SAE 2008 World Congress, April 14-17, 2008, Detroit, MI USA

Paper Number 2008-01-0195

A Markov Model for Driver Turn Prediction

John Krumm
Microsoft Research

ABSTRACT

This paper describes an algorithm for making short-term route predictions for vehicle drivers. It uses a simple Markov model to make probabilistic predictions by looking at a driver's just-driven path. The model is trained from the driver's long term trip history from GPS data. We envision applications including driver warnings, anticipatory information delivery, and various automatic vehicle behaviors. The algorithm is based on discrete road segments, whose average length is 237.5 meters. In one instantiation, the algorithm can predict the next road segment with 90% accuracy. We explore variations of the algorithm and find one that is both simple and accurate.

INTRODUCTION

Predicting a driver's near-term future path could be useful for giving the driver warnings about upcoming road situations, delivering anticipatory information, and allowing the vehicle to automatically adapt to expected operating conditions. This paper presents a prediction algorithm trained from a driver's past history. Specifically, we train an n^{th}-order Markov model to probabilistically predict future road segments based on a short sequence of just-driven road segments, usually 1-10. A sketch of this basic approach appears in Figure 1. We show that the resulting accuracy is much better than random guessing; for instance, looking at the 10 most recent road segments, we can predict the next road segment with about 90% accuracy. The algorithm is simple enough to be trained and executed on an in-vehicle navigation computer, and it needs no off-board network connection. We tested the accuracy of our predictions on GPS data from 100 drivers.

Our predictions are relatively short into the future, varying from a single upcoming road segment (about 237.5 meters ahead) to 10 segments. The predictions include merges and turns at intersections, which are the most likely to spur interesting applications. We envision applications like:

- Anticipatory driver warnings
  - Hazardous road conditions
  - Unusually slow traffic
  - Change in speed limit
  - Curve speed warning
  - Change in traffic patterns (*e.g.* due to construction)
  - Lane keeping assistance
  - Lane change prediction and warning
- Driver information
  - Points of interest
  - Advertising
- Automatic vehicle behaviors
  - Automatic turn signals
  - Headlight pointing
  - Wireless base station handoff
  - Gradual windshield darkening for expected glare
  - Engine load anticipation (*e.g.* cylinder deactivation)
  - Emergency preparation (*e.g.* pre-braking, seat belt tension, head rests)

This paper next describes our experimental GPS data and the road representation we used. It continues with a description of our prediction algorithm and accuracy results. In the results section, we explore how prediction accuracy is affected by various alternative training options in order to find an instance of the algorithm that

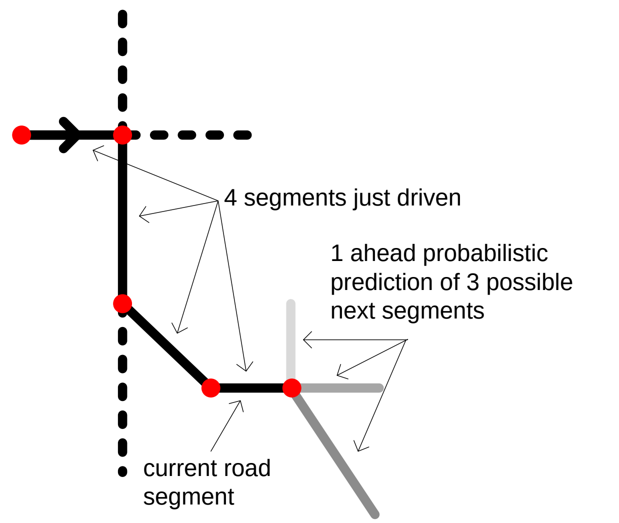

Figure 1: Our prediction algorithm looks at a recent sequence of road segments to probabilistically predict the next segments.

is both simple and accurate.

Previous work in route prediction includes Patterson *et al.* [6] who applied machine learning to GPS data to find a person's frequently visited locations and then inferred which one the person was traveling to. In [2], Karbassi and Barth predict routes taken by car-sharing vehicles going between pre-established stations. Torkkola *et al*. [9] learn frequent destinations and the regularly traveled sequence of GPS points between them to predict routes for giving traffic advisories. In work by Froehlich and Krumm ([1], under submission to SAE Congress), the goal is to find a driver's regular routes and predict which one he or she is currently driving. The work in the current paper differs from this previous work in that we are not concerned with destinations nor with beginning-to-end routes. Instead, our predictions are free-floating in that they are not necessarily dependent on repeated routes or destinations. Our Markov model can make a prediction based on only a single previous observation of a few road segments.

