Using Neural Networks for Route and Destination Prediction in Intelligent Transport Systems

Tomáš Miklusčák, Michal Gregor, and Aleš Janota

University of Žilina, Faculty of Electrical Engineering
Univerzitná 8215/1, 01026 Žilina, Slovakia
{tomas.mikluscak,michal.gregor,ales.janota}@fel.uniza.sk

Abstract. Route prediction and destination prediction based on the past routes are a missing piece in intelligent transport systems (ITS). These predictions can be useful in many areas: congestion prediction, traffic control, upcoming traffic hazards and targeting advertisements next to the roads are some of the obvious ones. Simply said, if we can estimate the future location of cars which are already on the road network, we will be able to estimate future congestions and upcoming traffic hazards. The GPS units in the new generation of smartphones provide a good data source for prediction algorithms. Google maps application already collects this data. This paper discusses several algorithms and methods which have been used in similar areas and a route prediction method based on artificial neural networks using the past routes of a vehicle.

Keywords: congestion prediction, neural networks, intelligent transport systems, prediction methods, location management.

1 Introduction

The aim of this paper is to transfer experiences from next location prediction problems which have been solved in smart environments, cellular networks and other to a very similar area: route and destination prediction problem. We propose a general route and destination prediction method which can be used to solve such problems. Several algorithms and methods for route prediction are discussed as well.

Traffic congestions are a major problem in big cities in developed and developing countries. Many different more or less successful approaches have been used to reduce congestions. Research indicates that traffic congestions can be alleviated by providing timely and accurate traffic information to drivers. With such information drivers can choose another route and avoid congested roads or even change departure time. The ITS able to predict congestions could potentially help traffic control to eliminate them, or even to spread information about future congestion so that motorists may adapt their decisions appropriately. Modern ITS are trying to collect data about actual traffic using variety of sensor systems like embedded roadway loop detectors, electronic toll collection transponders, automatic vehicle location, etc. They are also trying to predict congestion according to this data. Using this sensory data, ITS can predict congestions and even estimate route duration. Unfortunately, the

J. Mikulski (Ed.): TST 2012, CCIS 329, pp. 380–387, 2012.

capability of these systems to predict congestions is not sufficient. On the other hand, the ability to predict next routes and the final destination of vehicles may play a vital role in congestion prediction. Simply said, if we can predict next route and the final destination of vehicles which are on the roads already, we can also predict possible upcoming hazards, estimate future congestions and travel times with significantly higher accuracy. As a data source we can use data which is already being collected. Every modern smart phone and every navigation device is able to provide this data and commerce navigation companies do already collect it (e.g. Google maps). Among other sources of data, there are satellite tool systems.

Even though congestion prediction is the major concern of this paper, there are many other possible applications of route and destination prediction. Some of them will also be described in the following text.

The authors of [2] propose probabilistic reasoning from observed context-aware behavior (PROCAB) system. The authors envision a future navigation device that learns driving routes from a group of experts – taxi cab drivers – who have good knowledge of alternative routes and contextual influences on driving. This device is then navigating common drivers. In fact researches shows that drivers just rarely choose the fastest possible route. According to [3] just 35% and 38% according to [2] take the fastest way even if they are using commercial navigation software. The PROCAB includes three predictions: turn prediction, route prediction and destination prediction.

One of the most innovative applications of route prediction concerns improving efficiency of hybrid vehicles. Given knowledge of future changes in elevation and speed, a hybrid control system can optimize the vehicle's charge/discharge schedule. Researchers from Nissan showed that it is possible to improve hybrid fuel economy by up to 7.8% if route is known in advance [5]. While driver can be asked for his or her route before every drive, it is suspected that most drivers would tire of this quickly. This problem has been investigated in [3], where authors use K-order Markov Model to predict routes.

2 Prediction Methods

This part contains a small review of methods which have been used for route prediction or for prediction in similar areas of research.

