2015 IEEE Intelligent Vehicles Symposium (IV)
June 28 - July 1, 2015. COEX, Seoul, Korea

Driver Behavior Modeling near Intersections Using Support Vector Machines based on Statistical Feature Extraction*

Seifemichael B. Amsalu[1], Abdollah Homaifar[1], Fatemeh Afghah[1], Saina Ramyar[1], and Arda Kurt[2]

Abstract— **The capability to estimate driver's intention leads to the development of advanced driver assistance systems that can assist the drivers in complex situations. Developing precise driver behavior models near intersections can considerably reduce the number of accidents at road intersections. In this study, the problem of driver behavior modeling near a road intersection is investigated using support vector machines (SVMs) based on the hybrid-state system (HSS) framework. In the HSS framework, the decisions of the driver are represented as a discrete-state system and the vehicle dynamics are represented as a continuous-state system. The proposed modeling technique utilizes the continuous observations from the vehicle and estimates the driver's intention at each time step using a multi-class SVM approach. Statistical methods are used to extract features from continuous observations. This allows for the use of history in estimating the current state. The developed algorithm is trained and tested successfully using naturalistic driving data collected from a sensor-equipped vehicle operated in the streets of Columbus, OH and provided by the Ohio State University. The proposed framework shows a promising accuracy of above 97% in estimating the driver's intention when approaching an intersection.**

I. Introduction

The development of autonomous vehicles leads to a mixed-traffic environment where human driven, semi-autonomous and autonomous vehicles must cooperate to assure a safe traffic flow. This requires estimation and prediction of the drivers' behaviors of these vehicles effectively in order to achieve road safety and safe driving. The techniques designed to estimate the intention of drivers are also needed in the development of Advanced Driver Assistance Systems (ADAS) which results in automated driving [1]. The recent advancements in sensing, communication and computational technologies allow us to deploy the techniques designed for estimation of driver's intentions for active safety features such as antilock braking systems and adaptive cruise control to reduce road accidents [1].

The goal of this work is developing driver behavior models to make accurate prediction during pre-crash scenarios. The development of multi-agent models is based on a framework developed in Ohio State University (OSU) [2][3]. The framework takes dynamic inputs from the changing environment and other behaviors, and simulates the perception, attention, cognition, and control behavior of interest by applying different mathematical or symbolic methods [3][4].

In this paper, an algorithm that estimates the driver's intention near a road intersection controlled by a traffic signal is proposed and validated using naturalistic driving data. The objective is to estimate from a set of observations whether the driver will stop, turn right, turn left, or go straight safely according to traffic signal indicator. Drivers behave differently and the variations in observation must be taken into account in the classification process. For example, when a driver decides to "Turn Left" at an Intersection, the turn left signal blink, the speed reduces and brake light illuminate, and he/she turns left finally. The driver behavior estimation is done through observing the continuous vehicle dynamics and estimating the driver decisions that result in those observations. Here, the combination of vehicle and driver is referred as "driver" and the trajectory shown by the combination is referred as "driver behavior" [5]. The overall work is based on the Hybrid-State System (HSS) framework [6].

In this study, the Support Vector Machine (SVM) is used to estimate the driver's decisions that are the discrete- state in the HSS framework, as the vehicle approaches an intersection. The SVM technique is a relatively new type of pattern recognition methods for learning separating functions. The SVM was defined based on statistical learning theory that is the theory of structural risk minimization (SRM) and VC bounds, proposed by **V**apnik and **C**hervonenkis [7-9]. The SVM technique has been widely utilized in many applications including prediction, handwritten recognition, face detection, text categorization, speech recognition, etc. A comprehensive survey on the applications of SVM in pattern recognition is presented in [10]. In the near intersection problem, the driver has multiple decisions to choose from; consequently the basic SVM that is a binary classifier should be extended to make multiclass classification. There are two basic approaches of One-vs.-One and One-vs.-the-rest for multi-class classification with SVM [11][12]. In this work, One-vs-One multi-class classification method is used to estimate the driver's decisions near the intersection. Overall, the SVM is combined with the HSS framework to achieve driver behavior estimation. The SVM is utilized as a mathematical technique that relates the discrete-states (driver decisions) and continuous-states (continuous observations) of the HSS framework. Through this paper, the term HSS+SVM refers to this combination. The performance of the proposed

*Research was partially supported by US Department of Transportation under University Transportation Center (UTC) (DTRT13-G-UTC47).

