

ECI-###

Driver's Travel Intention Prediction

YILU ZHANG

Electrical & Controls Integration Laboratory

REID SIMMONS

Carnegie Mellon University

Approved by

Jayendra S. Parikh, Group Manager

Patrick I. Popp, Director

Electrical & Controls Integration Laboratory



RESEARCH REPORT

GM RESEARCH & DEVELOPMENT CENTER

December 10, 2003

GM INTERNAL USE ONLY — NOT TO BE DISTRIBUTED OUTSIDE GENERAL MOTORS

ABSTRACT

Contents

Purpose of the Research	1
Conclusions	1
Significance	1
1. Introduction	2
2. Problem statement	3
2.1. Historical data collection	3
2.2. Prediction model	4
3. System architecture	4
3.1. Map database and its server	5
3.2. GPS receiver and its server	6
4. The learning/retrieval engines	7
4.1. The model	7
4.2. Algorithm	8
4.3. Complexity analysis	8
5. Experimental results	8
6. Discussions and future work	8
References	9
Distribution List	10

PURPOSE OF THE RESEARCH

CONCLUSIONS

SIGNIFICANCE

1. Introduction

Driving is fun but not easy. Many things are on driver's worry list: direction, traffic, detour information, parking, weather, fuel, and etc. Drivers may obtain this information from various resources, such as internet, radio stations, talking with friends, or personal experiences. Since all the information usage happens in the vehicle, having the vehicle be the resource of information is an appealing approach to both the customers and the OEMs, which is the basic idea of telematics.

With the fast advancing wireless technology, vehicles are no longer separated islands from the information continents created by the internet. The availability of the infrastructure, such as the reliable and high-speed communication linkage, however, is not the whole solution to implement telematics. Different from the desktop information consumers, drivers are not able to devote much attention to dig the needed information because her primary task is to maintain safe driving. Therefore, there is a bottleneck on the way from information sources to the end users.

A careful examination of the driver's worry list shows that all the items are related to the destinations the driver is heading to and the routes the driver is going to take. This is not a surprising discovery since the primary function of a vehicle is to carry the drivers and/or the passengers from one place to another. What it does tell us is that the bottleneck will be resolved if the telematics system knows the driver's intention in terms of destinations and routes.

There are many destination entry solutions in the existing practice. One way is to have the driver set the destination and preferred route before a trip starts. Since the driver does not need to worry about driving, the offline destination entry can be done through any conventional computer input device, such as keyboard, GUI with mouse. And the device to accept the inputs can be either the vehicle directly or a home computer or PDA that delivers the inputs to the vehicle through wireless communication. This solution is relatively cheap and straightforward to implement. Examples of this destination-entry solution includes OnStar's myOnStar (???). The major drawback of the preset solution is not able to handle dynamic driving situations. For example, traffic conditions may change after the trip starts. Driver's destination may also change during a trip.

Real-time destination entry solutions have been developed based on automatic speech recognition (ASR) technology and haptic device. Unfortunately, both of these two technologies have some limitations. The acoustic noise in the driving environment is the major hurdle for the voice-based system. The resulted poor recognition performance may distract the driver significantly. For haptic device, the concern is the manual and visual workload it imposes to the driver.

This report presents a system that learns driver's routine behaviors and predicts travel destinations and routes, both from passive observations. While the presented system can stand alone as a real-time destination entry solution, it can also be a complementary approach to the voice-based and haptic solution by providing contextual information so as to improve the recognition performance and/or reduce the distraction. In the rest of this report, we first define the problem and discuss the related difficulties. The system architecture is presented in Section 3 followed by the detailed discussions on the algorithm. Experimental results are reported on two datasets, the Detroit dataset and the Pittsburgh dataset. Finally, we conclude with some discussions on future work.

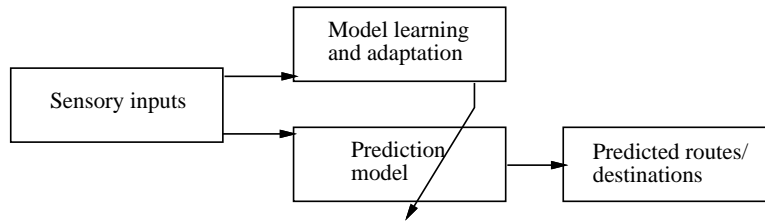


Figure 1: The diagram of the travel intention prediction system.

