STRC | 18th Swiss Transport Research Conference
Monte Verità / Ascona, May 16 – 18, 2018

Enhancing Discrete Choice Models with Neural Networks

Brian Sifringer
Virginie Lurkin
Alexandre Alahi

Ecole Polytechnique Fédérale de Lausanne **May 2018**

STRC | 18th Swiss Transport Research Conference
Monte Verità / Ascona, May 16 – 18, 2018

Ecole Polytechnique Fédérale de Lausanne

Enhancing Discrete Choice Models with Neural Networks

Brian Sifringer, Virginie Lurkin, Alexandre Alahi
Visual Intelligence for Transportation
Ecole Polytechnique Federale de Lausanne
Route Cantonale, 1015 Lausanne
phone: +41 21 69 32608
fax: N.A
{brian.sifringer,virginie.lurkin,alexandre.alahi}@epfl.ch

May 2018

Abstract

In this paper, we aim at bringing the predictive strength of Neural Networks, a powerful machine learning-based technique, to the field of Discrete Choice Models (DCM) without compromising interpretability of these choice models. We start by matching the mathematical derivation of the multinomial logit model (MNL) to its neural network equivalent. This allows us to write DCM problems in modern machine learning libraries and opens the way for our novel hybrid approach: we suggest to add a term arising from a dense neural network (DNN) in the utility function. This added value is obtained by using all discarded features from the original DCM model as input to the DNN. Not only does this greatly increases the predictive strength of the model, but it also keeps the strong parameters significance used in the original MNL. Lastly, we have reasons to believe this term fits very well in DCM theory when relating it to the random utility term ϵ, capturing all unknown or unused features of the model which may appear in the thinking process of an individual.

Keywords

Discrete choice modeling, neural networks, multinomial logit, convolution, hybrid, enhancing

1 Introduction

Deep learning has been revisiting many fields for the past few years such as signal processing, computer vision, finance and many more (LeCun *et al.,* 2015). Its ability to learn a non-linear mapping function from observed data to a desired output is second to none. However, in many fields, it comes with the drawback of being a black-box. When studying demand in travel applications, health care programs or market produce for example, it is of utmost importance we understand what are the key parameters in the decision-making process of the clients. This is why researchers have been using Discrete Choice Modeling (DCM), as they are specifically designed to capture in detail the underlying behavioral mechanisms at the foundation of this decision-making process (Ben-Akiva and Lerman, 1985).

Recently, researchers have started to study how to bridge the gap between Discrete Choice Modeling (DCM) and Machine Learning (ML) frameworks (Acuna-Agost *et al.,* 2017, Hagenauer and Helbich, 2017, Iranitalab and Khattak, 2017, Paredes *et al.,* 2017, Brathwaite *et al.,* 2017). There are recent attempts to combine them (Otsuka and Osogami, 2016, Yang *et al.,* 2017). However, DCM remains the most commonly used method due to the interpretability of its parameters. Hence, in this paper, we propose to enhance discrete choice modeling, using a neural network while keeping interpretability of the results. The method consists in adding an extra term in the utility function of a logit model estimated by a dense neural network (DNN) during the minimization of the negative log likelihood. The input to the DNN must be complimentary to that of the utility function. The goal is to keep the key parameters of interest in the DCM framework for behavioral interpretation and to use the remaining ones in order to improve predictability.
To evaluate our method, we use the openly available data, Swissmetro, and a Multinomial Logit (MNL) model described by Bierlaire *et al.* (2001). We then compare it with our new method and show a 15% increase in the final log-likelihood while keeping significance and interpretability of the important MNL parameters. Finally, we suggest an intuitive explanation on how this new term may be integrated within the DCM framework.

2 Method

In this section we present our new approach using a multinomial logit model. However the methodology is general and can be applied to more advanced logit models.