[1]S. B. Amsalu, A. Homaifar, F. Afghah and S. Ramyar are with the Department of Electrical and Computer Engineering at North Carolina A&T State University, Greensboro, NC 27411 USA, {samsalu@aggies.ncat.edu}, {homaifar, fafghah, sramyar} @ ncat.edu.

[2]A. Kurt, is with the Department of Electrical and Computer Engineering at The Ohio State University, Columbus, OH 43210, USA, kurt.12 @osu.edu.

HSS+SVM method for driver behavior estimations is compared with HSS + Hidden Markov Model (HMM) reported in [5]. The accuracy percentage of the SVM model excels the HMM one in estimating the driver's decisions near intersections.

Modelling of driver behaviors including near a road intersection problem have been studied by many researchers. In [13], the authors developed graphical models, HMMs and their extensions for different driver maneuvers with a focus on the context effect on the driver's performance. In addition to the real-time vehicular observations, the contextual traffic information is used to predict the maneuvers. In [14], a framework for a cognitive model of human behavior is proposed with a HMM method as the driver behavior recognition for emergency and normal lane changes. The authors used driver behavior data from a driving simulator to train the HMM models. In [15], a model for recognition of driving events using discrete HMMs is presented utilizing longitudinal and lateral acceleration and speed data from a real vehicle in a normal driving environment.

In this paper, a driver behavior estimation method is proposed based on observable parameters using the SVM and the HMM approaches based on the HSS framework. In section II, the problem of driver estimation is described with an example. A brief discussion on HSS, HMM, SVM, and Statistical Feature Extraction is provided. The flowchart of the proposed SVM with statistical feature extraction technique is also explained here. In section III, the data collection and data analysis results are described. Section IV discusses the results obtained from the proposed technique. The results are also compared with a HMM approach. Finally, the paper is concluded with a summary of the results and future work.

II. Driver Behavior Estimation Framework

In section I, the driver behavior estimation is described as: from observable continuous vehicle dynamics, estimating the driver's decisions that result those observations. Considering Fig. 1 that shows an example intersection scenario, where an autonomous red car tends to turn left as shown by the dashed red line when approaches the intersection. On-board sensors alert the vehicles of another human-driven vehicle with the right of way as shown in green. The red car must determine which path the green one will follow, before turning left. Here, the red car can turn left if the green car is turning right, otherwise it must stop and let the green one pass. Hence, the red car needs to determine the intention of the green car's driver from observations obtained through the use of on-board sensors such as GPS, lidar, and radar or vehicle-to-vehicle (V2V) communication [5].

A. Hybrid State System (HSS) Framework

A hierarchical interaction of a discrete-event system and a continuous-state plant has been modeled as a Hybrid-State System (HSS) in various applications including autonomous vehicles [16] and HSS estimation [6]. The behavior of a vehicle and its driver can be estimated, tracked, and predicted by representing the interaction of the vehicle and the driver in an HSS model. Fig. 2 shows the HSS setting that consists of a higher level discrete-state system (DSS) part and a lower level continuous-state system (CSS) part. According to discrete events a driver reacts and makes corresponding decisions on the higher level, and based on the decisions the vehicle follows a continuous trajectory. The HSS system equation and formulation is developed in [17].

B. Hidden Markov Models (HMM)

The relationship between the CSS and DSS parts of HSS discussed in the previous section can be mathematically modeled using HMMs [18], [19] that provides a stochastic relationship between the vehicle dynamics and driver state. Here, the DSS driver states are not predefined and the changes in CSS vehicle dynamics lead to changes in the driver state as determined by training observation data.

A HMM is a stochastic model that represents a dynamic process of two related random processes. An underlying stochastic process that is not observable (hidden states) can be observed through another set of stochastic processes that produces the sequence of observed symbols. A HMM consists of a set of N distinct "hidden" states of the Markov process $Q = \{q_1, q_2, \dots, q_N\}$ and a set of M observable symbols per state $V = \{v_1, v_2, \dots, v_M\}$. The overall HMM model is defined as follows (with $\mathrm{q_t}$ and $\mathrm{o_t}$ denoting the state and observation symbol at time t respectively).

