Travel Time Estimation on Arterial Roads using Probe Data and Bayesian Network Learning

Charitha Dias[*1] Marc Miska[*2] Masao Kuwahara[*3]
Chodai Co., Ltd.[*1]
(2-1-3 Higashi-Tabata, Kita-Ku, Tokyo 114-0013, Tel: +81-3-3894-3236, dias-c@chodai.co.jp)
Research Fellow, University of Tokyo, Institute of Industrial Science[*2]
(4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Tel: +81-3-5452-6419, mmiska@iis.u-tokyo.ac.jp)
Professor, University of Tokyo, Institute of Industrial Science[*3]
(4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Tel: +81-3-5452-6419, kuwahara@iis.u-tokyo.ac.jp)

In this paper we are introducing a self learning tool for travel time estimation in signalized urban networks based on probe data. The main feature of this tool is, that it can be applied with a basic network description instead of a detailed modeling of the network structure. We show how probe data can be utilized to train a Bayesian network to forecast the travel time on a route along an arterial road. In the conclusion we take a critical look on the limitations of such a system and possible extensions to increase its performance.

Keywords: *Travel time estimation. Probe data, data learning, Bayesian network*

1. Introduction

Probe vehicles gained recently a lot of momentum in the traffic engineering world, since they allow in an ideal case a fully covered network status. However, due to the variety of individual drivers and their behavior a sufficient sample size is needed to extract reliable information.

In this paper, we are introducing a self learning tool for travel time estimation in signalized urban networks based on probe data. The main feature of this tool is, that it can be applied with a basic network description instead of a detailed modeling of the network structure. We utilize a Bayesian network to forecast the travel time on a route along an arterial road.

A Bayesian network (BN) is a special type of diagram (called a graph) together with an associated set of probability tables. The nodes represent variables, which can be discrete or continuous and the arcs represent causal relationships between variables. The key feature of BNs is that they enable us to model and reason about uncertainty, similar to models like neural nets, fuzzy control. In the BN we model this by filling in a probability table for each node. To find the so called Node Probability Tables or NPTs, there are several ways of determining the probabilities of any of the tables. The beauty of BBNs is that we are able to accommodate both subjective probabilities (expert knowledge) and probabilities based on objective data.

If a node has more than one parent node in the graph, additional conditional probability tables are needed. BNs make explicit the dependencies between different variables. In general there may be relatively few direct dependencies (modeled by arcs between nodes of the network) and this means that many of the variables are conditionally independent. The existence of unlinked (conditionally independent) nodes in a network drastically reduces the computations necessary to work out all the

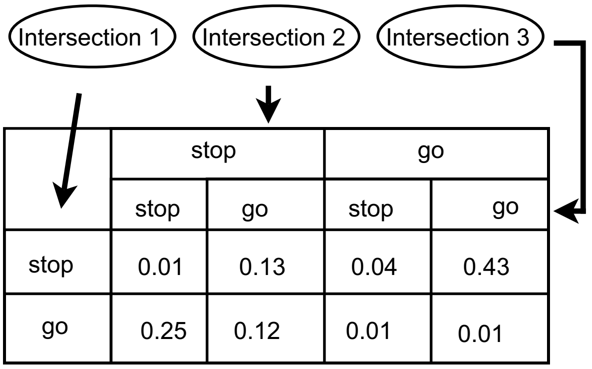
probabilities we require. In general, all the probabilities can be computed from the joint probability distribution. Crucially, this joint probability distribution is far simpler to compute when there are conditionally independent nodes. Imagine a large net with many dependencies and nodes that can take on more than two values. Doing the propagation in such cases is generally very difficult. In fact, there are no universally efficient algorithms for doing these computations (the problem is NP-hard). This observation, until relatively recently, meant that BNs could not be used to solve realistic problems. However, Rumelhart and McClelland [1] discovered propagation algorithms that were effective for large classes of BNs. With the introduction of software tools that implement these algorithms (as well as providing a graphical interface to draw the graphs and fill in the probability tables) it is now possible to use BNs to solve complex problems without doing any of the Bayesian calculations by hand. The most widely used BNs are the ones embedded in Microsoft's products as trouble shooter and technical support [2], and the Vista system, which is a decision-theoretic system that has been used at NASA Mission Control Center in Houston [3]. Others are medical diagnostic [4], traffic monitoring [5] and the Bayesian automated taxi [6].

