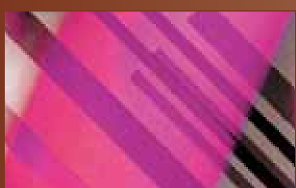
Predestination: Where Do You Want to Go Today?

John Krumm and Eric Horvitz
Microsoft Research



Predicting driver destinations could help target location-based services.

Mobile computers have the potential to provide a new array of location-based services to drivers as they travel around. However, naïve implementations of vehicle-based m-commerce applications could be more frustrating than helpful. For example, while people might welcome information about available discount parking several minutes before arriving at a crowded airport or urban center, they would not likely want to hear about a sale on bananas as they drive by a grocery store at 40 mph.

An automobile's navigation system could answer the question, "Where do you want to go today?" by using accurate predictions of driver destinations to deliver anticipatory notifications about traffic jams, alternate routes, and interesting sights. In addition, location-specific services could be better targeted to drivers, highlighting relevant geocentric opportunities as well as suppressing irrelevant marketing, much like an antispam filter.

The Predestination algorithm predicts a driver's destination based on both general trends, such as the likelihood that people will choose various types of destinations, and personalized data such as a list of previously visited locations (http://research.microsoft.com/users/jckrumm/predestination.pdf). We designed the algorithm to run on a vehicle's navigation system and learn a driver's habits based on logged GPS measurements.

Moving beyond classic machine-learning techniques, we added an *open-world modeling* component. Designed to boost prediction accuracy in the algorithm's early "out of the box" phase when little or no individual training data is available, the component also extends to longer-term usage. In the open-world model, the algorithm can predict destinations that the driver has not yet visited.

WHERE DO PEOPLE DRIVE?

In early 2004, we initiated the Microsoft Multiperson Location Survey (MSMLS) to collect GPS measurements from volunteer drivers with the goal of developing a corpus of driving data to support multiple location-centric research projects. Each MSMLS participant receives a GPS receiver that can record up to 10,000 GPS waypoints. The receivers are configured to be "hands off;" drivers in the study simply leave the unit unattended on their car's dashboard for two to four weeks.

To date, we have collected data from about 200 drivers, mostly in the Seattle, Washington, area, covering about 135,000 kilometers of driving in about 11,000 discrete trips. Each of these trips represents one destination that we use to train and test Predestination.

WHERE DO YOU GO?

Predestination represents space as a simple grid of 1-km squares, as Figure 1 shows. Square outlines indicate the relative popularity of destinations, with darker outlines representing higher destination probabilities—thus coastal and rural areas are clearly not as popular as urban and residential areas.

As a trip progresses, the algorithm computes a probability for each cell of the grid that the driver will end the trip there. Using Bayesian inference, it combines four different sources of evidence about the driver's behavior: ground cover, driving efficiency, driving time, and personal destinations. The algorithm recomputes all the probabilities each time the driver enters a new cell.

Computing probabilities, rather than a single-most-likely cell, exposes the inherent uncertainty in prediction, which applications can exploit in decision making.

Ground cover

The US Geological Survey categorizes the ground cover of each 30m × 30m square of the country into one of 21 different categories. In considering the destinations of our drivers, we found that the two most likely destination labels were, by far, "commercial" and "low-intensity residential." Among the least popular were "bare rock," "perennial ice," and "emergent herbaceous wetlands." Computing probabilities of destina-

Figure 1. Grid of 1-km squares showing the relative popularity of destinations in the Seattle, Washington, area. Darker outlines represent higher destination probabilities.

tion by type of ground cover yields the grid shown in Figure 1.

Driving efficiency

We have found that drivers are not optimally efficient in getting where they want to go, though efficiency tends to increase as they near their destination. Predestination captures this tendency by computing how often drivers transition to a grid cell that is actually farther away in time from their ultimate destination based on driving-time estimates from a map-based route planner.

As Figure 2 shows, destination probabilities start out uniform and then gradually drain away from regions where the driver is clearly not heading. The algorithm's driving-efficiency element updates destination probabilities by increasing the likelihood of those cells to which the observed path is a somewhat efficient route.

Driving time

According to the 2001 US National Household Travel Survey (http://nhts.ornl.gov/2001/index.shtml), the average car trip is 14 minutes long. Working from an NHTS distribution of trip times, along with estimated driving times between cells in our grid, Predestination computes the probability of each cell based on how long it would take to drive there. This tends to bound predictions to those relatively close to the starting point.

