Sensor Sequence Modeling for Driving

Kari Torkkola[☺], **Srihari Venkatesan**[†], **and Huan Liu**[☼]

[☺]Motorola Labs, 2900 S. Diablo Way, Tempe, AZ 85282, USA, Kari.Torkkola@motorola.com
[†]Overture Services, Pasadena ,CA 91101, USA
[☼]Arizona State University, Department of Computer Science, Tempe, AZ 85284

Abstract

Intelligent systems in automobiles need to be aware of the driving and driver context. Available sensor data stream has to be modeled and monitored in order to do so. Currently there exist no building blocks for hierarchical modeling of driving. By semi-supervised segmentation such building blocks can be discovered. We call them *drivemes* in analogy to phonemes. More parsimonious modeling of driving becomes now possible.

Introduction

There is increasing proliferation of portable and fixed devices in cars, such as navigation modules, entertainment devices, real-time information, and communication equipment. There are also increasing demands to manage the distractions they present to the driver. This in turn raises the need to develop intelligent driver assistance systems that, for example, manage the presentation of information to the driver from various devices or subsystems in the car, or alert the driver when his or her attention is not where it should be. The basic philosophy is never to take the control from the driver, but only assist the driver. One necessary sub-component of such an intelligent assistance system is a driving situation classifier that detects difficult driving situations requiring full attention of the driver, and then acts as a gate to information presentation from other devices to the driver. Another component could be a system detecting where the attention of the driver is directed. Such subcomponents can be designed in two fundamentally different ways. One can either program them heuristically, using common sense knowledge, or, such systems can be learned from data. The latter approach based on machine learning is what we describe in this paper.

Context detectors or classifiers observe the available sensor stream and produce a label, perhaps also a probability, of a meaningful context state as their output. The classifier can be constructed for instantaneous operation, that is, it operates on instantaneous snapshots of the sensor data labeling them independently of the preceding or succeeding time samples of sensor data vectors. This may be a preferable approach in time-critical applications, where no delay is tolerable. On the other hand, an instantaneous snapshot may not be sufficient when the context depends on earlier context. An alternative is sequential modeling. A driving maneuver typically consists of a sequence of actions that repeats with variations. These variations, as reflected in the sensor data stream, need to be captured in a sequential model. Speech recognition community has modeled the speech signal using stochastic graphical models, Hidden Markov Models (HMM). We take the same approach in modeling the sensor stream acquired from an automobile.

This paper describes attempts to discover useful subunits in sequential modeling of driving. The structure of this paper is as follows. We begin with a motivation drawing the analogy between driving maneuver recognition and speech recognition. We discuss HMMs in sequence modelling. The specifics of the driving data and its modeling are discussed next. The core of the paper is the section describing experimentation in discovering the drivemes. An overview of other relevant work and discussion concludes the paper.

Motivation

Subunits in Speech Modeling

In automatic speech recognition (ASR) the speech signal is converted into a sequence of acoustic parameter vectors. Most commonly, these are so called cepstral vectors, computed at a rate of 100Hz. ASR then models this stream of parameter vectors as a sequence of subunits (Rabiner, 1989). The basis of these subunits is linguistic, because the whole purpose of speech communication is, of course, to convey a message. The smallest subunit that can change the meaning of a speech message is a phoneme. Acoustic realizations of phonemes are called phones, which is typically the lowest level subunit modeled in ASR.

The existence of these well-defined subunits enables hierarchical modeling the speech signal. There is no need to construct a distinct acoustic model for each word in the language. It suffices to construct a model for each phone(me). Words are then modeled as concatenated phoneme models according to a pronunciation dictionary, and sentences are concatenations of word models according to some kind of a language model. Thus, there is a small number of subunits that are shared in the next level of modeling hierarchy. In learning these models, training data becomes utilized much better because of parsimony in representation.

