

# Predestination: Inferring Destinations from Partial Trajectories

John Krumm and Eric Horvitz

Microsoft Research  
Microsoft Corporation  
One Microsoft Way  
Redmond, WA USA 98052  
{jckrumm, horvitz}@microsoft.com

**Abstract.** We describe a method called *Predestination* that uses a history of a driver's destinations, along with data about driving behaviors, to predict where a driver is going as a trip progresses. Driving behaviors include types of destinations, driving efficiency, and trip times. Beyond considering previously visited destinations, *Predestination* leverages an *open-world* modeling methodology that considers the likelihood of users visiting previously unobserved locations based on trends in the data and on the background properties of locations. This allows our algorithm to smoothly transition between "out of the box" with no training data to more fully trained with increasing numbers of observations. Multiple components of the analysis are fused via Bayesian inference to produce a probabilistic map of destinations. Our algorithm was trained and tested on hold-out data drawn from a database of GPS driving data gathered from 169 different subjects who drove 7,335 different trips.

## 1 Introduction

Location has played a central role in ubiquitous computing research. Information about the location of users can enable numerous compelling location-based services. For example, location can be used to fetch relevant information such as nearby points of interest and available services. Beyond current location, services can be developed around predictions about future locations. For example, a driver may want to know about restaurants or traffic problems before encountering them to give time to prepare and make decisions. Location-based services could present their availability in anticipation of a user's arrival. In another application, a prediction of a person's destination can be helpful in deciding if the person is deviating from an intended route [2]. Cheng *et al.*[3] even speculate that destination prediction could be used to catch automobile thieves.

We present a methodology named *Predestination* that is aimed at predicting a driver's destination as a trip progresses. The probabilistic prediction is based on several sources of data, including the driver's history of destinations and an ensemble of trips from a group of drivers. We demonstrate how to combine these data sources in a

principled way to reason about drivers' ultimate destinations, resulting in inferred probability distributions over a geographic region. We trained and tested our algorithm on GPS data from 169 different drivers who participated in a data-collection effort that we call the *Microsoft Multiperson Location Survey* (MSMLS)[4].

## 2 Related Research

Previous work in predicting users' locations includes an application designed for a pool of shared vehicles. Karbassi and Barth[5] process historical GPS data from the vehicles to extract the most common routes between five pre-designated locations. Given the destination, their goal is to predict the route in order to estimate arrival times. In contrast, Predestination is designed to predict the destination, not necessarily the route. And, our algorithm works on a fully tiled geographic region, not just a handful of discrete locations.

Systems for predicting locations can be used to make smooth handoffs between wireless communication cells, like Wi-Fi and cellular telephones. Cheng *et al.*[3] review the work in this area. The defining characteristics are that locations are represented as antenna cells and that prediction is based on past behavior.

Like ours, much of the previous work on destination prediction is based on GPS sensing. In their *comMotion* work, Marmasse and Schmandt[6] predict a person's destination from a list of previously visited destinations using a Bayes classifier, histogram matching, and an HMM. Ashbrook and Starner[7] find potential destinations by clustering GPS data, then predict destinations from these candidates based on Markov models trained to find the next most likely destination based on the one(s) that were recently visited. In Project Lachesis, Hariharan and Toyama[8] present a location clustering algorithm that is sensitive to scale in both space and time. They model transitions between clustered locations with a Markov model. Liao *et al.*[9] present a hierarchical dynamic Bayesian network to predict destinations, which is shown to outperform a 2<sup>nd</sup>-order Markov model. Using a dynamic network model, Gogate *et al.*[10] incorporate time-of-day and day-of-week evidence to predict a driver's route and destination. All of this work shares the trait that candidate destinations are extracted from GPS histories, *i.e.* places that subjects have actually visited.

Although previously visited locations are one component of our prediction algorithm as in the work above, we also predict destinations that the user has not necessarily ever visited before. We do this with an examination of ground cover (*e.g.* middles of lakes are unpopular driving destinations), the fact that drivers attempt to take efficient routes, and a distribution of likely trip times. In Predestination, all this information is combined with Bayes formula to give a probability distribution of destinations over a geographic area. We also introduce the concept of an *open-world* model that addresses the incompleteness of our models in an explicit manner. With the open-world method, we acknowledge and model the possibility that behaviors observed in the future may not be represented in the current dataset. In the case of Predestination, drivers may go to destinations that have not been previously recorded in the training data. The open-world model allows us to reason about new destinations seen as



training progresses so as to capture a user's general patterns of trips and destinations, as well for capturing the ongoing background exploration of new locations that people may perform on an ongoing basis.

### 3 MSMLS: Multiperson Location Survey

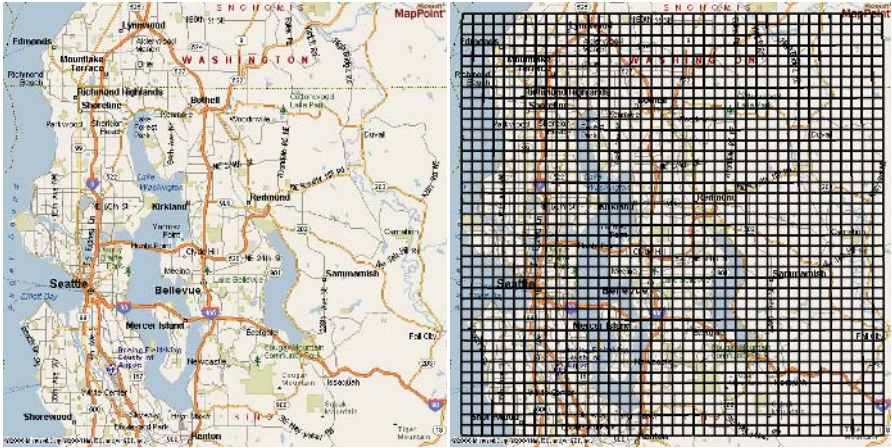
We trained and tested our destination prediction algorithm on driving trip data gathered from 169 subjects[4]. These subjects volunteered to place one of our 55 GPS receivers in their car for two weeks (and occasionally longer) as they drove normally. Nearly all the subjects live in the Seattle, WA USA area, and they include employees of our institution and their family members. The GPS receivers were Geko 201 models, capable of recording up to 10,000 time-stamped (latitude, longitude) coordinates. Each subject was given a cable to supply GPS power from the car's cigarette lighter. Using a simple hardware modification, we altered the GPS receivers so they would automatically turn on whenever power was supplied. This meant that the drivers did not have to remember to turn the receivers on or off, and could instead just set the receiver on the dashboard and neglect it for the entire two weeks. Because some cars' cigarette lighters are powered even when the car is off, we used a mode on the GPS receivers that only recorded points when the receiver is in motion, eliminating the accumulation of points when the cars were parked.

