In-Vehicle Route Recommendation System with Learning by Neural Networks

Toshiyuki Yamamoto, Shoichiro Nakayama, and Ryuichi Kitamura

***Abstract*— A two-stage route recommendation system is developed. The system consists of the minimum cost path search stage and the route choice stage. Several routes are selected according to multiple additive cost functions in the former stage, and a recommendation route is chosen from among the selected routes according to another function capable of nonlinear manipulations in the latter stage. These functions are developed as neural networks. The effectiveness of the system is examined using an experimental data set.**

***Index Terms*— geographical information systems, navigation, neural networks, route choice.**

I. Introduction

IN-VEHICLE route guidance systems are one of the most prevalent consumer goods that belong to the class of intelligent transportation systems (ITS). In Japan, the cumulative shipments since 1992 of in-vehicle navigation system units reached 9,390,000 units as of the end of May 2002 [1]. Recommending a route to take is the primary function of the in-vehicle navigation system, and the appropriateness of the recommended route is one of the key factors that determine whether the driver follows the recommended route. In many metropolitan areas of Japan, road networks are dense, and numerous alternative routes are in general available for any given combinations of the origin and the destination. It is often impossible for the driver to compare every available route and determine the optimal route among them within a reasonable amount of time. A route-guidance system is valuable in the sense it aids the driver in making route choice a practicable decision. The driver, however, may not necessarily be satisfied with the route recommended by the route guidance system; preferences of routes are most likely different among drivers, and could vary from time to time even for the same driver. Commercially available route guidance systems, on the other hand, select routes to recommend based on very simplistic rules. In order to offer route guidance that is of value to the user, the route guidance system must learn its user's preferences and continuously refine its knowledge of user preferences.

One approach to address this issue is to develop a route choice model that contains driver-specific parameters which represent his/her preferences in route choice, then update the parameters with data that contain cumulative information on many choices made by the same driver. Discrete choice models such as logit models have been applied to represent individuals' travel behavior for some time. The parameters of such models are usually estimated after collecting information from all sample cases. Such estimation procedures are called "batch learning" from the viewpoint of learning. Batch learning is not suitable for in-vehicle guidance systems because the capacity to store data and the computational speed available to estimate a model with the data are both limited; rather, on-line learning, where model parameters are updated recursively as data are obtained, is more suitable. It is possible to treat the driver's choices repeated over time very efficiently in on-line learning.

We develop a learning model of route choice behavior using neural networks, which is one of the typical on-line leaning models. Neural networks have been applied in many transportation research fields [2], [3]. Prior researches applying neural networks to the analysis of route choice behavior (e. g. [4]) observed their high ability of representing drivers' route choice. Adding to their ability of on-line learning, neural networks also allow us to represent non-linear relationships between variables easily, a task that requires considerably more effort with discrete choice models. For example, the perceived disutility of a trip may increase proportionally with travel distance up to a certain point, beyond which the marginal increase in disutility with respect to travel distance may start declining.

Many graph search algorithms have been developed and applied to find the minimum cost path in the road network (e. g. [5]–[7]). Recently, faster algorithms using the hierarchical nature in real road network is applied for in-vehicle route guidance systems to find the optimal route (e. g. [8]–[10]). In these algorithms, however, a certain cost is pre-assigned to each link according to a certain additive cost function and the attributes of the link, and the minimum cost path is sought according to the total cost summed over the links on the path. Because of the algorithm, the cost function is unable to include non-additive attributes of the route such as the rate of deviation

This work was supported in part by Sumitomo Electric Industries, Ltd. under research grant.

Toshiyuki Yamamoto was with Kyoto University, Kyoto, Japan. He is now with the Department of Geotechnical and Environmental Engineering, Nagoya University, Nagoya, 464-8603 Japan (phone: +81-52-789-4636; fax: +81-52-789-3738; e-mail: yamamoto@civil.nagoya-u.ac.jp).

Shoichiro Nakayama was with Kyoto University, Kyoto, Japan. He is now with the Department of Civil Engineering, Kanazawa University, Kanazawa, 920-8667 Japan (e-mail: snakayama@t.kanazawa-u.ac.jp).