From Phonemes to Drivemes

Modeling driving shares some aspects with speech recognition. From the sensor point of view, driving in a car produces a stream of parameters from various different sensors. From a driver assistance system point of view, the sensor stream needs to be segmented in time into different context classes that are relevant to driving. The "sentence" of driving should be segmented into "words" of driving, that is, maneuvers. However, no "phonemes" exist for driving. Thus each different maneuver has to be modeled as a discrete entity with no shared parts. We attempt to discover such subunits for the purposes of modeling driving sensor data. We call these subunits "drivemes".

Background

Hidden Markov Models

Hidden Markov Models (HMM) are a sequence modeling and recognition technique that is the dominant approach in automatic speech recognition. HMMs are doubly stochastic models where the underlying stochastic process is hidden and is only indirectly observed by another set of stochastic processes that produce the observation sequence (Rabiner 1989). They model observation sequences as states and transitions between states. An HMM is characterized by the following

1. N, the number of states in the model. The states are generally interconnected with each other. The states are identified by $S_1, S_2, \ldots, S_N$.
2. M, the number of distinct observation symbols given by $v_1, v_2, \ldots, v_M$.
3. State transition probability distribution $A = \{a_{ij}\}$ denotes the probability of transition from state i to state j.
4. Observation symbol probability distribution $B = \{b_i(k)\}$ denotes the probability of observing symbol v_k at state j.
5. Initial state distribution $\pi = \{\pi_i\}$ which gives the probability of choosing the i^{th} state as the first state for generating the sequence.

Efficient dynamic programming algorithms exist for calculating the probability of observing a given sequence by the model, and for computing the most likely state sequence generated by the model for the given observation sequence. The HMMs can be trained from the data by Maximum Likelihood estimation using an Expectation-Maximization-based algorithm. Rabiner (1989) presents a comprehensive overview of these algorithms.

These algorithms do not make any assumptions about the observation distributions used in each state or about the HMM topology. The presence of self transitions allows the model to repeat the same state several times and thus match a sequence of (almost) arbitrary length, even varying proportions of the observation sequence matching each state. These properties render HMMs their flexibility and theoretically make them possible to model any source of data. In this research, HMMs were used to model the various maneuvers of a driver and the drivemes.

Sequence Modeling for Driving

We define drivemes as data units that occur as common patterns among the various driving maneuvers. Since these are sequences on their own they could be modeled as separate Hidden Markov Models. The general idea is to model each maneuver via HMMs and find the states or sequence of states that are common to the maneuvers. These common states or state sequences represent the drivemes and can be used as building blocks to model the various maneuvers of a driver.

The steps involved in learning the drivemes can be summarized as follows:

1. Collect driving data.
2. Annotate the data for different maneuvers.
3. Build HMMs for each maneuver using the features/sensors that discriminate them the most.
4. Cluster the states of these HMMs. All those states that are similar will belong to the same cluster. These clusters will represent the common patterns among the various maneuver models.
5. Build tied state maneuver models using these cluster states. These clusters states or cluster state sequences that appear in common to more than one maneuver will characterize the various drivemes.

Each of the above steps are detailed in the following subsections.

Driving Simulator Environment

The experiments were conducted in a driving simulator lab, which is an instrumented car in a surround video virtual world with full visual and audio simulation (although no motion or G-force simulation) of various roads, traffic and pedestrian activity. The driving simulator consists of a fixed based car surrounded by five front and one rear screens. All driver controls such as steering wheel, brake, accelerator are monitored and affect the motion through the virtual world in real-time. Various hydraulics and motors provide realistic force feed back to driver controls to mimic actual driving.

The basic driving simulator software is a commercial product with a set of simulated sensors that, at the behavioral level, simulate a rich set of current and future onboard sensors in the near future. This set consists of a radar for locating other traffic, a GPS system for position information, a camera system for lane positioning and lane marking, a mapping data base for road names, directions,