2.1 Multinomial Logit as a Neural Network

In discrete choice modeling, a commonly used model is the multinomial logit (McFadden *et al.,* 1973). Given an individual n and a set of d variables $\mathcal{X}_n = \{\mathbf{x}_{1n}, ..., \mathbf{x}_{dn}\}$, we define a choice set C_n of $\mathcal{I}$ alternatives and matching utility functions:

$$U_{in} = \beta_1 \cdot x_{1in} + ... + \beta_d \cdot x_{din} + \varepsilon_{in} \qquad \forall i \in C_n \tag{1}$$

$$= V_{in} + \varepsilon_{in} \tag{2}$$

where $\varepsilon_{1n}, ..., \varepsilon_{pn}$ are i.i.d Extreme Value distributed and $\beta_1, .., \beta_d$ is a set of parameters to be estimated by minimizing the negative log-likelihood:

$$\mathcal{L} = -\sum_{n=1}^{N} \sum_{i \in C_n} y_{in} \log\left[P(i|C_n)\right] \tag{3}$$

with y_{in} equal to 1 if individual n chooses i and 0 otherwise. The probability of choosing $i \in C_n$ for multinomial logit is defined as:

$$P(i|C_n) = P(U_{in} > \max_j(U_{jn})) = \frac{\exp^{V_{in}}}{\sum\limits_{j \in C_n} \exp^{V_{jn}}} \tag{4}$$

This mathematical model has deep theoretical foundations (Ben-Akiva and Lerman, 1985) making extensive use of ε to define statistical properties. However, when minimizing equation (3), we can relate this act to training an artificial neural network in machine learning (LeCun *et al.,* 2015).

Indeed, if we define $\boldsymbol{\beta} = \{\beta_1, ...\beta_d\}$ as a kernel of size $(1 \times d)$ and a single set of variables $\mathcal{X}_n$ as an image of size $(I \times d)$, we get the observables of the utility functions $\mathbf{V}_n = \{V_{1n}, ...V_{In}\}$ by doing a convolution[1] between the kernel and the image, as seen on the right side of figure (1). The probabilities can be obtained by using a softmax activation function (Bishop, 1995) defined as:

$$(\boldsymbol{\sigma}(\mathbf{V}_n))_i = \frac{\exp^{V_{in}}}{\sum\limits_{j \in C_n} \exp^{V_{jn}}} \tag{5}$$

which can be identified as equation (4) for all probabilities. For the loss function, we use categorical cross-entropy Shannon (1948) written:

$$H_n(\boldsymbol{\sigma}, \mathbf{y}_n) = -\sum_{i \in C_n} y_{in} \log\left[\sigma(\mathbf{V}_n))_i\right] \tag{6}$$

[1] may be defined as a correlation depending on the kernel orientation or coding library used

which is the same as equation (3) when summed over all individuals. This method allows us to use conventional deep learning libraries to implement the multinomial logit model, giving us very high flexibility and efficiency in modifying the structure and in learning the parameters.

2.2 Enhanced Utility Functions

In this section, we take advantage of the neural network approach to add a value, u_{in}, to the corresponding utility function U_{in} for all $i \in C_n$ and $\mathbf{u}_n = \{u_{1n}, ..., u_{pn}\}$ such that:

$$\mathbf{u}_n = \psi(Q) \tag{7}$$

where Q is the ensemble of inputed features and $\psi : \mathcal{R}^{I \times d} \mapsto \mathcal{R}^I$ is the function defined by multiple neural network dense layers and their corresponding activation functions, such that the utility functions can be written as:

$$\mathbf{U}_n = \boldsymbol{\beta} \mathcal{X}^T + \mathbf{u}_n + \boldsymbol{\varepsilon}_n \tag{8}$$

2.2.1 Same Input

If we define $Q = \mathcal{X}$, where $\mathcal{X}$ is the inputed features of MNL, $\mathbf{u}_n$ can be interpreted as the best hyperparameter for each alternative $i \in C$ which maximizes the model's likelihood. Unfortunately, in this case, the neural network layers also overrun the simple linearity of MNL parameters in the utility functions making all betas insignificant. To avoid this problem, we need to select features for the neural network which aren't the same, or highly correlated, with the original MNL input.

2.2.2 Extra Input

To avoid that the new term overruns the DCM parameters, we define $Q = \mathcal{U}$, where $\mathcal{U}$ is all the unused features in $\mathcal{X}$ and contain distinct information. Such a setting may greatly increase the likelihood while keeping the original parameters highly significant.
The final model, combining both the convolution approach of writing MNL and the added neural network term from unused DCM features can be seen in figure (1).