Personal destinations

Drivers tend to revisit places they've been before. In fact, after logging destinations for two weeks, we found that the probability of a driver's trip terminating in a previously unvisited cell within the grid was only about 9 percent. The rate of a driver visiting a previously unobserved destination decays roughly exponentially with observation time, converging to an equilibrium rate of visiting new destinations. Interestingly, the rate that women visit previously unobserved destinations tends to decay slightly faster than the rate for men.

Our analysis also showed that destinations are often clustered, with drivers choosing destinations near each other, likely for reasons of efficiency and familiarity.

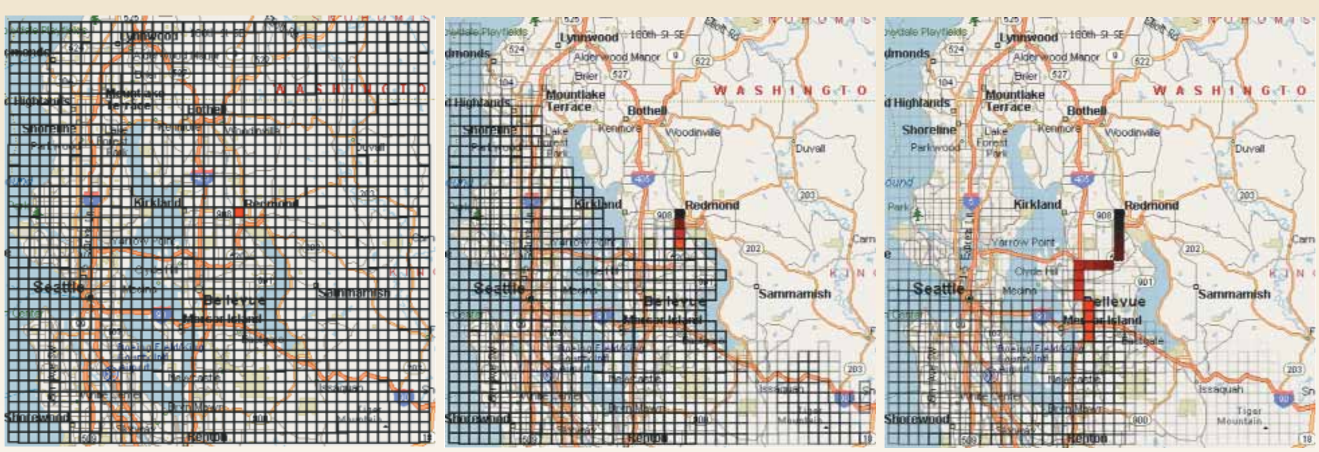


Figure 2. Predestination's driving efficiency element. The driver starts in the red cell on the left and leaves a path of filled cells as the trip proceeds. The algorithm updates destination probabilities by increasing the likelihood of those cells to which the observed path is a somewhat efficient route.

Predestination's "closed world" component considers drivers' habitual return to previously observed destinations, while the "open world" component calculates the learned likelihood that drivers will visit a previously unlogged destination. The algorithm folds the two analyses together to compute probability distributions over destinations.

The open-world modeling methodology has several useful properties, including the provision of a smooth transition over time between untrained, out-of-the-box behavior and more fully trained customization for an individual driver.

We trained Predestination on half of our data and tested it on the other half. Taking the predicted destination as the highest-probability cell, we found, unsurprisingly, that the algorithm's prediction accuracy improves as drivers progress on their trip. Median error begins at about 3 km, drops to around 2 km at the halfway point, and goes down to 1 kilometer at the end of the trip—it is not zero because Predestination doesn't know when the trip ends.

We imagine Predestination as a future component of vehicle navigation systems, which are already equipped with the necessary GPS, CPU, and map data. Predictions about destination can be an important component of ubiquitous computing, and we hope that such predictions might one day enhance the provision of information and services to people as they move through the world. ■

John Krumm is a researcher in the Adaptive Systems and Interaction Group at Microsoft Research. Contact him at jckrumm@microsoft.com.

Eric Horvitz is a principal researcher and research area manager in the Adaptive Systems and Interaction Group at Microsoft Research. Contact him at horvitz@microsoft.com.

Editor: Bill N. Schilit, Google; bill.schilit@computer.org, http://schilit.googlepages.com