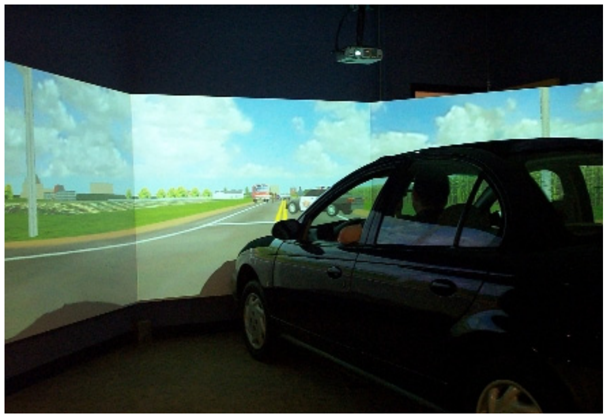
Fig 1. The driving simulator

locations of points of interest etc. There is also a complete car status system for determining the state of engine parameters (coolant temp, oil pressure etc), driving controls (transmission gear selection, steering angle, window and seat belt status etc.). The simulator setup also has several video cameras, microphones and eye tracking infrared sensors to record all driver actions during the drive that is synchronized with all the sensor output and simulator tracking variables. Altogether there are 425 separate variables describing an extensive scope of driving data – information about the auto, the driver, the environment, and associated conditions. An additional screen of video is digitally captured in MPEG2 format, consisting of a quad combiner providing 4 different views of the driver and environment. Combined, these produce around 400Mb of data for each 10 minutes of drive time.

Data Collection and Annotation

The simulation package also has an authoring tool component, which was used to create the driving world scenario. The virtual drive simulated a six kilometer square multi-lane beltway with on and off-ramps, overpasses, and two and three-lane traffic in each direction (separated by a grass median). Interior to the squared beltway, connecting to each mid-side overpass, were four varied two-lane roads - urban, suburban, industrial, and rural environments. These crossed in the middle at a light controlled intersection. All drivers used daytime dry-pavement driving conditions with good visibility. For a high-density driving environment, 59 "distracter" vehicles were added to the highway scenario along with high-density random "ambient" traffic. These distracter vehicles were randomly programmed to drive between ±10 percent faster/slower than the posted speed limit, providing the impression of a steady flow of normal traffic around the subject car. All distracter vehicles simulated alert, "good" driver behavior and reacted reasonably to any particular maneuver from the subject driver. This arrangement allowed a variety of traffic conditions and road types within a confined, but continuous driving space. Opportunities for passing and being passed, traffic congestion, and different levels of driving difficulty were thereby encountered during the drive.

Data was collected from four drivers each driving about 15 minutes in the simulated world. Drivers were instructed to engage in all possible maneuvers they could think of. The data is labeled for maneuvers by manually hand annotating the data using a special purpose tool that combines video playback with graphical visualization of the selected variables from the data.

The data was labeled with the following 12 maneuvers: ChangingLeft, ChangingRight, CrossingShoulder, Not-OnRoad, Passing, Reverse, SlowMoving, Starting, Stopping, Tailgating, TurningLeft, TurningRight, and U-Turn. The labels could be overlapping, say every Passing Maneuver is a sequence of ChangingLeft followed by ChangingRight.

TABLE I
BASE FEATURE SET

Feature	Type*	Description
Accelerator	Cont	Normalized accelerator input value
Brake	Cont	Normalized brake input value
Speed	Cont	Speed of the subject (m/s)
Steering	Cont	Normalized steering angle
TurnSignal	Disc	Status of Indicator lights
AheadLaneBearing	Cont	Bearing of the current lane 100 meters ahead
CrossLaneAcceleration	Cont	Acceleration during lane changes (m/s^2)
CrossLaneVelocity	Cont	Velocity during lane changes (m/s)
distToRightLaneEdge	Cont	Distance to the edge of the right lane (m)
distToLeftLaneEdge	Cont	Distance to the edge of the left lane (m)
laneOffset	Cont	Offset relative to the center of the lane
lateralAcceleration	Cont	Acceleration in the direction perpendicular to the motion of the vehicle (m/s^2)
HeadwayDist	Cont	Distance from the subject's front bumper to the rear bumper of any other vehicle ahead. (m)
HeadwayTime	Cont	Time in seconds to vehicle ahead.

*Type denotes the feature type indicating whether the feature is Discrete or Continuous.

