Where Will They Turn: Predicting Turn Proportions At Intersections

John Krumm

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

Abstract. Predicting a driver's route would be useful for warning a driver of upcoming road hazards, informing about traffic situations, and serving relevant advertising. There are many clues to a driver's future route, including their past behavior and likely destination. Another clue is the driver's turn choices at a sequence of intersections. Strung together, the turn choices form a route. This paper develops a basic algorithm, and variations, to predict the aggregate turn behavior of drivers at intersections. Given an intersection with a few different turn options, including the option to continue straight ahead, our goal is to infer the proportion of drivers who will take each option. For ground truth, we use raw turn counts gathered for government traffic studies by our local municipality at 40 different intersections. Our basic turn prediction algorithm is based on the assumption that drivers tend to choose roads that offer them more destination options. This matches our intuition that turning onto a short, dead end road is relatively rare compared to turning onto a highway "on" ramp. The best performing algorithm predicts turn proportions with a median error of 0.142.

Keywords: Location prediction, driver turn prediction, route prediction, driver behavior, intersections, turn counts, turn proportions

1 Introduction

Predicting a driver's route would be useful to warn the driver about impending road hazards, to inform of upcoming traffic situations, and to deliver relevant location-based advertising about soon-to-be-near businesses. With more vehicles carrying navigation systems, the sensing, computation, and display required for making and using route predictions is becoming widespread. Route prediction would be simple if the driver planned and followed a route using the vehicle's navigation system. We assert, however, that most of the time drivers know where they're going, and so do not use the navigation system for route planning. Quantitative support for this assertion comes from usage statistics of GM's OnStar system. OnStar is an onboard, pushbutton system for driving assistance that includes the ability to get turn-by-turn directions on the vehicle's display. As of July 2008, the service was handling 600,000 monthly requests for driving directions from its 5 million subscribers [10]. Combined

with the fact that American drivers take about 1500 driving trips per year [1], OnStar subscribers ask for directions for only about 1% of their trips. If this holds true for drivers in general, then predicting routes based on driving directions is not a solution that will work very often.

With only rare explicit clues about a driver's route, researchers have resorted to a variety of implicit clues. One clue is the driver's past behavior, normally measured with GPS, which is a reliable indicator of future routes if the driver is driving in a familiar area. For instance, Patterson *et al.* [7] used GPS traces and a dynamic Bayes net to infer, in real time, a moving person's mode of transportation (*i.e.* bus, foot, car) and a prediction of their route based on historical behavior. The work in [5] used a Markov model, trained from past behavior, to predict the next few road segments that a driver would take. Froehlich and Krumm [3] predicted entire routes from GPS traces by looking at which previous route a driver appeared to lock onto partway into the trip. The Predestination system [6] predicted a driver's destination based partly on past behavior and partly on geographic features. The research found, unsurprisingly, that land covered by water or ice made for less popular destinations than commercial and residential areas.

In this paper, we aim to augment route prediction by inferring which direction a driver will turn, if at all, at a given intersection. Instead of relying on the driver's personal GPS traces as in [3], [5], and [7], we instead look at general geographic features as in [6]. This leads to general, probabilistic predictions at each intersection that apply to all drivers. Specifically, for each intersection, we infer which proportion of drivers will chose each option, including the option of continuing along the same road.

The intuition behind our predictions is that drivers tend to choose the road that will give them more destination options. Thus, choosing to turn down a dead end road is likely rare compared to choosing to turn onto a highway "on" ramp. This is illustrated more generally with the intersection shown in Figure 1. Here, a westbound driver has

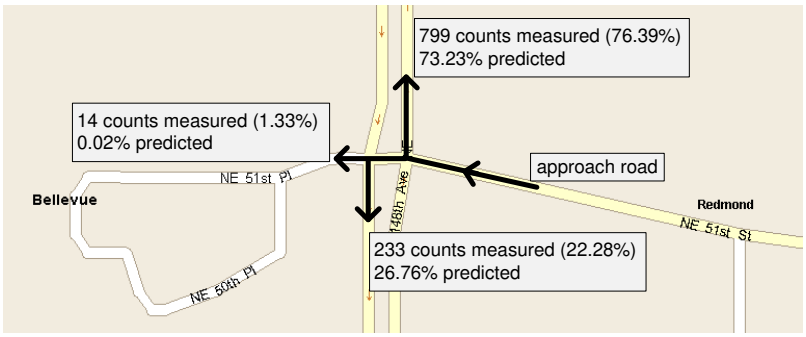


Figure 1: Driving westbound, counts show drivers' choices at the intersection. Our turn prediction algorithm is based on our intuition that it is less likely to turn toward directions with fewer possible destinations. In this example, the proportion of westbound vehicles going straight into the small loop was measured at 1.33% by road sensors. Our basic algorithm estimated the proportion at 0.02% because of the paucity of destinations compared to the other choices at the intersection. Our goal is to accurately compute the percentages at arbitrary intersections