We gathered a total of 1,228,237 (latitude, longitude) points for an average of 6,267 points per person. The points were separated by a median distance of 63 meters and 6 seconds. We also gathered demographic data from each subject: 75% were male, 71% had a domestic partner, 48% had children, and the average age of drivers was 36.

As the goal of this research is to predict destinations, we segmented our GPS data into discrete trips. We identified trips by looking for places in the sequence that met either of the following criteria:

- Gap of at least five minutes: A five-minute gap indicates that the GPS was not moving and, because of its adaptive recording mode, not recording new points. Such a gap can also come from vehicles where power to the device turns off with the car.
- At least five minutes of speeds below two miles per hour: Identifying such a period of low speed accounts for the fact that, even when parked, GPS noise can make it appear that the vehicle is moving slightly. We took measures of five minutes or more of extremely slow apparent motion to be a split between trips.

We deleted discrete trips whose maximum speed did not exceed 25 miles per hour in an effort to eliminate pedestrian and bicycle trips, because some of our study participants took their devices out of their cars. We also eliminated trips of less than one kilometer or with less than 10 GPS points. After this segmentation and culling, we had 7,335 discrete trips. We found that the average length of these trips is 14.4 minutes, and that subjects took an average of 3.3 trips per day.



**Fig. 1.** We used a grid over the Seattle, WA area as the basis for modeling driver behavior and destination predictions. The left map shows the area without the grid. The cells are squares, one square kilometer in size.

## 4 Destination Probabilities

This section explains the probabilistic analysis of drivers' destinations. We first describe the spatial representation and the underlying probabilistic model for the Pre-destination method. Then we describe the four constituent components of the probabilistic analysis. Finally, we describe the probabilistic integration of the sources of information to produce a posterior probability distribution over destinations.

### 4.1 Probabilistic Grid on Map

We represent space as a 40x40, two-dimensional grid of square cells, each cell 1 kilometer on a side, as shown in Figure 1. Each cell represents one discrete location. Our destination prediction is aimed at picking the cell in which a driver will conclude his or her trip based on which cells the driver has already traversed and on the characteristics of each cell. This particular discretization of space is a heuristic choice, and we could have chosen a different tiling, size, and number of discrete cells. Each of the  $N = 1600$  cells is given an index  $i = 1, 2, 3, \dots, N$ .

Because our methods are probabilistic, we ultimately compute the probability of each cell being the destination, *i.e.*  $P(D = i | \mathbf{X} = \mathbf{x})$ , where  $D$  is a random variable representing the destination, and  $\mathbf{X}$  is a random variable representing the vector of observed features from the trip so far. While we shall focus on trajectory-centric observations, other factors can be included in  $\mathbf{X}$ , such as time of day and day of week.

We decompose the inference about location into the prior probability and the likelihood of seeing data given that each cell is the destination. Applying Bayes rule gives

$$P(D = i | \mathbf{X} = \mathbf{x}) = \frac{P(\mathbf{X} = \mathbf{x} | D = i)P(D = i)}{\sum_{j=1}^N P(\mathbf{X} = \mathbf{x} | D = j)P(D = j)} \quad (1)$$

Here  $P(D = i)$  is the prior probability of the destination being cell  $i$ . We shall compute the prior with two sources of map information, detailed in Sections 4.2 and 4.3.  $P(\mathbf{X} = \mathbf{x} | D = i)$  is the likelihood of cell  $i$  being the destination based on the observed measurements  $\mathbf{X}$ . We compute this with two other sources of map information, detailed in Sections 4.4 and 4.5 below. The denominator is a normalization factor computed from the collected data.

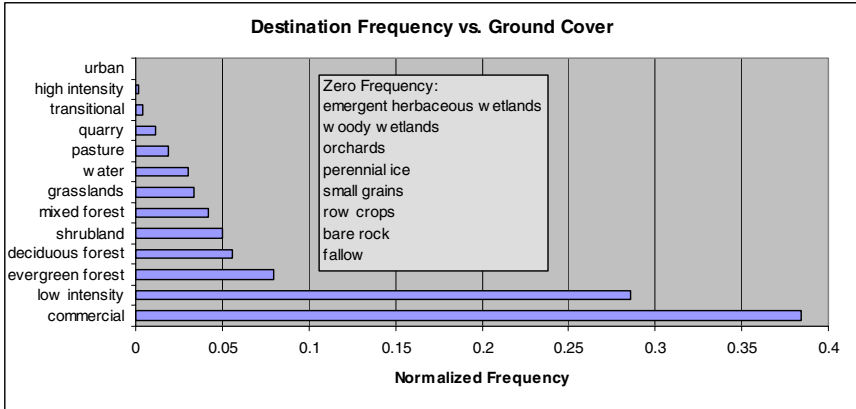
## 4.2 Ground Cover Prior

We can use ground cover information as one source of information about the probability of destinations. For example, we can assert that the middles of lakes and oceans are unlikely destinations for drivers, and that commercial areas are more attractive destinations than places that are perennially covered with ice and snow. In order to verify these suspicions and use them for destination prediction, we characterized each cell in our grid based on a United States Geological Survey (USGS) ground cover map, available for free download [11]. These maps categorize each 30m x 30m square of the U.S. into 1 of 21 different types of ground cover whose types are given in Figure 2.

