Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 170 (2020) 115–122

Procedia Computer Science

www.elsevier.com/locate/procedia

The 11th International Conference on Ambient Systems, Networks and Technologies (ANT)
April 6 - 9, 2020, Warsaw, Poland

Using Neural Nets to Predict Transportation Mode Choice: An Amsterdam Case Study

Ruurd Buijs[a], Thomas Koch[b,*], Elenna Dugundji[a,b]

[a] Vrije Universiteit Amsterdam, De Boelelaan 1111, Amsterdam 1081 HV, The Netherlands
[b] Centrum Wiskunde en Informatica, Science Park 123, Amsterdam 1098 XG, The Netherlands

Abstract

In the Amsterdam metropolitan area, the opening of a new metro line along the north south axis of the city has introduced a significant change in the region's public transportation network. Mode choice analysis can help in assessment of changes in traveler behavior that occurred after the opening of the new metro line. As it is known that artificial neural nets excel at complex classification problems, this paper aims to investigate an approach where the traveler's transportation mode is predicted from a choice set through a neural net. Although the approach shows promising results, it has been found that its performance can be attributed partly to the presence of differences in data patterns between the actual and generated trips, which the neural net is able to detect. By adding generated user characteristic attributes, the performance of the model can be boosted slightly overall, and significantly concerning prediction of whether or not a trip was made by car.

Keywords: Transportation mode choice; artificial neural nets; machine learning; public transportation network change; travel behaviour

In 2018, the region of Amsterdam witnessed the most comprehensive structural change in their public transportation network in more than a century. The opening of a new metro line serving the entire length of the north-south axis of the city has led to rigorous changes in the existing tram and bus network.

Analyzing the behaviour of transport movements of individuals is an effective way to assess the impact of a rigorous network change. A standard approach carried out to map transportation behavior is discrete choice mode analysis. This method uses statistical techniques for parameter estimation.

This study explores a relatively new method that can contribute to behavioral analysis of transport movements, based on a data set collected in Amsterdam. Considering the fact that Artificial Neural Networks (ANNs) are extremely capable of performing well when assigned complex classification tasks, this study looks into possible application of this technique within the field of behavioral analysis in transportation.

[*] Corresponding author. Tel.: +31-20-592-4132
E-mail address: koch@cwi.nl

10.1016/j.procs.2020.03.015

The paper is structured as follows: Firstly, a brief literature review1 is presented, followed by a description of the Amsterdam case study. Next, the data set used and methodology to process it will be discussed. After that, suggestions are made for a neural net implementation to classify mode choice. Finally, results are presented and discussed and recommendations for future research are made.

1. Literature review

For several decades, discrete choice modeling has been dominated statistical models, such as the logit and probit models. This paradigm dates as far back as the 1970s (Coslett [7]), and is an approach that is built upon in recent publications (e.g. Angelo Guevara and Ben-Akiva [4]).

In 2003 Vythoulkas and Kotsopoulos [13] applied a different approach to this problem. Trying to beat the results obtained by conventional statistical methods, they introduced a neural net structure based on fuzzy set theory to model discrete choice behaviour in transportation. They then applied this algorithm to a small data set that obtained by the Dutch Railways. The data was related to transportation mode alternatives in the Dutch city of Nijmegen. The proposed model performed slightly better in the case study than a logit model constructed for the same purpose. One of the key underlying assumptions in this study was that travelers decide based on simple underlying rules rather then complicated functions $F : X \rightarrow Y$. Those rules were then incorporated into a neural net system.

Recently in 2019, Van Cranenburgh and Alwosheel [12] have been among a growing kernel of researchers to again use neural nets in practice in a similar context. In their paper, they describe how an ANN can be trained to investigate decision rule heterogeneity. Their method trains a multinomial classification network to assign users to one of four quintessential decision rules, based on theoretical choice data, where each user was presented a series of choices in order. The results of each user are then combined and fed to the network that classifies the user into one of the four categories.

Another interesting paper on extracting decision rules from data using a neural net has been presented by Hayashi et al. [9]. They make use of the *Re-RX* algorithm to extract rules from a pruned neural network. The dataset used by the authors contains user characteristics and preferences on eating behaviour, which is also an application of neural nets in a behavioral context.