the choice of turning left or right or proceeding straight through the intersection. The option of proceeding straight is likely unattractive for most drivers, because it leads to a small loop road. Turning left or right leads to many more possible destinations. Indeed, the measured proportion of westbound drivers going straight through this intersection was about 1.3%. Our basic algorithm, explained subsequently, estimated the proportion at 0.02%.

The ground truth for the example in Figure 1, and for all our tests, came from our local, municipal government's traffic study data and our own road map data, which we describe next.

2 Turn Counts and Road Map Data

The ground truth data for our research came from road sensors temporarily installed by our local municipality to study traffic volumes. We combined this data with a digital road map to develop our algorithm for predicting turn proportions. This section describes the ground truth and road map data.

2.1 Turn Counts

Our ground truth data consisted of turn counts measured at the 40 intersections shown in Figure 2. Our local city of Redmond, Washington USA contracts with companies to set up tube sensors on the road to count the number of vehicles passing in a given

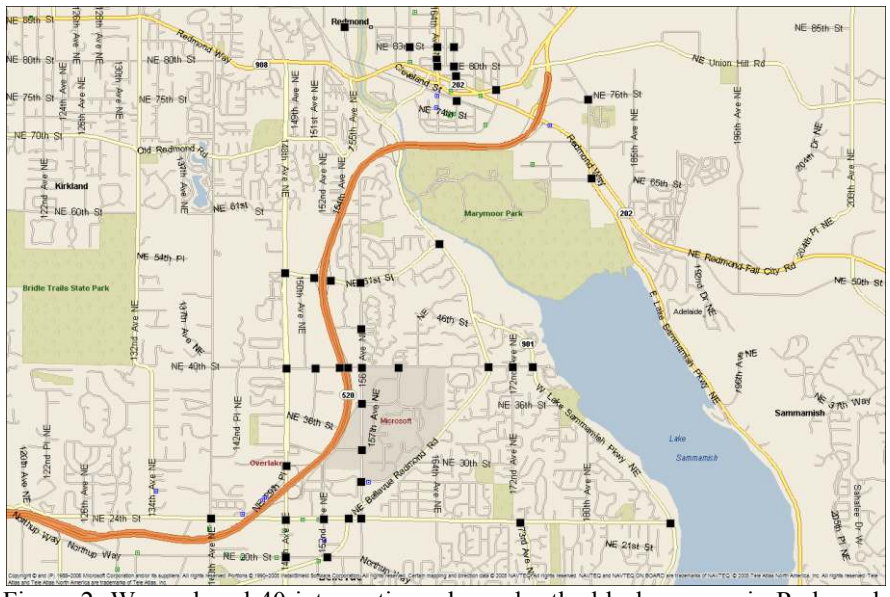


Figure 2: We analyzed 40 intersections shown by the black squares in Redmond, Washington USA.

City of Redmond Turning Movement Counts

Year	Mo	Day	Day_of Week	Peak Period	Northbound Left	Northbound Thru	Northbound Right	Southbound Left	Southbound Thru	Southbound Right	Eastbound Left	Eastbound Thru	Eastbound Right	Westbound Left	Westbound Thru	Westbound Right	Source
020148	NE 20 St and 148 Ave NE																
2001	7	19		MD	187	779	117	238	696	260	317	578	158	217	568	171	BEL
2002	8	1		AM	146	808	112	112	689	190	92	148	56	74	301	121	BEL
2002	6	13		PM	139	736	86	190	1109	209	267	527	160	220	452	109	BEL
2004	9	20		AM	152	1016	115	126	788	139	130	196	60	72	343	138	BEL
2004	8	19		MD	163	710	103	212	729	282	308	519	121	185	534	155	BEL
2004	7	26		PM	167	820	74	211	1164	216	335	554	193	243	553	130	BEL
2005	09	27	Tues	PM	167	753	55	192	1159	152	269	518	161	305	610	118	RED

