

Traffic Advisories Based on Route Prediction

Kari Torkkola, Keshu Zhang, Haifeng Li, Harry Zhang, Christopher Schreiner, Mike Gardner

Motorola, Intelligent Systems Lab, Tempe, AZ, USA

Phone: (602)-438-3467

harryzhang@motorola.com

ABSTRACT

Traffic advisories can be delivered to assist travelers in avoiding congested areas and reaching their destinations in a timely manner. Instead of relying on a user to manually enter travel routes and times, we propose a prototype application that learns from the user's travel history and automatically predicts the user's destinations and routes. Traffic flow information is then accessed to determine whether congestions will occur along the predicted routes to warrant the delivery of a traffic advisory alert. User acceptance and trust are evaluated in a field study.

Categories and Subject Descriptors

H.5.2 User Interfaces: *User-centered design*

General Terms

Algorithms, Design, Human Factors.

Keywords

Context aware, user modeling, mobile assistance.

1. INTRODUCTION

Highway congestion has become a way of life in many parts of the world. This problem stems from several factors. First, the limited highway capacity simply cannot keep up with the population growth in many metropolitan areas. In U.S.A., driving is still the dominant method of transportation, despite the availability of public transportation. Second, urban sprawling has resulted in a rush hour nightmare when driving to metropolitan areas from the suburban areas during the morning commute and when driving in an opposite direction during the afternoon commute. The use of the high occupancy vehicle (HOV) lane can mitigate the congestion problem, but carpooling is not always available or convenient to everyone. Finally, highway constructions often lead to lane restrictions, and highway crashes can bring the traffic to a standstill. In U.S.A., there are over 6.4 million police-reported highway crashes every year.

Several commercial companies provide traffic flow information that is gathered by the department of transportation at the state level, the local police department, and their own helicopter fleet. Roadway sensors and cameras have been the prevalent method of acquiring the traffic flow information. The traffic flow information is made available to the public via radio or TV broadcast, Internet portals (e.g., www.traffic.com, maps.yahoo.com, maps.google.com), RSS feeds, email messages, or SMS messages on mobile phones. Radio or TV broadcast appeals to a large-scale audience, whereas a user can manually set up a traffic monitoring service on Internet portals that will send a custom report (via an email, an SMS message, or an RSS feed)

when the traffic congestion exceeds a threshold for a particular route during a time window.

A traffic report that is customized for a user is desirable because it tends to provide information that is more pertinent to the user. However, the initial manual set-up using Internet portals is tedious. If travel routes and times are changed, they will need to be modified manually on the Internet portals. The manual set-up and modification process is difficult to manage when the number of travel routes and times grows to accommodate mobile lifestyles. This process can be simplified or largely bypassed by using an automatic algorithm that predicts the routes that a user will take at a particular time. This paper describes one such algorithm that learns from a user's travel history, predicts likely routes and destinations using context information such as the current location and time, determines traffic conditions along the predicted routes, and delivers timely advisories and alerts to the user when there are unexpected slowdowns.

Krumm and Horvitz and their colleagues [3,4] have attempted to predict a user's routes from the user travel history. We adopt a similar approach to learn a user's usual locations and routes from the usage history and predict likely routes based on the location and time information. Unlike Krumm and Horvitz, we do not attempt to handle destinations that the user has not previously traveled to. The resulting traffic advisory system is a functional prototype which uses time and location information in conjunction with learned patterns of user behavior to determine when a user is leaving a location for a learned destination. The application reviews past routes that the user has taken to traverse from the current location to the predicted destination and checks available traffic monitoring services to determine if there are any unusual alerts.

2. DATA COLLECTION

Location and route data was collected from fourteen participants, representing 6 undergraduate students, 5 office employees, and 3 independents (stay-at-home parents or businessmen such as realtors who did not work consistently in an office environment). Each participant was provided with a commercially available mobile phone equipped with Bluetooth, along with a separate Bluetooth GPS receiver to acquire location data. Data logging software was installed on the mobile phone. Each participant used the phone and the GPS receiver for a two to three-month period. Context data logging software is "ContextPhone", public domain software from University of Helsinki, Finland [6]. The logged context variables consist of the following:

- GPS data from a Bluetooth GPS receiver,
- Current GSM cell ID,
- Bluetooth devices around the phone,
- Phone profile,

- Active phone application,
- Phone idle/active time,
- Battery status and charger status,
- Incoming/Outgoing calls, SMS or multimedia-messages,
- User interaction with the Phonebook and recent call log,
- Media captured with the device (e.g., photo, video).

The data is uploaded over the air to a web server, transparent to the participants. Logged data consists of events, a time stamp followed by the event. GPS and Bluetooth environment are scanned every 30 seconds. Other variables are logged as their status changes.