In the context of Market Share forecasting, a study has been carried out by Agrawal and Schorling [3] in 1996, regarding a comparison between the ANN and multinomial logit method. In brand choice analysis, a hybrid model has been suggested by Bentz and Merunka in 2000 [5].

2. Case Study

In Amsterdam, a new metro line has been opened in 2018 serving the north-south axis of the Dutch capital. In order to improve integration of the new metro line, the existing transportation network underwent significant changes. A large number of the bus and tram lines were re-routed to connect different areas. One particular aim of these changes was to create more east-west links, that connect to the new north-south line at one of the metro stations in the centre of the city [1]. The design moved away from a network that was heavily focused on lines to and from the central train station to a network where instead the new North South metro line forms a spine.

For many inhabitants of the city, these changes in the network meant that their personal travel itineraries were affected. At the same time, car drivers were also confronted with the introduction of new restrictions in the inner city and around Amsterdam Central train station to avoid through-traffic in the inner city.

Policy makers from the regional transportation authority in Amsterdam and the city of Amsterdam are keen to assess the impact of the introduction of this new network. For this analysis, data was collected using a smart phone GPS application that was installed by a panel of participants recruited via several existing survey panels. Additional participants were recruited on the street. The smart phone application tracks the activities of the user in the background of the smart phone using sensors on the phone such as GPS and acceleration sensors.

3. Data

From the dataset with the GPS survey we took a specific sample limited to data concerning the 78 users that have used the North-South metro line at least once during the tracking period. The distribution of the number of trips registered over the users is shown in Figure 1. It has to be noted that 21 users have only one trip entry, meaning no predictions can be made for these users. Also, only 21 out of 78 users have more than ten trip entries. The small number of users that are suitable for prediction does not allow for a predictive model design that consists of each user being a single entry, due to the sparse number of data entries that would remain after a train/test split.

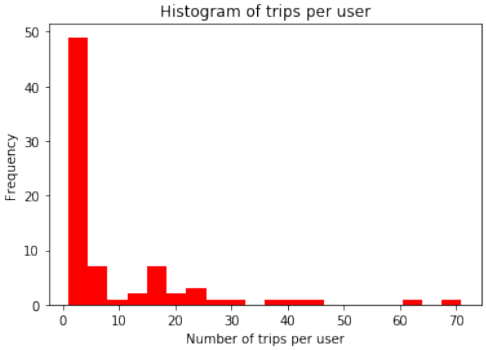

Figure 1. This histogram shows the distribution of number of trips registered over the users in the data sample.

3.1. *Choice set generation*

In order to explore what other transportation modes were available for each user for each of their observed choices, we generated a number of alternatives using an open source library developed by Conveyal [2], R5 - Rapid Realistic Routing on Real-world and Reimagined networks. This router has been used previously by other studies such as Conway et al. [6] and de Freitas et al. [8]. R5 is able to return a large set of feasible, fast routes within a given time-range. This permits a more realistic assessment about accessibility than would be possible using estimations based on fixed frequencies. We used two separate General Transit Feed Specification (GTFS) data files to feed the router with the correct timetable before and after the opening of the metroline. For the street network we used a temporally appropriate extract from OpenStreetMap. Additionally, we directed R5 to generate transit routes specifically including and excluding metro. For each observation and alternative we then categorized a route into one of 5 different non-overlapping strata: 1. a walk trip (generated if walking stays under 60 minutes) 2. a car trip (generated if destination is reachable by under 60 minutes) 3. a bicycle trip (generated if bicycling stays under 60 minutes) 4. a transit trip, without use of metro 5. a transit trip, with use of metro To generate choice sets, we looked at the observations and categorized each observation with a stratum and subsequently took the best (fastest) from the alternatives that fit each alternatives. In some cases not each alternative was available, for example walking is not always an option if the distance between origin and destination is long.

3.2. *Feature engineering*

From the observations and the generated alternatives we collected a number of explanatory variables such as modes used, the egress and access mode to transit, start- and end-times and distances/durations broken out per modality. We used a walking speed of approximately 5 km/h and a bicycling speed of 14.4 km/h. We based our car speed on the speed limits in OpenStreetMap.

