Learning to Associate Observed Driver Behavior with Traffic Controls

Christopher A. Pribe [1]   Seth O. Rogers [2]

DaimlerChrysler Research and Technology Center
1510 Page Mill Road
Palo Alto, CA 94304

[1]Phone: (650)845-2547 e-mail: pribe@rtna.daimlerbenz.com, pribe@pribe.net
[2]Phone: (650)845-2533 e-mail: rogers@rtna.daimlerbenz.com

Abstract

Adaptive techniques support the development of new tools to help traffic engineers classify and evaluate traffic flow at intersections. We describe a tool that learns to associate driver behavior with a subset of traffic controls, e.g. stoplights and stop signs. In the case where the traffic controls for an intersection are not readily available or are unknown, the tool automatically identifies the traffic controls present at an intersection from observed driver behavior. This capability may be used to augment digital maps with traffic control locations. In the case where traffic controls are known or have previously been classified, the tool flags instances where driver behavior is inconsistent with the traffic controls actually present. This capability might be used by various services for drivers such as dynamic routing and new safety systems. It might also be used by traffic engineers to evaluate control placement in real or simulated road networks by finding situations that elicit unusual driver behavior. We calibrate the tool with driving data on a set of segments with known controls. The tool first learns to identify controls present on individual road segments, then uses hand-crafted rules to verify control consistency across segments at intersections. The data set comprised real-world position data collected during normal daily driving. The tool accurately identified 100% of the data that passed verification. These results encourage us to believe that the system can provide traffic engineers with a reliable mapping between driver behavior and traffic controls.

1 INTRODUCTION

This paper describes a prototype system that learns to associate driver behavior with traffic controls. The system combines adaptive and hand-crafted components in a way typical of many industrial applications. The adaptive component is a supervised neural network trained to identify controls on individual road segments using measures of driver behavior as input. We preferred accurate knowledge extraction over complete area coverage during system design because applications that affect driver safety may employ the system[1]. The hand-crafted component increased accuracy by using rules about roads and intersections to reject suspicious results.

We intend our system to locate traffic controls and detect anomalous driver behavior. The system can serve to improve many telematic services, such as trip information and safety services[2]. One application is finding the locations of traffic controls and augmenting digital maps with these locations. Another application is detecting traffic disruptions by comparing expected behavior, as defined by controls stored in the augmented map, with dynamically measured behavior.

The adaptive and hand-crafted components of the system are described below in §2. The prototype system described here was designed to identify those controls related to stopping, namely stop signs, traffic lights, and clear intersections. System performance was excellent as discussed in §3. We discuss possible future work and applications in §4. Finally, we summarize our contribution in §5.

2 ADAPTIVE AND HAND-CRAFTED COMPONENTS

We divided learning to associate driver behavior with traffic controls into two tasks. The first task is to identify the control present on a road segment. The second is to validate identification by checking the consistency of controls at segment connections. We treated the identification task as a supervised learning problem. We solved it with a supervised neural network as described below in §2.1. We wrote rules that compare the controls on immediately connected segments and used them to validate identification as described in §2.2.

2.1 Control Identification on Individual Road Segments

We trained a supervised neural network to identify controls on individual road segments. The inputs to the neural network were statistics about driver behavior on a given segment. In order to compute these statistics, we collected geographic position data from drivers as described below in §2.1.1. We computed road segment data from geographic position data using a digital road map[3]. We used both geographic and road segment data to compute measures of driver behavior related to stopping. As described in §2.1.2, we applied a number of filters to the data at various stages and computed statistics on the data from all traversals, or passes of a vehicle over a segment. Direction matters, so traversals of a given two-way road segment in one direction are kept separate from traversals in the other direction.

The types of controls we chose to identify — stop signs, traffic lights, clears — govern stops at intersections. We designed our input representation to reflect stopping behavior. We chose inputs that showed how often and how long drivers stopped. We also designed our representation to support discrimination between multiple short stops and one or two long stops so as to capture the qualitative differences between stops at stop signs and lights.

The neural network had eleven input values. These included the average and the standard deviation of each of the following: number of times stopped, total duration of all stops, the durations of the three stops closest to the end of the segment. We also computed the percentage of traversals that included at least one stop for each segment. Though we did not formally analyze the utility of the inputs, we observed that the most meaningful predictor was the percentage of traversals that included a stop.

