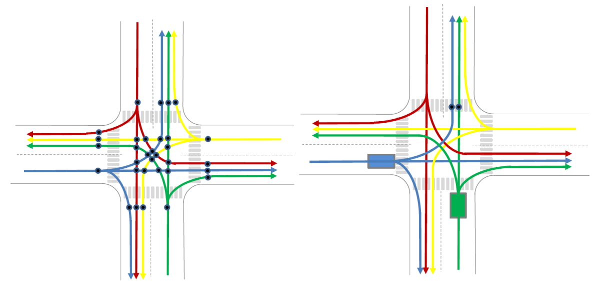
2015 IEEE Intelligent Vehicles Symposium (IV)
June 28 - July 1, 2015. COEX, Seoul, Korea

Turn Prediction at Generalized Intersections

Bo Tang, Salman Khokhar, Rakesh Gupta

***Abstract*— Navigating a car at intersections is one of the most challenging parts of urban driving. Successful navigation needs predicting of intention of other traffic participants at the intersection. Such prediction is an important component for both Advanced Driver Assistance Systems (ADAS) and Autonomous Driving (AD) Systems. In this paper, we present a driver intention prediction model for general intersections.**

Our model incorporates lane-level maps of an intersection and makes a prediction based on past position and movement of the vehicle. We create a real-world dataset of 375 turning tracks at a variety of intersections. We present turn prediction results based on Hidden Markov Model (HMM), Support Vector Machine (SVM), and Dynamic Bayesian Network (DBN). SVM and DBN models give higher accuracy compared to HMM models. We get over 90% turn prediction accuracy 1.6 seconds before the intersection. Our work advances the state of art in ADAS/AD systems with a turn prediction model for general intersections.



(a) fig 1 (b) fig 2

Fig. 1. Collision depends on the lane and the layout of the road. As shown in (a) there are 32 possible collisions at the intersection including rear ending. As shown in (b) for a specific configuration of two cars only a couple of these possibilities are relevant (best seen in color).

I. INTRODUCTION

Recent studies show that 45% of injuries and 22% of roadway fatalities in US are intersection related [10]. The primary factor in these accidents is the driver's inability to correctly observe a signal, stop sign, or another traffic participant at the intersection. Advanced Driver Assistance Systems (ADAS) that build a model of surrounding cars to predict collisions and warn the driver when there is high collision risk are desirable.

Each intersection has unique characteristics such as its shape, number of intersection legs, road turn types, and number of lanes. While most previous work built intersection models at specific intersections [12] [5], our work builds a general driver intention prediction model that works at any intersection. Applications of our model include Intersection Movement Assist (IMA) and Left Turn Assist (LTA). Intersection Movement Assist warns the host vehicle when it is not safe to enter an intersection due to high collision probability with other vehicles. Left Turn Assist warns the host vehicle when it is not safe to make a left turn at the intersection due to high collision probability with other vehicles.

Collision detection at intersections is a challenging task. Time-To-Collision (TTC) [8] assumes constant vehicle velocity to predict collisions. TTC works well for straight roads, and fails to detect collision for curved lanes and intersections. This is because the collision depends on the lane and the geometry of the road. The vehicle lane restricts the vehicle's turning options, and the geometry of the road indicates more realistic potential trajectories. Hence, it is essential to use accurate lane-level maps with localization and a technique to predict motion of other cars to avoid collisions and safely navigate the intersection.

In an intersection there are 32 ways to get into a collision [6] as shown in the Figure 1(a). Given a specific configuration of two cars and intended routes, only a couple of these cases are relevant. For example, for one car turning left and the other car going straight, there is only one possible collision as shown in Figure 1(b). To predict a collision at an intersection, we need to know:

- The lane for traffic participants at the intersection.
- Traffic participant intention to go straight, left or right at the intersection.

Given these pieces of information, one can estimate the potential collision location and time.

The intersection layout is incorporated via a high resolution lane-level map of the intersection. There are several reasons to use lane-level maps in our model: firstly, the complex structure of intersection can be reduced to a lane model; secondly, the traffic rules at the intersection can be incorporated in the lane model; thirdly, the lane allows computation such as inside/outside lane, distance to centerline; lastly, the lane-level maps provide a driver friendly visualization. In this paper, we propose a driver intention model based on intersection layout encoded via high resolution maps and features of the vehicle such as its location and velocity.