By looking at the (latitude, longitude) of each trip destination in our dataset, we created a normalized histogram over the 21 ground cover types, shown in Figure 2. As expected, water is an unpopular destination, although still more popular than some other categories, and commercial areas are more attractive than those covered with ice and snow. The two most popular destinations are “commercial” and “low intensity residential”, which the USGS describes as:

- Commercial/Industrial/Transportation – “Includes infrastructure (*e.g.* roads, railroads, etc.) and all highly developed areas not classified as High Intensity Residential.”[12]
- Low Intensity Residential – “Includes areas with a mixture of constructed materials and vegetation. Constructed materials account for 30-80 percent of the cover. Vegetation may account for 20 to 70 percent of the cover. These areas most commonly include single-family housing units. Population densities will be lower than in high intensity residential areas.”[12]

The “water” category was nonzero likely because a 30m x 30m USGS square is categorized as water even if it is up to 25% dry land, which could include beaches and waterfront property depending on how the squares are placed. We expect this distribution to be not generally applicable outside the immediate area of our testing, because different regions will have different mixes of ground cover and its people will have possibly different behaviors.



**Fig. 2.** Destination probabilities vary with the type of ground cover at the destination

The normalized histogram in Figure 2 represents  $P(D = i | G = j)$ , which gives the probability of a destination cell if it were completely covered by ground cover type  $j$  for  $j = 1, 2, 3, \dots, 21$ . In reality, our 1 km x 1 km cells each contain about 1,111 30m x 30m ground cover labels, and they are usually not completely covered by the same type. To account for this, we compute the distribution of ground cover types for each cell, calling it  $P_i(G = j)$ . We compute the probability of each cell being a destination, based only on the ground cover information, by marginalizing the ground cover types in the cell:

$$P_G(D = i) = \sum_{j=1}^{21} P(D = i, G = j) = \sum_{j=1}^{21} P(D = i | G = j) P_i(G = j) \quad (2)$$

$P_G(D = i)$  is the probability of a destination cell based on ground cover. A plot of this likelihood on a map is shown in Figure 3(a), which shows that water and more rural areas are lower-probability destinations.

### 4.3 Personal Destinations and Open World Modeling

We now turn to the incorporation of more informative probabilistic information based on the prior history of driver's destinations. We build on the intuition that drivers often go to places they have been before, and that such places should be given a higher destination probability. This is the main principle behind much previous work in pervasive computing on modeling and predicting transportation routines. For instance, Murmasse and Schmandt[6] used the loss of a GPS signal to indicate that a user had entered a building. If the user enters the same building a number of times, that location is marked as candidate destination for future prediction. Ashbrook and Starner[7] cluster GPS-measured locations where a user spent more than 10 minutes to extract likely

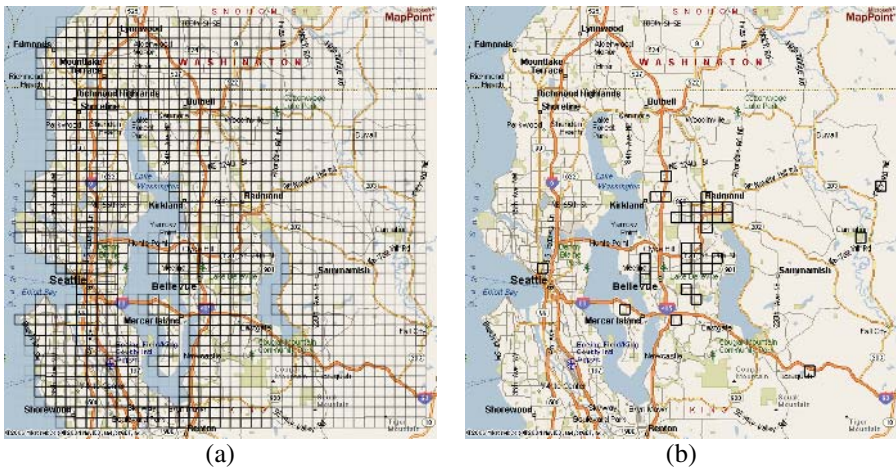
destinations. In their work on learning and modeling transportation routines, Liao *et al.*[9] extracted destinations by clustering locations of long stays. In Project Lachesis, Hariharan and Toyama[8] infer potential destinations in a similar way, but explicitly account for variations in scale of a destination’s size and duration of stay.

We model personal destinations as the grid cells containing endpoints of segmented trips. As such, the spatial scale of a candidate destination is the same as a cell’s size, and the required stay time to be considered a destination is determined by our trip segmentation parameter, which is currently five minutes.

### 4.3.1 Closed-World Assumption

We first consider the case where drivers only visit destinations that they have been observed to visit in the past. We refer to this assumption as the *closed-world* assumption, and to corresponding analyses as closed-world analyses. Much of the prior research on predicting a person’s location based on GPS data, including that described above, centers on closed-world analyses.

Making a closed-world assumption, we examine all the points at which the driver’s trip concluded and make a histogram over the  $N$  cells. Normalizing gives a probability mass function  $P_{closed}(D=i), i=1,2,3,\dots,N$ , where the “closed” subscript indicates that this probability is based on personal destinations. Figure 3(b) shows the cells with nonzero  $P_{closed}(D=i)$  for one driver in our study. If a user has never visited a cell, the personal destinations probability for that cell will be zero. This is because, as explained in Section 4.6, this probability will be multiplied by other probabilities over the  $N$  cells in the Bayesian calculation to compute the posterior destination probability for each cell. If any cell has a zero prior, that cell will not survive as a possible destination.



**Fig. 3.** (a) Probabilities based solely on ground cover information, with darker outlines showing cells of higher probability of being destinations. Water and rural regions are less probable destinations. (b) The destination cells recorded for one driver in the MSMLS study.