In our approach, we use the BN to learn from actual probe data the values for stop and go patterns along the journey on an arterial road. The details will be described in the next chapter, followed by a feasibility test of such a system in undersaturated traffic conditions and a testing of the system performance under real world conditions. In the conclusion we take a critical look on the limitations of such a system and possible extensions to increase its performance.

2. Methodology

Travel time in urban networks is largely determined by intersection delay. Along an arterial road we can observe patterns of arriving during the green phase and stopping at an intersection depending on the demand and signal control settings. In an ideal situation with well synchronized signal settings (green wave), a vehicle will experience no delay passing the trajectory. However, this situation one may only find in the middle of the night. On all other times, vehicles will experience a delay and that delay depends on the pattern of needed stops along the intersections of the arterial. Assuming a sequence of three intersections, there are 2^3=8 possible patterns.

To identify the probability of these patterns to occur we feed the experienced patterns from probe vehicles to a Bayesian learning tool. In this study we used UnBBayes, a software package by Fernandez [7]. For the calculation of the conditional probabilities (see Figure 1) in the Bayesian network, UnBBayes utilizes the K2 Algorithm, which heuristically searches for the most probable belief network structure given a set of events to estimate conditional probabilities [8].



	stop		go	
	stop	go	stop	go
stop	0.01	0.13	0.04	0.43
go	0.25	0.12	0.01	0.01

Figure 1. Bayesian network to determine pattern probability

Additional to the experienced stopping pattern, the delay at each intersection is recorded. The average delay of all probe vehicles for a specific intersection is stored. These delay values multiplied with the stopping probability of the intersection result into the expected delay time on a journey along the arterial road. Figure 2 illustrates the procedure.

To create a self learning system, the Bayesian network is updated continuously with newly available probe vehicle records. In this way, we are able to grasp the average delays since outliers will have less and less influence. Since a probe vehicle must have completed the trip along the arterial before the data set can be fed to the Bayesian learning tool, there will be a time lag depending on the actual travel time of the journey.

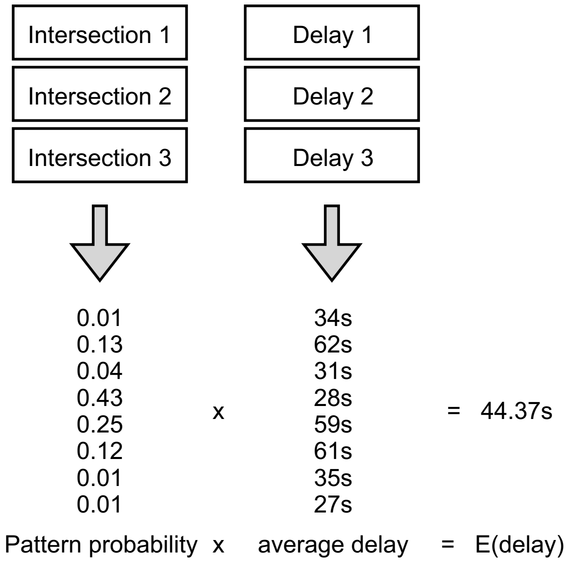


Figure 2. Illustration of calculating the expected delay value for a journey on arterial road

Summarizing the method so far, we are creating a Bayesian network structure including the number of intersections on the arterial road and calculate the probability of stopping patterns along the arterial road as well as the average experienced delay for the event of stopping at a certain intersection. The Bayesian network gets updated continuously, so that outliers are weighted less.

As noted in the beginning, the experienced stopping patterns and delays are highly dependent on the demand level, we have to incorporate the demand into the Bayesian network.