2.1 To What Extent It Is Possible to Predict Routes?

The assumption underlying all route prediction systems is that human behaviour and particularly routing on roads can in principle be predicted. The analysis of inhabitant's daily lifestyle will tend to show identifiable patterns: these can be learned and therefore actions can, to a certain extent, be predicted. What is more, the underlying hypothesis that human behaviour and human movement can be predicted has to date been proven by many experiments, for example [2] and [3] where success rate of predicted route of common persons is higher than 85% in both cases. The ability to predict person moving have been proved in a similar area – smart house projects [6], [7] or smart office projects [8] as well. These researches indicate that in spite of the fact, that there exist a number of possible routes, the driver will tend to have their own routine, and their own routes.

2.2 From Cellular Networks and Smart Houses to ITS

The location prediction algorithms have been mainly developed and tested in cellular networks and smart houses. Pioneering in this area is Bhattacharya and his assumption that the inhabitant's mobility is a piecewise stationary ergodic stochastic process, as hypothesized in [4]. This work concerns tracking mobile users in a wireless cellular network. It has motivated a number of researchers to explore this area and a few very good algorithms and methods have been proposed. Among methods with higher accuracy of prediction that have been be used in smart homes are Active LeZi [6], Neural networks [8, 7], Bayesian networks [9], Hidden Markov models [10] and even algorithms based on game theory [11]. Later, route prediction has come into focus and K-order Markov models [3] and Markov decision process with inverse reinforcement learning [2] have been employed to accommodate this problem.

Algorithms based on game theory have been developed with the purpose of handling multi-inhabitant environment. Human behaviour is different if there are more persons in the same environment. Route prediction does not suffer from similar issues, so we will not consider this hereinafter. The major difference between using location prediction (person movement prediction) in smart pervasive environment (SPE) and route prediction in road nets lies in the size. While one will hardly find a net with more than 100 nodes in SPEs, 10 000 node road nets are not unusual. The Active LeZi is designed to work with this amount of data and in principle, Bayesian networks, or Hidden Markov models should be able to handle work with this amount of data as well. Our aim then is to explore whether an artificial neural network would be able to handle such a large amount data (the number of input and output neurons will be rather large when using the usual representation).

3 Using Artificial Neural Networks for Prediction

There is a number of papers that investigate applications of artificial neural networks (ANN) to forecasting, or, if you like, to time series prediction. Several points are to be addressed when investigating such applications, including:

- Selecting an appropriate architecture of the ANN;
- Finding an appropriate data representation;
- Specifying a way in which the data is to be coded into inputs for the ANN.

3.1 Architecture of the Artificial Neural Network

Perhaps the first choice to be made when designing an ANN-based forecaster is whether to use feed-forward architecture of recurrent architecture. In recurrent networks, as opposed to feed-forward network, neurons are allowed to have connections to outputs of neurons from the following layers.

Recurrent networks would seem to be especially well-suited to perform the task at hand. It is quite obvious that knowledge of several previous values (states) will generally be required in order to determine the forecast. It should be noted, however, that recurrent networks tend to be significantly more difficult to train.

Also, if we are willing to provide for time delay of input signals explicitly, feed-forward networks represent a viable alternative. In fact, according to [12], of these two, this is the more widely used approach.

To summarize the approach based on the feed-forward architecture, one may say that the data is fed to the ANN in the following fashion: Let $f(k)$ be a time series we are trying to predict (and k be the discrete time step). Then, if we leave coding of the values out of consideration for the time being, the following is to hold for data fed into the ANN at point k :

$$X(k) = \{f(k-1), f(k-2), \ldots, f(k-n)\}. \tag{1}$$

where n is the number of previous values of $f(k)$ we present to the network, that is, the order of the forecaster (or, if you like, the size of its memory). When specifying the value of n , one should therefore be reasonably certain that the system producing the outputs is of order equal or smaller than n .

3.2 Data Representation and Coding

As mentioned, the issue of coding has been left aside in the previous section. Let us now return to this point and give some attention to the various options that face us.

Many works concerning ANN-based forecasting deal with continuous data such as currency exchange rates, river flow, rainfall, financial and demographic indicators, etc. For such applications, what we have said so far is actually quite accurate.