In the next section we describe related work, and our problem formulation. We then describe statistical models for driver intention prediction including Hidden Markov

B. Tang is with Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881, USA. Email: btang@ele.uri.edu. S Khokhar is with Department of Electrical Engineering and Computer Science at University of Central Florida, Orlando, FL 32816, USA. Email: skhokhar@cs.ucf.edu. R. Gupta is with Honda Research Institute USA Inc., 375 Ravendale Drive, Suite B, Mountain View, CA 94043, USA. Email: rgupta@honda-ri.com.

Model (HMM), Support Vector Machine (SVM), and Dynamic Bayesian Network (DBN). We then discuss our data collection and how we create a real-world driving dataset at a variety of intersections with lane-level maps. We then describe our experiments, classification features, results and conclusions.

A. Related Work

Existing research work on driver intention prediction focuses on freeway scenarios, due to its relative simplicity. More recently, driver intention prediction at intersections has been an important topic of interest in autonomous driving. Many models have been proposed in literature, including HMM [12], Gaussian Process [7], DBN [9] [4], SVM [1], Case-Based Reasoning [5], to name a few.

Streubel and Hoffmann [12] presented a prediction framework based on HMM and examined the variation of model parameters. Their HMM model is similar to ours, and we also use the velocity, acceleration and yaw of vehicles as features to train three individual HMM models. However, ours incorporates lane-level map at general intersections, while their model is limited to a specific intersection. Laugier et al. [7] used a two layer HMM and Gaussian process to estimate and predict collision risks and likely behaviors of multiple agents in road scenes. Possible trajectories can then be sampled from a Gaussian process.

Rossler [11] localized the host vehicle using an Environment Model (EM) to take the current sensor data. They present a similar lane model as ours to model the road network with road, lanes, and intersection entities. The driver model contains both Macroscopic behavior (MaB) and Microscopic Behavior (MiB). MaB defines roads before and after the intersection and is formulated as a graph with road element as nodes and transition probabilities on the edges. MiB is a more detailed view and defines behavior in terms of lanes rather than roads. High level prediction considers decisions such as intention to turn left or stopping at red light. Low level prediction considers trajectories and speed profiles.

Aoude et al. [1] designed a threat assessment module based on SVM and random trees. SVM was used as the learning method for intention prediction, while random Trees were applied to evaluate the threat of errant drivers. In [5], Graf et al. introduced the concept of Case-Based Reasoning (CBR) with context information for a specific intersection. With CBR, a solution for the current situation is found by adapting it to the most similar case.

While most existing works are based on a specific intersection, in this paper, we propose a general intention prediction framework in which a variety of learning approaches can be used. Our implementations of HMM, SVM, and DBN demonstrate the effectiveness of our proposed framework.

B. Problem Formulation

Given a sequence of observations for a vehicle, such as GPS and CANbus data, and a road map with lane boundaries, we build statistical models for human intention. Our goal is to predict driver intention for other traffic participants (e.g. cars turning right/left or going straight at an intersection). We assume vehicle-to-vehicle (V2V) communication to identify and transfer data between self car and other traffic participants. We further assume that the self car path is known. An alternative to V2V communication is vision or laser sensors with object recognition. Velodyne sensor with 3D maps is often used for localization with DGPS sensors providing ground truth to cm accuracy.

The problem of driver intention prediction can be usually formulated as classification problem or as time-series analysis problem. We consider it as a classification problem, and extract features with a fixed number of dimensions from sequential observations for a classifier, such as SVM. The sequential observations can be further modeled and interpreted with unobserved (hidden) states using time-series analysis methods, such as Hidden Markov Model and Dynamic Bayesian Network.

II. Statistical Models for Driver Intention Prediction

To predict the driver intention (right turning, left turning, and straight going) at an intersection, both classification-based approaches and dynamic inference-based methods are tested in our real-world driving tracks dataset.

A. Hidden Markov Model (HMM)

HMM can be used to represent a dynamic process observed through Markov Chains. It is based on an assumption that the observation sequence $Y = [y_1, y_2, \ldots, y_t]$ is determined by a discrete hidden state sequence $Q = [q_1, q_2, \ldots, q_t]$. The variables used in our formulation are described in a later section. HMM is a suitable learning model, as one can divide a turning at the intersection into several stages with discrete hidden states. For example, a right turn process can be described by an HMM with 5 hidden discrete states: approach the intersection, start turning, do right turning, finish turning, and drive away from the intersection. One can use a similar process for left turn and going straight.