Feature Selection for Maneuver Classification

From many of the available features, a subset of 15 features was selected as base features that could be foreseeable to be built into a car. They are described in Table I. An economically very viable alternative to adding more sensors is to calculate newly derived variables from existing sensors. Of course, one cannot add new information to the sensor signal just by processing it, but it is possible to make important information more explicit. In order to enhance the feature set, the following features were added to the base feature set:

1. Quadratic terms, i.e., all the cross products and squares, of the numeric variables
2. First order time derivatives of the numeric features.
3. Second order time derivatives of the numeric features.
4. Short-time running entropies for steering, brake, and accelerator prediction errors. Entropy is calculated as described by Boer (2000), but within a sliding window.
5. Multivariable stationarity with delta=2 and 3 samples.
6. The output of a quadratic classifier trained using standard least-squares approach.

Adding theses features makes the total size of the feature set to 138. Feature selection is applied to this dataset for selecting the most relevant sensors for maneuver classification. In addition to selecting the best base sensors, we also evaluate what derived features are relevant to this problem. Random Forests based feature selection (Breiman 1989) has given satisfactory results for instantaneous classification of driving data (Torkkola et al 2004). The top 32 most significant features were selected for training the maneuver models.

HMM Modeling of Maneuvers

In the case of driving, each observation is a time sample of all available sensor readings as a vector. The basic assumption is that the sequence of observations from a maneuver is different from a sequence of observations from a non-maneuver. One HMM for each class is trained to model these time sequences, or, trajectories in parameter space. Fig 2 depicts maneuver-based HMM.

In the training process, example sequences of the desired classes are first collected and are then used to train, or estimate, the parameters of corresponding models. In the recognition phase, an unknown sequence of sensor data is presented to each of the trained models, each of which produces a probability of that sequence having been generated by the model. Maximum probability provides the classification result. In practice, since we do not have isolated sensor data sequences with the task of determining whether they are a maneuvers or not, but rather a continuous stream of sensor data, HMMs are used to detect the most probable segmentation of the stream to maneuvers and non-maneuvers, that is, to detect where there is a maneuver in the sensor stream. A simple sentence grammar allows any maneuver to follow any other maneuver.

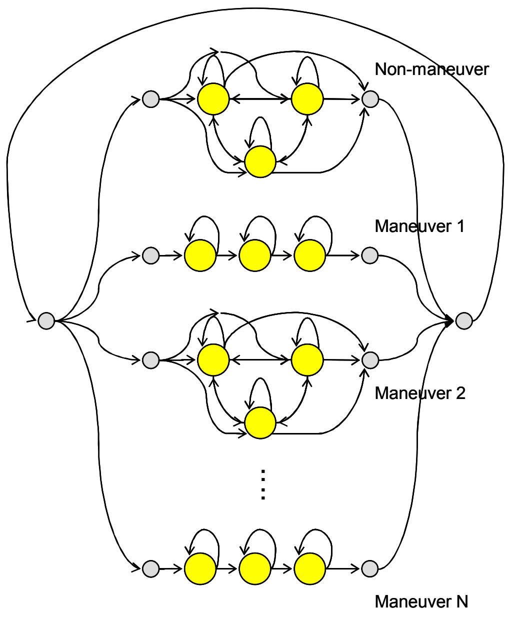

Fig 2. Modeling discrete driving maneuvers

HMM Experiments

Dataset Generation & Feature Selection

Data from the driving simulator was hand annotated for different maneuvers. For the purpose of experimentation we selected the maneuvers that occurred most frequently in the dataset. These maneuvers are specified in Table II.

A tabular labeled dataset was generated using the sequence data from the simulator and the hand annotations. It is possible for certain maneuvers to be overlapping with other maneuvers. For instance, Passing maneuver will usually be composed of ChangingLeft and ChangingRight maneuvers, hence the dataset is a multi-label multi-class data. In order to perform random forest based feature selection the dataset needed to be converted to single class dataset. In order to overcome this multi-class problem, a single pass algorithm was developed which duplicates all the overlapping instances and creates one instance for each class label. The algorithm also tries to preserve the sequential nature of the input by not interleaving the class labels of overlapping instances. This unfolded data is passed to the feature selection algorithm to pick the features that best discriminates the different maneuvers.