We also note that the work in this paper is different from our previous work on destination prediction [3, 4]. In the previous work, our goal was to predict the location of the driver's destination, but not anything about the future route. The work in this paper is aimed at predicting which roads the driver will drive on next.

The work most closely related to our work is from Simmons *et al.* [8]. They train a hidden Markov model (which is different from our simpler Markov model) to predict a driver's future road segments while simultaneously predicting the driver's destination. We will explain the distinctions between their work and ours scattered over the remainder of this paper.

EXPERIMENTAL GPS DATA

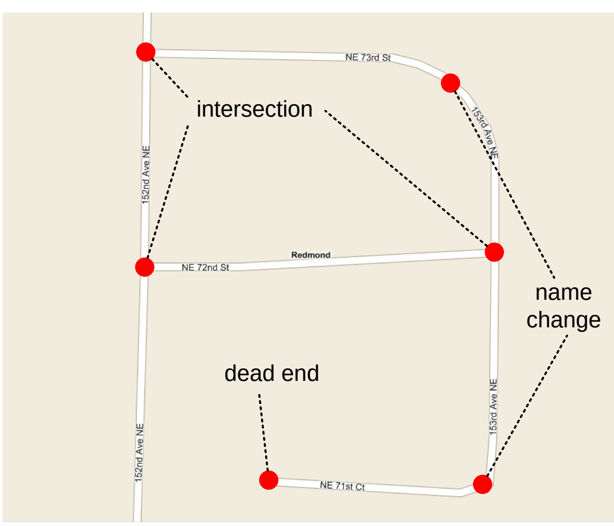

Figure 2: Roads are represented as discrete segments. In this example, the ends of the segments are shown as dots which mark intersections, name changes, or dead ends.

Our prediction algorithm is based on observations of where drivers drive measured from GPS receivers. We have been gathering GPS data from volunteer drivers in our Microsoft Multiperson Location Survey (MSMLS) starting in March of 2004. Volunteer drivers are loaned one of our 55 Garmin Geko 201 GPS receivers, capable of recording 10,000 time-stamped latitude and longitude measurements. We instruct drivers to leave the GPS on their vehicle's dashboard, plugged into the cigarette lighter for power. On some vehicles, power to the cigarette lighter is cut when the vehicle's key is removed. For this reason, we inserted a small wedge holding down the GPS receiver's "on" button to ensure that it always turns on when it receives power. Otherwise it would remain off without attention from the driver. The GPS receivers are set to an adaptive recording mode that records more points when the vehicle is moving and accelerating. The median interval between recorded points is 6 seconds and 62 meters. Although we have data from 252 subjects, we chose a subset of 100 subjects for this study to reduce the computation time required for our experiments.

ROAD REPRESENTATION

We use a Markov model to predict a vehicle's near term future route. More specifically, we use a discrete Markov chain representation (see [7]), which is explained in the next section. Applied to our problem, this scheme represents the state of the vehicle as being located on one of a discrete set of road segments, as shown in Figure 2. Road segments come from our digital map representation of the road network. Segments terminate at intersections, dead ends, and changes in the name of the road. For the road segments covered in our study, the median length was 237.5 meters (0.15 miles). Our 100 drivers covered a total of 43,893 distinct road segments while under observation.

We do not attempt to predict where on the road segment a vehicle will be, nor do we attempt to predict when it will arrive at a road segment. Our goal is predict the chain of road segments that a vehicle will next encounter. In this way, our predictions show the vehicle's future path, elevation, and turns.

As described above, our experimental data is a set of time-stamped latitude and longitude pairs. In order to make predictions about road segments, we need to convert this data into sequences of road segments. We begin by segmenting the GPS data into discrete trips, splitting the sequence at any point with more than five minutes between adjacent points in time. This sometimes leads to tiny garbage trips due to short bursts of GPS data from a parked vehicle. Thus, we filter out any trips less than one kilometer (0.62 miles) and any trip with less than ten GPS points. We also eliminate any trip whose maximum speed is below 25 mph to help eliminate walking and biking trips.