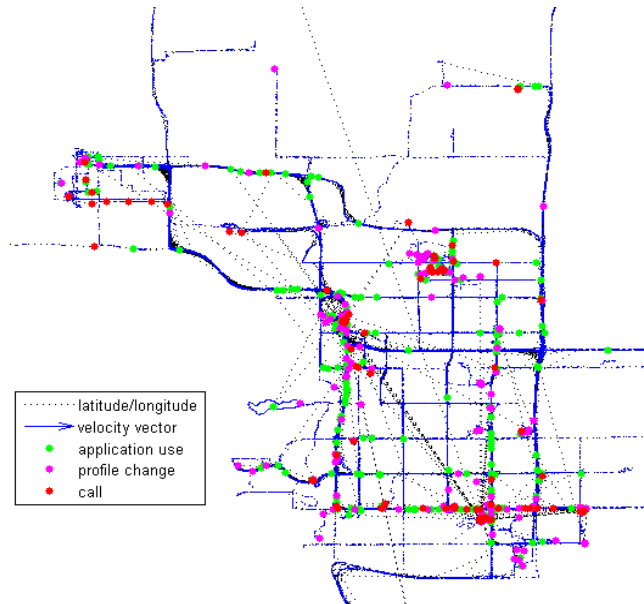


Figure 1. GPS coordinates are plotted together with selected phone application events for a typical participant.

As our focus in this paper is on a user's locations and routes, GPS coordinates can be plotted together with the use of phone applications to visualize the roads to which the participant has traveled and where the mobile phone is used (Figure 1). Figure 1 is cropped to cover only the local metropolitan area. It indicates that there are a small number of seemingly important locations where the phone has been used. It also indicates that the user's daily commute routes are regular and predictable.

3. ADAPTIVE LOCATION LEARNING

3.1 Adaptation of location clusters

Having alluded to the existence of regularities in the user behavior data, we describe our experimentation in discovering those regularities. Based on the collected context data, we employ machine learning techniques to determine the important locations where the user spends most of his/her time or most often is engaged in some activities. Location learning is mainly based on the collected GPS data. The GPS signal may be uncertain, however, when the user enters a building, a concrete canyon in an urban area, or when the GPS device is powered off [7]. To mitigate this problem, we use cell IDs as a secondary source to resample GPS data when GPS data loss occurs. The void GPS data is made up by repeating the last valid GPS reading, if the

area in which a GPS signal disappeared and reappeared, is covered by the same group of cell IDs.

Although a batch (non-adaptive) clustering method can automatically identify important locations, it is computationally expensive and requires processing available data as a whole [3,4]. It cannot adjust to changes in a user's lifestyle pattern. In order to track changes, adaptive (or incremental/sequential) algorithms are adopted. An adaptive algorithm is trained with a small amount of data initially, and adapts continuously as new data is entered. There are several advantages of using adaptive algorithms over batch algorithms. Adaptive algorithms are computationally efficient. They require less storage because a large set of previously used training data do not need to be kept in memory. The algorithm parameters can vary continuously to accommodate lifestyle changes.

For our traffic advisory system, a Gaussian mixture model is used to identify important locations. Assume for the moment that the true (but unknown) density of the GPS value g_t is of the form

$$p(g_t; \theta_t) = \sum_{\ell=1}^{N_t} \varpi_t^\ell \phi(g_t; \mu_t^\ell, \Sigma_t^\ell)$$

Where the number of components $N_t < \infty$ is known, parameter vector is $\theta_t = \{\varpi_t^1, \mu_t^1, \Sigma_t^1, \dots, \varpi_t^{N_t}, \mu_t^{N_t}, \Sigma_t^{N_t}\}$, the nonnegative mixing coefficients ϖ_t^ℓ sum to unity, and $\phi(g_t; \mu_t^\ell, \Sigma_t^\ell)$ is the normal probability density function with 2 dimensional mean μ_t^ℓ and standard deviation Σ_t^ℓ . The goal is to estimate the parameter vector θ_t , which consists of $3N_t$ components.

At time t , the number of Gaussians N_t in the mixture model represents the number of significant location candidates, the weight of Gaussians ϖ_t^ℓ denotes the significance of the learned locations, and the mean μ_t^ℓ and variance value Σ_t^ℓ in each Gaussian specifies the important location center and size.

In order to make model parameters adaptive, Titterton [8] has provided a recursive update formula based on maximum likelihood estimation. With Priebe's adaptive mixture estimator [5], we have designed an adaptive location clustering method which includes the update rule, creation rule, pruning rule and a model group switch rule. The details of the adaptation procedure will be described in a forthcoming paper. The adaptive location clustering method can recursively adjust not only the location parameters, such as the center and size, but also the number of location clusters needed to fit the data. This adaptive mixtures approach does not require intensive computational power and storage capacity and produces a consistent small sample estimator.