The HMM is specified by a set of parameters (A, B, π):

1. The prior probability distribution $\pi = \{\pi_i\}$ where $\pi_i = P(\mathrm{q_1} = s_i)$ are the probabilities of s_i being the first state in a state sequence.
2. The transition probability matrix $\mathrm{A} = \{\mathrm{a_{ij}}\}$ where $\mathrm{a_{ij}} = \mathrm{P}(\mathrm{q_{t+1}} = s_j |\, \mathrm{q_t} = s_i)$ are the probabilities to go from state s_i to state s_j.
3. The emission (observation) probability matrix $\mathrm{B} = \{\mathrm{b_{ik}}\}$ where $\mathrm{b_{ik}} = \mathrm{P}(\mathrm{o_t} = v_k |\, \mathrm{q_t} = s_i)$ are the probabilities to observe v_k if the current state is $\mathrm{q_t} = s_i$.

The aforementioned emission probability matrix is defined for the case of discrete HMM. In our problem, the observation symbols are continuous vehicle dynamics, so the HMM development is based on probability density functions. The emission pdfs for a continuous valued observation is described by a set of probability density functions (PDFs), $\mathrm{b}_i(\mathrm{o_t}) = p(\mathrm{o_t} | \mathrm{q_t} = s_i)$, over the observation space for the system being in state s_i. In this paper, the most general emission distribution, that is modeled as a joint Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, called Gaussian Mixture Model (GMM) is used, with

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_M \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 & \cdots & \sigma_{1M}^2 \\ \sigma_{21}^2 & \sigma_{22}^2 & \dots & \sigma_{2M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{M1}^2 & \sigma_{M2}^2 & \dots & \sigma_{MM}^2 \end{bmatrix} \quad (1)$$

The GMM is of the form

$$\mathrm{b_i}(\mathrm{o_t}) = \mathrm{p}(\mathrm{o_t} |\, \mathrm{q_t} = s_i) = \sum_{\mathrm{m=1}}^{\mathrm{M}} \mathrm{c_{im}} \mathcal{N}(\mathrm{o_t}; \mu_{\mathrm{im}}, \Sigma_{\mathrm{im}}) \quad (2)$$

where $\mathrm{c_{im}}$ stands for the mixture coefficient for the m^{th} mixture in the i^{th} state and $1 \leq \mathrm{i} \leq \mathrm{N}$. The $\mathcal{N}$ stands for the pdf of a normal distribution with mean μ and covariance Σ

measured at observation o_t. Mixture coefficient c_{im} satisfies the following constraints:

$$\sum_{m=1}^{M} c_{im} = 1, \qquad 1 \le i \le N, 1 \le m \le M \tag{3}$$

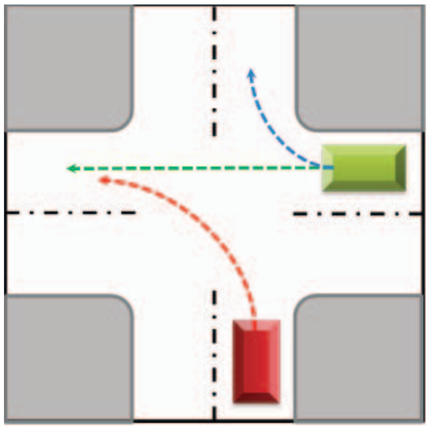

Figure 1. A simple intersection senario. Given the right of way is for the green car, for the red car to turn left, it must determine if the green car is going straight or turning right.

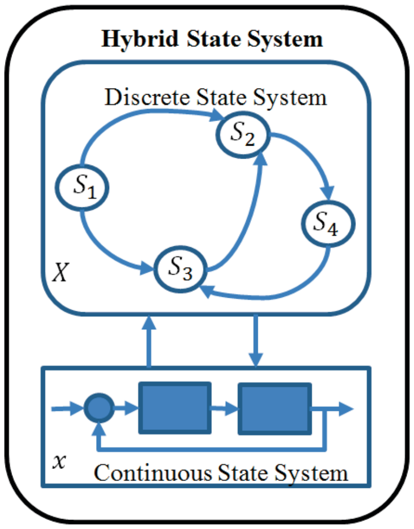

Figure 2. Hybrid State System Setting

Such that the pdf is normalized to be

$$\int_{-\infty}^{\infty} b_i(o)do = 1, 1 \le i \le N \tag{4}$$

With the above definitions, an HMM can be completely specified by the pdf of a normal distribution $\mathcal{N}$, and the five probability measures $\lambda = (A, \pi, c, \mu, \Sigma)$.