4. Methodology

In order to cope with the problems related to the small number of different users in the data set, a three step method is suggested that allows for a larger number of data entries. A single data entry that will be fed to the neural net is in this method not a collection of choice data regarding a single user, but rather a frame containing information about a single choice, along with characteristic user-specific information. This method deals with the problem of having too few entries to build a reliable model.

4.1. *Data Preparation*

4.1.1. *Combining data*

The first step is to combine the data from different sources. As described before, for each individual A to B trip in the transportation data, a single routing possibility has been generated for each mode of transport (i.e. if a realistic routing using the respective mode of transport exists), and stored in a different data file. In order to integrate the two data sets, both files are merged. Initially, duplicates exist in this merged data set, i.e. for a single trip, there can be two routes with the same transportation mode: one that corresponds to the trip that was made originally by the user, the other one is the generated trip having the same transportation mode. In order to have unbiased data that can be used in a machine learning model, one of the two entries must be deleted, so that for each trip only one option per transportation mode remains. Since the data that most corresponds to reality is contained in the data set consisting of trips that were actually made by the users, it is opted for to preserve this data and discard the generated duplicates.

4.1.2. *Splitting data*

We used the common practice within the field of machine learning is to split the data before it is being used. We divided the data-set into a training set with 50% of the data, a validation set of 20% and a test set with the remaining 30% of the data. In order for the model to be able to take into account individual user preference characteristics, it is important that data from all users is contained in the train set. Because most users have very few data entries, the final distribution of entries over the set slightly deviates from this proposed distribution, whereas the eventual distribution of the 705 entries is as follows: the training set: 372 entries (including the 21 single-entry users); 52.8%, the validation set: 121 entries; 17.1% and the test set: 212 entries; 30.1% The training set is the only set for which the target variable (in this case Strata) is not hidden. Hence, all operations used for setting up the model that are described in the following sections, apply to the training set only.

4.1.3. *Classifying choices*

It is clear that individual users have different preferences. Those individual preferences should be taken into account in any predictive model, as becomes clear in the literature. As it has been concluded that this is not possible by considering each user as a single data entry, the some user-characteristic data must be fed to the model together with each choice data entry. The literature suggests different methods in order to classify users as decision-makers, yet all are based on assumptions. The most important notion that comes clear from this is that different individuals decide differently, and can be divided into classes or groups that share similar decision characteristics. Regardless of what the underlying decision functions might be (it will be nearly impossible to approximate them due to the few entries available per user), it is possible to divide the choices made into different classes based on comparative measures regarding the alternative modes. The comparative measures have been computed by normalizing the attributes *transfers, duration, bicycle_distance, bicycle_duration, walk_distance, walk_duration, car_distance, car_duration* and *waiting_time* for each choice set individually and extracting the values corresponding to the chosen mode. In this way, for each scenario each attribute can obtain a value between 0 and 1, where 0 is obtained if the alternative with the lowest value of an attribute is chosen and 1 is obtained if the alternative with the highest value for this attribute is chosen.

4.1.4. *k-means Clustering*

In order to form described classes without having a clear target to aim for, a method called k-means clustering is used (Steinley [11]). With this particular data set, there are two main challenges in terms of clustering: The most obvious underlying structure is the strata classification itself, which tells something about the choice, yet is not the particular information structure we are looking for. And secondly Some of the variables are clearly correlated, for example *bicycle_distance* and *bicycle_duration*. This might lead to a bias when classifying the choices. In order to overcome these obstacles, the following solutions have been suggested: Choose the number of clusters k such that k exceeds the number of strata (5) by a comfortable margin (but not higher than necessary) to create substantial 'classes' that are composed of entries from different strata. In this case, k was set to 10. And secondly perform principal components analysis prior to performing k-means clustering Joliffe and Cadima [10]. This method creates linearly independent vectors (i.e. vectors that have covariance 0). The resulting vectors are then used as input for the k-means classification algorithm.