### 4.3.2 Open-World Analysis

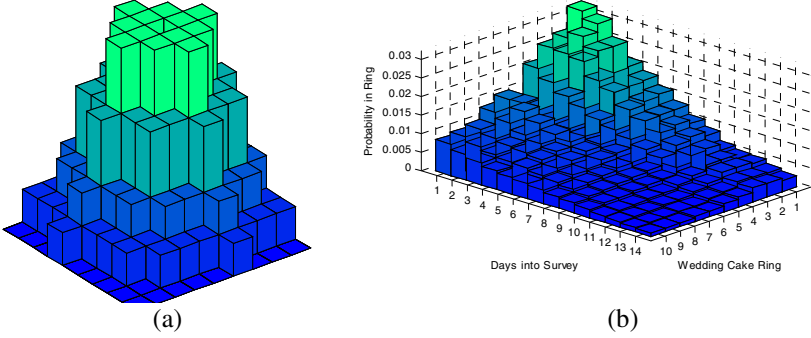
The closed-world assumption is naïve in that people actually can visit locations they have never been observed to visit. This is the case in general, but such observations of new destinations are especially salient in the early phases of observing a driver. On the latter, “new” locations include places a driver has visited before, but had not been observed to visit during the course of a study, as well as genuinely new destinations for that driver. Thus, a more accurate approach to inferring the probability of a driver’s destinations would consider the likelihood of seeing destinations that had not been seen before, thus leveraging an “open-world” model. If we can correctly model this effect, we can transform a closed-world probability mass function taken at an early point in the survey into an approximation of the steady state probability that we would have observed at the end of the survey and beyond. This open world model then replaces  $P_{closed}(D=i)$ , and we have a more accurate model of the places a subject tends to visit.

Focusing on open-world modeling, we model unvisited locations in two ways. The first is based on our observation that destinations tend to cluster, as seen in the example in Figure 3(b). Our intuition is that drivers tend to go to places near each other to save time, or to overall regions they are familiar with, *e.g.* drivers might chose gas stations and grocery stores that are near their place of work. We modeled this effect as a discretized probability distribution over the distance from previously visited points. This distribution has the overall shape of a tiered “wedding cake” as in Figure 4(a). Each tier gives the probabilities of new destinations around previously visited ones. Each tier of the wedding cake is a concentric ring of constant probability at some radius from center, and it is intended to model the eventual clustering of destinations in the steady state.

We measured this clustering tendency by looking at the normalized histograms of destinations on our grid over the days of each subject’s GPS survey. For each destination on a given day, we computed the probability that an as-yet-unvisited destination would appear in the eventual steady state for each ring of a 10-tier wedding cake around that destination. Each tier is a ring of width one kilometer and a center radius of  $r = \{1, 2, \dots, 10\}$  kilometers, and the steady state was taken from all the destinations visited over the whole survey. The results are shown in Figure 4(b) for days 1-14 of the survey. On day 1 of the survey, the probabilities of finding unvisited steady-state destinations near already-visited destinations are relatively high. As the days go on, each subject gradually visits most of their usual destinations, so the probabilities drop. For each day, tiers near the center are higher than near the outer edge. Operationally, for a given closed-world probability  $P_{closed}(D=i)$  from a given day, we compute another probability with the unvisited neighbors of each nonzero  $P_{closed}(D=i)$  replaced by a wedding cake with probability values for the appropriate day taken from Figure 4(b). This simulates the spread we expect to see in the steady state. After normalizing to one, we refer to the wedding cakes as  $W(D=i)$ . This is done separately for each subject

Although the steady state destinations tend to cluster, isolated destinations also occur. We characterized this effect by computing the probability that a steady state destination would *not* be covered by a 10-tier wedding cake around a destination visited





**Fig. 4.** (a) A 4-tier probability distribution, showing the discretization over four threshold radii from a previously visited location. We used a 10-tier version to model the clustering of destinations. (b) Probability distribution showing allocation of probability to each tier with each passing day. These probabilities were computed from our GPS data.

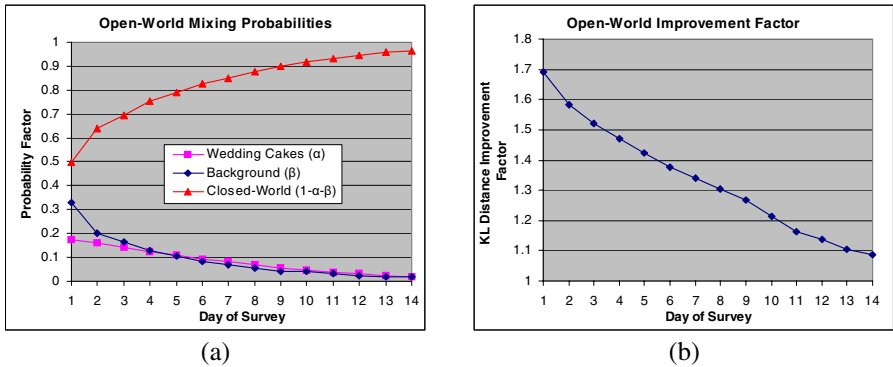
before steady state. This probability is shown as  $\beta$  in Figure 5(a). As expected, the probability of new, isolated destinations drops with time. One way to model this background probability is with a uniform distribution over all grid cells. However, this would have the undesirable effect of contributing probability to places where no one goes, like middles of lakes. Instead of a uniform distribution, we take the background as  $P_G(D=i)$ , which is the ground cover prior as described previously in Section 4.2.

