 10$), simplicity dominates, with only travel time and travel cost retained (100%). Headway is included in just 7% of models, and all other variables are excluded, highlighting a stark contrast to the "true" model's specification. This suggests that high penalties risk oversimplifying the model and omitting critical features.

Overall, the results demonstrate the framework's ability to retain key variables under varying penalties. However, the exclusion of headway at higher α values highlights the trade-off between simplicity and fidelity to the "true" model, underscoring the importance of carefully tuning α.



5. Conclusions and discussion

This study introduces a semi-automated framework for specifying utility functions in discrete choice models (DCMs) by combining the interpretability of analytical forms with data-driven flexibility. Using a grammar-based approach within a Grammatical Evolution (GE) framework, the method enables systematic selection of variables, transformations, and interactions, ensuring theoretical soundness and interpretability.

The framework was validated using a synthetic dataset generated from a predefined "true" model. Results demonstrated its ability to recover the "true" utility structure with high similarity indices and comparable performance. Parsimony penalties, which reduce model

complexity, were shown to prioritize critical variables like travel times and costs while maintaining acceptable performance. This balance between simplicity and fit allows the framework to align closely with criteria like AIC and BIC.

By formalizing the specification process through grammar rules, this approach offers a systematic alternative to traditional trial-and-error methods, reducing modeler bias and enhancing consistency. The study also highlights future directions, such as extending the framework to more advanced logit models and incorporating soft constraints to handle real-world complexities.

In conclusion, this grammar-based framework bridges the gap between interpretability and data-driven exploration in DCMs, offering a robust tool for streamlining model development in fields like transportation, marketing, and policy analysis.

References

Ben-Akiva, M. E., & Lerman, S. R. (1985). Discrete Choice Analysis: Theory and Application to Travel Demand. MIT Press.

Bierlaire, M., Axhausen, K., & Abay, G. (2001). The Acceptance of Modal Innovation: The Case of Swissmetro. Proceedings of the Swiss Transport Research Conference (STRC), Ascona, Switzerland.

Haj-Yahia, S., Mansour, O., & Toledo, T. (2023). Incorporating Domain Knowledge in Deep Neural Networks for Discrete Choice Models. *arXiv preprint arXiv:2306.00016*.

Han, Y., Zegras, C., Pereira, F. C., & Ben-Akiva, M. (2020). A Neural-embedded Choice Model: TasteNet-MNL Modeling Taste Heterogeneity with Flexibility and Interpretability. http://arxiv.org/abs/2002.00922

Kling, C. L. (1989). The importance of functional form in the estimation of welfare. Western Journal of Agricultural Economics, 168–174.

McFadden, D. (1974). Conditional Logit Analysis of Qualitative Choice Behavior. In Frontiers in Econometrics.

Ortelli, N., Hillel, T., Pereira, F. C., de Lapparent, M., & Bierlaire, M. (2021). Assisted specification of discrete choice models. Journal of Choice Modelling, 39, 100285.

Sifringer, B., Lurkin, V., & Alahi, A. (2020). Enhancing discrete choice models with representation learning. Transportation Research Part B: Methodological, 140, 236–261.

Torres, C., Hanley, N., & 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. https://doi.org/10.1016/j.jeem.2010.11.007