3.2 Evaluation of adaptive location learning

To evaluate the location learning feature in the traffic advisory system, we compare our adaptive location learning algorithm to a batch learning method. As the batch method we employ the DBSCAN (density-based clustering) algorithm, because it can automatically find the number of clusters, identify outliers, work well for arbitrary-shaped clusters and efficiently for large datasets [1]. We first evaluate the algorithms on one office employee who worked in regular business days at a fixed location. Both batch and adaptive algorithms are applied to the re-sampled GPS data,

the comparison results for important location discovery are generated for the first day, the first week, and the first month. Figure 2 presents the visualization results of the discovered significant locations after one week.

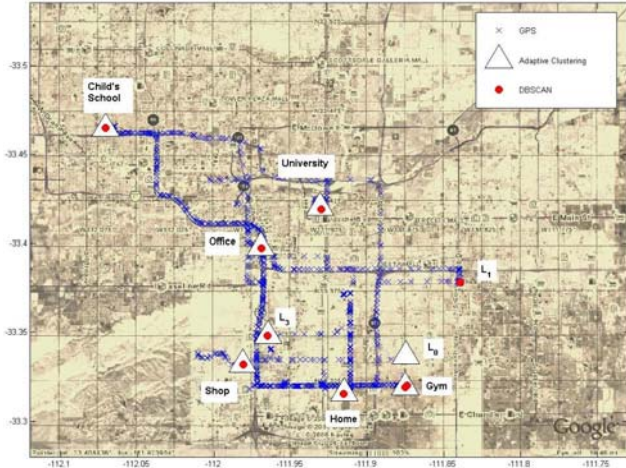


Figure 2. Location identification after one week

After the first day, the discovered locations by both batch and adaptive methods match very well. After the first week, compared with the batch method, the adaptive method discovered location L0 instead of L1 (Figure 2). Since the adaptive method adjusts the parameters online, location L1 which was visited early in the week is already discarded from the significant location candidates due to the short visit time and low visit frequency (only one visit for that week). Instead, location L0 which has been visited most recently had higher initial weight and then becomes more important than location L1. Meanwhile, both algorithms can discover several nearby locations in certain large areas, i.e. the “University” campus and the “Gym”. This comparison shows that the proposed location learning algorithm is effective in location identification and can adapt to the user’s daily life.

4. ROUTE LEARNING AND PREDICTION

After learning the important locations, the next step is to learn routes between a pair of locations. Currently, most transportation route learning methods are based on geographic information systems (GIS). Because our system runs on cell phones with limited computing and storage resources, we instead use raw GPS data sequences to represent routes [3,4]. A GPS string between two locations is denoted as a route. The starting point and ending point of the route involves detecting of the user leaving one important location boundary and entering into another important location boundary.

Because travelers often take different routes from location A to location B, GPS data strings between these two locations may not represent one single route. In order to determine if two GPS data strings represent the same route, the data strings must be compared. If the difference is small (with a MinMax criterion), then the two GPS strings belong to the same route. Here is the mathematical formula. Suppose two GPS data strings are

$$S_1 = \{g_1^1, g_2^1, \dots, g_n^1\} \text{ and } S_2 = \{g_1^2, g_2^2, \dots, g_m^2\}$$

The best match of S_1 in terms of S_2 is

$$\hat{S}_1 = \{\hat{g}_1^1, \hat{g}_2^1, \dots, \hat{g}_n^1\}, \text{ where } \hat{g}_i^1 = \arg \min_{g_j^2} (\|g_j^2 - g_i^1\|).$$

Similarly, $\hat{S}_2 = \{\hat{g}_1^2, \hat{g}_2^2, \dots, \hat{g}_m^2\}$ is the best match of S_1 in terms of S_2 , where $\hat{g}_j^2 = \arg \min_{g_i^1} (\|g_i^1 - g_j^2\|)$.

Two GPS strings are declared to belong to the same route if

$$\max(\|g_i^1 - \hat{g}_i^1\|) < d_1 \text{ and } \max(\|g_j^2 - \hat{g}_j^2\|) < d_2,$$

where d_1 and d_2 are the largest sampling intervals of R1 and R2, respectively.

With the adaptive location clustering, the significant locations are gradually discovered and updated. Once an important location pair is identified, the commute routes between them are recorded and learned. Using the preceding comparison process, unique routes are identified and saved in the database. The unique routes are non-directional. That is to say, the same route may be traveled from location A to location B, or from location B to location A. Frequencies of the unique routes are tallied in an hourly basis. After two month learning, 47 unique routes are obtained for one user. The route frequency is derived by considering all different routes starting from Location A to Location B. Table 1 presents the frequency of routes from “home” to “office” for one user. Note that routes 4, 8, and 26 are the most frequently traveled routes.