To get road segments from our GPS data, we submit each trip to a map matching algorithm we developed [5].

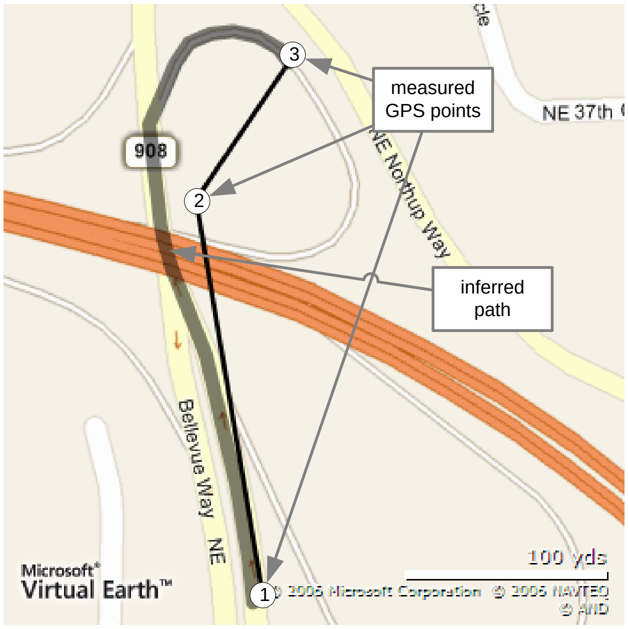

Figure 3: Map matching converts from noisy GPS coordinates to road segments (from [5]).

The basic map matching process is illustrated in Figure 3. This algorithm matches each GPS point to a road segment, taking into account which roads are nearby as well as constraints imposed by the connectivity and speed limits of the road network. Adhering to these constraints significantly reduces nonsense matches that come from the inevitable inaccuracies in GPS data. For instance, matching to the nearest road without constraints sometimes results in a path that jumps between parallel, opposing lanes of traffic. Our map matching program minimizes these errors.

After map matching, we process the road segments to first eliminate any adjacent repeated road segments. These come from having more than one GPS sample on the same road segment. We also fill in gaps so the road segments for each trip are contiguous. With this processing, each trip is represented by a connected sequence of road segments with no adjacent repeats.

One difference between this work and some previous work is that we use discrete road segments as the basis for our predictions, as was also done in [2]. By using a symbol-based description of routes, rather than 2D points in terms of latitude and longitude, we are able to use a symbol-based predictive method like the Markov model, described next.

MARKOV MODEL

Our prediction of a vehicle's near term future route is based on its near term past route. We model the sequence of traversed road segments as $X(i)$, with i representing a discrete time variable and $X(\cdot)$ representing a road segment (*e.g.* an integer unique among all the road segments). Whenever we want to make a prediction, we denote the vehicle's road segments as $\{\dots, X(-2), X(-1), X(0), X(1), X(2), \dots\}$, where $X(0)$ is the current road segment, $X(-1)$, $X(-2)$, … are the immediately preceding road segments, and $X(1)$, $X(2)$, … are the unknown future road segments that we are trying to predict. The road segments are not necessarily encountered at even time intervals. The discrete time variable i serves as an index over the segments in the order they are encountered. When the vehicle moves to a new road segment, that segment becomes the new $X(0)$. At any time, we know the current road segment $X(0)$ and the past road segments $\{\dots, X(-3)\ X(-2), X(-1)\}$ back to the beginning of the trip. The unknowns are the future road segments that we are trying to predict, $\{X(1), X(2), X(3), \dots\}$.

At any point along a trip, the driver can choose which road segment to drive on next. In light of this choice, our predictions are probabilistic. For instance, $P[X(1)]$ represents a discrete probability distribution over all the road segments giving which road segment will be encountered after the current one, $X(0)$. $P[X(2)]$ is the distribution for which road segment will be encountered after $X(1)$, and so on.

The Markov model gives a probabilistic prediction over future road segments based on past road segments. The standard, first order Markov model says that the probability distribution $P[X(1)]$ for the next road segment is independent of all but $X(0)$, the current road segment:

$$P[X(1)|X(0), X(-1), X(-2), \dots] = P[X(1)|X(0)] \qquad (1)$$

For a given driver, we can build $P[X(1)|X(0)]$ easily. For each road segment $X(0)$, we build a histogram of which road segments were encountered immediately after, and then normalize to get a discrete probability distribution. There is a separate probability distribution for each road segment that a driver has ever driven on.