4.1.5. *Principal Component Analysis*

Our next step is to gather information about users using the obtained choice classification. Based on the outcomes of the labeling phase described earlier, each user now has a characteristic 'label distribution'. The relative frequencies of the different choice types are stored in a DataFrame for each user. Again, PCA is conducted to reduce these values to a set of 7 vectors that aim to capture the users' behavior and taste.

4.2. Prediction

In order to predict which strata will be chosen in different situations for different users, we feed the acquired data concerning trips made, alternatives, and user preference to an artificial neural network (ANN).

4.2.1. Neural networks for multiclass classification

Neural nets have extended the scope of machine learning beyond linear models. A feedforward neural network consists of one or more hidden layers, that each consist of a number of nodes. In a basic neural net, each node gets input from all nodes in the previous layer, and outputs to all nodes in the next layer. Each layer is assigned a type of activation function, i.e. a function that generates the output of a node based on the input from a previous node. Commonly used non-linear activation functions include *sigmoid, tanh* and *reLU* functions, with respective domains (0,1), (-1,1) and [0,∞). For a multiclass classification, the method that will be used in this study, another activation function is usually used in the final (output) layer. The so-called *softmax* activation takes exponents of the output of the previous layer and scales them such that they sum to 1. The network is trained by a back-propagation algorithm that works upon a chosen loss function. Commonly used loss functions include least-squares and cross-entropy loss. The network is trained with *rate* η. After each iteration the weights are adjusted in the direction of the gradient of the chosen loss function, based local derivatives and the chain rule. The *training rate* η is the parameter determining the magnitude of the change of weights after each iteration. Choosing a higher value for η increases the training speed but may result in not being able to find optimal values.

4.2.2. Data shape and processing

In this case, the shape of the individual data entries fed to the network is a table of 5 by 17; 17 attribute values for each of the 5 different alternative strata. Each *groupid* in the training set corresponds to one mode choice scenario and therefore to one of these tables. The values of the 17 attributes are not always available for every stratum as not every mode of transportation is possible on every trajectory. If no route was generated for a certain stratum in a certain scenario, all attributes corresponding to this stratum (including user-specific attributes that are essentially known even for alternatives that do not have a route generated) are set to 0 in this scenario, and will be passed to the network as such. This is done because the network requires the data entries passed to it to have a consistent shape (*Strata 1= row 1, Strata 2 = row 2 etc*), while in the meantime strata without a generated route option must not affect the working of the model. Before creating the data entries, all data has been normalized using the minimum and maximum values of the entire combined dataset.

5. Results

5.1. Classifying choices

Table 1 shows the composition of the clusters that result from the k-means clustering algorithm with $k = 10$. After inspection of the clusters, a description has been added based on the values of the used comparative measures observed among the choices in each cluster.

5.2. Prediction

The neural network model has been trained for 300 epochs. The initial architecture that consisted of a model containing 2 hidden layers, both containing 5 nodes, with *ReLU* (Rectified Linear Unit) and *softmax* activation, respectively, resulted in a model which reported promising results. Eventually, the accuracy for the training set was reported to be 98.7%. The accuracies for the validation and test sets attained values of 93.4% and 94.6% respectively. The sparse categorical cross-entropy loss, that has been used as the loss function that the model was trained upon, attained values of 0.087,0.252 and 0.205 for train, validation and test set. These values show that even though the model overfits slightly, it still performs quite well. Table 3 shows how well the model performed per Strata. Especially the scenarios where public transportation is chosen are almost always correctly predicted by the model.

6. Discussion

At first sight, the results of the model may look very promising. It needs to be verified however, that the model is suitable to use on real choice sets, and thus can identify which choice is more likely, rather than distinguishing between real and generated data. Since the chosen Strata is always linked to data from the real trips dataset, whereas

Table 1. Composition and description of clusters obtained by k-means clustering after applying PCA