Ryuichi Kitamura is with the Department of Civil Engineering Systems, Kyoto University, Kyoto, 606-8501 Japan and with the Department of Civil and Environmental Engineering, University of California, Davis, CA 95616 USA. (e-mail: rkitamura@term.kuciv.kyoto-u.ac.jp).

from the geometrically shortest route. The additivity assumption is convenient for solving shortest path problem, and especially for traffic assignment problem, but unfortunately there are many situations in which the additivity assumption is inappropriate. Gabriel and Bernstein [11] suggested that these situations include nonlinear valuation of travel time, non-additive tolls and fares, and emission fees.

Artificial neural networks automatically identify such non-linear relationships and calibrate the parameters to represent them. Calibrated parameters, however, cannot be used to find the minimum cost path generally because of the non-additivity. A two-stage route recommendation system, therefore, is developed in this study. The system consists of the minimum cost path search level and route choice level. The former level selects several routes according to multiple additive cost functions, and the latter level determines a recommendation route among the selected routes according to the function capable of accounting for non-additive attributes.

Functions in both stages are formulated as neural networks. Prior researches suggested that several factors influence the preferences of the route [12]. The factors are, however, not fully known to us yet. Thus the combination of neural networks with GIS (Geographical Information System) based digital maps which are used for in-vehicle navigation system, has a potential to provide a better representation of the preference of the route than other methods such as logit models. Neural networks are capable of automatically and quickly identifying the contributing factors among the huge number of variables stored in the digital map. The parameters of the neural networks are estimated using an experimental data set in this study. Future extensions of the study are also described.

II. A Two-Stage Route Recommendation System

The proposed two-stage route recommendation system is shown in Fig. 1. At first, the user inputs the destination of the trip to the recommendation system, while the origin of the trip is inputted automatically as the current location of the vehicle.

At stage 1, N alternative routes for the inputted origin-destination pair are searched from the road network on the digital map using N additive cost functions, respectively. In stage 1, the neural networks themselves are not used to select alternative routes. Rather, the framework of neural networks is used for learning the preference of the user.

N two-layer feed-forward neural networks with one input layer, one output layer and no hidden layer are used in stage 1. Two-layer feed-forward neural networks with a sigmoid function become identical to binary logit models, when the differences between two routes in explanatory variable values are inputted to the nodes in the input layer. The cost functions correspond to the observable components of the utility functions in binary logit models, which are represented by the weighted summations of the inputs in the neural networks. Thus, the additivity of the cost function is maintained whilst the learning in the two-layer feed-forward neural networks, which represent N additive cost functions, is performed as

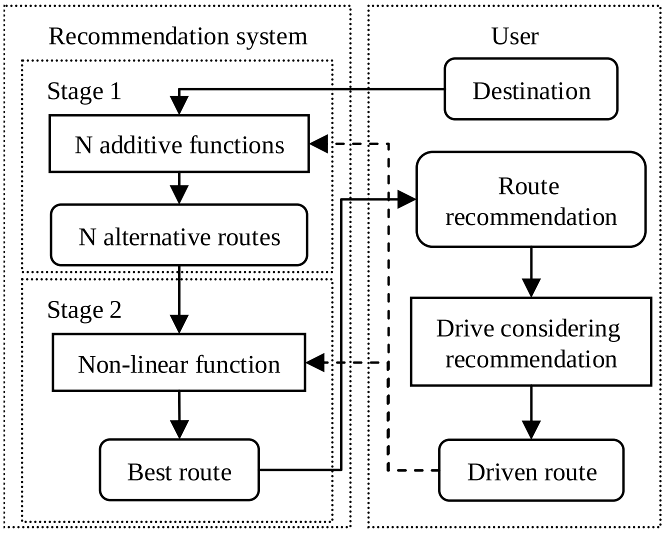

Fig. 1. Structure of a two-stage route recommendation system. Dashed arrows represent the feedback loop from the driver that neural networks use for on-line learning.

updating of the weights that apply to the respective variables in the cost functions. N least cost routes are selected from the road network based on the N additive cost functions. Note that the same routes may sometimes be selected even when N different cost functions are adopted. The identical routes are used as different alternatives in stage 2 in this case, and the computational process in stage 2 remains unchanged.