A second order Markov model is sensitive to the *two* most recent road segments, *i.e.* $P[X(1)|X(-1), X(0)]$. We build this model in a similar way, except we create a histogram over all two-element, ordered sequences $\{X(-1), X(0)\}$. In our results, we check to see if using higher order models helps prediction accuracy, which it does. In particular, looking at the two most recent road segments gives a sense of the direction of travel along a road, which helps significantly. For a first order model, the direction of travel is not encoded by observing only the current road segment.

The Markov model can be used to predict beyond just the next road segment. We can clearly build $P[X(2)|X(0)]$, which is the distribution over the road segments after the next one, given the current one. We can also user higher order models to make these farther out predictions, *e.g.* $P[X(2)|X(-1), X(0)]$. In general, we can build an n^{th} order Markov model $(n \geq 1)$ to

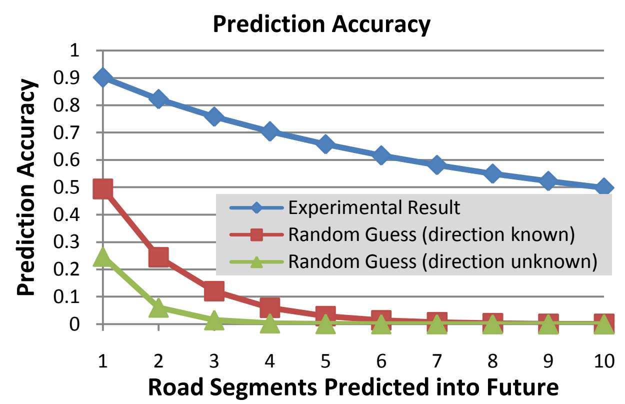

Figure 4: Prediction accuracy drops as predictions go farther into the future. These experimental results are based on looking at the last 10 road segments. The two lower curves show how accurate prediction would be with random guessing. It quickly drops to near zero.

predict the m^{th} next encountered segment ($m \geq 1$). We denote our general n^{th} order model as

$$P_n[X(m)] = P[X(m)|X(-n+1), X(-n+2), \ldots, X(0)] \qquad (2)$$

In our results, we look at how prediction accuracy changes as we increase n, the number of segments we look at into the past (better). We also look at how prediction accuracy changes as we increase m, the number of segments we predict into the future (worse).

We note that the Markov model does not explicitly constrain a vehicle to adhere to the connectedness of the road network. A trained model could conceivably predict that a driver will jump over several road segments. However, since the model is trained from real data, where such jumps do not occur, the Markov model implicitly prevents such nonsense predictions.

One advantage of probabilistic predictions is that the algorithm has a measure of its own uncertainty that can be usefully reported to in-vehicle applications. For instance, automatically engaging a turn signal might depend on near 100% prediction certainty, while presenting a point of interest would not require the same level of confidence.

RESULTS

We tested our prediction algorithm on data from 100 drivers in our MSMLS study. On average, we observed each of the 100 drivers for a total of 12.21 days. In testing a given set of parameters, e.g. m and n, we used leave-one-out testing, where we trained the Markov model with all but one trip and tested it on the remaining trip. Doing this once for each trip gives an average accuracy figure over all the trips. Leave-one-out testing comes very close to predicting how the algorithm would work in a real application where all the previous trips would be used for training.

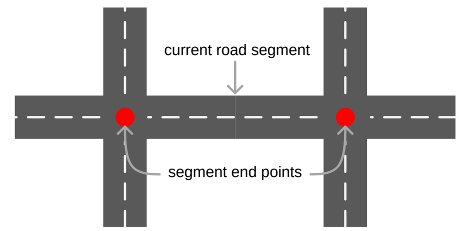

Figure 5: From the current road segment in this example, there are six choices for the next road segment if U-turns are disallowed and if the direction of travel is unknown. If the direction is known, then there are only three choices.