Two procedures are needed in HMM: learning and evaluation. In learning stage, given a set of observation sequences, the parameters of state transition probability matrix $\boldsymbol{\pi} = \{\pi_{ij}\}_{M\times M}$ and the distribution of observations $p(y_t|q_t)$ need to be learnt with M hidden states. For each HMM, we assume that $p(y_t|q_t)$ has a Gaussian Mixture Model (GMM) distribution with parameter set $\boldsymbol{\theta}$. Because there is no closed form of the estimation of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$, we apply the Baum Welch algorithm to approximate the optimal solution. The Baum Welch algorithm is an Expectation-Maximization (EM) algorithm. The expectation step provides the expectation of the log-likelihood given the current system states and transition probabilities, and the maximization step adapts the model parameters to maximize the expectation of log-likelihood. By iteratively applying the expectation and maximization steps, the maximum likelihood estimations of $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ can be approximated. In evaluation stage, we calculate the probability of a given observation sequence $y_{1:t}$ that is created by a HMM $hmm(\boldsymbol{\pi}, \boldsymbol{\theta})$, i.e., $p(y_{1:t}|hmm(\boldsymbol{\pi}, \boldsymbol{\theta}))$.

In our HMM-based classification method, we build three individual HMMs for right turn $hmm_1(\boldsymbol{\pi}_1, \boldsymbol{\theta}_1)$, left turn $hmm_2(\boldsymbol{\pi}_2, \boldsymbol{\theta}_2)$, and going straight $hmm_3(\boldsymbol{\pi}_3, \boldsymbol{\theta}_3)$, and apply the Baum Welch algorithm to learn the parameters $\boldsymbol{\pi}_i$ and $\boldsymbol{\theta}_i$ $(i = 1, 2, 3)$ for each HMM. Given a new sequence of observation $y_{1:t}$, we classify it into three potential driving behaviors with the Maximum a Posteriori (MAP) rule:

$$\begin{aligned} i &= \arg\max_{i=\{1,2,3\}} p(hmm_i(\boldsymbol{\pi}_i, \boldsymbol{\theta}_i)|y_{1:t}) \\ &\propto \arg\max_{i=\{1,2,3\}} p(y_{1:t}|hmm_i(\boldsymbol{\pi}_i, \boldsymbol{\theta}_i))p(hmm_i(\boldsymbol{\pi}_i, \boldsymbol{\theta}_i)) \end{aligned} \tag{1}$$

where $p(hmm_i(\boldsymbol{\pi}_i, \boldsymbol{\theta}_i))$ is the prior probability for each turn class. Assuming $p(hmm_i(\boldsymbol{\pi}_i, \boldsymbol{\theta}_i))$ has uniform distribution, we have

$$i = \arg\max_{i=\{1,2,3\}} p(y_{1:t}|hmm_i(\boldsymbol{\pi}_i, \boldsymbol{\theta}_i)) \tag{2}$$

B. Support Vector Machine (SVM)

We also use a supervised learning method, support vector machine (SVM), for classification. The features used are described in a later section. SVM offers a robust and efficient classification approach for driver intention prediction by constructing a hyperplane in a higher dimensional space. The hyperplane provides a separation rule with the largest distance to the "support" training data of any class. We use the toolkit of LIBSVM for training the SVM for our three-class classification problem [2]. For the details of SVM and more information about LIBSVM, we refer the interested readers to the papers [2] [3].

C. Dynamic Bayesian Network (DBN)

While both HMM and SVM can be applied to predict driver intention at the intersection in a manner of classification, we further propose a Dynamic Bayesian Network (DBN) approach for driver intention inference. Motivated by the fact that driver intention at an intersection can be reflected by the driving tracks through controls of vehicle (e.g., braking, accelerating and steering), our proposed DBN approach infers the driver intention considered as a latent state I from the observations of position and velocity of the vehicle over time. In addition to the driver intention I (a discrete variable including straight driving, right turning and left turning), other hidden states include the position and velocity of vehicle ($\mathbf{s}$ and $\mathbf{v}$), the control variables of vehicle (the acceleration a and the change of driving direction $\Delta\alpha$), and the driving lane index L. Denote by $\mathbf{s} = [s_x, s_y]$ the x and y coordinate and $\mathbf{v} = [v_x, v_y]$ the velocity. Note that only the state of the position $\mathbf{s}$ and velocity $\mathbf{v}$ are observed, denoted as $\hat{\mathbf{s}}$ and $\hat{\mathbf{v}}$, respectively. To simplify the notation, we define the variable vector $\mathbf{y} = [s_x, s_y, v_x, v_y]$ for the observation state of Y, the variable vector $\mathbf{o} = [\hat{s}_x, \hat{s}_y, \hat{v}_x, \hat{v}_y]$ for the hidden state of O, and the variable vector $\mathbf{c} = [a, \Delta\alpha]$ for the hidden state of C. The graphical model of our proposed DBN is shown in Fig. 2.