In stage 2, the best route is chosen, from among the N alternative routes, according to a neural network. The best route is given to the user as the recommended route. A 3-layer feed-forward neural network with sigmoid functions is used at stage 2, which is capable of representing the non-additivity of the preference of routes.

A neural network with N × K input nodes is a straightforward implementation at stage 2 if K explanatory variables of each route and N alternative routes are used in the recommendation system. However the number of input nodes tends to become large, causing the typical problem known as “over-fitting”; a model fits the data better as the number of nodes increases, but it does not necessarily predict well for cases which are not in the teaching data. Indeed, we examined this type of neural networks with 8 explanatory variables and 5 alternative routes, and obtained the results that the neural network perfectly replicated all 39 cases in the training data, but predicted correctly only in 5 cases out of 10 test cases. The results imply that the over-fitting occurred with 40 (= 8 × 5) input nodes.

Thus a neural network with 2 × K input nodes is used in this study for comparing two alternative routes in order to keep the number of input nodes small. 2 × K input nodes are for the sums and differences of the attributes of paired routes, i.e., ($X_{ik} + X_{jk}$) and ($X_{ik} - X_{jk}$), where X_{ik} is the kth attribute of route i. The best route is chosen from the round robin competition, in which two alternatives are compared with each other at a time, for all the pairs, the better route get one point in each comparison, and the route with the most points is chosen as the best route.

The user drives to the destination while taking the recommended route into consideration. If the user is satisfied with the recommended route, he/she would drive through the recommended route. Otherwise, the user would drive through a route other than the recommended route. In the latter case, the parameters of the cost functions in stage 1 and the neural network in stage 2 are adjusted to reflect that fact that the user preferred the driven route to the recommendation route.

The adjustments are performed by the learning mechanism of neural networks. Specifically, a back propagation algorithm [13] is used in this study. To update the cost functions in stage 1, the differences in the attribute values between the recommendation route and the driven route are inputted, and the weights are adjusted to calculate the weighted sum of the inputs such that the driven route will be outputted as the better route. The updated function is used as the new cost function in the minimum path search algorithm. Even if the cost functions are set to have completely different values from one another initially, the learning may result in the identical functions producing only one alternative route outputted in stage 1. One approach to avoid this inefficiency may be to let only one function be updated by learning, and let the other functions be set to be fixed as the values initially set. Alternatively, a mutation process like ones used in genetic algorithms may be included to keep the functions different from each other.

To update the parameters in stage 2, the attributes of the route driven by the user and the route recommended by the system are inputted similarly as in stage 1, but the sums and differences of the attributes between the two routes are used in stage 2. The neural network in stage 2 is also adjusted to output the driven route as the better route.

III. Sample Calibration

A. Data Set

The data set used in this study contains the attributes of 5 routes which are selected for each origin-destination pair on the basis of 5 criteria: minimize travel distance, maximize expressway (tolled) use, maximize arterial road use, maximize road width, and minimize turns. The data set represents 5 alternative routes for each of 49 origin-destination pairs selected for the study. The origin-destination pairs are chosen from Kinki region in Japan, and local governmental offices or train stations are used as the origins and destinations. The origin-to-destination distance ranges from 10 to 400 kilometers. These pairs were selected through discussions with engineers of the Sumitomo Electric Industries Ltd. (SEI) who were in charge of developing in-vehicle route guidance systems and who knew the traffic conditions along the 5 alternative routes for each origin-destination pair. The attributes of each route are represented by: total distance, distance of expressways, distance of wide roads (wider than 13 m), distance of narrow roads (narrower than 5.5 m), the number of turns, the number of traffic lights, the number of segments (a sequence of roadways that belong to one type of road width (wide road/narrow road/other) is defined as one segment), and the amount of tolls on expressways. The values of most of these variables were determined for each route based on the geographical information system (GIS) database implemented in the SEI's in-vehicle route guidance systems. Five alternative routes for each of the 49 origin-destination pairs are ranked according to their appropriateness by the SEI's engineers. These rankings are used in the study to indicate the preference of the user for the sample calibrations.