TABLE II
FREQUENT MANEUVER SET

Maneuver	Description
ChangingLeft	changing to a lane in the left, changing to the rightmost lane while entering a freeway.
ChangingRight	changing to a lane in the right, exiting to a service lane from a freeway.
Passing	passing a car / many cars, by moving the left lane passing the car and moving back to the original lane.
Starting	starting the car from a stopped state
Stopping	bringing the car to halt – to stop in a signal, etc.
Tailgating	any maneuver the driver undertakes in response to a car in the front.

Maneuver Classification

Generic maneuver classification experiments were conducted with different type of HMM topologies, and various types gaussian mixtures modeling the states of the HMMs. All the experiments were conducted with diagonal covariance matrices for the gaussians. The experiments were conducted using Hidden Markov Model Toolkit (HTK). The preliminary experimental results showed that the models fluctuated between high precision - low recall, and low precision – high recall values. To strike a balance between the two van Rijsbergen's F-measure was used as

$$F = \frac{1+b^2}{\frac{b^2}{recall} + \frac{1}{precision}}$$

an indicator to identify the best model:
b is the parameter that controls the degree of inclination to precision/recall. In our domain, recall is more important than precision. Setting the value of *b* to be 2, makes recall twice more important than precision.

The experiment that gave the best results for F-measure was using an ergodic HMM with 6 emitting states and a gaussian of 3 mixtures for modeling the states of the Maneuver, and an ergodic HMM with 6 states and single gaussian for the states modeling the NonManeuver. The values for F-measure, Recall, and Precision for the Maneuver was 84.5, 87.9%, and 73.1% respectively. The experiment conducted using the same Maneuver model as above and an ergodic model with 8 emitting states with 2-mixture gaussian for states of NonManeuver gave the results of 82.3, 82.4%, and 82% for F-measure, Recall, and Precision respectively. This model strikes a nice balance between all the three. These values suggest that HMMs are certainly useful and capable of modeling the driving behavior of a driver.

Towards the Discovery of Drivemes

Due to the small size of the driving database, more rigorous classification experiments on fine-grained maneuvers could not be performed. But our approach to discover the drivemes seemed promising even with this tiny dataset. In the following sub-section we give our preliminary experimental results and justify that our approach to discover the drivemes is quite promising.

The fine-grained maneuvers were modeled using HMMs with linear topology. HMMs with 6 emitting states were used for modeling all the maneuvers except for Passing maneuver, for which a 10 state model was used. Since the Passing maneuvers are usually longer in time than other maneuvers, a longer length HMM was chosen to model it. The states were modeled as a single Gaussian with diagonal covariance matrix.

Separate models were trained for each maneuver. Fig 5 illustrates the fact that the models were actually able to learn driving behavior. The mean of the variable "*Speed* " is plotted against the different states of the Starting and Stopping maneuver. It can be seen that as we move from state 1 to state 6 (i.e., as time progresses) the speed value increases for Starting and decreases for Stopping maneuver, which matches our intuition.

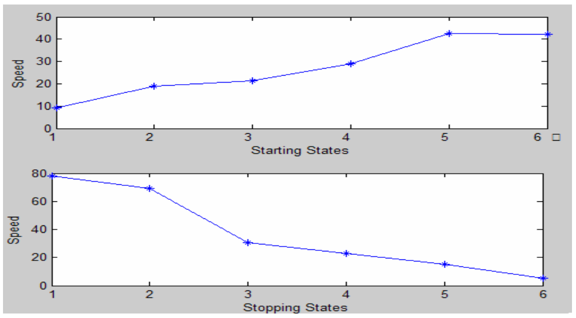

Fig 5. Speed Vs Starting, Stopping States

To justify the fact that other maneuvers are also modeled accurately, distToRightLaneEdge was plotted against the states of ChangingLeft Maneuver. See Fig 6 for the plot.