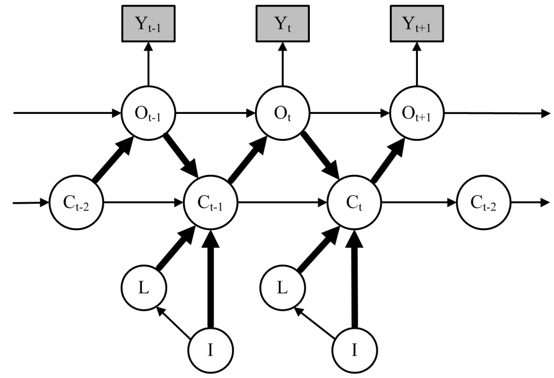


Fig. 2. The graphic model of the proposed DBN. Denote by I the driver intention, L the lane index, Y the observed position and velocity of O, and C the vehicle controlling.

In Fig. 2, the nodes represent random variables and the arcs represent the conditional independence. While Y, O and C are time-dependent continuous variables, both driver intention I and driving lane index L are discrete variables and are assumed to be constant when the vehicle drives through the intersection. More specifically, the lane index L indicates the lane from which the vehicle drives into an intersection. Compared to the HMM, DBN represents the distribution $p(\mathbf{y}(t), \mathbf{o}(t), \mathbf{c}(t)|\mathbf{y}(t-1), \mathbf{o}(t-1), \mathbf{c}(t-1), L, I)$ in a more compact way. We assume that the likelihood of observation given the hidden states has a Gaussian distribution:

$$\begin{aligned} &P(\mathbf{y}(t)|\mathbf{o}(1:t), \mathbf{c}(1:t), L, I) = P(\mathbf{y}(t)|\mathbf{o}(t)) \\ &= \frac{1}{2\pi\sigma_s\sigma_v} \exp\left\{-\frac{||\mathbf{s}-\hat{\mathbf{s}}||^2}{2\sigma_s^2} - \frac{||\mathbf{v}-\hat{\mathbf{v}}||^2}{2\sigma_v^2}\right\} \end{aligned} \tag{3}$$

where the operator $||\mathbf{x}||$ denotes the Euclidean norm of a vector $\mathbf{x}$. The position and velocity of the vehicle can be updated:

$$\begin{cases} s_x(t) = s_x(t-1) + v_x(t-1)\Delta T + w_x \\ s_y(t) = s_y(t-1) + v_y(t-1)\Delta T + w_y \\ ||\mathbf{v}(t)|| = ||\mathbf{v}(t-1)|| + a(t-1)\Delta T + w_a \\ \angle\mathbf{v}(t) = \angle\mathbf{v}(t-1) + \Delta\alpha(t-1) + w_\alpha \\ v_x(t) = ||\mathbf{v}(\mathbf{t})|| \cos(\angle\mathbf{v}(t)) \\ v_y(t) = ||\mathbf{v}(\mathbf{t})|| \sin(\angle\mathbf{v}(t)) \end{cases} \tag{4}$$

where we have $w_x \sim \mathcal{N}(0, \sigma_x^2)$, $w_y \sim \mathcal{N}(0, \sigma_y^2)$, $w_a \sim \mathcal{N}(0, \sigma_a^2)$ and $w_\alpha \sim \mathcal{N}(0, \sigma_\alpha^2)$. The dynamic process of control variables $\mathbf{c}(t) = [a(t) \;\; \Delta\alpha(t)]$ are very important because these hidden states reflect the driver intention I directly with specific turning restrictions from the lane L. Given the states of vehicle $\mathbf{o}(t-1)$ and $\mathbf{c}(t-1)$ at time $t-1$, the lane number L, and the underlying driver intention I, the control variables at time t are sampled from the conditional distribution of $p(\mathbf{c}(t)|\mathbf{c}(t-1), \mathbf{o}(t-1), I, L)$. Due to the fact that the control of vehicle is usually determined by the position of vehicle in a lane through which the vehicle drives into the intersection, we have

$$p(\mathbf{c}(t)|\mathbf{c}(t-1), \mathbf{o}(t-1), I, L) = p(\mathbf{c}(t)|\mathbf{c}(t-1), \mathbf{x}(t-1), I) \tag{5}$$

where $\mathbf{x}(t-1) = [d_{intersection}\ d_{center}\ \alpha]$ in which $d_{intersection}$ is the distance to the intersection, d_{center} is the distance to the lane center, and α is the angle of vehicles driving direction relative to the lane direction. Given $\mathbf{o}(t-1)$ and L at time $t-1$, the features $\mathbf{x}(t-1)$ can be calculated. For each class of turning (i.e., I is given), we assume the distribution $p(\mathbf{c}(t)|\mathbf{c}(t-1), \mathbf{x}(t-1))$ satisfies a Gaussian mixture model (GMM). The parameters of these GMMs and variances σ_x^2, σ_y^2, σ_a^2 and σ_α^2 are estimated offline by using Expectation-Maximization (EM) method.