TABLE I
Calibrated Neural Network and Logit Model for Stage 1

Variable	Neural Network Weight	Logit Model Coef.	Logit Model s.e.
Total distance	-12.02	-13.61^{**}	2.97
Expressway distance	62.15	45.39^{**}	10.73
Wide road distance	2.74	1.19	1.75
Narrow road distance	-24.68	-8.82^{**}	3.14
Number of turns	-3.54	-3.66^{**}	1.19
Number of traffic lights	-2.35	-2.14	1.73
Number of road segments	-26.38	-21.02^{**}	6.41
Fee of expressway	-4.39	-2.49^{*}	1.05
Sample size	245		245
L(0)			-169.8
L(β)			-109.5
Hit ratio	29/49		29/49

**: $p < 0.01$, *: $p < 0.05$

B. First Stage

Here, the learning ability of the two-layer feed-forward neural networks with a sigmoid function are examined, so only one neural network is used for the experiment. A total of 245 pairs of alternative routes are used to train the 2-layer feed-forward neural network to output the higher ranked route as the better route. The input variables are normalized to have values between 0.0 and 1.0 as is normally done in neural network analysis. The learning rate is set as 0.1. The results are given in Table I.

All the weights of the trained neural network have the theoretically expected signs. The results suggest that expressways are preferred and that narrow roads tend to be avoided. The results also suggest that traffic lights and turns are negatively valued almost equally.

A binary logit model with the same explanatory variables as the neural network is estimated by maximum likelihood estimation to compare the estimated coefficients with the weights trained by the neural network. The results are also given in Table I.

Some of the variables have similar values between the weight of the neural network and the coefficient of the binary logit model. The statistical tests on the differences of the values between them are difficult because the standard errors of the weights in the trained neural network are not given in general. Bootstrap methods [14] may be a possibility to obtain standard errors of the weights in neural networks, but it remains as a further task. Here, the standard errors of the weights are assumed to be the same as those of the binary logit model in order to grasp rough inferences on the differences between them. Then, assuming no correlations of the values between the weight of the neural network and the coefficient of the binary logit model, the t-statistics, t_i's, are calculated to examine the differences of the two statistically by

TABLE II
HIT RATIOS FOR TRAINING CASES AND LEAVE-ONE-OUT CASES

Number of nodes in hidden layer	1	2	4	8	16
Training cases (/49)	35	38	45	48	49
Leave-one-out cases (/49)	31	30	30	32	33

TABLE III
HIT RATIOS FOR TRAINING CASES AND LEAVE-ONE-OUT CASES WITH 8 INPUT NODES

Number of nodes in hidden layer	1	2	4	8	16
Training cases (/49)	26	28	30	37	42
Leave-one-out cases (/49)	28	28	30	31	33

$$t_i = \frac{w_i - \beta_i}{\sqrt{2}\sigma_i} \quad (1)$$

where w_i is the weight of the ith variable in the neural network, and β_i and σ_i are the coefficient and the standard error of the ith variable in the binary logit model, respectively. The calculated t-statistics suggest that all the variables except narrow road distance have statistically indifferent values between the neural network and the binary logit model.

The hit ratio, which is defined as the fraction, in the total origin-destination pairs, of those origin-destination pairs where the highest ranked route has the highest value of the trained cost function, is calculated for the neural network and the binary logit model. Both models have the same hit ratio as 29/49, indicating that the two models have similar ability of replicating the ranking of alternative routes.

C. Second Stage

The same 245 pairs of alternative routes as used in stage 1 are used to train the 3-layer feed-forward neural network. The learning rate of the neural network in stage 2 is also set as 0.1. Alternative neural networks with 16 input nodes are developed in this study by setting the number of the nodes in the hidden layer as 1, 2, 4, 8, and 16.

Table II summarizes the results of network training. The hit ratio for training cases is defined as before, with the total of 245 cases from the full 49 origin-destination pairs. On the other hand, the hit ratio for leave-one-out cases is defined with cases for the 48 origin-destination pairs, excluding from the full pairs the pair for which the prediction is made. The latter requires developing 49 neural networks to test each origin-destination pair. This indicator is said to show the predictability of the model more appropriately than the hit ratio for training cases.