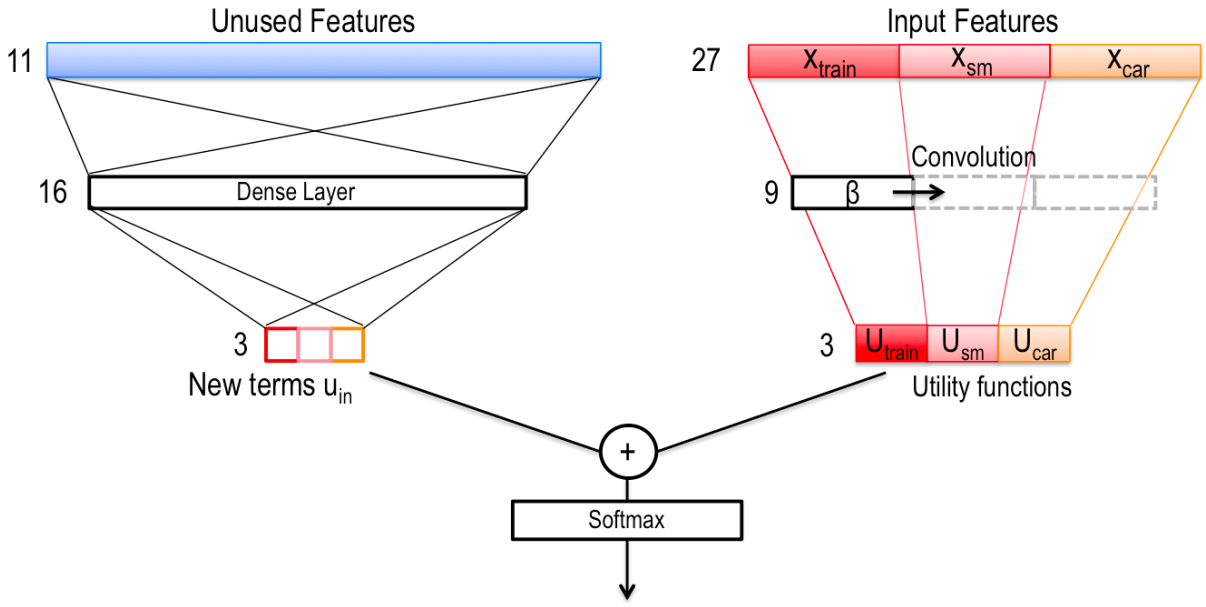

Figure 1: *With DCM written in modern machine learning libraries, one can flexibly change the model, optimizers in training and more. On the right-hand side, the weights of the kernel correspond to the β_j parameters, and applying a convolution layer to the input features gives us the same utility functions as in MNL. The left hand side is the DNN component, producing a single term for each utility function and highly increasing predictive accuracy.*

2.3 Dataset and Modelling

To present our new method, we follow the multinomial logit model from Bierlaire *et al.* (2001) on the openly available Swissmetro dataset. It is based on a stated preference survey on transport modes, gathering 10'700 entries from 1'190 different participants. Each individual informed of his choice in transportation for various trips including the car, the train or an innovative project: the Swissmetro. The Swissmetro is the name given to an attempt to build a very fast underground transport mode to connect the biggest cities in Switzerland.
Unfortunately, the original dataset used in Bierlaire *et al.* (2001) is not the same as the one currently available, which will give some differences in the parameters found in the benchmark MNL model. The utility functions are defined in table (1) as done previously by Bierlaire et al. using the same variable descriptions.