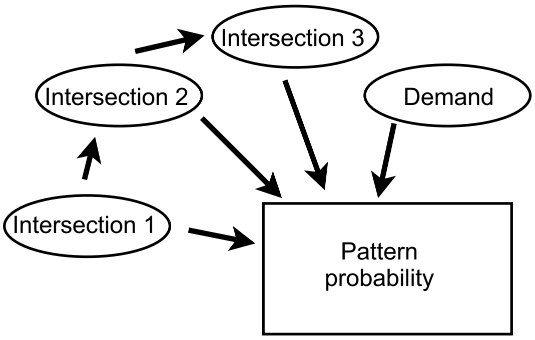


Figure 3. Including the demand into the Bayesian network to calculate the stopping pattern probability

In Figure 3 the Bayesian network for three intersections is illustrated. In comparison with the network from Figure 1, the conditional probability table contains now the 8 values for each demand level. We categorize the demand in 5 categories: very low, low, transition, congested, heavily congested. With this updated system we are now able to take different traffic situations into account.

The system is designed to stabilize itself during runtime and to improve the estimation quality with the amount of collected probe vehicle records. If we assume laboratory conditions, this is legit, but considering the real world, the system has to be able to deal with incidents and changes in the network (e.g. maintenance works, signal plan changes). To cope with these changes, we are running two parallel systems. While the first system (main system) keeps track of all probe vehicle records, the second system (sub system) will flush its memory except for a minimum of information periodically. This gives new probe vehicle records a higher weight and adaptable to new situations. A comparison of both system estimates will reveal changes in the network (capacity changes) or signal changes. If the difference in both systems is detected over a time period t, the estimation procedure will be taken over by the second system and the first system gets flushed. In this case the roles of the systems switch.

Now we have a Bayesian network that calculates pattern probabilities depending on the demand level and stores average delay values. The calculation gets updated with every new vehicle record. The final step left to provide route travel time estimation is to sum up the expected delay and the free flow travel times between the intersections.

3. Feasibility study

To check if the described methodology leads to feasible results, a 1 km stretch of an arterial road was simulated. The stretch includes four intersections with signal control.

To collect the training data for the bayesian network, 7% of the vehicles were selected as probe vehicles and the network was simulated for a series of 10 different demand levels.

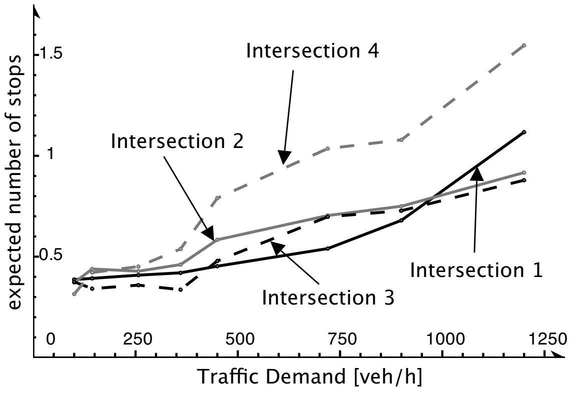


Figure 4. Expecting numbers of stops per vehicle given the demand level from an upstream detector

Based on these simulation the data from each probe vehicle was extracted to determine the number of stops and the caused delay at each intersection. The results are shown in Figure 4 and Figure 5.

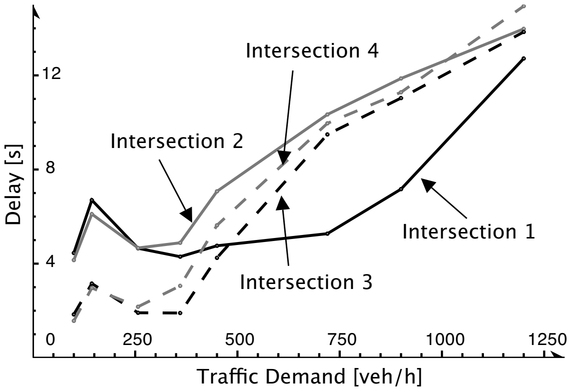


Figure 5. Expected delays calculated from the probe vehicle records.