We trained the neural network to recognize clear road segments (those with no restrictions), segments with stop signs, and segments with traffic lights. We determined ground truth, i.e. the control locations, from direct observation and from maps obtained from the Transportation Department of the City of Palo Alto.

2.1.1 Collection of Driver Behavior Data

An on-board computer monitored driver behavior by recording Global Positioning System (GPS) position signals during trips. GPS receivers pinpoint geographic position by triangulating on radio signals from GPS satellites. These receivers generate a position estimate using latitude, longitude, and altitude. They also provide other information, such as a time stamp for each position detection.

We used real-world, differential GPS data collected from two different drivers and vehicles during the course of normal daily driving. Differential GPS provides more accurate estimates of position than basic GPS. Differential GPS uses fixed location receivers on earth to determine the current error in the GPS signal. A radio station transmits a correction for this error to be used by receivers to improve the accuracy of basic GPS. We used a Garmin integrated satellite antenna/receiver, a DCI differential corrections unit, and a LG Phenom hand-held PC to detect and record position. Estimated position accuracy is 5-10 meters.

We employed a map matching procedure to reconstruct the path taken during a trip. This procedure assigns each recorded position to exactly one of the road

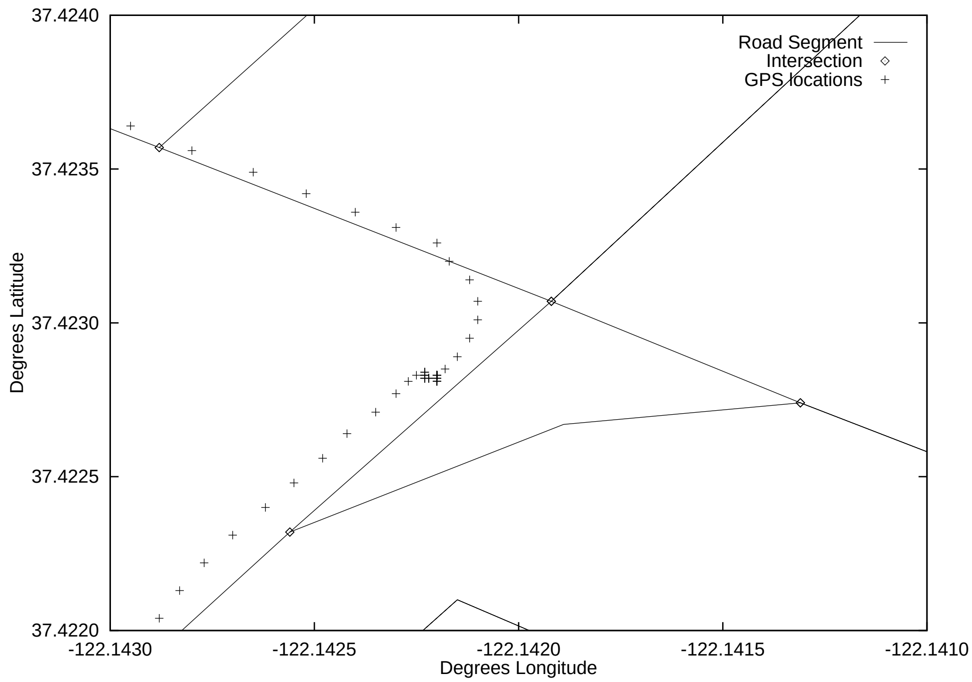

FIGURE 1: A road trip is recorded as a set of geographic positions that must be assigned to road segments in order to be used to represent driver behavior.

segments stored in the digital road map. Since the segment sequence describes the path taken during a trip, each segment in the sequence must be connected the next. Noise in the position estimates and the segment locations stored in the digital map introduce uncertainty into map matching. We implemented a backtracking best-first-search algorithm to find the most likely path. This algorithm finds the path nearest to entire sequence of positions for a trip. The algorithm reduces overall uncertainty and eliminates discontinuous paths by summing distance over many segments rather than over just one segment.