Moreover, as seen in 2.2, we will take all unused features from our survey as input for the neural network component of the enhanced method. There are a total of 8 extra features which are as follows:

Table 1: Utility functions

Variable		Alternative		
		Car	Train	Swissmetro
ASC	Constant	Car-Const		SM-Const
TT	Travel Time	B-Time	B-Time	B-Time
Cost	Travel Cost	B-Cost	B-Cost	B-Cost
Freq	Frequency		B-Freq	B-Freq
GA	Annual Pass		B-GA	B-GA
Age	Age in classes		B-Age	
Luggage	Pieces of luggage	B-Luggage		
Seats	Airline seating			B-Seats

- Travel purpose: Discrete value between 1 to 9 (Business, leisure, travel,...)
- First class: 0 for no or 1 for yes if passenger is a first class traveler in public transport
- Ticket: Discrete value between 0 to 10 for the ticket type (One-way, half-day, ...)
- Who: Discrete value between 0 to 3 for who pays the travel (self, employer, ...)
- Male: Traveler's gender, 0 for female and 1 for male
- Income: Discrete value between 0 to 4 concerning the traveler's income per year
- Origin: Discrete value defining the canton in which the travel begins
- Dest: Discrete value defining the canton in which the travel ends

3 Results

3.1 Multinomial Logit as Benchmark

We start by running our model on simple Multinomial Logit as a benchmark. As we can see in table (2), the original model defined by Bierlaire *et al.* (2001) no longer holds the same values when applied to the new dataset, but the utility functions are still well defined with all significant parameters.

Table 2: MNL parameter values

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	t-stat	p-value
1	ASC_{Car}	1.20	0.183	6.58	0.00
2	ASC_{SM}	1.19	0.182	6.53	0.00
3	β_{age}	0.175	0.0512	3.41	0.00
4	β_{cost}	-0.00690	0.000577	-11.97	0.00
5	β_{freq}	-0.00704	0.00116	-6.09	0.00
6	β_{GA}	1.54	0.168	9.17	0.00
7	$\beta_{luggage}$	-0.113	0.0479	-2.36	0.02
8	β_{seats}	0.432	0.115	3.76	0.00
9	β_{time}	-0.0129	0.000842	-15.34	0.00

Number of observations = 7234

$\mathcal{L}(\hat{\beta}) = -5766.705$

3.2 Improvements with enhanced method

A common difficulty in modeling utility functions is adding statistically significant features. However, when successful, this translates into the strength of DCM, giving insight on the importance of chosen parameters in the decision making process. In the following results, we release some of this informative power by allowing the machine to find the best possible hyperparameters for each single utility function.

To first maximize the model's likelihood, we implement the multinomial logit model in a deep learning library (Abadi *et al.,* 2015*,* Chollet *et al.,* 2015) as seen in section (2.1) by adding the term defined in equation (7). Statistical properties of the parameters are obtained thanks to Biogeme (Bierlaire*,* 2009). Since the new term is a MNL feature, we redefine equation (8) as $\mathbf{U}_n = \tilde{\boldsymbol{\beta}}\tilde{X}^T + \boldsymbol{\varepsilon}_n$ where $\mathbf{u}_n$ is added to X as a variable and has now its own parameter β_{NN} added to $\boldsymbol{\beta}$. The results of these steps can be found in table (3).

As we can see, the log-likelihood ratio test is very high, reaching up to 1515.6 with only one added parameter. However, this parameter holds all unused features, which is a total of 8, and goes through a neural net with about 250 trained variables. So one could argue to use a X^2 value with much higher degrees of freedom to accept the log-likelihood ratio test as significant.

Table 3: Hybrid Model parameter values

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	t-stat	p-value
1	ASC_{Car}	0.0652	0.179	0.37	0.71
2	ASC_{SM}.	0.327	0.171	1.92	0.06
3	β_{age}	0.376	0.0464	8.12	0.00
4	β_{cost}	-0.0141	0.000595	-23.63	0.00
5	β_{freq}	-0.00807	0.00123	-6.55	0.00
6	β_{GA}	0.130	0.181	0.72	0.47
7	$\beta_{luggage}$	0.0153	0.0505	0.30	0.76
8	β_{seats}	0.207	0.106	1.95	0.05
9	β_{time}	-0.0157	0.000952	-16.53	0.00
10	β_{NN}	1.24	0.0524	23.74	0.00

Number of observations = 7234

$\mathcal{L}(\hat{\beta}) = -5008.996$

Fortunately, even with a thousand degrees of freedom, our value is above $\chi^{2,0.001}_{1000} = 1143.9$ and the model can be accepted with confidence higher than 99.999%

Concerning the parameters, we see that some have lost their significance. This problem may arise when the neural network component has learned highly correlated information to these linear parameters. The non linear strength of this new term can overrun parts of the original MNL description. For example, the amount of luggage taken may be correlated to the travel purpose such as going on holidays. Or having an annual pass may be obvious in many conditions, when combining the origin, destination and purpose of travel. In the following, we show how we can select which parameters are most important for interpretation and yield even better results with the neural network enhancing method.