While during low flow conditions the number of stops and the expected delay remain low, the stops and delay grow rapidly with the demand. While previous work only considered undersaturated conditions [9], this paper extends the work for higher flow rates.

Based on this, the developed system can now estimate the expected travel time on the arterial stretch, with only the demand of an upstream detector. To evaluate the system we simulated the arterial road again with a typical peak hour demand as shown in Figure 6.

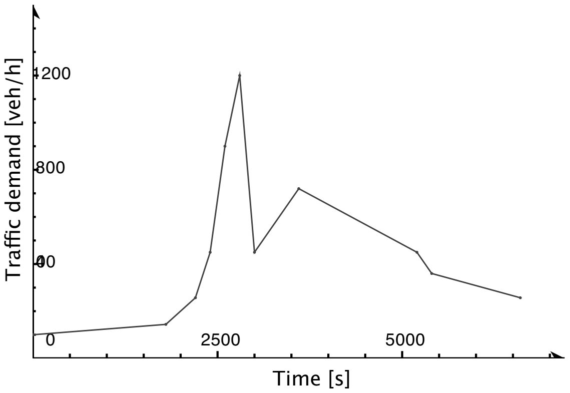


Figure 6. Demand profile of a peak hour for the simulation of the arterial road stretch.

During the simulation the actual travel time of each vehicle was recorded and is plotted in Figure 7. As one expect, the variation of individual travel times is high and the estimation of the travel time for traveler information systems can only provide average travel times.

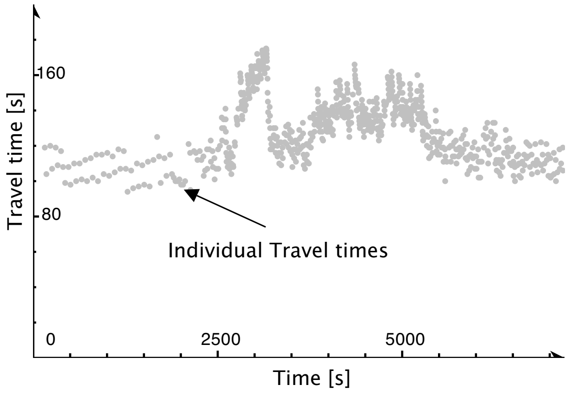


Figure 7. Individual travel time record during the simulation

Now, with the record of the upstream detector during the simulation, we fed the system to estimate the expected travel time. Figure 8 is showing the comparison between the data.

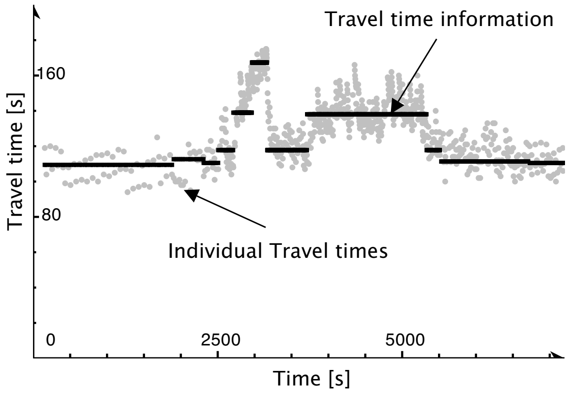


Figure 8. Estimated travel time information compared with the record of the individual travel time records

The system is able to give a good estimation of the experienced travel times. Certainly it does not cover the whole range, but gives a good estimation of the average travel times. To improve the result, one can take into account the change in the demand level and interpolate the expected travel time between these levels. The interpolation should take into account the update cycle of the traveler information system as well as the expected travel time itself.

Further simulations using two parallel systems to detect and react on anomalies in the traffic conditions or changes to the network structure (e.g. maintenance sites, signal program changes) did not lead to further information than already described in the original work by Dias [9].