An example result is shown in Figure 4. This shows the accuracy of predicting one road segment ahead, two ahead, and up to ten segments ahead based on the last ten observed road segments. The predicted road segment for our experiments is the one with the highest probability form the Markov model. In this case, the Markov model predicts the next road segment with slightly over 90% accuracy. As expected, the prediction accuracy drops the farther it looks into the future. Since the road segments average 237.5 meters long, a one-ahead prediction corresponds to predicting over the next 237.5 meters (0.15 miles). At 10 segments ahead (2375 meters or 1.5 miles), our prediction accuracy is 50%.

Figure 4 also shows the accuracy of prediction using random guessing as a way to assess the relative accuracy of the Markov model. Random guessing would involve randomly predicting the next road segment at each choice point. For instance, on a road segment that connects two four-way intersections (Figure 5), the random algorithm would assign a probability of 1/6 to each of the six possible next road segments and randomly pick one for the next segment. (We discount the possibility of a U-turn in our analysis of random guessing, which makes it appear somewhat more accurate.) Predicting two segments into the future would involve making two of these random choices. In fact, the road segment choices are conveniently represented as a tree, as shown in Figure 6. The branching factor for this tree is two, given that each node splits into two as the tree gets deeper. If the branching factor is b, and if we are predicting m segments into the future, the number of possible choices at that future point is b^m. Of course the branching factor changes from segment to segment, but

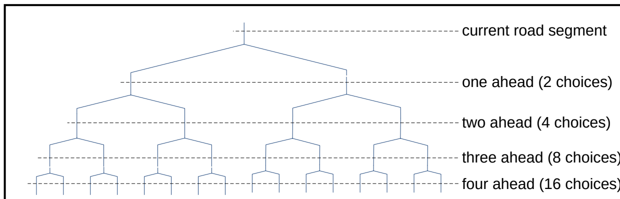

Figure 6: The choice for future road segments can be represented as a tree. In this example, the branching factor is two.

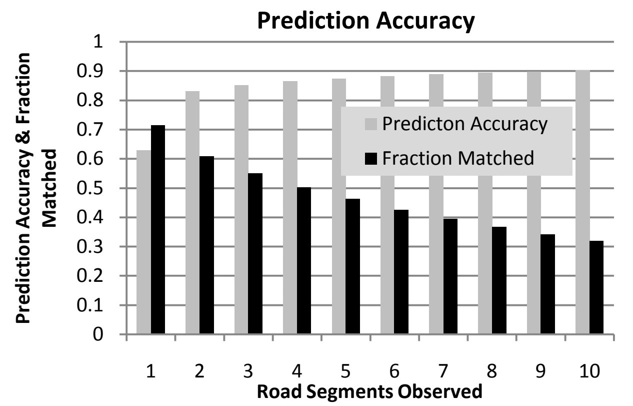

Figure 7: Accuracy for predicting the next segment rises with the number of past segments matched, but longer past sequences are harder to find.

the average branching factor for the roads encountered in our study was 4.04.

For a given prediction depth m and branching factor b, the probability of randomly guessing the right segment is $1/b^m$. If we know the direction of travel, then the branching factor is halved, because we know which end of the current segment is next. In this case, the probability of making the correct prediction by random guessing is $1/(0.5b)^m$. These two curves are shown in Figure 4, demonstrating that our predictions are significantly better than random guessing, as they should be.

Simmons *et al.* [8] achieved a next-segment prediction accuracy of up to 99% in a test on a driver with 46 trips using a hidden Markov model that accounts for road segments and a destination prediction. They note the issue of random guessing as well. In their road network representation, 95% of the road segment end points were connected to only one other road segment, presenting the prediction algorithm with a single forced choice that was guaranteed to be correct. In contrast, only about 28% of our segment end points have such a forced choice.

SENSITIVITY TO NUMBER OF PAST SEGMENTS

We showed above how accurate our predictions are based on a 10th order Markov model, *i.e.* one that looks at the previous ten road segments in order to predict the future (Figure 4). Sometimes, there are fewer past road segments to examine, either because the trip just started or because there isn't a long enough match sequence in the driver's recorded history of road segments.

Figure 7 shows how prediction accuracy varies with the number of past road segments observed. The prediction here is just the *next,* single road segment. Prediction accuracy increases as the number of observed segments increases, meaning that a longer sequence of road segments is more indicative of the future. There is a significant jump in prediction accuracy between one and two observed segments. This is because observing only one segment is not indicative of a driver's direction, meaning that the vehicle could be heading toward either end of the current segment. Observing two or more past segments indicates the driver's direction and reduces the possibilities for future segments.