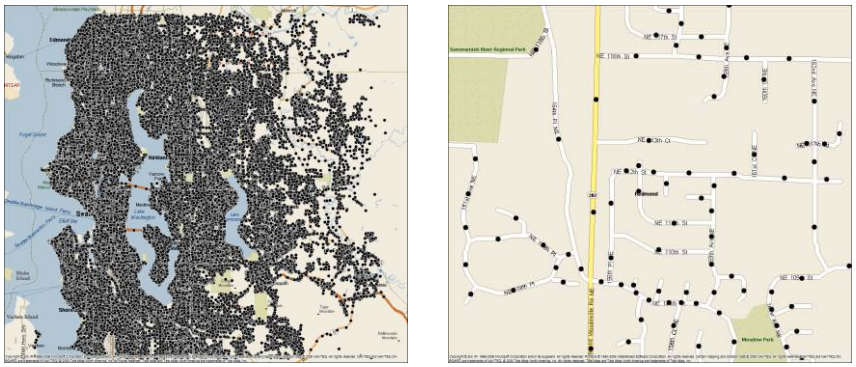
Figure 3: This is a sample of the original pdf file containing the turn count data that we used. We subjected it to an OCR program to extract the numbers.

period of time. With tube sensors deployed on all roads approaching and departing an intersection, it is possible to compute the number of vehicles that took any given departure road for any given approach road. The counts for each intersection are split by the approach road, date, and period of the day (morning, midday, evening). Figure 1 shows example counts for one approach to one intersection during the evening of one day. The turn proportions, which we are ultimately trying to predict, are computed in the obvious way from the count data. Since count data is not available at all intersections in the world, our algorithm seeks to compute the proportions using other data. Turn count data can also be derived from cameras [8].

The turn data we used is available on our city's Web site [2] as a pdf file. A sample of the file is shown in Figure 3. We used an optical character reader, along with some manual editing, to convert the pdf document into a digital spreadsheet. For the 40 intersections we examined, there were a total of 1224 different turn count studies performed dating back to 1994, spread over different days and different times of day.

2.2 Road Map

We used a digital road map as an integral part of our prediction algorithm. Our road map data came from our institution's mapping product group. It gives the geometry and topology of the road network. Roads are represented as road segments, where



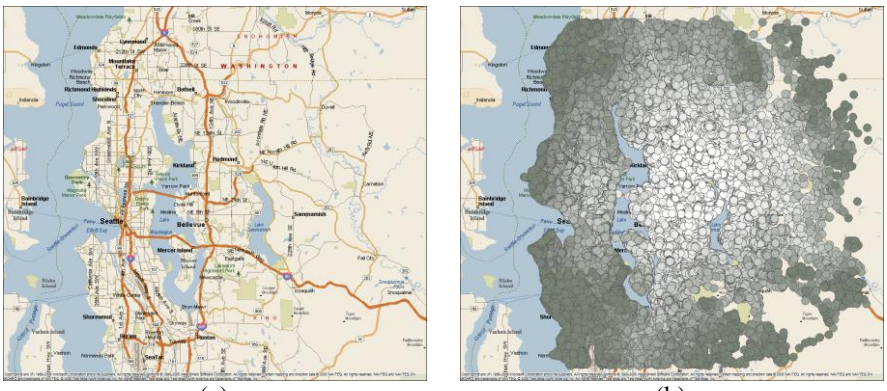
(a) (b)

Figure 4: (a) The black dots mark the centers of the 83,353 road segments we used in our analysis. (b) Close-up

each is typically a short section of road that goes between two intersections. Road segments can also terminate at road name changes and dead ends. Almost all the segments go between two connected intersections, and the average length of a road segment in our area of interest was 131 meters. Each road segment contains its length, approximate speed limit, and allowable driving directions (*i.e.* one-way *vs.* two-way). The two ends of each road segment contain a list of attached road segments, thus representing the road network as a classic graph with nodes (intersections) and edges (road segments).

Our turn prediction algorithms, detailed subsequently, look at the driving times between pairs of road segments to assess which ones are closest in time after each turn choice. Toward this end, we need to know about the locations and driving times on the roads in our area of interest. Since we do not have the storage nor computational power to look at all the road segments in our digital map, which covers North America, we limited our analysis to a 45km × 45km square approximately centered on the 40 intersections we analyzed. The centers of the 83,353 road segments in this square are shown in Figure 4(a), with a close-up in Figure 4(b).