We combine these effects to compute a probability distribution of destinations that more accurately models the steady-state probability. The three components are the closed-world prior  $P_{closed}(D=i)$ , the parameterized spread as represented by the wedding-cake shaped distributions  $W(D=i)$  described above, and the background probability  $P_G(D=i)$  to model isolated destinations. We apportion a fraction  $\alpha$  of the total probability to  $W(D=i)$ , where  $\alpha$  is the sum of the tiers from Figure 4(b) for the appropriate day, shown in Figure 5(a). A learned fraction  $\beta$  of the probability, capturing the probability that users will travel to places beyond the tiered distributions is allocated to the background, with  $\beta$  also shown in Figure 5(a). The open-world version for the probability of a driver's destinations, which we shall take as a prior probability for our analysis, is then computed as

$$P_{open}(D=i) = (1 - \alpha - \beta)P_{closed}(D=i) + \alpha W(D=i) + \beta P_G(D=i) \quad (3)$$

We refer to this as the *open-world prior probability distribution*.

Figure 5(a) shows the behavior of  $\alpha$ ,  $\beta$ , and  $1 - \alpha - \beta$  with time. As time goes on,  $\alpha$  and  $\beta$  tend to decrease, deemphasizing the adjustment for clustering and



**Fig. 5.** (a) The mixing of probabilities for computing the open-world prior. As time goes on, the prior emphasizes the subject's previous destinations more strongly. (b) The multiplicative improvement in KL distance between the closed-world and open-world priors when compared to the true steady state.

background probability in favor of each subject's actual learned destinations. This represents the richness of an open-world model that takes into appropriate consideration the fact that people can visit new locations, especially early on in the observation period, but also in the long run.

The open-world prior probability distribution,  $P_{open}(D=i)$ , is designed to approximate a subject's steady state distribution of destinations better than the naïve, closed-world prior  $P_{closed}(D=i)$ . To test this, we computed the Kullback Leibler (KL) distance<sup>1</sup> between both of these models and the actual steady state prior for each subject. The steady state prior is simply the closed-world prior computed with each subject's entire survey data. Figure 5(b) shows the improvement factor as a function of days into the survey. At day one, the KL distance between the naïve, closed-world prior and the actual steady state is about 1.7 times as great as the distance between the open-world prior and the steady state. This factor decreases with time as the naïve prior approaches the steady state prior. The advantage of the open-world prior is that the system works with a prior much closer to the actual steady state than with the closed-world model.

Equation (3) is the prior probability that we use in Bayes formula in Equation (1). The next two subsections discuss the two likelihoods of the form  $P(\mathbf{X}=\mathbf{x}|D=i)$ , where  $\mathbf{x}$  is some measured feature of the current drive, that we use to model other sources of information for destination prediction.

<sup>1</sup> The KL distance between a true distribution  $p(x)$  and an approximate distribution  $q(x)$  is  $\sum p(x) \log_2[p(x)/q(x)]$ . If the approximate distribution is zero anywhere, this results in a division by zero. The closed-world prior often has zeros, which we accounted for by modifying it in the standard way:  $q'(x) = (1-\lambda)q(x) + \lambda/N$  with  $\lambda = 0.01$ .

#### 4.4 Efficient Driving Likelihood

Drivers tend to take purposeful and somewhat efficient routes to their destinations[13]. Intuitively, if a driver appears to be taking a very inefficient route to a candidate destination, then we can reason that that destination is unlikely. The efficient driving parameter is intended to capture this behavior in order to help narrow the set of likely destinations. We developed a different destination prediction algorithm using only efficient driving in [14].

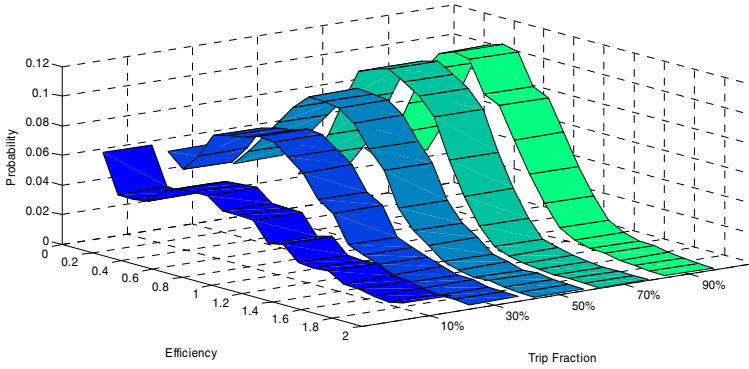
We quantify efficiency using the driving time between points on the driver's path and candidate destinations. Thus, for each pair of cells  $(i, j)$  in our grid, we estimate the driving time  $T_{i,j}$  between them. A first approximation to the driving time could come from a simple Euclidian distance and speed approximation between each pair of cells. Instead, we used Microsoft MapPoint desktop mapping software to plan a driving route that MapPoint considers to be ideal between the center (latitude, longitude) points of all pairs of cells. MapPoint provides a programmatic interface that returns the estimated driving time of planned routes. Using a driving route planner takes into account the road network and speed limits between cells, giving a more accurate driving time estimate.

For  $N$  cells, there are  $N(N-1)$  different ordered pairs, not including pairs of identical cells. Our route planning software plans routes at the rate of about four per second on a 2.8 GHz PC, meaning it would take about 7.4 days to plan routes for all  $N(N-1) \approx 2.6 \times 10^6$  pairs. We cut this time in half by assuming that the travel time from cell  $i$  to  $j$  is the same as from cell  $j$  to  $i$ , *i.e.*  $T_{i,j} = T_{j,i}$ . The computation time for route planning was the main barrier to increasing the resolution of our grid. Fortunately, this computation must be done only once for the grid.