This plot also shows how often we found matches in the driver's history for past sequences of a given length. While accuracy goes up with a longer match sequence, it is generally harder to find these longer sequences in the driver's history. Based on the accuracy results in Figure 7, an optimal prediction algorithm would look for the longest match available to make a prediction.

SENSITIVTY TO TIME OF DAY

It may be that the time of day can help predict a driver's future road segments. Some regular trips might begin by traversing the same set of road segments, only to diverge at a certain intersection depending on a regularly scheduled commitment. We explored this possibility by building a series of Markov models whose training data were limited to a certain time interval centered around the test trip.

The results of this test are shown in Figure 8. In this case, we tested a 2nd-order Markov model, *i.e.* one that looks at the past two road segments, and we predicted the next, single road segment. We tested Markov models whose training data was limited to 1 hour around the test trip, 2 hours around the test trip, and up to 24 hours around the test trip. The time tolerance ignores the date of the trip, instead looking at just the time of day. The plot shows that prediction accuracy is not very sensitive to the time of day, so this is not an important factor to consider when making turn predictions. We found a similar lack of sensitivity for Markov models up to order 10. Simmons *et al.* [8] found a similar lack of sensitivity to both time of day and day of week, although they found that speed is a significant factor in increasing prediction accuracy.

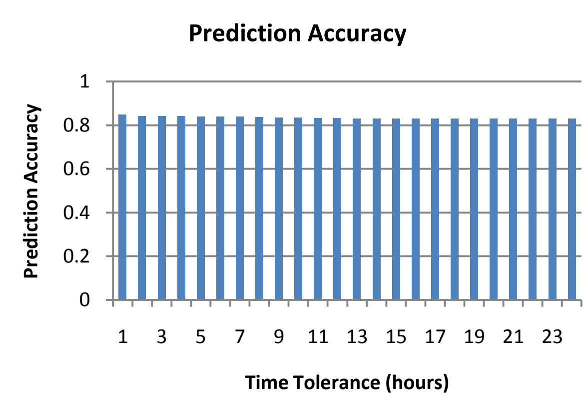

Figure 8: Considering the time of day had a negligible effect on our prediction accuracy. This shows results of a 2nd order Markov model predicting the next road segment.

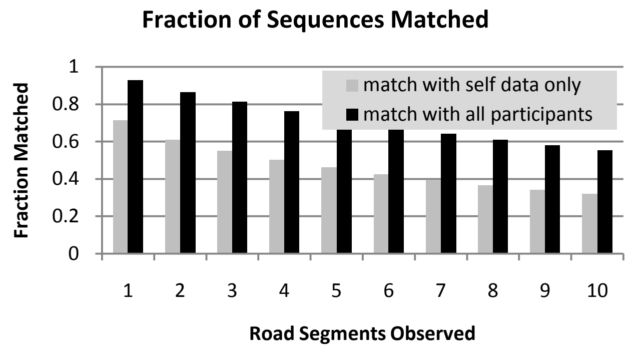

Figure 9: Using other drivers' data to create the Markov models gives a better chance of finding a matching sequence in the training data.

SENSITIVITY TO OTHERS' TRAINING DATA

Until now, we have trained each driver's Markov models using only that driver's data. We can also make Markov models using data from all the drivers in our study. The clear advantage of this is that we can usually find more training data for a given observed sequence. We can also can find more long sequences, which, according to Figure 7, gives more accurate predictions. The disadvantage is that using other peoples' driving data makes the models less specific to a given driver's behavior.

We explored these questions by creating Markov models using only the test driver's data (as we have been doing all along), and using all drivers' data. The results are shown in Figure 9 and Figure 10. Figure 9 shows that, as expected, we are able to find more matches to past sequences using all drivers' data. Figure 10 shows that prediction accuracy drops slightly when using all drivers' data, at least on the task of predicting the next, single road segment. This drop in accuracy is small, but consistent. It is small likely because there are not too many overlapping road segments from our sample of 100 drivers. But, even with small overlaps, adding other's data dilutes a driver's turn probabilities enough to consistently reduce prediction accuracy.