We used these road segment centers for two purposes. One was to anchor the text description of the intersections in the pdf file (Figure 3) with a digital representation that we could compute with. Specifically, for 40 intersections listed in the pdf file, we manually clicked on our map of road segment centers to indicate the road segments corresponding to the intersection's approach and departure roads. The second use of the road segment centers was to compute driving times between all pairs of road segment centers. For this we used the standard Dykstra algorithm, with the cost of driving each road segment computed as the time to traverse it driving at its speed limit. With $N = 83{,}353$ road segments, there were $N(N-1)$ driving times to compute, or about 7 billion. Note that the driving time between two road segments depends on which segment is the start point due to one-way road restrictions. We computed these driving times on a 100-node computing cluster over a period of about one week and stored the results in a database. In the end, this computation gave us a



(a) (b)

Figure 5: Map (a) shows our area of interest. The dots in (b) are shaded proportionately with the computed driving time from a point near the center of the map. Darker shading indicates a longer drive.

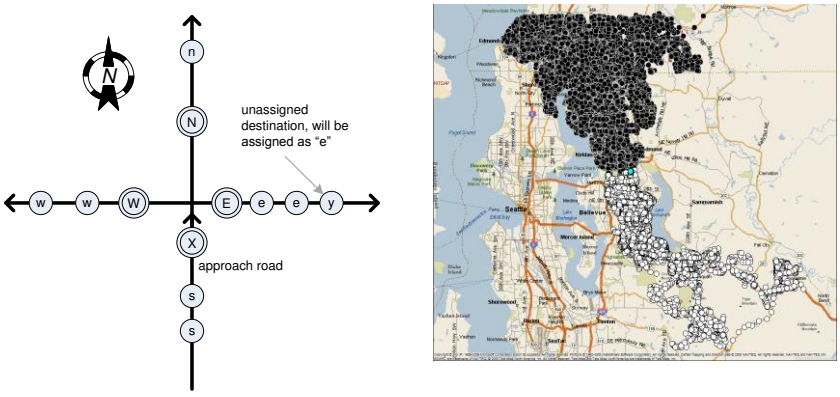
table of driving times between any pair of roads segments in our area of interest. A visualization of the driving times from a point near the center of our map is shown in Figure 5.

These driving times were an integral part of our basic prediction algorithm, which we describe in the next section.

3 Turn Proportion Prediction: Basic Algorithm

Our basic turn prediction algorithm stems from the assertion that drivers rarely turn down dead-end roads, because there are not many destinations available there. This is illustrated in Figure 1, where measurements show that drivers rarely chose to drive into the small loop to the west of the intersection. A second assertion is that drivers take the most efficient route, in terms of time, to their destination. We know there are exceptions to both these assertions, but they do lead to a simple algorithm that works well.

The schematic diagram in Figure 6(a) is the basis of a simplified example to explain our algorithm. The black lines are roads forming an intersection. Entering the intersection from the approach road to the south, a driver has three choices about which departure road to chose: west (left turn), north (straight ahead), and east (right turn). Each circle on the roads represents the center of a road segment, and the driver is currently on the road segment marked with an uppercase X, traveling north toward the intersection. (In actuality, road segments are normally separated by intersections, but we omit the other intersections in this schematic for simplicity.)



(a) (b)

Figure 6: (a) A schematic intersection for explaining our prediction algorithm. The driver is on road segment X and will go toward W, N, or E with probability proportional to the number of destinations along each direction. (b) For the intersection in Figure 1, the black dots show the destinations reachable most efficiently for a turn to the north, and the white dots for a turn to the south. The four barely visible gray dots show the destinations for going straight through the intersection.

Importantly, we assume that each road segment represents a possible destination for the driver. Clearly this is not exactly correct, since a single road segment may contain zero or several actual destinations, like stores, houses, *etc.*. We attempt to mitigate this problem in the first variation of our basic algorithm. But, using this assumption, the centers of the road segments in the schematic diagram serve as possible destinations for the driver, and we assume the driver is headed toward one of them along the most time-efficient route. Our goal is to assign a symbol to each possible destination where the symbol corresponds to which way the driver would turn at the intersection to get there most efficiently in terms of time. For instance, in the schematic, some destinations are marked "e", meaning the driver would turn east (right) to get to those in the least possible time.