2. Problem statement

Unless the driver explicitly expresses them, driver's intentions are not directly observable. Our work of inferring driver's intention is based on the assumption that the driver has routine travel patterns, such as usual routes, usual destinations, stable detour preference, typical travel time, and etc. We also assume the driver to be reasonable in obeying traffic regulations, such as no off-road driving. This is a reasonable and important constrain. It reduces the complexity of our problem from the order of line segments in a plan to the order of the number of road segments and makes our problem much more traceable.

In order to do the inference, the system first needs to collect the driver's historical information such as the routes she has taken, the destinations she has been to, the time of the trips, and etc. Other information, such as the weather and the traffic, is also helpful because they are factors that influences the decisions made by the driver. A prediction model can then be set up based on the historical information. With the prediction model and the current driving information, such as current vehicle position and current time, the system predicts where the driver is heading and the subsequent road segments she may take. Fig. 1 shows the flow diagram of the above process.

There are three major issues in this project.

2.1. Historical data collection

How should the historical data be collected? One way to do this is to have the driver specifying her preference, such as the usual routes and destinations. While this approach makes the job of system engineers relatively easy in terms of setting up the prediction model, it may not be the most convenient way for drivers. A driver usually just wants to sit in the vehicle and start driving. Even if the driver wants to spend sometime doing the initialization, she may only cover the places she already knows or has been to. As time moves on, her experience will expend and so should her historical information in order to have an up-to-date prediction model. With this approach, she has to sit down and do the historical information updating periodically.

Another way of collecting historical data is to have the vehicle continuously recording the trip information in real-time. This approach guarantees that the driver's historical information and the prediction model are up to date. Plus, it minimizes the inputs from the driver, and thus improves the usability. The convenience comes with a price though. The plain sensory data does not explicitly tell driver's preference. The system has to infer the preference by certain criterion, such as the frequency of taking a particular route. In addition, the sensor noise may introduce uncertainty, although this problem is not unique to this approach because we also need to deal with the sensor

noise when doing the prediction.

Our solution is to combine the above two approaches. Given a route and/or the address of a destination and a map database, it is not difficult to synthetically generate a trip as a sequence of points represented by the longitude and latitude coordinates. The preferred routes and destinations explicitly specified by the driver then can be converted to a representation that is the same as the one used in real-time travel data collection. The degree of preference on a particular route can be changed by arbitrarily repeating the route in the training dataset.

2.2. Prediction model

A modern ground transportation system is essentially a network of road segments¹. Each trip made by a driver, therefore, consists of a list of connected road segments. To finish a trip, a driver makes a sequence of decisions to select the road segments that eventually link the origin to the destination. The nature of driver's navigation behavior decides that driver's intention prediction model should be a sequential decision-making model.

Driver's selection of road segments are affected by various factors. Some of the factors are observable but noisy or incomplete, such as the traffic condition, the weather, the road condition, and the time. Others are not observable, such as the intended destination and the detour preference. Although they may be resolved by the observations of driver's actual travel behavior, such as the vehicle's current position, the road segments taken, more uncertainty is introduced. Therefore, the prediction model should have a strong power of handling uncertainty.

In practice, we adopted the hidden Markov model (HMM). Hidden Markov model (HMM) is a well-developed method. It has advantages over other classification methods, such as expert systems or decision trees, in that it is better suited to dealing with uncertainty and sequential behavior. HMM has been widely used to model sequential human and robot behaviors, such as speech. In the application of automatic speech recognition, it successfully demonstrated the power of handling the uncertainty caused by the variant speech speeds, the variance between speakers, variance of speakers, the environment noise, the signal transformation channels, and etc.

There is a difference of trip predict from ASR though. An ASR system does recognition after a whole speech sequence, such as a word, is observed. On the contrary, a trip should be recognized as early as possible during the trip instead of after the trip is finished. (partial recognition ???)

In addition to the representation power of the model, another important issue is how to adapt the model the changes. The change may be introduced by a relocation of the driver, the detour caused by construction, or simply changes of travel habit. To deal with changes, the system should be able to conduct incremental learning and constantly update the model as new data is available.