3.3 Model Redefinition

In Bierlaire *et al.* (2001), two important values which are used to compare multiple models are Value of Time (VOT) and Value of Frequency (VOF). In many DCM problems, we aim to find the most accurate model possible, with many significant parameters, such that the post-estimation indicators make sense. Indeed, a model which is too simplified will perform poorly not only for

prediction but also when trying to understand the human decision process and forecasting how it will change when the settings are different, such as a rise in market price or increase in transport times. Therefore, keeping an eye on important values across different methods allows us too validate or not the chosen features.

If we consider our MNL model so far, the most interesting parameters in forecasting are most likely cost, time and frequency. All other parameters are mostly here to get good values on the important features as explained above. However, with the hybrid model, we can simply let the dense neural network select which parameters and non-linear combinations are interesting with unused features and allow us to concentrate on the inputs we want to interpret. This is what we have done in table (4) where only a few desired features are kept, and all others are sent to the DNN component of our model as extra features. As we can see, the values found are closer to our previous models, closer than if we hadn't used the hybrid model as seen in table (5), which is what we mentioned above as being oversimplified.

Table 4: Hybrid model containing only values of greater interest

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	t-stat	p-value
1	ASC_{Car}	0.966	0.0977	9.89	0.00
2	ASC_{SM}	1.13	0.0941	11.97	0.00
3	β_{cost}	-0.0165	0.000666	-24.71	0.00
4	β_{freq}	-0.00820	0.00129	-6.38	0.00
5	β_{time}	-0.0171	0.000853	-20.05	0.00
6	β_{NN}	1.25	0.0854	14.65	0.00

Number of observations = 7234

$\mathcal{L}(\hat{\beta}) = -4894.539$

3.4 Results summary

In table (6), we compare the VOT and VOF between our hybrid models and the MNL benchmarks. As we can see, both hybrid methods give values close to each other, and as for the small MNL, it is closer to the benchmark. However, as mentioned above, utility functions with low dimensionality may not give the best minimum for its parameters.

Table 5: MNL containing only values of greater interest

Parameter number	Description	Coeff. estimate	Robust Asympt. std. error	t-stat	p-value
1	ASC_{Car}	0.533	0.0883	6.04	0.00
2	ASC_{SM}	0.753	0.0889	8.47	0.00
3	β_{cost}	-0.00840	0.000596	-14.09	0.00
4	β_{freq}	-0.00704	0.00116	-6.09	0.00
5	β_{time}	-0.0124	0.000827	-14.95	0.00

Number of observations = 7234

$\mathcal{L}(\hat{\beta}) = -5862.549$

When it comes to interpretation, we stay in the same order of magnitude and the same sign in every situation.

Table 6: Parameter ratio comparison

Parameter	MNL	Hybrid	Simple Hybrid	Simple MNL
β_{cost}	100.0%	204.3%	239.1%	121.7%
β_{freq}	100.0%	114.6%	116.5%	100.0%
β_{time}	100.0%	121.7%	132.5%	96.1%
Value of Time	0.54	0.89	0.96	0.68
Value of Frequency	0.98	1.75	2.01	1.19
Final Log-Likelihood	-5766.71	-5009.00	-4894.54	-5862.55
Number or parameters	9	10	6	5

4 Discussion

4.1 Analysis

As we have seen, enhancing a MNL model with a neural network needs a strategic choice in parameters. Indeed, the predictive strength of the dense layers component may easily overrun the original linear parameters due to redundant information read through the data. As such, it

is important to keep the features of greatest interest for the MNL component, and then give an independent set of unused features to the NN part. By doing so, we were able to increase the final log-likelihood by at least 15% compared to the benchmark. Moreover, the t-statistics of the betas are still significant.

It is important to note, that the dataset if fairly optimized for DCM as we have few and discrete features. We believe this method may perform much better with bigger datasets, since neural networks are, by nature, data-driven approaches.