Table 1. Frequency of travel routes from home to office

Route ID	4	8	25	26	27	28	29	30
Frequency	0.6	0.1429	0.0286	0.1143	0.0286	0.0286	0.0286	0.0286

5. ROUTE PREDICTION-BASED TRAFFIC ADVISORY SYSTEM

Once routes are predicted, a traffic advisory system can be built. We designed the system to ensure that the user experience was unobtrusive and well accepted by the users. Ideally, the system is designed to assist users during transition times – as the user is transitioning out of one location and likely into a vehicle environment. Previous research has shown that messages from context aware devices are better received during transitions [2], and in our case it is during transitions that the information provided by the application is of the greatest relevance.

The application is designed to automatically detect when the user is leaving a location and predict to where they are likely headed, and the most likely routes the user will take. When this occurs, a pop-up window will appear on the visual display informing the user that the application has detected the transition and is searching for traffic information. This will usually occur before driving. Soft keys will be allocated to allow the user to edit or cancel this function. No audio or haptic feedback will be provided to alert the user in case the user is not interacting with the mobile device during this time. To adopt a user-centric approach and minimize the obtrusiveness, traffic history may be studied to derive traffic averages and variances and traffic alerts are only provided when current or predicted traffic flow exceeds the average by a certain margin. If a traffic alert is detected that is out of the ordinary for that route and time of day, then the user will receive an audio or haptic alert, depending upon the phone’s current profile settings. In addition, a pop up window such as

Figure 3 will appear alerting the user that there is an alert, and offering the user the choice of ignoring the alert or obtaining more information. If the user selects the latter, more detailed information on the alert (e.g. type of incident and location) will be provided with an option for the application to recommend a new route to the predicted location. Because of driver distraction concerns, alerts are issued only when it is necessary to avoid traffic delays.



Figure 3. Screen shots of the application. The left side depicts a textual pop-up alert, and the right side a map view of the alert.

The developed prototype is a client-server system. The client software is running on a cell phone. The client collects GPS data and cell IDs and learns semantic locations and routes between them with our adaptive algorithm. An easy-to-use user interface is developed to alert users about heavy traffic situations. The server side includes a set of machines to provide map, real-time traffic, and other information to clients based on Java EE. The communication between clients and the server side is based on standard HTTP protocol.

All specific aspects of the implementation of the design of the traffic advisory system, as well as the overall acceptance of the application are validated by real users from specific demographic areas of interest in a large-scale field evaluation of the application.

6. CONCLUSION

Context knowledge such as the user's needs, locations, and travel routes is a major enabler for providing the user with unobtrusive assistance for mobile device. In this paper, the focus has been on a location-based traffic advisory system that is based on learning and predicting the user's patterns using data available from the mobile devices. We used a data-driven approach in the

development of the traffic advisory system. We collected usage data from multiple users on phones equipped with sensors over an extended period of time, mined the data to identify various lifestyle patterns across different forms of context, and attempted to predict the patterns. The predicted patterns were used to design applications that satisfy user needs.

We have demonstrated a successful implementation of a traffic condition assistant that learns user's important locations, routes between those, and learns to predict those routes. In order to maximize the system effectiveness and user acceptance, advisories are provided to the user only when there is a real need (e.g., a traffic problem on the predicted route). We envision that context aware features like this including a user modeling component will become essential in supporting new applications and concepts that connect consumers with media and services through multiple devices in the office, car, or home.

7. REFERENCES

- [1]. Ester, M. et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. KDD, 1996.
- [2]. Ho, J., Intille, S. Using Context-Aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. CHI 2005. April 2-7, 2005. Portland, OR.
- [3]. Krumm, J., Horvitz, E., Predestination: Inferring Destinations from Partial Trajectories. UbiComp 2006: The Eighth International Conference on Ubiquitous Computing, September 17-21 (2006), Orange County, CA.
- [4]. Letchner, J., Krumm, J., Horvitz, E. Trip Router with Individualized Preferences (TRIP): Incorporating Personalization into Route Planning. Eighteenth Conference on Innovative Applications of Artificial Intelligence (IAAI-06), July 2006.
- [5]. Priebe, C. E. Adaptive mixtures. *Journal of American Statistical Association*, 1994, pp. 796-806.
- [6]. Raento, M., Oulasvirta, A., Petit, R., Toivonen, H., ContextPhone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing* 4:51-59, 2005.
- [7]. Rukzio, E., Hamard, J., Noda, C., De Luca, A. Visualization of uncertainty in context aware mobile applications. MobileHCI'06, September 12-15, 2006, Helsinki, Finland.
- [8]. Titterton, D. M. Recursive parameter estimation using incomplete data. *Journal of Royal Statistical Society*, 1984, pp. 257-267.