3. System architecture

Fig. 2 shows the system architecture, where the arrows represents the direction of the information flow.

The whole system has three major hardware components, the computer, the input device, and the output device. In current implementation, we only utilize the GPS receiver as the input device.

¹We do not deal with off-road driving situations because they usually violate the routine-behavior assumption of this project.

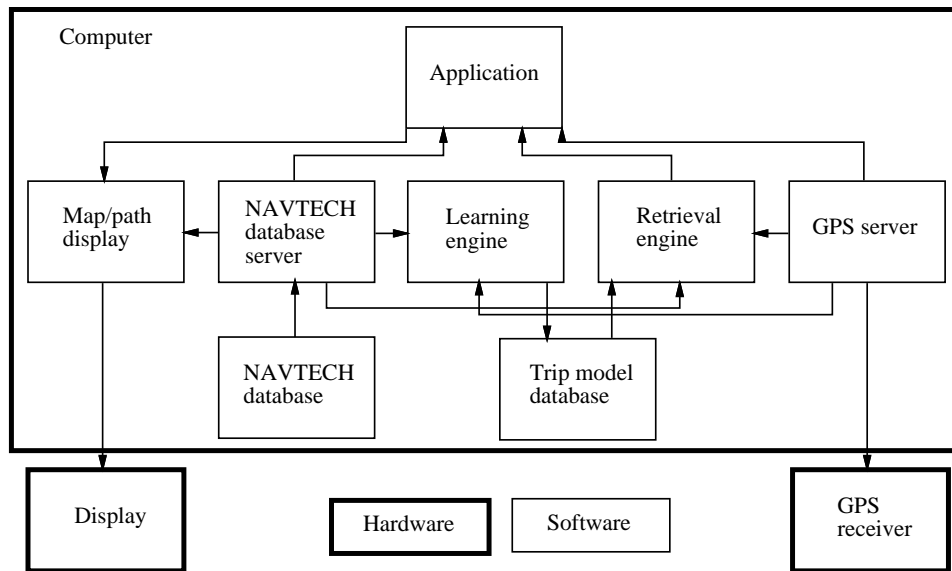


Figure 2: The architecture of the travel intention prediction system.

Other sensory inputs, such as wireless communication module to provide traffic, weather, road condition information, can be added as needed. The output device implemented right now is the graphic display. Depending on the ultimate application, a voice output device can be added.

The software of the system is written in Java and runs on Windows 2K/XP platform. The application module can be any application that requests route/destination prediction. Other than executing codes for a specific application, the application module receives the vehicle position information (longitude and latitude) from the GPS server, queries the future route/destination from the retrieval engine module, and provides appropriate information to the driver. Since the focus of this project is the trip model learning and retrieval, the user interface is not optimized. We simply draw the actual route (path) and the predicted route (path) on the map through the Map/Path display module. For the debug purpose, a brief text description of the predicted routes is generated every time when there is a change in the prediction.

The learning engine module continuously accumulates the vehicle position information (historical data), and creates or updates HSMM trip models in the trip model database. The retrieval engine collects the current vehicle position, retrieves the trip model database, identifies the most possible trip, and provides the predicted routes and destinations to the application module. In addition to collecting information from the sensor module (the GPS sever), the learning/retrieval engine modules communicate with the map database server in order to locate the positions of the vehicle in the map, such as the road segments.

A screen shot of the running software is shown in Fig. 3.

3.1. Map database and its server

We use the map database provided by Navigation Technologies Corporation (NavTech). NavTech is a leading provider of digital map information and related software and services. The NAVTECH map database is a digital representation of the road network that depicts what is on the road in