In order to train the HMM models, the continuous observations are used to infer the discrete-state of the driver using an iterative procedure called the Baum–Welch method (Expectation Maximization) [20][21]. It computes the maximum-likelihood estimates of the HMM model parameters $\lambda = (A, \pi, c, \mu, \Sigma)$ using the *forward* and *backward algorithms* [22]. Once the models, $\lambda_1, \lambda_2, \dots, \lambda_n$, are trained, the *forward algorithm* is used to determine the driver states. For a given sequence of observations, $O = \{o_1, o_2, \dots, o_t\}$, the forward algorithm computes the probabilities $P(O|\lambda_i), i = 1, 2, \dots, n$ for each model. The model λ_i with the highest likelihood probability at a given time step represents the estimated driver state. Thus, to determine the drivers state at time t

$$S(t) = \arg\max_i P(o_1, o_2, \dots, o_t|\lambda_i) \; i = 1, \dots, n \tag{5}$$

Fig. 3 shows the general idea of this technique. Equation (5) is represented in the final compare step. Since the focus of this paper is in vehicle actions near an intersection, models λ_i: straight, left, right and stop are the possible driver decisions near an intersection.

C. *Support Vector Machine (SVM)*

The SVM is a method for learning separating functions in pattern recognition tasks. It is unlike classical pattern recognition algorithms that work in an L1 or L2 norm and minimize the absolute value of an error or of an error square, the SVM performs SRM that builds a model with a minimized VC dimension [7-9].

In the simplest pattern recognition tasks, the SVM uses a linear separating hyper-plane to create a classifier with a maximal margin. In order to do that, the learning problem is cast as a constrained nonlinear optimization problem. In this setting, the cost function is quadratic and the constraints are linear (i.e., one has to solve a quadratic programming problem). Consider the problem of binary classification, the training data are given as:

$$(X_1, y_1), (X_2, y_2), \dots, (X_l, y_l), \quad X \in R^n, y \in \{+1, -1\}. \tag{6}$$

Let us consider a two-dimensional input space, $X \in R^2$, for ease of visualization. When data is linearly separable, many different hyper-planes can perform separation as shown in Fig. 4. The question here is how to find the best one.

Using the given training examples during the learning stage, the SVM finds parameters $W = [w_1\ w_2\ \dots\ w_n]^T$ and b of a decision function $d(X, W, b)$ given as

$$d(X, W, b) = W^T X + b = \sum_{i=1}^{n} w_i x_i + b, \tag{7}$$

where $X, W, \in R^n$, and the scalar b is called a bias. (Note the solid lines in Fig. 4 represent $d(X, W, b) = 0$).

From analytical geometry, the margin can be derived to be $M = {}^2/_{\|w\|}$. In order to find the optimal separating hyper-plane having a maximal margin, a learning machine should minimize $\|w\|^2$ subject to inequality constraints $y_i[W^T X_j + b] \ge 1, \ i = 1, \dots, l.$ The objective function to be minimized becomes

$$\min\{{}^{\|w\|^2}/_2\} \tag{8}$$

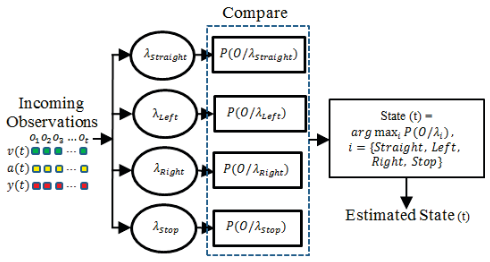

Figure 3. Estimation of the state that best describes the incoming observations sequence.

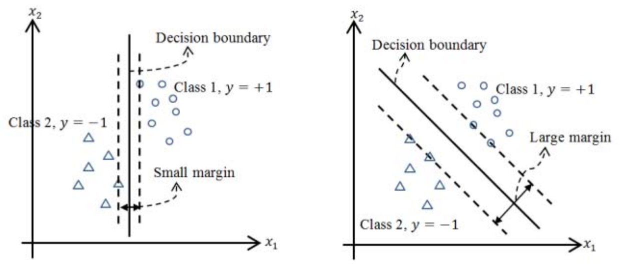

Figure 4. Two out of many separating lines: right, a good one with a large margin, and left, a less acceptable one with a small margin.