To begin, the algorithm adds those segments in the digital road map that are closest to the first point in the position sequence to a path list. It iteratively expands paths on the list starting with the path with lowest average distance between the positions so far and the segments to which they are assigned. It expands a path by 1) adding all neighboring segments to the path list and 2) computing a new average distance. The list is pruned at each iteration so that at most 100 nearest paths remain. At the end, the output of the algorithm is a list of segment identifiers, the direction traveled on the segment, and a time stamp indicating when the driver entered that road segment.

2.1.2 Computation and Preprocessing of Segment Statistics

We took the following steps to preprocess segment data and compute segment statistics:

1. Insure consistency of data: The data preprocessor, comprised of several UNIX shell scripts and 'C' programs, excluded data that reflect a turn at a four way intersection. Drivers can behave differently when turning than when continuing straight through an intersection. For example, it is legal to make a right-hand turn after a making a stop when the light is red at some stop lights in California.

2. Detect stops: The data preprocessor detected a stop whenever a driver's speed dropped below a threshold. It used a threshold greater than zero to compensate for variability in the GPS position estimate and to allow detection of stops whose duration was less than the position sampling rate of 1 Hz.

3. Record stop data: The data preprocessor counted and recorded the number and duration of stops. It recorded the three stop times for the stops closest to the end of the segment.

4. Match data to ground truth: Drivers drove on some roads for which we did not possess ground truth; we did not use this data. The preprocessor selected

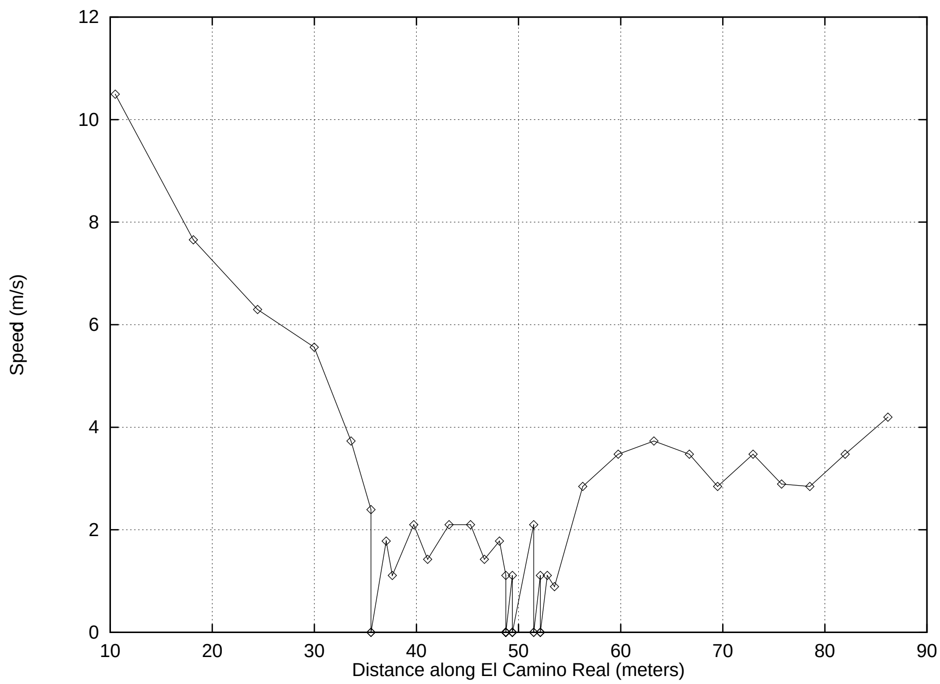

FIGURE 2: This graph of driver speed along a road segment shows that speed estimates derived from GPS data exhibit noise. The system records a 'stop' when speed drops below a threshold rather than when it reaches zero to compensate for this noise and to detect stops whose duration is smaller than the 1 Hz GPS sampling rate.

those segments for which we had ground truth from files containing segment traversal data.

5. Average traversal data: The preprocessor computed the average and standard deviation statistics for each measurement. It computed the percentage of traversals in which at least one stop occurred. If there were 5 or more samples, it created an input instance from the computations. That is, a single input instance comprises the statistics computed from all the different traversals of a given road segment in a particular direction.

6. Reject known ambiguous instances: The preprocessor filtered out instances that exhibited infrequent stops. It removed cases where the percent of traversals with a stop was greater than zero but less than 30%. This data is ambiguous. We explain why and discuss how to reduce this ambiguity below in §4.