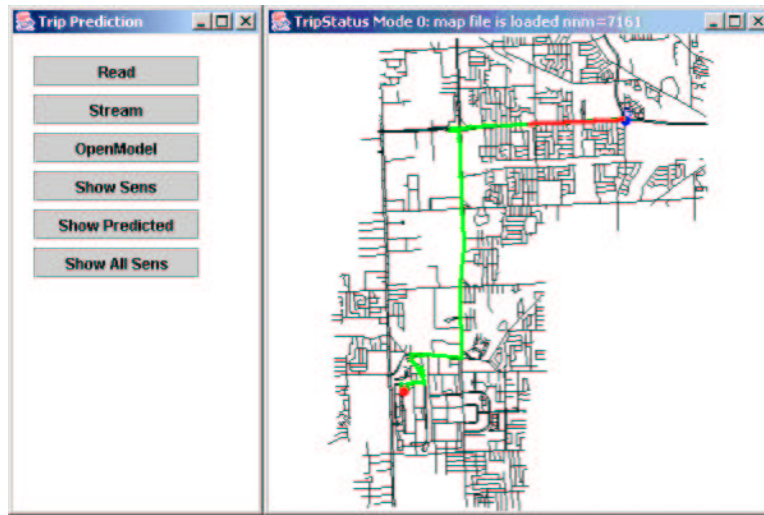


Figure 3: The screen shot of the travel intention prediction software.

detail. From turn restrictions to points of interest, every road segment has up to 150 data attributes. In addition to the database, NavTech provides a suite of software tools that facilitates the development of navigation systems, such as geocoding, route calculation, route guidance, and vehicle positioning. In our project, we mostly use the vehicle positioning tool to translate the position of a vehicle based on sensor inputs, the GPS readings, into a position in the NAVTECH database.

The NAVTECH database defines two essential navigable features, road network and junction. We are primarily interested in the road network, in which each road is represented by a series of road segments. A road segment is defined to be a segment of road between two consecutive intersections which can be either major intersections with traffic lights or minor intersections such as the U-turn spots. Each road segment is assigned a unique 32-bit ID number in the NAVTECH database. 256 discrete sub-segments, called *spots*, are defined within each road segment. Given a longitude/latitude position, the vehicle positioning tool returns the segment ID of the nearest segment and the spot ID of the nearest spot in the NAVTECH map database.

The NAVTECH database server (Navserver) was developed at GM R&D. Navserver is a multifunctional navigation server that fulfills requests from clients connected to it over a network. It provides geocoding, route calculation, route guidance and map-data generation services. For a more detailed description, the reader is referred to [1].

3.2. GPS receiver and its server

The accuracy of GPS receivers depends on a number of issues. The two most important ones are the number of satellites visible and their geometric configuration. A minimum of four satellites are needed to obtain a good GPS position. The measurement of geometric configuration is dilution of precision (DOP). The lower the DOP is, the better the accuracy is. A good DOP occurs when the satellites being tracked and used are evenly distributed throughout the sky. A bad DOP occurs when the satellites being tracked and used are not evenly distributed throughout the sky, i.e., they are grouped together in one part of the sky. Signal multipath also degrades the accuracy. This



Figure 4: The Deluo USB GPS receiver.

occurs when the GPS signal is reflected off objects such as tall buildings or large rock surfaces before it reaches the receiver. This increases the travel time of the signal, thereby causing errors. Other issues affecting the accuracy are the receiver clock errors, the weather that changes the speed the satellite signals passing through the atmosphere. U.S. Department of Defense intentionally degrades the satellite signals to prevent military adversaries from using the highly accurate GPS signals, which is called selective availability (SA). But the government turned off SA in May 2000, the accuracy of civilian GPS receivers is significantly improved. Nowadays, the accuracy of GPS receivers without calibrating with base stations (differential GPS or wide area augmentation system) is about 5 to 15 meters on average.

We use the USB GPS receiver (GPSU) built by Deluo Electronics (See Fig. 4). GPSU has a built-in antenna. To get good exposure to satellite signals, the receiver should be placed on top of the roof of the vehicle. After powered up, GPSU continuously transmits the data in the NMEA-0183 format at an update rate of 1Hz. The claimed position accuracy is 5 to 25 meters. Since we do not have the ground truth of the position, we could not assess the actual accuracy. But the average error between the GPS readings and the map database is within 10 meters.

4. The learning/retrieval engines

The core part of this architecture are the modules of the learning/retrieval engines.