To assign a symbol, we compute the most efficient turn that it would take to get there, and assign the symbol according to that turn. Referring to the schematic, we compute the time cost of each turn from the approach road segment X to the three departure roads, marked with uppercase W, N, and E. These time costs are X→W (left), X→N (straight), and X→E (right), which come from the driving time computations described in the previous section. Of course, these time costs are relatively short, usually a few seconds, because the points are so close together. In the schematic, there is an unassigned road segment marked with y. To assign a symbol to this segment, we compute the time cost of driving to it via each of the three possible turns. These time costs are X→W→y, X→N→y, and X→E→y. We find the minimum of these time costs and give the corresponding symbol to y, which in this case would be e, because X→E→y is the minimum time cost.

We also check to see if a candidate destination can be reached more efficiently by making a U-turn. In the schematic, these are marked "s", which means that X→s was less than going through any of W, N, and E.

The predicted turn proportions are simply the proportion of destinations marked with the same symbol. In the schematic, there are three destinations marked W or w, two marked N or n, and four marked E or e (including the just-assigned y). Thus the predicted turn proportions would be W: 3/9, N: 2/9, and E: 4/9. We do not include the U-turn destinations, because U-turns are rare.

Figure 6(b) shows results from this basic algorithm applied to real data for the intersection shown in Figure 1. The black dots are reachable most efficiently by turning north, and these make up 73.23% of the total dots. The white dots (22.28%) are reachable most efficiently by turning south. The four barely visible gray dots (0.02%) are reachable most efficiently by going straight. Note that the dots do not cover all the possible destinations shown in Figure 4(a), because many of the dots are reachable most efficiently by making a U-turn. People familiar with the area would recognize that a U-turn leads quickly to a highway, which is the most efficient way to get to destinations on the western side of the map, where there are no dots in Figure 6(b).

Processing all turn reports, the median absolute error between our predicted proportions and the actual proportions was 0.219. If we process only the reports from the most recent day for each intersection, then the median absolute error drops slightly to 0.192. This is likely because the more recent data reflects changes in the road network, *e.g.* new roads and closed roads.

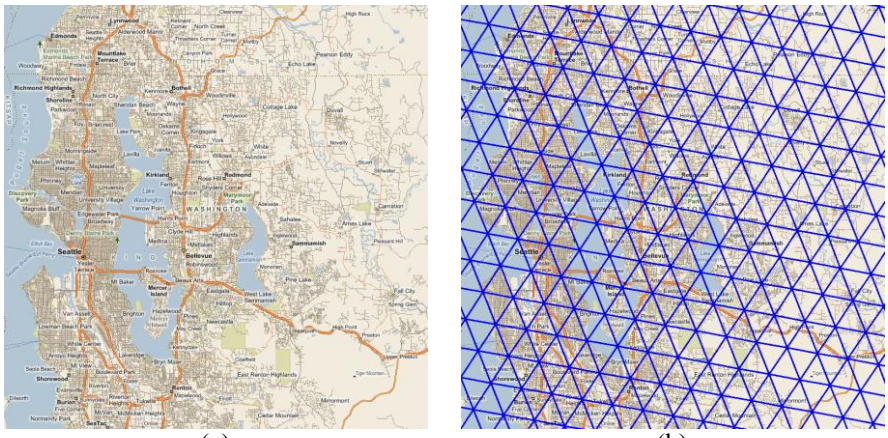
4 Normalizing Out Destination Density: Triangles Variation

One potential problem with our basic algorithm is the non-uniform density of the road segments that we use to represent destinations. A neighborhood with many intersections will have more road segments, and will thus artificially inflate the attractiveness of turning toward it. This non-uniformity is apparent in the road segment centers shown in Figure 4(b).

One way to alleviate this problem is to create a uniform tessellation of the map to normalize the density of the road segments. A convenient tessellation is the Hierarchical Triangular Mesh (HTM) [9], which uses a covering consisting of triangles, as shown in Figure 7. Different levels of the hierarchy give different, nearly uniform sized triangles. Figure 7 shows level 12, and the triangles covering our area of interest have an average area of 5.1 square kilometers. To move to a finer level in the hierarchy, each triangle at the current level is split into four smaller triangles. For our experiment, we used level 12, which covered our region of interest with 640 full or partial triangles.

In the basic algorithm, destinations (road segments) are associated with the turn direction that gives the most time efficient route to the destination. (Destinations that are most efficiently reachable by a U-turn are eliminated.) Each associated destination casts a vote for its associated turn direction, and the votes are normalized to get turn proportions. In this new variation of the algorithm, triangles cast votes instead of destinations. More specifically, a triangle casts one vote for a turn direction if it contains one or more destinations that are associated with that turn direction. If a triangle contains destinations associated with more than one turn direction, it casts one vote for each turn direction.