In this case, the two classes are not completely separable, but a hyper-plane that maximizes the margin while minimizing a quantity proportional to the misclassification errors can still be determined. This can be done by introducing positive slack variables ξ_i, which represents the magnitude of the classification error in the constraint $y_i[W^T X_j + b] \geq 1 - \xi_i,\ i = 1, \dots, l, \xi_i \geq 0$. If an error occurs, the corresponding ξ_i must exceed unity, so $\sum_i \xi_i$ is an upper bound for the number of misclassification errors. Hence, the objective function to be minimized becomes

$$\min\{\frac{\|w\|^2}{2} + C \sum_{i=1}^{l} \xi_i\} \tag{9}$$

where C is a parameter chosen by the user and is used to balance between the margin and the misclassification errors. A larger value of C means that a higher penalty for misclassification errors is assigned. The minimization of equation (9) under the constraint $y_i[W^T X_j + b] \geq 1 - \xi_i,\ i = 1, \dots, l, \xi_i \geq 0$ results in the Generalized Separating Hyperplane. This still remains a Quadratic Programming (QP) problem and standard techniques (Lagrange Multipliers, Wolfe dual) can be applied [22][10].

In cases when the given classes cannot be linearly separated in the original input space, the SVM first nonlinearly transforms the original input space into a higher-dimensional feature space. This transformation can be achieved by using various nonlinear mappings (kernels): polynomial, sigmoidal, RBF mappings [22]. After this nonlinear transformation step, the task of an SVM in finding the linear optimal separating hyper-plane in this feature space is relatively trivial. This technique is called *Kernel Trick*. The resulting hyper-plane in feature space will be optimal in the sense of being a maximal margin classifier with respect to training data.

Since the intersection problem is a multi-class classification problem (Left, Right, Straight, and Stop), the SVM cannot be applied directly. There are different methods to apply SVM for multi-class classification. The most common ones are one-vs.-one and one-vs.-the-rest [22]. The one-vs.-one trains a classifier for each possible pair of classes. For M classes, this results in $(M-1)M/2$ binary classifiers. When it tries to classify a test pattern, it evaluates all the binary classifiers, and classifies according to which of the classes gets the highest number of votes. The one-vs.-the-rest gets M-class classifiers, it constructs a set of M binary classifiers each trained to separate one class from the rest, and combine them by performing the multi-class classification according to the maximal output. In this work, one-vs-one multi-class SVM classification method is used to estimate the driver decisions near the intersection.

D. *Statistical Feature Extraction and Classification*

Since the time series observation is too large to be taken as a feature vector for classification using the SVM, feature extraction technique is applied to capture the relevant information from the time series. Also, it allows predicting each time step using the previous observations (makes use of history).

In this study, statistical features (mean, median, min, max, variance, sum, range, mode, kurtosis and skewness) of the time-series maneuver data at a given time $t = \tau$ are extracted for the observations: Velocity, Acceleration and Yaw-rate. These measures that are used to describe the dataset show the central tendency (mean, median, and mode), variability (variance and range) and shape of the probability distribution (kurtosis and skewness) of the observations. This enables us to obtain the main features of the observations that are relevant for classifying the different maneuvers accurately. At $t = \tau$, the time series maneuver data is represented by the 10 statistical features mentioned for the three observations. This forms 30 component feature vectors for a given maneuver at $t = \tau$ i.e. 10 statistics for the 3 observations.

The calculated statistical features of the time series maneuver data at $t = \tau$ are presented to the multi-class SVM classifier. In the classification stage, the dataset (the feature vectors with their corresponding labels) is divided into 70% training and 30% testing set. The training set is used to build the Multi-class SVM classifier and the testing set is used to calculate the classifier performance. The flowchart of the algorithm of the SVM based on statistical feature extraction method is shown in Fig. 5.

In addition to developing method that gives the best features, which represent the time series data, formulating an appropriate SVM kernel function for the problem is still an open research question. There are predefined kernel functions including linear, polynomial, and radial basis function (RBF) kernels [22].