Cluster	Strata 1	Strata 2	Strata 3	Strata 4	Strata 5	Total	Description
0	0	0	0	28	15	43	Public transport with transfers, where access/ egressmode is usually walking
1	0	51	0	2	1	54	Trips by car
2	0	0	0	18	31	49	Trips by public transport that are mostly direct: smaller number of transfers and lower waiting time
3	0	0	10	13	21	44	Trips with a large biking component. The trips often take more time than alternatives at hand
4	0	0	0	34	10	44	Trips with more transfers and higher waiting time than alternatives at hand.
5	10	0	0	26	14	50	Trips where walking is required. Similar to cluster 0, but with less transfers
6	0	9	0	19	5	33	Trips where car is used, but waiting time is also involved, like public transport trips with access/ egress-mode car.
7	0	4	0	0	2	6	Trips where both car and bicycle are used.
8	1	0	0	21	13	35	Trips with a relatively high waiting time, but not a high number of transfers.
9	0	0	2	4	6	12	Trips where cycling and walking are combined
Total	11	64	12	122	161	370	

Table 2. The confusion matrix obtained by predicting the chosen modes in the test set scenarios

		Predicted					
		Strata 1	Strata 2	Strata 3	Strata 4	Strata 5	Predicted correctly
Actual	Strata 1	7	0	0	0	1	87.5%
	Strata 2	0	26	0	3	2	83.9%
	Strata 3	0	1	1	2	0	25.0%
	Strata 4	0	1	1	142	0	98.6%
	Strata 5	0	0	0	0	125	100.0%

the alternatives stem from the generated set, this needs to be investigated. This is done by training a NN-model in the exact same way as before, but excluding certain attributes from the set. For this we distinguish between attributes that are distance related (*distance, car_distance, walk_distance, bicycle_distance*), attributes that are duration related (*duration, car_duration, walk_duration, bicycle_duration*), user characteristic attributes that have been extracted from

Table 3. The performance (in terms of accuracy and sparse categorical cross-entropy loss) of the NN-model after some attributes have been excluded from the data

Attributes excluded	Accuracy train set	Accuracy validation set	CE-loss train set	CE-loss validation set
None (*original* model)	98.7%	94.2%	0.087	0.252
User characteristic attributes, duration related attributes	94.6%	95.9%	0.259	0.303
User characteristic attributes	94.6%	94.2%	0.228	0.301
User characteristic attributes, *waiting_time*	94.1%	93.4%	0.250	0.329
Distance related attributes	96.2%	93.4%	0.155	0.334
waiting_time	94.9%	92.6%	0.182	0.310
Duration related attributes	97.0%	92.6%	0.128	0.273
Distance related attributes, *waiting_time* (proposed *revised* model)	96.0%	89.2%	0.188	0.373
User characteristic attributes, distance related attributes	89.8%	87.6%	0.364	0.494
Distance related attributes, duration related attributes	89.0%	84.3%	0.277	0.448
User characteristic attributes, distance related attributes, *waiting_time*	86.6%	76.9%	0.527	0.622

Table 4. The confusion matrix obtained by predicting the chosen modes in the test set scenarios, revised model

		Predicted					
		Strata 1	Strata 2	Strata 3	Strata 4	Strata 5	Predicted correctly
Actual	Strata 1	7	0	0	1	0	87.5%
	Strata 2	0	22	0	6	3	71.0%
	Strata 3	0	1	1	2	0	25.0%
	Strata 4	0	5	3	135	1	93.8%
	Strata 5	0	1	0	1	23	92.0%

the training data using k-means clustering and PCA, and the attributes *waiting_time* and *transfers*. The results are shown in Table 4.

These results show that the original model performs equally well if it is trained and tested without giving user characteristic features as input. The results also hint that some attributes may cause the model to learn to indeed recognize which transportation mode alternatives were put in as generated data and which alternative contained the real trip data. Evidence for this is that the model performs better in terms of accuracy if solely these attributes are taken into account, even though they it is not presumed that these attributes encompass a way larger predictive potential when compared to, for example, the duration of a trip. This problem occurs because we are not dealing with survey data, but rather actual information about the location and movements of individuals. In this case, presumably the best way to deal with the confounding variables is to look at a *revised* model where the attributes related to distance and the attribute *waiting_time* are discarded.

When looking at the revised model, it turns out that the presence of user characteristic attributes does indeed have a positive impact on the predictive performance. As can be seen in Table 5 and 6, the performance drops slightly overall but significantly considering the ability of the model to correctly predict the case of a trip by car.