We measure efficiency based on the trip's starting cell  $s$  and a candidate destination cell  $i$ . If the driver's route is efficient, then the total time required to go between these two cell should be about  $T_{s,i}$ . If the driver is currently at cell  $j$ , then the time to reach the candidate destination  $i$  should be about  $T_{j,i}$ . If  $i$  really is the destination, and if the driver is following an efficient route, then the driver should have taken a time of  $T_{s,i} - T_{j,i}$  to reach the current cell  $j$ . We know that the driver's actual trip time to this point is  $\Delta t$ , which will be longer than  $T_{s,i} - T_{j,i}$  if the driver is taking an inefficient route. Thus our measure of efficiency is the ratio of how much time the driver should have spent moving toward the candidate destination divided by how much time has actually transpired:

$$e_i = \frac{T_{s,i} - T_{j,i}}{\Delta t} \quad (4)$$

We expect this to be about one for an efficient trip between  $s$  and  $i$ . Using our GPS survey data, we computed the distributions of efficiency values based on known trips and their corresponding destinations. The efficiency likelihood  $P_E(E = e | D = i)$



**Fig. 6.** These curves show the measured distributions of driving efficiency, which vary with the amount of the trip completed

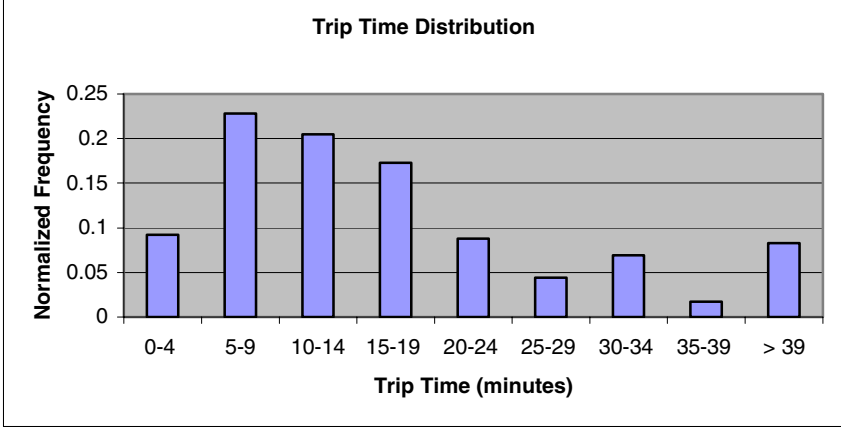
represents the efficiency that drivers actually produce on their way to a destination. If a candidate destination results in a low-likelihood efficiency, its posterior probability will be corresponding low when  $P_E(E = e|D = i)$  is incorporated in Bayes rule. As a function of the fraction of the trip, the efficiency likelihood is shown in Figure 6. The distribution near the beginning of the trip is unrealistic likely due to MapPoint’s inability to give accurate travel times for short trips. For all trip fractions, some drivers are able to boost their efficiency beyond 1.0, either due to speeding or mistakes in the ideal trip time estimates from MapPoint. The effect of using this likelihood for destination prediction is that if a driver appears to be driving away from a candidate destination, that destination’s probability will be lowered.

#### 4.5 Trip Time Distribution

The final component we use for destination prediction is a distribution of trip times. Intuitively, we know that most car trips are measured in minutes, not hours, which limits the range of likely destinations. To quantify this intuition, we used data from the U.S. 2001 National Household Transportation Survey (NHTS)[1]. The NHTS collected data on daily and longer-distance travel from approximately 66,000 U.S. households based on travel diaries kept by participants. The survey results are available via a Web interface[15], from which we created a histogram of trip times, a normalized version of which is shown in Figure 7.<sup>2</sup>

The likelihood governing trip times is  $P_{\Delta T}(\Delta T = \Delta t|D = i)$ , where  $\Delta T$  is the random variable representing the trip time so far. For use of this likelihood, we quantize our trip times according to the bins in Figure 7. Figure 7 actually represents the

<sup>2</sup> Specifically, from <https://nhts.ornl.gov/2001/Login.do>, we created a table whose analysis variable was “Annual vehicle trips (VT)” and whose row variable was “Calculated Time to complete trip (min.) (TRVLCMIN)”.



**Fig. 7.** This is a distribution of trip times taken from the 2001 U.S. National Household Transportation Survey[1].

distribution of destination times before a trip has started, *i.e.*  $P(T_D = t_D)$ , where  $T_D$  represents the total trip time. Once some time has passed on a trip, the probability of times passed drops to zero, so we renormalize to get

$$P_{\Delta T}(\Delta T = \Delta t | D = i) = \begin{cases} 0 & \text{if } t_D < \Delta t \\ P(T_D = t_D) / \sum_{t_D \geq \Delta t} p(T_D = t_D) & \text{if } t_D \geq \Delta t \end{cases} \quad (5)$$

To compute the likelihood for a candidate destination, we take  $\Delta t$  as the length of the trip so far and  $t_D$  as the estimated time to the candidate destination from the current cell, based on the  $T_{i,j}$  estimated trip times explained in Section 4.4.

#### 4.6 Inferring Posteriors Over Destinations

In Section 4.3, we described the generation of a prior probability distribution via the combination of visited locations and geographic data, combined with an open-world approach. Sections 4.4 and 4.5 introduced two likelihoods of seeing data given the truth of target destinations.

If we assume independence of the driving efficiency and the trip duration likelihoods given the destinations, we can combine these two elements and the prior into a single posterior probability for each destination using Bayes rule from Equation (1), giving the destination probability as

$$P(D = i | E = e, \Delta T = \Delta t) = \frac{P_E(E = e | D = i) P_{\Delta T}(\Delta T = \Delta t | D = i) P_{open}(D = i)}{\sum_{j=1}^N P_E(E = e | D = j) P_{\Delta T}(\Delta T = \Delta t | D = j) P_{open}(D = j)} \quad (6)$$

Considering such independencies is referred to as the naïve Bayes formulation of Bayesian updating. Relaxing the independence assumptions to allow richer probabilistic dependencies would likely enhance the accuracy of the predictions because introducing realistic dependencies minimizes “overcounting” of probabilistic influences. In this case, we are not considering the relationships between driving efficiency and duration.