The problem at hand is not, it should be obvious, of the same nature – identifier of a crossroad, route, etc. is a discrete value. There are several approaches one can take when devising a representation for such data:

- Feed the numeric code, such as it is, into a single artificial neuron (in the input layer);
- Have a separate input (and output) neuron for every room times every time step;
- Convert the numeric code into its binary form and assign one neuron to every bit of the resulting binary code.

The first option is the same as used for forecasting of continuous numeric variables. Such approach has some significant downsides, the most important of which is the fact that the ANN will not generalize correctly [7].

If we select the second option, we have to provide the ANN with $n.r$ input neurons, where r is the number of distinct routes. Input patterns would then be generated in such way that a given route would be represented by setting the input of its corresponding neuron to 1 while setting the inputs of all other input neurons to 0. While this approach is very useful for some smaller and medium-sized tasks, an ANN for our task would be prohibitively large.

Thus, we select the third approach, which provides something of a middle ground between the other two methods.

4 Experiments

The experiments presented hereinafter are based on data sets generated using a simulation model briefly described in the following sections. The experiments themselves compare the results achieved on two such datasets: *dataset 1*, which is an easy data set with very little stochasticity, and *dataset 2*, which, in contrast, represents a very difficult problem with a considerable amount of random decisions. Dataset 1 consists of 54031 samples. Dataset 2 has 69338 samples.

4.1 Simulation Model

Advantages of simulation when compared with experiments in real environments are obvious: a simulation model saves money and time. We have developed a simulation model which includes common driving of a common employed person in a small city (1000 crossroads). According to the day and time, the simulated person drives to work, to a shopping centre, church, football stadium and to visit parents, or friends. The average distance between these points is 50 crossroads. The simulated person sometimes just randomly drives to other places in the city.

It is obvious that there some downsides to this approach: we cannot really say how stochastic a real dataset would be and – on the other hand – how pronounced would the driving patterns be. Therefore we are working with two datasets that hopefully represent the boundaries. The main motivation behind this work is to ensure that such approach is feasible, and that it is possible to do this in real time for a reasonably large number of vehicles. It is obvious, however, that to gain some more realistic insights into what accuracy to expect, real datasets would serve better.

4.2 ANN Architecture, Programming Environment, Reference Hardware

In the following experiments a feed-forward network has been used to implement the forecaster. The ANN is trained using Rprop. The experiments have been carried in the Matlab environment.

The hardware platform used to perform the experiments is composed of the following components: 8GB of RAM; 1.6GHz Intel Core i7 720QM (quadcore, each core divided into 2 virtual threads).

Only a portion of the available memory and processor cores have been utilized in the course of learning and prediction. It is to be noted that Matlab parallelizes the computation to a certain extent, when presented with all data at once.

4.3 Results – Performance of the Forecaster

As for accuracy, the following results have been achieved when varying size of the training data set (the number of samples used to train the network; the rest of the samples is used to test network's performance). The performance is a ratio of correctly predicted routes to the total number of predictions.

For all of these results, architecture with two hidden layers, each with 500 neurons and sigmoid activation functions has been used. In every case, 5 previous values are presented to the network.

Table 1. Performance with varying number of samples used for training

Dataset 1		**Dataset2**	
Training dataset size	Performance (training dataset / testing dataset)	Training dataset size	Performance (training dataset / testing dataset)
2000	98.95% / **98.74%**	2000	98.95% / **64.27%**
1500	98.53% / **97.86%**	1500	99.27% / **64.45%**
1000	99.20% / **98.25%**	1000	99.20% / **57.29%**
500	100% / **89.41%**	500	99.40% / **56.69%**
100	100% / **59.79%**	100	100% / **44.39%**

4.4 Results – Elapsed Time

In this section, the training dataset size and the number of neurons in hidden layers vary. We compare the resulting training and execution times. The number of neurons is represented in the following way: [size of the first hidden layer, size of the second hidden layer]. We are using dataset2 in this comparison.