4.2 Intuitive Interpretation

When we make use of modern supervised machine learning, we are letting a deep network finding the best mapping it can, given a set of features and labels. If we have a good architecture, it will excel at predicting the correct answer of new data. However, getting to know what exactly the network learned is a field of research in itself.
In our case, by using all unused features, we are providing as much information as we can to the neural network. In this sense, the new term u_{in} in our utility function can be seen as an estimation of all uncaptured information in the decision making process of an individual. This description resembles closely to the random utility term ε_{in} which is the randomness of our model, arising from the fact that we cannot take into account all the information which goes into the decision making of each individual. So in a way, $u_{in} = \bar{\varepsilon}_{in}$, where $\bar{\varepsilon}_{in}$ is an estimation of ε_{in} given the data at hand. Since a survey may never acquire all the necessary information every single individual will use for making a decision, and since deep neural networks are powerful with manipulating data, but far from flawless, we may define the random utility term ε^*_{in} such that:

$$\varepsilon_{in} = \bar{\varepsilon}_{in} + \varepsilon^*_{in} \tag{9}$$

$$\Leftrightarrow \quad U_{in} = V_{in} + \bar{\varepsilon}_{in} + \varepsilon^*_{in} \tag{10}$$

where ε^*_{in} captures the randomness in the decision making, unforeseen by the gathered data itself as well as the imperfect prediction of the neural network.
With this interpretation, we defend the idea that the new beta parameters find a better value as they don't compensate for lacking information.

5 Conclusion

We have suggested a new method combining both statistics and intuition from the field of discrete choice modeling with the predictive strength of modern machine learning. This is achieved by adding a single term, originating from a fully connected neural network, in each utility function of a MNL model. The input of the network component must be unused features and independent from the DCM framework. This method allows for a great increase in the final log-likelihood while keeping parameters interpretable. We suggest that they actually converge to a better value, which cannot be obtained through simple models. Moreover, we open the way for an interpretation of this added term in the utility function, which is an estimator for uncaptured decision-making information. Future research would be to verify the mathematical foundations of this proposed theory. This method would also be much more efficient when applied to a larger dataset. This hybrid model could bring intuition and statistics to neural network problems or higher efficiency to cases solved with DCM framework.

6 References

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng (2015) TensorFlow: Large-scale machine learning on heterogeneous systems, `https://www.tensorflow.org/`. Software available from tensorflow.org.

Acuna-Agost, R., T. Delahaye, A. Lheritier and M. Bocamazo (2017) Airline itinerary choice modelling using machine learning, paper presented at the *International Choice Modelling Conference 2017*.

Ben-Akiva, M. and S. Lerman (1985) *Discrete Choice Analysis: Theory and Application to Travel Demand*, MIT Press series in transportation studies, MIT Press, ISBN 9780262022170.

Bierlaire, M. (2009) Estimation of discrete choice models with biogeme 1.8.

Bierlaire, M., K. Axhausen and G. Abay (2001) The acceptance of modal innovation: The case of swissmetro, (TRANSP-OR-CONF-2006-055).

Bishop, C. M. (1995) *Neural networks for pattern recognition*, Oxford university press.

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

Chollet, F. *et al.* (2015) Keras, `https://github.com/keras-team/keras`.

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

Iranitalab, A. and A. Khattak (2017) Comparison of four statistical and machine learning methods for crash severity prediction, *Accident Analysis & Prevention*, **108**, 27–36.

LeCun, Y., Y. Bengio and G. Hinton (2015) Deep learning, *nature*, **521** (7553) 436.

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

Otsuka, M. and T. Osogami (2016) A deep choice model., paper presented at the *AAAI*, 850–856.

Paredes, M., E. Hemberg, U.-M. O'Reilly and C. Zegras (2017) Machine learning or discrete choice models for car ownership demand estimation and prediction?, paper presented at the

Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on, 780–785.

Shannon, C. E. (1948) A mathematical theory of communication, part i, part ii, *Bell Syst. Tech. J.*, **27**, 623–656.

Yang, J., S. Shebalov and D. Klabjan (2017) Semi-supervised learning for discrete choice models, *arXiv preprint arXiv:1702.05137*.