7. Exclude intersections with short segments from training: Some intersections included very short segments on the digital road map. It is difficult to correctly assign GPS data to road segments in intersections with short segments. We excluded this data from training, but allowed it during testing.

2.1.3 Supervised Neural Network Classifier

A supervised neural network classifier learned to associate driver behavior with roadway traffic controls. The input layer of the two layer network had 11 nodes and the output layer had three nodes. The neural network was fully connected: all output nodes received input from every one of the input nodes.

We constructed the neural network classifier using the Aspirin/Migraines package [4]. It used a learning rate of 0.05 and inertia of 0.95. It ran 50,000 iterations to train and updated the weights each time training data were presented.

We used K-fold cross validation, where K = 10, to evaluate the performance of the network. K-fold cross-validation works as follows:

1. Split a data set of N instances into K cuts containing (N/K) random instances.

2. Form a testing set with each cut.

3. Form a training set for every testing set with the remaining (N - N/K) instances.

4. Train and test the neural network using each of the pair of training and testing sets.

5. Record and average the results for the testing sets to determine the performance of the network.

2.2 Control Consistency Checking Using Rules

Drivers expect the controls on all of the roads entering an intersection to be consistent. That is, when drivers see a green light they expect that cross traffic also has a traffic light (and that it is red). We used rules to insure that identified controls were consistent at segment connections. Roads are broken into segments in the digital road map. The points at which road segments connect are called nodes. Nodes typically indicate either an intersection or the continuation of a road. We wrote a set of rules for intersections which also encompasses continuations.

Note that there are some situations, such as that shown in Figure 3, where more than one node in the digital map may correspond to what most drivers would consider a single intersection. These intersections can produce data that violate our rules. These intersections are indicated by short segments in the digital road map. We did not use segment data from intersections with short segments during training as discussed in §2.1.2.

Segments for intersections that violate the rules below are eliminated by the post processor. We assume that any type of control may be used at any intersection.

Intersection Rules – Only the following intersections are acceptable:

1. n-way clear,
2. n-way light,
3. n-way stop, and
4. (n-2)-way stop.

In order to perform multi-segment analysis at an intersection, data must be available for at least two of the segments entering the intersection. We only used the intersection rules to reject suspicious data in the work reported here. However, they can also correct control identification errors, if data for all or most of the segments is available.

3 PERFORMANCE ON REAL-WORLD DRIVING DATA

The system was 100% accurate at single segment control identification for those identifications that also passed the intersection rules. Table 1 shows the performance results. In the table, “Raw recognition” refers to the performance of the