With this technique, a densely clustered set of destinations should not become over-represented in the final result. In fact, the algorithm modification helps, but only slightly. The median proportion error for the modified algorithm was 0.198 compared



(a) (b)

Figure 7: For our region of interest in (a), (b) shows the 640 level 12 triangles of the Hierarchical Triangular Mesh. The average area of these triangles is 5.1 square kilometers.

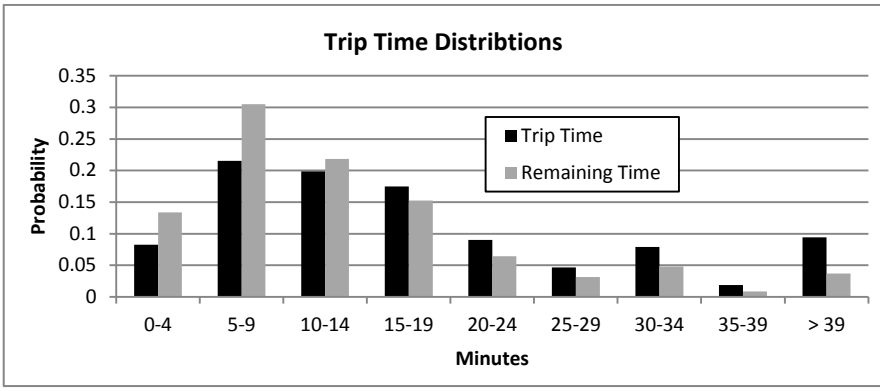


Figure 8: The black bars show a distribution of trip times from [4]. The gray bars show a distribution we derived giving the amount of time remaining in a trip if the elapsed time is not known.

with 0.219 for the basic algorithm. Using only the most recent turn counts at each intersection, the modified algorithm improved the median error from 0.192 to 0.183.

5 Considering Trip Time Expectations: Trip Times Probabilities Variation

We felt justified in limiting our set of candidate destinations to a certain local area, because we know that most vehicle trips are not overly lengthy. In fact, statistics from the U.S. 2001 National Household Transportation Survey (NHTS) [4] give the histogram of driving times for typical trips shown in Figure 8. It seems natural to reduce the effect of candidate destinations that are far away from the intersection in question, since there is a smaller chance of them being the driver's actual destination. This is the modification we explore in this section.

While the black bars in Figure 8 give the probabilities of trips of certain lengths, the drivers we are trying to model have already begun their trip. We do not know how long they have been already driving, so we cannot apply a simple trip time distribution to their candidate destinations. Instead, we must compute a distribution of *remaining* trip time, given that we do not know the elapsed time.

We derived a probability distribution of remaining trip times by imagining two random choices:

1) The driver chooses a trip time according to the trip time distribution in Figure 8 (black bars). Of course drivers do not consciously chose a trip time when they begin a trip, but the NHTS says that their actual trips do implicitly follow the distribution.

2) The imaginary observer picks a random observation time uniformly distributed from zero to the driver's total trip time. For our scenario, this is the time at which the driver reaches the intersection in question.

With these two random variables, the distribution of remaining trip times is computed as the gray bars in Figure 8. Note that the distribution of remaining times skew less than the distribution of trip times. In fact, not counting the ambiguous ">39" bin, the mean trip time is 16.9 minutes, while the mean remaining trip time is 12.2 minutes. This matches our intuition in that a driver's remaining trip time will always be less than the total trip time, unless the trip was observed at the beginning.

We used the probabilities of remaining trip times to modulate the voting weight of destinations associated with different turn directions. Recall that in the basic algorithm, all destinations associated with a given turn direction had an equal vote for that turn direction. In this modified algorithm, for a given candidate destination, we first find the driving time to that destination from the point of the intersection. This is the remaining driving time. We use this to look up the probability of that destination from our probability distribution of remaining trip times. This probability is the destination's vote magnitude. We normalize these magnitudes to compute predicted turn proportions. The overall effect of this scheme is to reduce the weight of votes from distant destinations that the driver is unlikely to have chosen as a destination.

Using the same test data as for the previous two algorithms, the median proportion error of this algorithm variation was 0.204, slightly better than the basic algorithm, but slightly worse than the triangles algorithm. Using only the most recent turn data from each intersection, the median error was 0.183, the same as the triangles algorithm, and better than the basic algorithm's 0.192.