III. DATA COLLECTION AND ANALYSIS

In this study, the driver behavior information data set collected by the Ohio State University researchers [5] is used to train the mathematical models discussed in section II. Several participants were recruited to drive a sensor fitted vehicle around the streets of Columbus, OH. This work focuses on estimating the states that represent the high-level behavior of the driver from the observable parameters. It is assumed that we are able to collect the low-level continuous observations such as velocity, yaw-rate, and acceleration, given the correct combination of sensors.

The data is collected from a 2012 Honda Accord equipped with the following sensors [5]: *NovAtel GPS unit-*: it provides GPS latitude, GPS longitude, timestamp of reading and others; *Honda Accord controller area network (CAN) bus-* it provides timestamp of reading, yaw rate, lateral acceleration, speed, steering wheel angle, odometer, headlights, brake lights, turn signals and others. *Three HD* *cameras*- they provide views of the front, left side, and right side of the vehicle.

The participants drive the vehicle through selected paths that represent a typical road situations encountered in daily driving tasks. The scenarios of interest are intersection (Straight, Left turn, Right turn and stop) and highway driving (merging and entering/exiting ramps).

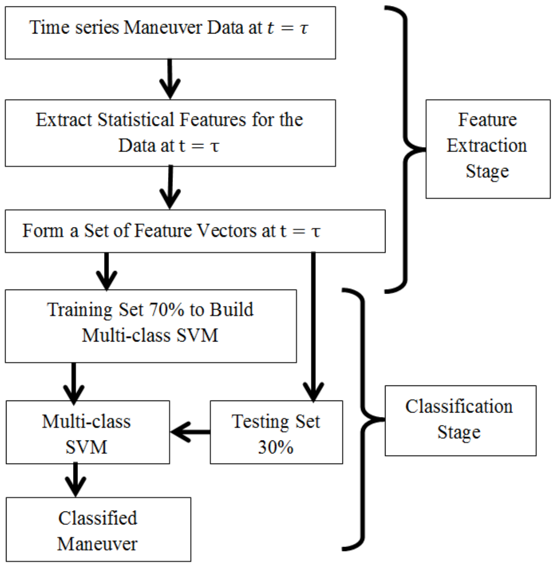
Time series Maneuver Data at $t = \tau$
Extract Statistical Features for the Data at $t = \tau$
Form a Set of Feature Vectors at $t = \tau$
Feature Extraction Stage
Training Set 70% to Build Multi-class SVM
Multi-class SVM
Testing Set 30%
Classification Stage
Classified Maneuver

Figure 5. Flowchart of the Multi-class SVM based on Statistical Feature Extraction clasifier

The ground truth representing each scenario was determined from the collected data by manually marking the videos. For instance, in the video when a vehicle approaching an intersection and turn right, the time span was marked as "Right Turn" and the corresponding time series observations of velocity, yaw-rate and acceleration were extracted within this time span. These observations accurately described an intersection driving behavior as shown in [5][23]. The time-series observations were used to train the HMMs and also the SVM based on statistical feature extraction. The two models are tested and compared on the accuracy of estimating the states at each time step.

IV. Results

In this work, the SVM is applied to model the intersection driving scenario as a multi-class classification problem and compared with the HMM based model. Statistical features are extracted from the time series observations at time t=τ. This enables us to use the historical information before the time t=τ as an input for classifying the vehicle maneuver. As a result the vehicle trajectory before the given time t=τ is well represented in the feature vectors. The performance of the models is evaluated using Confusion Matrix where each column represents the number of instances in a predicted class and each row represents the number of instances in an actual class.

The confusion matrix on Table-I shows the performance of the SVM with One-vs.-one multi-class classification method with the RBF kernel and statistical features for 70% training and 30% testing set at t=τ. The Statistics Toolbox of MATLAB is used to implement the multi-class SVM with the default termination criterion. Here, the time series data for 18 Straight, 6 Left Turn, 5 Right Turn and 21 Stop maneuvers converted to 1964 Straight, 655 Left Turn, 545 Right Turn and 2279 Stop feature vectors at a given time t=τ. The conversion is done at time step of 0.1 seconds for an 11 seconds maneuver that is 6 seconds before and 5 seconds after entering the intersection. This makes the total number of feature vectors in the data set to be 5443 with corresponding labels that is divided into 70% training and 30% testing set.