4.1. The model

HMM is a variant of a finite state machine. It models a doubly stochastic process. One of the underlying processes is the transition between states that are usually not directly observable, represented by the solid arrows in Fig. 5. The other stochastic process is the emission of observations from each state, represented by the dash arrows in Fig. 5. For simplicity, an HMM is usually defined

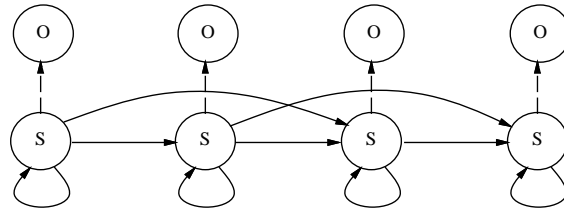


Figure 5: An example of linear hidden Markov model, where O represents an observation vector and S means a state. The solid and dash arrows represent transition and observation probabilities, respectively.

by a triplet, (A, B, π) , where A is the transition probabilities between states, B is the observation probabilities specifying the chance certain observation occurs under a particular state, and π is the probabilities of the initial states.

The observations in this project can be the GPS readings, the road segments, time, and the weather/traffic information. Given the dataset of the sequential observations, the HMM training phase deals with identifying the sequential patterns, such as routine trips, that can be represented by HMMs and learns the probabilities of the HMM for each pattern. When new observation data comes in during the test phase, the HMM with the highest likelihood to fit the new data is selected and the corresponding pattern (trip) is considered to represent the new data.

In current implementation of the system, we only deal with GPS readings. However, the presented framework can be extended to multiple sources of information. ???

... In addition, HMMs lend themselves to autonomous decision-making, using either exact or heuristic methods to solve the corresponding partially observable Markov decision process ... ???

...

4.2. Algorithm

... dealing with GPS reading errors ...

... training algorithm ...

... retrieval algorithm ...

4.3. Complexity analysis

...

5. Experimental results

6. Discussions and future work

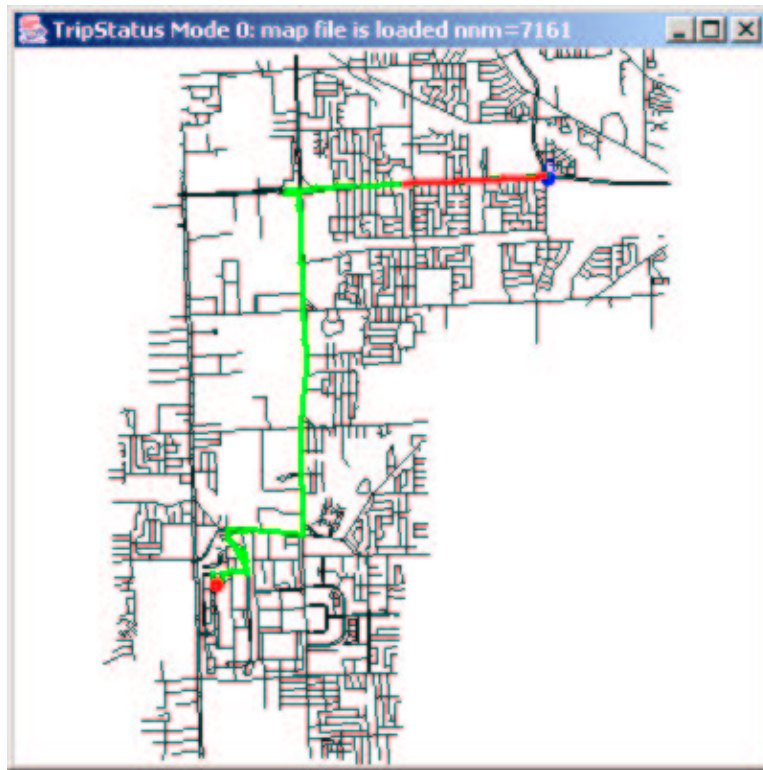


Figure 6: .

References

- [1] R. Roesser, "Navigation server and map client for vehicle information technology," Tech. Rep. ECI-###, under review, General Motors R&D and Planning, 30500 Mound Rd, Warren, MI, December 2003.

DISTRIBUTION LIST

GM R&D Center

Executive

Alan Taub: alan.taub@gm.com

Electrical & Controls Integration Lab

Patrick I. Popp: patrick.i.popp@gm.com

Jayendra S. Parikh: jayendra.s.parikh@gm.com

Technology Management & Technology Intelligence