The actual change in distToRightLaneEdge while there was a changing left maneuver is plotted in Fig 7. In Fig 7 distToRightLaneEdge is plotted across Time. The portion marked represents the driving situation where there are two successive ChangingLeft maneuvers. It is seen that during the maneuver period, distToRightLaneEdge increases to a maximum value, suddenly decreases to a minimum value

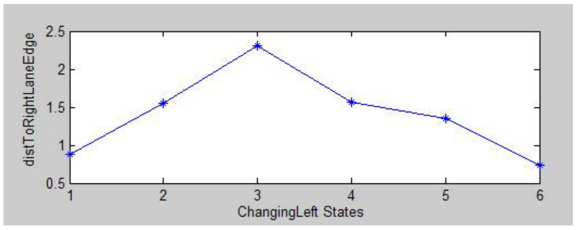

Fig 6. distToRightLaneEdge Vs ChangingLeft States

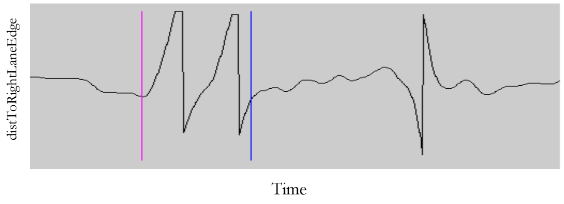

Fig 7. distToRightLaneEdge vs. time. The plot marks the portion where there are two successive ChangingLeft maneuvers.

(as the car's right wheel just crosses the lane) and again increases as the car moves towards the center of the left lane. Though the HMM in Fig 6 does not capture this sudden change it can be clearly seen that it smoothes over the values between the states and the HMM has been able to capture the overall change in the feature.

Finding Drivemes via HMM State Clustering

In our approach to find the drivemes, we cluster the state of the HMMs modeling the different maneuvers to get common subunits, called *drivemes*, among the various maneuvers. The states of different maneuvers are clustered using a hierarchical clustering algorithm with complete linkage to combine the intermediate clusters. The distance between a state in cluster i and a state in cluster j is calculated using the following formula.

$$d(i,j) = \sum_{k=1}^{N} \frac{(\mu_{ik} - \mu_{jk})^2}{\sigma_{ik}\sigma_{jk}}$$

All the states whose distances fall below a specified threshold form a cluster. In our experiments the threshold was given a value 1, so that only very close states are clustered together. The 38 states of the above specified 6 maneuvers clustered into 17 clusters. More than 50% reduction in the number of states indicates that there are certainly common subunits among the different maneuvers. Tied state models were built for each maneuver from these subunits of driving data. The *driveme* to which each state of the model was tied is given in Table IV.

From Table IV it is also seen that there are sequences of subunits which repeat. For instance, the driveme sequence (10, 1) appears in both ChangingRight and Passing. This shows that there are similarities among different maneuvers in multiple hierarchical levels, which has to be explored into in more detail.

TABLE IV
TIED STATE MODELS

Maneuver	Cluster Center Sequence
ChangingLeft	1 15 17 16 12 12
ChangingRight	12 1 12 12 10 1
Passing	1 15 14 13 12 12 12 11 10 1
Starting	5 6 6 9 8 8
Stopping	7 2 6 5 4 3
Tailgating	2 2 2 2 2 1

Fig 8 shows a plot of how the states of ChangingLeft, ChangingRight, and Passing maneuvers are mapped to the driveme subunits (cluster centers). The X-axis represent the space of drivemes, the Y-axis represent the state space of each maneuver. The plot shows the mapping between the states of each maneuver to the corresponding drivemes. It is seen from the figure that in the driveme space Passing maneuver is composed of ChangingLeft and ChangingRight maneuvers. This also supports our intuition and is strong evidence that the driving domain is composed of drivemes and the driving behavior can be decomposed into a sequence of drivemes. Fig 9 shows a plot of the distToRightLaneEdge values for the various driveme subunits of the ChangingLeft, ChangingRight, and Passing maneuvers. It can be seen from the figure that the feature's values at the overlapping portions of the Passing and ChangingLeft/ChangingRight maneuver are very close to each other. This also confirms our belief that driving is composed of drivemes and out approach to discover these subunits is essentially a right direction to pursue.