Table 2. Training time for [500, 500]; execution time for 67337 samples

Execution time		**Training time**	
Architecture	Execution time [s]	Training dataset size	Training time [s]
[500, 500]	4.233	2000	35.898
[200, 200]	1.575	1500	26.152

4.5 Evaluation of Results and Future Research

When looking at the results concerning performance, the need for testing on real data becomes increasingly obvious. The results achieved for the highly stochastic dataset are not very good – the performance approaches 60%. This is to be expected: the theoretical maximum is in fact not much greater for the data in question. It is shown that with an increasing size of the training data set, the results become better: this can be ascribed to the fact that the ANN is able to make better guesses concerning the probabilities related to the stochastic process.

As for the elapsed time comparison, we may conclude that the approach could be used in practice. The initial training can for the most part be done offline. The execution times are reasonable even for a large number of samples. If the computation were fully parallel and re-implemented in a more low-level language such as C or C++, this would be even more apparent.

5 Conclusion

The route and the destination prediction is a novel approach, able to solve many problems of current traffic systems. In this paper we have discussed a few methods, which have been used in a similar area: location prediction (person movement prediction) in smart pervasive environments (SPEs).

The major difference between using location prediction in smart pervasive environments and route prediction in the road net is the size of the problem. We have shown in this paper that it is possible to use artificial neural networks for route prediction. From the perspective of training and execution times, this approach should be tractable. As for performance of the forecaster itself, the results vary with the degree of stochasticity and the size of the training data set. To gain better understanding of this particular problem, testing on real data (as opposed to simulation data) would be advisable.

Acknowledgments. The paper has been elaborated with the support of the Slovak scientific grant agency VEGA within the research project "VEGA-1/0453/12 Study of interactions of a motor vehicle, traffic flow and road".

References

1. Karbassi, A., Barth, M.: Vehicle route prediction and time of arrival estimation techniques for improved transportation system management. In: IEEE Intelligent Vehicles Symposium, IV 2003, pp. 511–516 (2003)
2. Ziebart, B.D., Maas, A.L., Dey, A.K., Bagnell, J.A.: Navigate Like a Cabbie: Probabilistic Reasoning from Observed Context-Aware Behavior. In: UbiComp 2008 Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 322–331 (2008)
3. Letcher, J., Krumm, J., Horvitz, E.: Trip router with individualized preferences: Incorporating personalization into route planning. In: Proc. IAAI, pp. 1795–1800 (2006)
4. Froehlich, J., Krumm, J.: Route Prediction from Trip Observations. SAE Technical Paper 2008-01-0201, 13 p. (2008)
5. Deguchi, Y., et al.: Hev Charge/Discharge Control System Based on Navigation Information. In: SAE Convergence International Congress & Exposition on Transportation Electronics, Detroit, Michigan USA (2004)
6. Roy, A., Das, S.K., Basu, K.: A Predictive Framework for Location-Aware Resource Management in Smart Homes. IEEE Transactions on Mobile Computing 6(11), 1270–1283 (2007)
7. Mikluščák, T., Gregor, M.: Person movement prediction using artificial neural networks with dynamic training on a fixed-size training data set. Applied Computer Science: Management of Production Processes 7(2), 43–56 (2011)

8. Vintan, L., Gellert, A., Petzold, J., Ungerer, T.: Person Movement Prediction Using Neural Networks. Universität Augsburg, Institut für Informatik, Technical Report 2004-10, 16 p. (2004)
9. Petzold, J., Pietzowski, A., Bagci, F., Trumler, W., Ungerer, T.: Prediction of Indoor Movements Using Bayesian Networks. In: Strang, T., Linnhoff-Popien, C. (eds.) LoCA 2005. LNCS, vol. 3479, pp. 211–222. Springer, Heidelberg (2005)
10. Gellert, A., Vintan, L.: Person Movement Prediction Using Hidden Markov Models. Studies in Informatics and Control 15(1), 17–30 (2006)
11. Roy, N., Roy, A., Das, S.K.: Context-aware resource management in multi-inhabitant smart homes: a Nash H-learning based approach. In: IEEE Int'l Conference on Pervasive Computing and Communications, vol. 11, pp. 148–158 (2006)
12. Wong, W.K., Xia, M., Chu, W.C.: Adaptive neural network model for time-series forecasting. European Journal of Operational Research 207(2), 807–816 (2010)