In the general case, more sophisticated models come at the cost of more complex representations and data collection. Naïve Bayes has been shown to perform relatively well in a variety of domains (e.g., see [16]). Angermann *et al.*[17] used a similar technique for combining probabilistic location estimates defined on a grid. Elfes[18] introduced the probabilistic grid approach to robotics for sensor-based mapping. Practically, we implement this equation by computing a grid of scalars for each of the probabilistic components, multiplying the scalars in corresponding cells, and normalizing to make the sum of the products one.

The probabilistic formulation of destination prediction means that uncertainties about the driver’s true destination are represented in a coherent manner. This means that applications built on a system like ours can account for the inevitable uncertainty in a driver’s destination. For instance, an application that shows restaurants or gas stations near a driver’s destination could progressively show more detail and less area as the destination becomes more certain. Warnings about traffic problems could be held until the certainty of encountering them exceeds a certain threshold. Cognitively impaired people deviating from their intended destination could be warned only when the deviation becomes nearly certain.

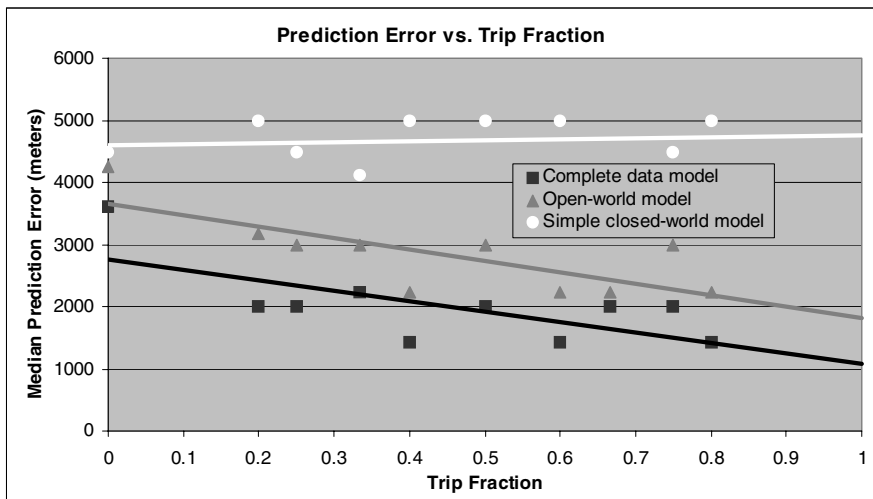
## 5 Results

We tested our Predestination algorithm on trips from our database of GPS traces. We split the data into two halves, using one half for training the efficiency distributions  $P_E(E = e | D = i)$  as explained in Section 4.4. Testing on the remaining half of the data (3667 trips), we iterated through each trip of each subject. For each trip, we trained the closed world prior  $P_{closed}(D = i)$  on that subject’s trips, omitting the one we were testing, resulting in a leave-one-out testing strategy. We tested the algorithm in three different modes:

- Simple closed-world model – This model uses only a closed-world prior based on survey data from days before the test day. It does not use the efficient driving likelihood nor the trip time likelihood, only a simple prior. This naïve model represents a first order attempt at predicting destinations based only on where a subject has gone in the past.
- Open-world model – This uses Equation (6) and the open-world prior from Equation (3). For training the  $P_{closed}(D = i)$  part of the prior, it uses only survey data from days before the test day. This represents a realistic scenario of user starting with no training data.
- Complete data model – This uses a closed-world prior, but it is based on *all* the survey data (except the holdout trip), including survey data taken after the test

day. We expect this model to perform best since it has the benefit of using the steady state destination prior. This is how the system would work after it has been trained sufficiently to have seen nearly all the subject's destinations.

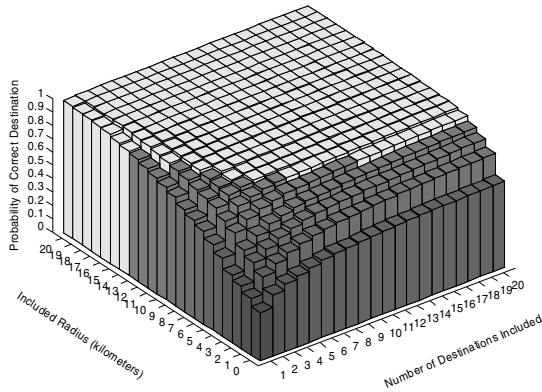
The result of computing a destination prediction from Equation (6) is a probability mass function over all the cells. Sorting the cells by this probability gives a ranking, from highest to lowest, of the posterior probability of each cell being the actual destination. It is this sorted list that Predestination would return to an application that needed to take an action as a function of the user's destination. Predestination makes such a prediction at every point along the trip. A simple way to examine the result is to consider the maximum *a posteriori* (MAP) estimate, which is the cell with the maximum probability. Figure 8 shows the median error between the MAP estimate and the actual destination as a function of the fraction of the trip completed. The complete data model has a median error of two kilometers at the halfway point, while the open-world model has an error of three kilometers at the same point. The simple open-world model gives an error that is consistently higher, around five kilometers. This result shows that our open-world model approaches the accuracy of the complete data model, and that the simple closed-world model is noticeably worse.



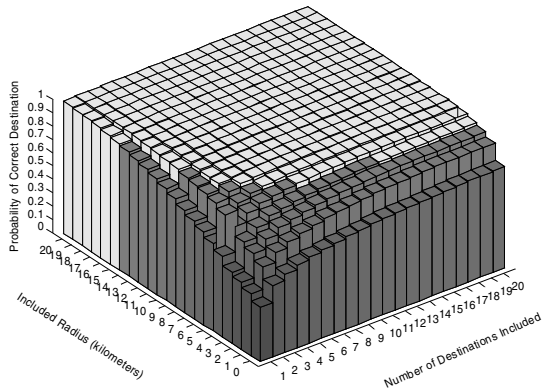
**Fig. 8.** The median prediction error using the MAP estimate drops with the fraction of the trip completed. At the halfway point of the trip, the complete data model's error is two kilometers. The open-world model comes within about one kilometer of the accuracy of the complete data model. The simple closed-world model is consistently poor.