5. Conclusions

In this paper we have described the setup and methodology of a self learning system to estimate the travel time on an arterial road stretch based on probe vehicle records. While the system learns continuously from incoming probe vehicle data, which can be performed offline, only needs real time detector counts to estimate the expected travel time.

The results we have presented show that the system is capable to forecast accurately the average travel times and can be used for traveler information systems. The detector counts can additionally be used to indicate the travel time development, either by a simple indication (e.g. increasing or decreasing demand) or by interpolating the shown travel time between the changing demand levels.

A critical reader might point out that the probe vehicle information needed (stopping pattern and delay) is difficult to extract. However, since this data can be collected and uploaded to the system offline, it is possible to use an in car equipment that uses more than just a GPS system. Useful information could be collected for instance from a taxi company fleet. Further research however could investigate the possibility to apply less location precise probe vehicles.

6. References

[1] Rumelhart, D., Geoffrey Hinton, and Ronald Williams. 1986. Learning internal representations by error propagation. In Rumelhart and McClelland, eds, Parallel Distributed Processing, vol 1. MIT Press

[2] Microsoft Support Technology, 1999, Microsoft Bayesian Networks, Basics of Knowledge Engineering, Microsoft, Redmont

[3] Horvitz, E. J., 1987, Problem-solving design: Reasoning about computational value, trade-offs, and resources. In Proceedings of the Second Annual NASA Research Forum, pages 26--43, Moffett Field, California, NASA Ames Research Center

[4] Wiggins M., Saad A., Litt B., and Vachtsevanos G., Genetic Algorithm-Evolved Bayesian Network Classifier for Medical Applications, Tenth World Soft Computing Conference and Transactions (WSC'10 2005), Sept 19-Oct 7, 2005.

[5] Huang, T., D. Koller, J. Malik, G. Ogasawara, B. Rao, S. Russell and J. Weber, 1994, Automatic Symbolic Traffic Scene Analysis Using Belief Networks. Proceedings of the Twelfth Conference on Artificial Intelligence

[6] Russell, S. et al, 1997, Batmobile: Towards a bayesian automated taxi. In Proceedings of the 14th Int. Joint Conf. on Artificial Intelligence, pp. 1878— 1885

[7] Fernandez, C.M., An Algorithm to Handle Structural Uncertainties in Learning Bayesian Network, Brazil, 2004.

[8] Ruiz, C., Illustration of the K2 Algorithm for Learning Bayes Net Structures, Lecture Notes on Machine Learning, Department of Computer Science, Worcester Polytechnic Institute, Massachusetts, 2005.

[9] Dias, Ch., Self learning tool for travel time estimation in signalized urban networks based on probe data, MSc thesis from the Department of Civil Engineering of the University of Tokyo, 2007

Charitha Dias currently works at Chodai Co., Ltd. After receiving his bachelor's degree in civil engineering from University of Moratuwa, Sri Lanka, he completed his master degree in civil engineering (transportation engineering and planning) at the university of Tokyo. After receiving his master degree he worked at i-Transport lab Co., Ltd for several months.

Marc Miska is a postdoctoral Research Fellow at Kuwahara Lab, Institute of Industrial Science, University of Tokyo. After studying applied computer science in civil engineering at the University of Hannover in Germany, he worked at the Franzius Institute for coastal engineering and hydraulics in the field of container transshipment and Hinterland transport before moving to the Transportation and Planning department of the Delft University of Technology, where he received his PhD in Traffic Engineering in 2007.

Masao Kuwahara is Professor of Traffic Engineering at the center for collaborative research, University of Tokyo. After obtaining a Ph.D from University of California, Berkeley, he started teaching and researching at the Department of Civil Engineering, University of Tokyo. He transferred to the institute as Associate Professor in 1987, became Professor in 2000 and transferred to the center for collaborative research as Professor in 2001.

- *Received date: March 19, 2008*
- *Received in revised form: November 14, 2008*
- *Accepted date: November 27, 2008*
- *Editor: Seungjae Lee*