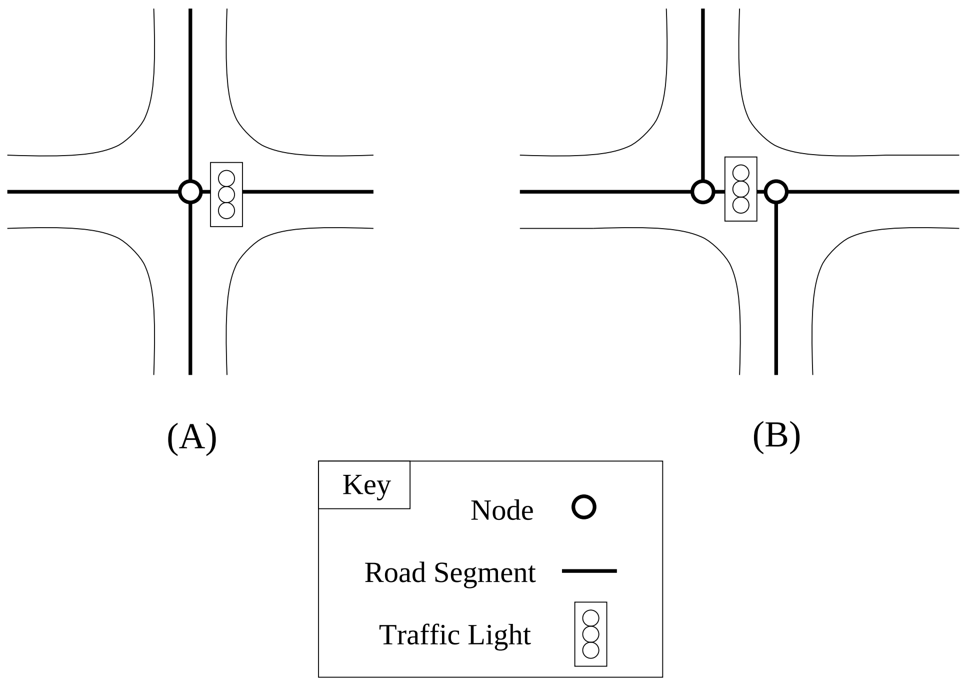

FIGURE 3: Normally, a traffic light at an intersection will correspond to a single node in a digital map as in **(A)**. However, there are some cases where a single intersection may be covered by more than one node as in **(B)**. Correctly assigning the traffic light to the road segments can lead to intersections which violate the rules used by the system.

	Instances	Correct	Percent
Raw Recognition	100	90	90.0
Intersection Threshold	59	56	94.9
Legal Ground Truth	57	54	94.7
Legal Test Results	50	50	100.0

TABLE 1: Filtering the single segment control identifications using the multi-segment intersection rules yields 100% accuracy.

classifier alone. Although raw recognition was good, it is not high enough for some applications, such as vehicle safety.

Some intersections only had one entering segment that passed the minimal sampling criteria. Since these intersections trivially satisfy one of the intersection rules, they were excluded. “Intersection Threshold” refers to the requirement that at least two segments be available to include all the segments of an intersection.

In rare cases, the ground truth for some segments violated the intersection rules. “Legal Ground Truth” refers to system performance after the system has rejected these segments.

Finally, “Legal Test Results” refers to the system’s performance after it had rejected results for test data that violated the intersection rules. A sample legal intersection is all four entering segments classified as “stoplight.” A sample illegal intersection is three segments classified “stoplight” and one classified “stop sign.”

4 DISCUSSION

The system is useful: It can identify controls at many intersections with a high degree of confidence. It can also reject questionable identifications. The limitations of the work are described in §4.1. The system developed is suitable for applications such as those described in §4.2.

4.1 Limitations and Next Steps

The small size of the data set limited our work. Various processing stages eliminated almost half of the raw data. This was due in part to the fact that real-world driving samples roads unevenly. The drivers tended to move quickly to major thoroughfares during normal driving, so they visited many small roads only rarely.

We restricted ourselves to the identification of only three traffic controls. A small fleet of instrumented vehicles would sample a larger number of roads and would provide sufficient traversals to eliminate under-sampling. It would also provide the data to explore other types of controls, such as traffic lights for turns.

We eliminated data exhibiting infrequent stops, because it is ambiguous. It is difficult to discriminate between a clear segment that shows an occasional, random stop from a segment with a traffic light that is infrequently red. Confusion can also result from traffic backup from segments with lights onto clear segments.

Other solutions, besides simply eliminating the data, are possible. For example, the multi-segment post-processor might exploit characteristics of the distribution of stop positions to disambiguate data exhibiting infrequent stops. The frequency histogram in Figure 4 shows that the distribution of stop positions is qualitatively different on clear segments than on segments with traffic lights. This approach requires many more samples for each segment than we had available for the work reported here.

4.2 Example Applications

The major advantage to our technique is that is reflects the actual behavior of drivers, so it allows traffic engineers to observe the effect of the current distribution of traffic controls. Our segment labelling component enables the following three applications:

1. Populate a map database with traffic controls as motorists react to them. There is no need to manually encode information from paper maps or observation, or to integrate information from local, county, state, and national agencies. There is no danger of outdated information or data entry error. In a scenario where GPS traces are available for many cars, adequate coverage for a target area may be built up very quickly.

2. Detect road incidents by monitoring changes in driver behavior. If drivers on a segment that had previously been labelled clear begin to behave as if there was a stoplight or stop sign, there may be an accident or other traffic interruption. If drivers on a segment with a stoplight act as if it was a stop sign, the stoplight may be broken.