6 Trip Time Weights: Trip Time Weights Variation

The algorithm above using trip time probabilities showed that considering trip times improved performance over the basic algorithm. This algorithm weighted each destination by the probability of the remaining trip time. We also investigated a weighting that simply applied a decaying exponential to the trip times. After associating a candidate destination with a turn choice, that destination's vote for the turn was computed as e^{-at} where t is the computed driving time in seconds. This has the effect of down-weighting more distant destinations. After sweeping through possible values of a, we found that values in the range [0.00,0.05] worked best. Figure 9 shows the median proportion error for our 40 test intersections as a function of a. Note that these values were computed with only the most recent turn counts for each intersection. The minimum occurs at $a = 0.018$, which gave a median error of 0.142. For all the turn counts, including the older reports, the median error was 0.163. Both of these are better than the basic algorithm, whose corresponding median errors were 0.219 and 0.192.

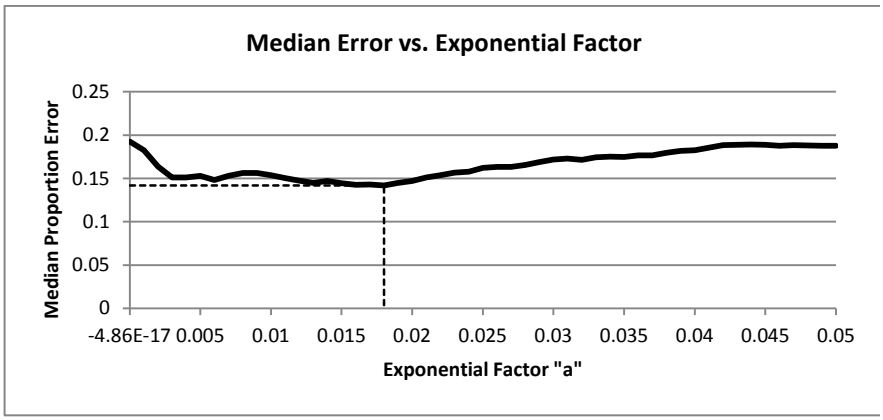
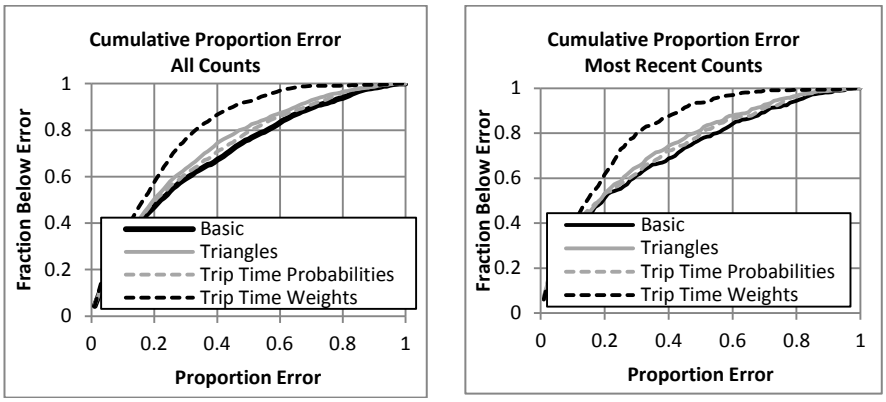


Figure 9: This shows the median proportion error as a function of the exponential weighting factor a using only the most recent turn reports from each intersection. The minimum error is 0.142 at a=0.018.

7 Discussion

The performance of the basic algorithm and the three variations, triangles, trip time probabilities, and trip time weights, is summarized in Table 1 and Figure 10. From these it is apparent that the basic algorithm works fairly well, and that the first two variations help slightly. The fact that the basic algorithm works as well as it does supports our original assertion that the number of available destinations that are



(a) (b)

Figure 10: These are the cumulative error distributions in proportion predictions for the basic algorithm and three variations. (a) shows the error for all the turn count reports, and (b) shows the error for only the most recent daily report(s) from each intersection.

Table 1: Median proportion errors with different algorithm variations and different data sets. Performance is generally better using only the most recent turn count data. "Trip Time Weights" performs best.