CONCLUSION

Our goal was to find a simple and accurate prediction algorithm. Based on GPS data from 100 drivers, we have shown that a Markov model is a simple, effective way to predict near-term, future road segments. Looking at the most recent 10 segments into the past, we can predict the next segment with about 90% accuracy. By comparison, random guessing with a known direction of travel would give about 50% accuracy. Since our mean road segment length is 237.5 meters (0.15 miles), predicting the next segment translates to predicted travel over the next 237.5 meters. We can predict ahead three segments with approximately 76% accuracy, which translates to 3×237.5 = 712.5 meters (0.44 miles).

Our experiments show that a fairly simple model works best. Prediction accuracy increases by considering a

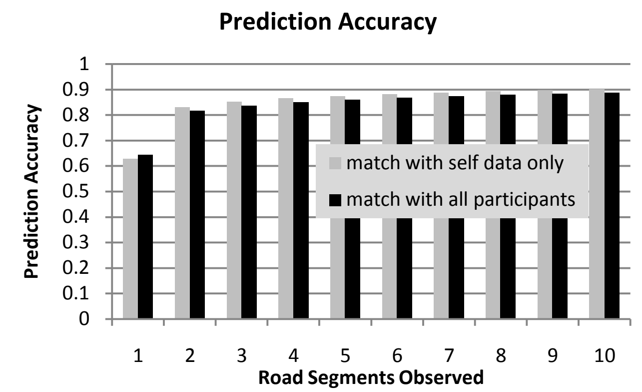

Figure 10: Prediction accuracy for one-ahead prediction drops slightly when using all drivers' data over data from just the driver in question.

longer sequence of just-driven road segments. Thus, a practical algorithm would maintain a set of n^{th} order Markov models, with, say, $1 \leq n \leq 10$. We also show that time of day is a negligible factor in the prediction accuracy and that using other drivers' data for predictions reduces prediction accuracy. Thus the model should ignore the complexity of considering time of day, and it should be trained only from the driver's vehicle.

We envision that predictions of this type could be used to warn drivers of upcoming traffic disruptions, provide anticipatory information, and trigger automatic vehicle behaviors.

Future work along the direction outlined in this paper could include experiments that show how prediction accuracy changes with the number of days that a driver is observed. We expect prediction accuracy to generally rise with observation time, eventually reaching a plateau. However, sometimes drivers change their habits, so an algorithm based on long term data should be able to recognize sudden changes and respond quickly to new habits.

CONTACT

John Krumm
Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
USA

jckrumm@microsoft.com

REFERENCES

1. Froehlich, J. and J. Krumm, *Route Prediction from Trip Observations*, in *Society of Automotive Engineers (SAE) World Congress*. 2008, Paper 2008-01-0195: Detroit, Michigan USA.
2. Karbassi, A. and M. Barth, *Vehicle Route Prediction and Time of Arrival Estimation Techniques for Improved Transportation System Management*, in *Intelligent Vehicles Symposium*. 2003. p. 511-516.
3. Krumm, J., *Real Time Destination Prediction Based on Efficient Routes.* SAE 2006 Transactions Journal of Passenger Cars - Electronic and Electrical Systems, 2006.
4. Krumm, J. and E. Horvitz, *Predestination: Inferring Destinations from Partial Trajectories*, in *Eighth International Conference on Ubiquitous Computing (UbiComp 2006)*. 2006: Orange County, California, USA. p. 241-260.
5. Krumm, J., J. Letchner, and E. Horvitz, *Map Matching with Travel Time Constraints*, in *Society of Automotive Engineers (SAE) 2007 World Congress*. 2007, Paper 2007-01-1102: Detroit, Michigan USA.
6. Patterson, D., et al., *Inferring High-Level Behavior from Low-Level Sensors*, in *UbiComp 2003: Ubiquitous Computing*. 2003, Springer: Seattle, Washington USA. p. 73-89.
7. Shanugan, K.S. and A.M. Breipohl, *Random Signals: Detection, Estimation and Data Analysis*. 1988, New York: John Wiley & Sons. 664.
8. Simmons, R., et al., *Learning to Predict Driver Route and Destination Intent*, in *2006 IEEE Intelligent Transportation Systems Conference*. 2006: Toronto, Canada. p. 127-132.
9. Torkkola, K., et al., *Traffic Advisories Based on Route Prediction*, in *Workshop on Mobile Interaction with the Real World (MIRW 2007)*. 2007: Singapore.