3. Assist in redesigning road networks to improve traffic flow. If a segment is clear, but is often relabelled as a stoplight, it is a good candidate for installing stop lights to reduce congestion. Alternatively, traffic engineers may define the acceptable characteristics of stoplights and stop signs, using the metrics defined in Section 2.1. If analysis shows that a particular intersection's characteristics are unacceptable for its label, the engineer may consider re-

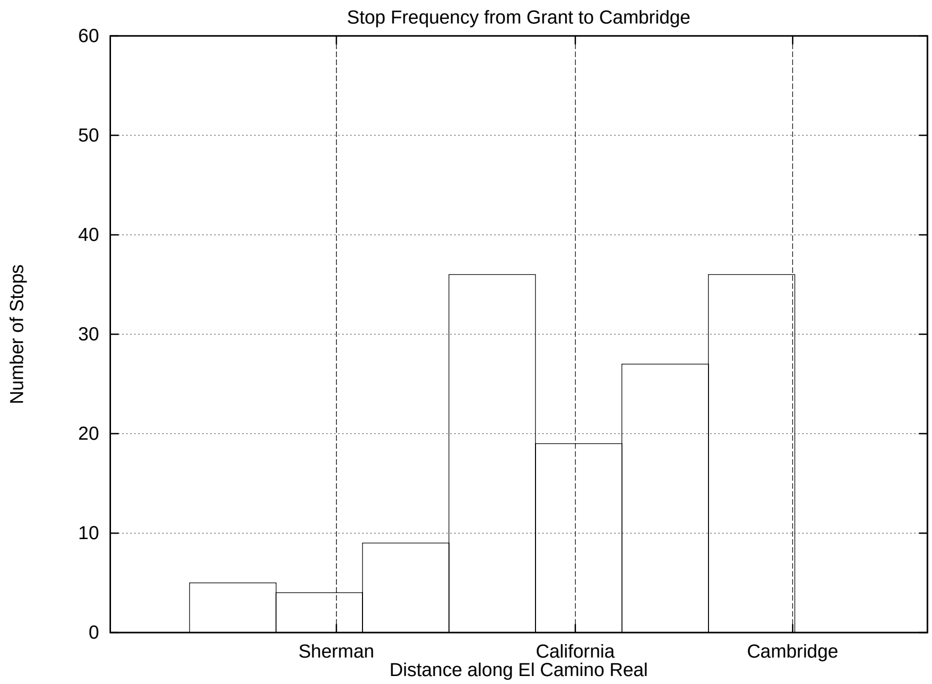

FIGURE 4: This frequency histogram of stop positions shows that the number of stops jumps prior to each light. In this figure, traffic travels from left to right along El Camino Real. There are lights at the labels for the California and Cambridge cross streets, but no control at the label for Sherman cross street. The jump flags clear segments that abut segments with traffic lights that are prone to traffic back-ups. That is, drivers may occasionally stop on a clear segment due to a traffic backup, but the frequency histogram of stop positions for that segment will not exhibit the characteristic jump that occurs before a light.

designing the intersection. If a suitably accurate traffic flow simulation is available, hypothetical road configurations may be tested.

5 CONCLUSION

The system presented forms associations between measures of driver behavior and various traffic controls. Among other things, the cost of fielding a dynamic routing service might be reduced by using the system as an event detector. The system can detect how and perhaps why drivers are stopping, in addition to detecting that they have stopped. Inferences from position traces, such as demonstrated in this system and others[5], can improve digital road maps and provide driver and roadway sensitivity to future driver warning safety systems.

References

[1] Christopher K. H. Wilson, Shawn Weisenburger, Seth Rogers, and Christopher A. Pribe. Position Aware Safety Systems: Interim report on technical progress at DaimlerBenz RTNA. Technical Report RTC Report 11/1998, Daimler Benz Research and Technology North America, Inc., 1998.

[2] Christopher A. Pribe. Integration of distributed telematic sensors and services. Technical Report RTC Report 20/1998, DaimlerChrysler Research and Technology Center, 1998.

[3] Navigation Technologies, Sunnyvale, CA. *Software Developer's Toolkit*, 5.7.4 solaris edition, December 1996.

[4] Russel R. Leighton. *The Aspirin/MIGRAINES Neural Network Software User's Manual.* The MITRE Corporation, release v6.0 edition, 1992.

[5] Christopher K. H. Wilson, Seth Rogers, and Shawn Weisenburger. The potential of precision maps in intelligent vehicles. In *IEEE International Conference on Intelligent Vehicles*, pages 419–422, Stuttgart, Germany, October 1998.