		Algorithm			
		Basic	Triangles	Trip Time Probabilities	Trip Time Weights
Median Error	All Turn Counts	0.219	0.198	0.204	0.163
	Most Recent Turn Counts	0.192	0.183	0.183	0.142

efficiently reachable after a turn is a good indication of the turn's popularity. There is a slight improvement in performance when using only the most recent turn count data, which is likely because newer turn counts represent the most current configuration of the road network.

It is somewhat surprising that two of the more sophisticated versions of the algorithm, Triangles and Trip Time Probabilities, improved performance only slightly. The triangle algorithm compensates for the uneven density of destinations, but it may be that, given enough destinations, the density approaches uniform anyway. The trip time probabilities algorithm was designed to account for the expected distribution of trip times. It may be that the intersections we chose to study did not represent uniform samples of drivers' elapsed trip times. This problem is plausible, since our intersections were near a retail center and a large office campus, both of which are likely the start or end points of driving trips.

The last variation of our algorithm, Trip Time Weights, worked best. This algorithm encodes our intuition that distant destination candidates are less important that nearby ones. It also demonstrates the advantage of having an adjustable parameter (the constant a in the exponent) to optimize over.

Other variations of our basic algorithm may work better. Some candidate destinations may be much more attractive than others. Our Trip Times Probabilities algorithm variation tried to assess the relative attractiveness of destinations based on their distance in time, with nearby destinations being generally more attractive. But, other attractiveness factors are probably more predictive, and their attractiveness likely varies by the time of day and the day of the week. On weekdays, commuters drive toward easily describable types of destinations in the morning and evening. Likely more popular on weekends are restaurants and entertainment venues. The locations of different types of businesses as potential destinations are available via digital Yellow Pages listings. An analysis of GPS traces or even machine learning on turn counts could be used to discern drivers' destination preferences. This would probably lead to skewed results using our limited data, however, since we had only 40 intersections in a fairly limited area. In fact, being familiar with the region, the two dominant destinations are a shopping mall and a large software maker.

8 Conclusion

This paper presented algorithms that predict drivers' turn proportions at road intersections. This can be a useful adjunct to algorithms that predict a driver's route,

which is in turn useful for giving advanced warnings and advertising. The basic algorithm is based on the assumption that drivers tend to turn toward directions that give them more possible efficiently reachable destinations. Using road segments as candidate destinations, along with a table of drive times between these destinations, our basic algorithm finds which destinations are most efficiently reachable via each possible turn direction. One variation of the basic algorithm, called Triangles, attempts to normalize out the effects of variations in the density of candidate destinations. Another variation, called Trip Time Probabilitiess, weights destination candidates according to a distribution of drivers' trip times. The basic algorithm worked well, with slight improvements from the first two algorithm variations. A third variation that down-weights distant destinations worked best. We speculated that the general approach could be improved more with a richer representation of the attractiveness of candidate destinations.

References

1. *National Household Travel Survey Daily Travel Quick Facts*. Available from: http://www.bts.gov/programs/national_household_travel_survey/daily_travel.html.
2. *Redmond Turning Movement Counts*. Available from: http://www.redmond.gov/connectingredmond/resources/pdfs/redmondturningmovementcounts.pdf.
3. Froehlich, J. and J. Krumm, *Route Prediction from Trip Observations*, in *Society of Automotive Engineers (SAE) 2008 World Congress*. 2008: Detroit, MI USA.
4. 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.
5. Krumm, J., *A Markov Model for Driver Turn Prediction*, in *Society of Automotive Engineers (SAE) 2008 World Congress*. 2008: Detroit, MI USA.
6. Krumm, J. and E. Horvitz, *Predestination: Inferring Destinations from Partial Trajectories*, in *UbiComp 2006: Ubiquitous Computing*. 2006: Orange County, CA USA. p. 243-260.
7. Patterson, D.J., et al., *Inferring High-Level Behavior from Low-Level Sensors*, in *UbiComp 2003: Ubiquitous Computing*. 2003: Seattle, WA USA. p. 73-89.
8. Song, K.-T. and J.-C. Tai, *Image-Based Turn Ratio Measurement At Road Intersection*, in *IEEE International Conference on Image Processing (ICIP 2005)*. 2005: Genoa, Italy. p. 1077-1080.
9. Szalay, A.S., et al., *Indexing the Sphere with the Hierarchical Triangular Mesh*. 2005, Microsoft Research Technical Report MSR-TR-2005-123.
10. Weissler, P., *Inside OnStar*, in *Automotive Engineering International*. July 2008, Society of Automotive Engineers. p. 34-36.