As shown in Table II, the hit ratio for training cases increases with the number of nodes in the hidden layer. The neural network with 16 nodes in the hidden layer perfectly replicates all 49 cases. However, the neural network with a larger number of nodes in the hidden layer tends to become "over-fit" to the data, too. The hit ratio for leave-one-out cases doesn't increase so clearly with the number of nodes in the hidden layer. The results imply the "over-fitting" occurs in the neural networks examined in this study.

In order to examine the "over-fitting", neural networks with 8 input nodes are also developed. The input variables are the sum of total distance, the difference of total distance, the difference of distance of expressways, and the difference of the number of turns. The results of network training are shown in Table III. The results suggest that the neural networks with 8 input nodes have lower hit ratios for training cases, but keep similar hit ratios for leave-one-out cases. These indicate that the neural networks with 8 input nodes have a comparable predictability with those with 16 input nodes.

IV. CONCLUSION

A two-stage route recommendation system is proposed, and the sample calibration is presented in this study. The proposed system is capable of learning the user's preference of routes through neural network training. The system is capable of representing the non-additivity in the preference, while conventional minimum cost path search algorithms which are limited to the linear cost functions can be used.

The results of model calibration show a high predictability of the neural networks, but imply that the neural networks examined in this study have a typical problem of "over-fitting". Thorough examinations to find the best structure of neural networks in terms of the number of nodes in the input and hidden layers and the learning rate remain as a further task.

ACKNOWLEDGMENT

The data set used in this study was provided to the authors by Sumitomo Electric Industries, Ltd. Comments received during the course of the study from Satoshi Fujii of Tokyo Institute of Technology, and Yoichi Doi and Masanobu Nagao of Sumitomo Electric Industries Ltd. are gratefully acknowledged.

REFERENCES

[1] http://www.its.go.jp (Accessed on 2002/08/10).
[2] M. Dougherty, "A review of neural networks applied to transport," *Transportation Research Part C*, Vol. 3, No. 4, pp. 247-260, 1995.
[3] D. Shmueli, I. Salomon, and D. Shefer, "Neural network analysis of travel behavior evaluating tools for prediction," *Transportation Research Part C*, Vol. 4, No. 3, pp. 151-166, 1996.
[4] H. Yang, R. Kitamura, P. P. Jovanis, K. M. Vaugh, and M. A. Abdel-Aty, "Exploration of route choice behaviour with advanced traveler information using neural network concepts," *Transportation*, Vol. 20, No. 2, pp. 199-223, 1993.
[5] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, Vol. 1, pp. 269-271, 1959.
[6] I. Pohl, I. "Bi-directional search,", *Machine Intelligence*, Vol. 6, pp. 127-140, 1971.
[7] D. Galperin, "On the optimality of A*," *Artificial Intelligence*, Vol. 8, pp. 69-76, 1977.
[8] J. F. Campbell, "Selecting routes to minimize urban travel time," *Transportation Research*, Vol. 26B, pp. 261-274, 1992.
[9] E. A. Huang, E. A. Rundensteiner, and N. Jing, "Evaluation of hierarchical path finding techniques for ITS route guidance," *Proceedings of the 6th Annual Meeting of ITS America*, pp. 340-350, 1996.
[10] K. Kim, S. Yoo, and S. K. Cha, "A partitioning scheme for hierarchical path finding robust to link cost update," *Proceedings of the 5th World Congress on Intelligent Transport Systems*, CD-ROM, 1998.
[11] S. A. Gabriel, and D. Bernstein, "The traffic equilibrium problem with nonadditive path costs," *Transportation Science*, Vol. 31, No. 4, pp. 337-348, Nov. 1997.

[12] COMSIS Corporation, “Analysis of travelers’ preferences for routing: literature review,” Report submitted to Federal Highway Administration, U.S. Department of Transportation, DTFH61-95-C-00017, 1995.

[13] D. Rumelhart, G. Hinton, and R. Williams, “Learning Internal Representations by Error Propagation,” in *Parallel Distributed Processing*, Vol. 1, D. Rumelhart and J. McClelland, Eds. 1986, pp. 318-362.

[14] B. Efron, and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York: Chapman & Hall, 1993.