TABLE I. Confusion Matrix for multi-class SVM Using 30% testing set at T=τ

Actual Maneuvers at t=τ	**Predicted Maneuvers at t=τ**			
	Straight	***Left Turn***	***Right Turn***	***Stop***
Straight	646	1	0	7
Left Turn	4	208	3	3
Right Turn	9	0	172	0
Stop	6	1	4	748

Accuracy = 97.90%

The proposed method is compared with its HMM counterpart as shown in the following confusion matrix using the 30% testing set. The HMM is trained with M=3 GMMs for the velocity, yaw-rate, and acceleration observations and N=4 possible hidden states using the modified version of [24]. Table II show the confusion matrix based on the HMM model. The accuracy of the SVM model in estimating the states at each time step as shown in Table I is far better than the HMM one. Even though the HMM represents the stochastic process that produces the observations with the intuition of Markov chains, the proposed technique uses the statistical features to represent the sequence of observations and utilizes the classification power of SVM to achieve the indicated accuracies.

TABLE II. Confusion Matrix for HMM Using 30% testing set at T=τ

Actual Maneuvers at t=τ	**Predicted Maneuvers at t=τ**			
	Straight	***Left Turn***	***Right Turn***	***Stop***
Straight	361	46	111	136
Left Turn	4	174	38	2
Right Turn	16	1	142	22
Stop	95	103	114	447

Accuracy = 62.03%

Fig. 6 shows the estimated states for four individual example maneuvers of straight, left turn, right turn and stop observation sequences at an intersection using the HMM and the SVM based methods discussed above. This figure describes the estimators output at a given time t=τ for HMM and Multi-Class SVM. For instance, Fig. 6 (a) shows the state estimation of an example "straight maneuver sequence" with ground truth corresponding to a vehicle going straight through an intersection using the developed models, the HMM and the SVM based. At every time step, in the HMM method, equation (5) is evaluated and the HMM that best describes the sequence until that point is the estimated state at a given time t=τ. The same way, at every time step, the state estimation for the sequence at t=τ using the SVM based method is shown in Fig. 6. This was similarly repeated for sample maneuvers observation sequences with ground truth corresponding to a vehicle turning left, right and stopping at an intersection. The figures show that overall the SVM based method outperforms the HMM one.

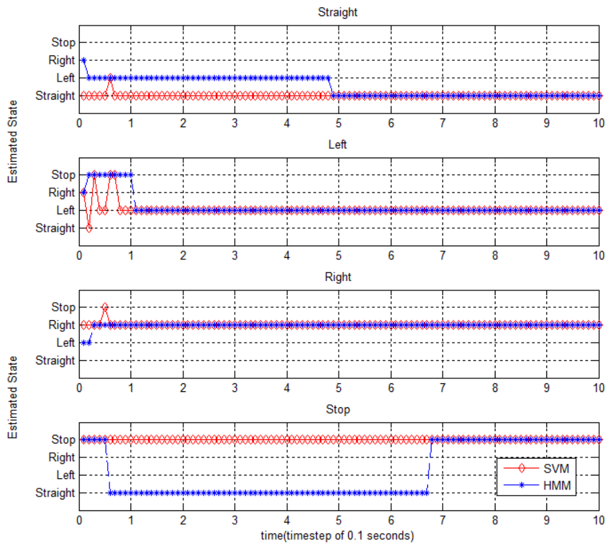

Figure 6. Estimation of state at each timestep for four different individual vehicle manuevers using HMM and SVM based techniques (a) Straight, (b) Left, (c) Right (d) Stop.

V. Conclusion and Future Work

This work has presented a SVM+HSS approach to driver behavior estimation near intersections. It provides accurate results that perform close to that of a human observer. This work combines the HSS framework with multi-class SVM based on statistical feature extraction. The statistical feature extraction enables the SVM to represent the historical information as the HMM represents sequence. A one-vs-one multi-class SVM classification method with the RBF kernel is used here. The proposed method is also compared with HMM based on HSS in estimating the state of the driver at an intersection. In many applications, SVM outperforms HMM in generalizing the pattern recognition problems. Also, the SVM has shown that higher performance rate of above 97% is achieved in driver behavior estimation near an intersection. In future work, the driver's unclear decision near the intersection will be taken into consideration in the estimation technique, to achieve a comprehensive solution. In addition to the intersection scenario, the proposed method will be applied in different scenarios of interest including merging and entering/exiting ramps, lane change and other near crash events using large scale dataset from SHRP2.