Discussion

Recognition of typical driving events and their long-term prediction using acceleration and location sensors with HMMs has been studied in (Mitrovic, 1999). Predicting the intent of the driver has been attacked with several machine learning architectures. Liu and Pentland (1997) use an architecture based upon a hidden Markov dynamic model. A hidden Markov model is used to control transitions between different dynamical models; each dynamical model is targeted towards a certain driveme such as "turn left", "turn right", etc. Oliver and Pentland (2000) use dynamical graphical models to predict and recognize driver behavior. None of this work has attempted to build a hierarchical framework for driving modeling.

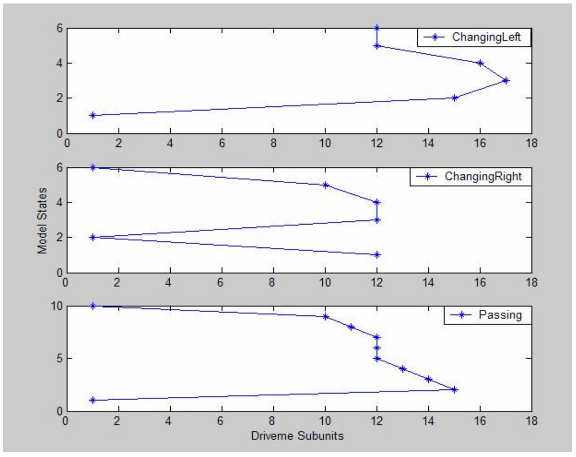

Fig 8. Mapping of ChangingLeft, ChangingRight, and Passing maneuvers onto its driveme components

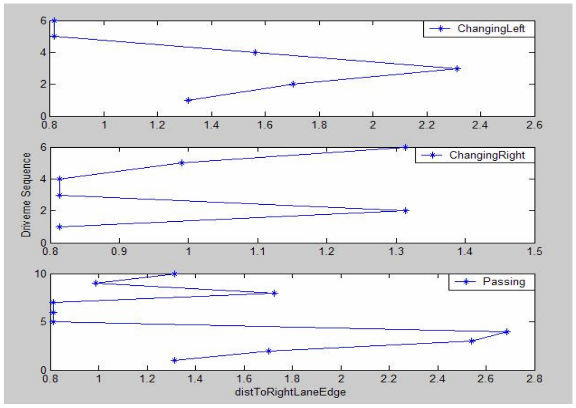

Fig 9. Plot of distToRightLaneEdge for different driveme units in ChangingLeft, ChangingRight, and Passing maneuvers.

Machine learning is based on the premise of availability of data from the problem domain. These results are based on a small sample of drivers in a simulator environment, and should be taken only as indicative of what can be achieved in reality. Collection of a large naturalistic database is underway.

References

Rabiner, L. R. 1989. A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*.

Torkkola, K. et al, 2004. "Sensor Selection for Maneuver Classification", *IEEE International Conference on Intelligent Transportation Systems*.

Breiman, L. 2001. "Random Forests", Technical Report, University of California, Berkeley.

Boer, E. R. 2000. Behavioral Entropy as an Index of Workload, *Proceedings of the IEA 2000 / HFES 2000 Congress*.

HTK Hidden Markov Model Toolkit. http://htk.eng.cam.ac.uk/

Liu, A. and Pentland, A. 1997. "Towards real-time recognition of driver intentions," *Proc. of the IEEE Conf. on Intelligent-Transportation Systems*, pp. 236-241.

Oliver, N. and Pentland, A. 2000. "Graphical models for driver behavior recognition in a smartcar," *Proc. of the IEEE Intelligent Vehicles Symposium*, pp. 7-12

Mitrovic, D. 1999. Driving event recognition by Hidden Markov Models, 4th *International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, pp. 110-113.