In some instances, an application could make use of a list of highly probable destinations rather than just the single MAP estimate. For instance, in giving anticipatory point-of-interest data, a user might be willing to pick from a list of the top several likely destinations. In this case, Predestination would return a list of destination





(a) Using open-world model



(b) Using complete data model

**Fig. 9.** These plots show the probability of getting the correct destination at the halfway point of the trip. Each vertical bar shows the probability of the correct destination being somewhere in the sorted list of the given number of destinations included and within the given radius of the computed destination(s). The light-colored bars show probabilities  $\geq 0.9$ .

predictions sorted by probability. Figure 9 shows how the probability of finding the correct destination cell changes with the length of the sorted list and the distance tolerance for both the open-world model and the complete data models at the halfway point of the trip. Specifically, each bar in these figures shows the probability of the correct destination being both:

- Somewhere in the list of the most probable computed destinations, as a function of the length of the list, and
- Somewhere within a radius of the most probable computed destinations, as a function of radius.

As either the length of the list or the radius is increased, the probability of finding the correct destination goes up. The light gray bars show which probabilities are at least 0.9. Figure 9(a) shows the probabilities using the open-world model, where training on trips after the day in question was disallowed. Figure 9(b) shows the probabilities using a complete data distribution. The open-world model achieves nearly the same level of performance as the complete data model, validating our attempt to model the steady state in the early days of the GPS survey.

## 6 Conclusions

“Predestination” predicts a driver’s destination as the trip progresses. In the approach described here, we considered four different probabilistic cues, and we combined the cues in a mathematically principled way to create a probability grid of likely destinations. The best performance on 3667 different driving trips gave a median error of two kilometers at the trip’s halfway point. We introduced an open-world model of destinations that helps the algorithm work well in spite of a paucity of training data at the beginning of the training period. Applications of Predestination include proactively delivering information about upcoming points of interest and traffic problems. This could reduce cognitive load on the driver by eliminating information about places that he or she is unlikely to encounter. Destination prediction can also be used to detect if a user is deviating from the route to an expected location.

Future work in this area includes exploring the value of relaxing assumptions of probabilistic independence and incorporation of additional prediction features. As examples of the latter, the structure of the road network and locations of specific points of interest could be considered. In addition to absolute destinations, destination classes such as coffee shops, dinner spots, and antique stores could be learned for individual users. Destinations also likely show temporal dependence, so we expect that inclusion of variables representing time of day, day of week, month of year, and a user’s calendar data would enhance predictive accuracy, especially with increasing amounts of data, providing representative coverage for different periods of time .

## Acknowledgments

We thank Kelli McGee for assisting with the administration of the MSMLS study.

## References

1. Hu, P.S. and T.R. Reuscher, *Summary of Travel Trends, 2001 National Household Travel Survey*. 2004, U. S. Department of Transportation, U.S. Federal Highway Administration. p. 135.
2. Patterson, D.J., et al. *Opportunity Knocks: A System to Provide Cognitive Assistance with Transportation Services*. in *UbiComp 2004: Ubiquitous Computing*. 2004. Nottingham, UK: Springer.
3. Cheng, C., R. Jain, and E.v.d. Berg, *Location Prediction Algorithms for Mobile Wireless Systems*, in *Wireless Internet Handbook: Technologies, Standards, and Application*. 2003, CRC Press: Boca Raton, FL, USA. p. 245 - 263.

4. Krumm, J. and E. Horvitz, *The Microsoft Multiperson Location Survey*. 2005, Microsoft Research (MSR-TR-2005-103): Redmond, WA USA.
5. Karbassi, A. and M. Barth. *Vehicle Route Prediction and Time of Arrival Estimation Techniques for Improved Transportation System Management*. in *Proceedings of the Intelligent Vehicles Symposium*. 2003.
6. Marmasse, N. and C. Schmandt, *A User-Centered Location Model*. *Personal and Ubiquitous Computing*, 2002(6): p. 318-321.
7. Ashbrook, D. and T. Starner, *Using GPS To Learn Significant Locations and Predict Movement Across Multiple Users*. *Personal and Ubiquitous Computing*, 2003. 7(5): p. 275-286.
8. Hariharan, R. and K. Toyama. *Project Lachesis: Parsing and Modeling Location Histories*. in *Geographic Information Science: Third International Conference, GIScience 2004*. 2004. Adelphi, MD, USA: Springer-Verlag GmbH.
9. Liao, L., D. Fox, and H. Kautz. *Learning and Inferring Transportation Routines*. in *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI 2004)*. 2004. San Jose, CA, USA.
10. Gogate, V., et al. *Modeling Transportation Routines using Hybrid Dynamic Mixed Networks*. in *Uncertainty in Artificial Intelligence (UAI 2005)*. 2005.
11. <http://landcover.usgs.gov/ftpdownload.asp>.
12. <http://landcover.usgs.gov/classes.asp>.
13. Letchner, J., J. Krumm, and E. Horvitz. *Trip Router with Individualized Preferences (TRIP): Incorporating Personalization into Route Planning*. in *Eighteenth Conference on Innovative Applications of Artificial Intelligence*. 2006. Boston.
14. Krumm, J. *Real Time Destination Prediction Based on Efficient Routes*. in *Society of Automotive Engineers (SAE) 2006 World Congress*. 2006. Detroit.
15. <https://nhts.ornl.gov/2001/Login.do>.
16. Rish, I. *An Empirical Study of the Naive Bayes Classifier*. in *IJCAI-01 Workshop on Empirical Methods in AI*. 2001.
17. Angermann, M., et al. *Software Representation for Heterogeneous Location Data Sources Using Probability Density Functions*. in *International Symposium on Location Based Services for Cellular Users (LOCCELLUS 2001)*. 2001. Munich.
18. Elfes, A., *Using Occupancy Grids for Mobile Robot Perception and Navigation*. *IEEE Computer*, 1989. 22(6): p. 46-57.