7. Conclusion

To summarize, this study explored the possibilities of using neural nets for mode choice prediction in transportation. Two datasets were combined and after user feature extraction using k-means clustering and PCA, a neural net was

Table 5. The confusion matrix obtained by predicting the chosen modes in the test set scenarios, revised model excluding generated user characteristic attributes

		Predicted					
		Strata 1	Strata 2	Strata 3	Strata 4	Strata 5	Predicted correctly
Actual	Strata 1	6	1	0	1	0	75.0%
	Strata 2	0	14	0	13	4	45.2%
	Strata 3	0	1	0	3	0	0.0%
	Strata 4	2	4	0	128	10	88.9%
	Strata 5	0	0	0	2	23	92.0%

trained to identify the mode chosen out of alternatives. After testing the model on a separate batch of data, it turned out that 94.6% was classified correctly. As removing the user-specific variables did not result in a decrease in model performance, while removing seemingly less significant attribute combinations from the data did deteriorate model performance, it has been found that this technique is prone to small differences in the data patterns between the two data sets, which may hint at which Strata was chosen. After removing attributes related to this possibly confounding variable, it was found that the presence of generated user characteristic attributes in the dataset prevented the model from under-performing in case car travel was the chosen mode. As the neural net that was trained in this case study had quite simple architecture, it is not unthinkable that a more refined, deep-learning model can perform better, especially when it is being used on a more complex and heterogeneous dataset.

8. Further research

As presented in the discussion section, one of the challenges of using the proposed approach on cases like the one presented in this study, is data quality. Since we are dealing with real data concerning the location and movements of individuals, rather than survey data, a major focal point for further research could be to investigate methods that circumvent the differences between generated and genuine trip data that a neural net might pick up on. Such methods are necessary for an unbiased assessment of the model and techniques to be carried out. Furthermore, it was mentioned before that the method has only been applied to a selective sample of one dataset. It can be seen for example that for only very limited number of times bicycle is chosen for transport. The users in the sample are all North-South metro line users, which makes the sample less representative. Especially when training a model for general use, a representative sample can be of crucial importance. Therefore, it is advisory to apply the method to bigger datasets that are more heterogeneous for a more thorough assessment in further research. When using bigger and more complex datasets, the importance of choosing the right hyperparameters and architecture of the neural network increases. Further research could be done to investigate upon this.

References

[1] , 2016. Ov lijnennetvisie 2018. https://www.vervoerregio.nl/pagina/20160131-ov-lijnennetvisie. Accessed: 2019-08-01.

[2] , 2017. Conveyal r5 router. https://github.com/conveyal/r5. Accessed: 2019-08-01.

[3] Agrawal, D., Schorling, C., 1996. Market share forecasting: An empirical comparison of artificial neural networks and multinomial logit model .

[4] Angelo Guevara, C., Ben-Akiva, M.E., 2013. Sampling of alternatives in logit mixture models .

[5] Bentz, Y., Merunka, D., 2000. Neural networks and the multinomial logit for brand choice modelling: a hybrid approach .

[6] Conway, M.W., Byrd, A., van der Linden, M., 2017. Evidence-based transit and land use sketch planning using interactive accessibility methods on combined schedule and headway-based networks. Transportation Research Record 2653, 45–53.

[7] Coslett, S., 1981. Efficient estimation of discrete choice models .

[8] de Freitas, L.M., Becker, H., Zimmermann, M., Axhausen, K.W., 2019. Modelling intermodal travel in switzerland: A recursive logit approach. Transportation Research Part A: Policy and Practice 119, 200–213.

[9] Hayashi, Y., Hsieh, M., Setiono, R., 2010. Understanding consumer heterogeneity: A business intelligence application of neural networks .

[10] Joliffe, I.T., Cadima, J., 2016. Principal component analysis: a review and recent developments .

[11] Steinley, D., 2006. K-means clustering: A half-century synthesis .

[12] Van Cranenburgh, S., Alwosheel, A., 2019. An artificial neural network based approach to investigate travellers‘ decision rules .

[13] Vythoulkas, P.C., Kotsopoulos, H.N., 2003. Modeling discrete choice behaviour using concepts from fuzzy set theory, approximate reasoning and neural networks .