Acknowledgment

This research is partially supported by the US Department of Transportation (USDOT), Research and Innovative Technology Administration (RITA) under University Transportation Center (UTC) Program (DTRT13-G-UTC47). And also, authors would like to thank Ümit Özgüner (PhD) from Ohio State University for helpful discussions.

References

[1] W. D. Jones, "Keeping cars from crashing," IEEE Spectrum, vol. 38, no. 9, pp. 40–45, 2001.

[2] A. Kurt, and Ü. Özgüner, "A probabilistic model of a set of driving decisions," In 14th International IEEE Conference Intelligent Transportation Systems (ITSC), pp. 570-575, 2011.

[3] B. Song, and D. Delorme, "Human driver model for smart AHS based on cognitive and control approaches," In ITS America conference, 2000.

[4] Ü. Özgüner, T. Acarman, and K. Redmill, *Autonomous ground vehicles*, Artech House Publishers, 2011.

[5] V. Gadepally, A. Krishnamurthy, and Ü. Özgüner, "A Framework for Estimating Driver Decisions near Intersections," Trans. Intell. Transp. Syst., vol. 15, no. 2, pp.637-646, 2014.

[6] A. Kurt, J. Yester, Y. Mochizuki, and U. Ozguner, "Hybrid-state driver/vehicle modelling, estimation and prediction," In Proceedings 13th IEEE ITSC, pp. 806–811, 2010.

[7] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," In Proceedings of Fifth Annual Workshop on Computational Learning Theory, pp. 144-152, 1992.

[8] C. Cortes, and V. Vapnik, "Support vector networks," In Proceedings of Machine Learning, vol. 20, pp. 273-297, 1995.

[9] V. Vapnik, "The nature of statistical learning theory," Springer, 1995.

[10] S. Lee, and A. Verri, *SVM,* Springer-Verlag Berlin Heidelberg, LNCS 2388, 213-236, 2002.

[11] B. Zhao, Y. Liu, and S. Xia, "Support vector machines and its application in handwritten numerical recognition," In Proceedings of 15th Int. Conference on Pattern Recognition, vol. 2, pp. 720-723, 2000.

[12] S. Bernhard, C. Burges, and A. Smola, "Pairwise classification and support vector machines," The MIT Press, C. Massachusetts, London England, pp. 255-268, 1999.

[13] N. Oliver and A. Pentland, "Graphical models for driver behavior recognition in a smart car," in Proc. IEEE IV Symp., 2000, pp. 7–12.

[14] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu, "A driver behavior recognition method based on a driver model framework," SAE Trans., vol. 109, no. 6, pp. 469–476, 2000.

[15] D. Mitrovic, "Reliable method for driving events recognition," IEEE Trans. Intell. Transp. Syst., vol. 6, no. 2, pp. 198–205, Jun. 2005.

[16] A. Kurt and U. Ozguner, "Hybrid state system development for autonomous vehicle control in urban scenarios," in Proceedings of the IFAC 2008 World Congress, pp. 9540–9545, 2008.

[17] M. Dogruel and U. Ozguner, "Discrete and hybrid state system modeling and analysis," Turkish Journal of Electrical Engineering and Computer, vol. 5, no. 2, pp. 263–286, 1997.

[18] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *IEEE ASSP Mag.*, vol. 3, no. 1, pp. 4–16, Jan. 1986.

[19] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," Proc. IEEE, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[20] L. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," Ann. Math. Stat., vol. 37, no. 6, pp. 1554–1563, Dec. 1966.

[21] J. Bilmes,"A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Int. Comput. Sci. Inst., Berkeley, CA, Tech. Rep. ICSI-TR-97-021, 1997.

[22] B. Scholkopf, and A. Smol, "Learning with Kernels- Support Vector Machines, Regularization, Optimization and Beyond," Massachusetts Institute of Technology, 2002.

[23] G. Aoude, V. Desaraju, L. Stephens and J. How, "Behavior Classification Algorithms at Intersections and Validation using Naturalistic Data," IEEE Intell. Veh. Sympo, 2011.

[24] K. Murphy, Hidden Markov Model (HMM) toolbox for Matlab. [Online]. Available: http://www.ai.mit.edu/~murphyk/Software/HMM/hmm.html