To infer the unknown hidden states, we employ particle filter approach to approximate the posterior distribution of hidden states $p(I, L, C_t, O_t | Y_{1:t-1}, C_{1:t-1}, O_{1:t-1})$. Compared to the classification methods, such as HMM and SVM presented in previous sections, the proposed DBN is an online inference approach. Meanwhile, because the driving lane L is conditionally dependent on the intention I, the knowledge of available turning option for a lane can be incorporated into our learning model.

Compared to the HMM and SVM approaches, the proposed DBN method offers many advantages: firstly, because the lane index is inferred, we can easily incorporate the traffic rule (i.e., the allowed turning options from that lane) to infer driver's intention, thereby improving the prediction performance; secondly, when the observation of turning signal of one vehicle is available, we can easily extend our DBN model by connecting the observation of turning signal to the hidden state of driver's intention directly, which helps in the inference of driver's intention.

III. Experiments

In our experiments, we evaluate the performance of HMM, SVM and DBN models with our collected dataset. We use 80% driving tracks for training and the remaining 20% tracks for testing.

A. Intersection Driving Data Collection

We collect driving tracks at intersections using a high resolution GPS. We use NavCom GPS with Inertial Plus IMU system. The intersections in our dataset include 5 three way T-junctions and 1 four way intersection. Of these 3 have red lights and the other 3 are stop signed intersections. Some of the intersections in our dataset do not have 90 degree turns making our dataset very challenging.

We use accurate lane-level maps provided by Zenrin in our work. Fig. 3 gives examples of straight driving at two different intersections with extents of lanes on the roads. We adopt the lane model of [11] to model the complex structure of intersections. In our lane model, the intersection consists of roads as ways, and the roads are composed of lanes. Each lane of an intersection is indexed with an identifying number. Based upon the Zenrin map, the turning lanes are automatically built to connect two lanes at the intersection using semi-automated scripts. Fig. 4 shows an example of Zenrin lane data overlaid on Google map data and Figure 5 shows the generated turning lane center-lines at the intersection.

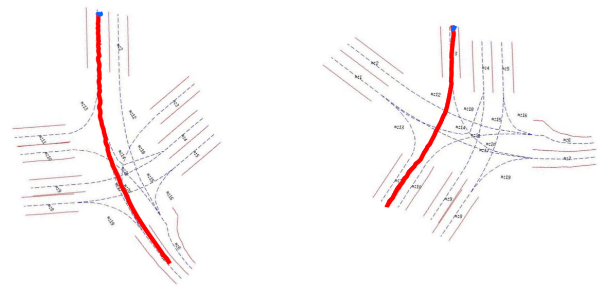


Fig. 3. Examples of two driving tracks in a lane-level map in our real-world dataset

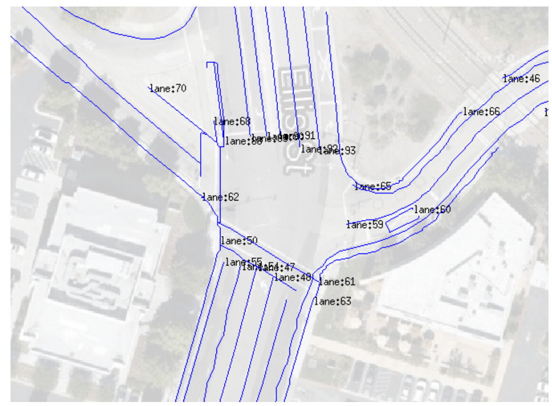


Fig. 4. Lane-level Zenrin lane boundary data overlaid on Google map (best seen in color)

The driving tracks collected in our dataset include Canbus data (velocity, acceleration), and GPS/IMU data (position, car orientation) using 1 car. We collect a total of 403 turning tracks through intersections. Because of errors in the GPS device, such as missing time stamps and drifting, some GPS tracks are found to have errors. We eliminate some of the tracks with major errors, leaving us with 375 tracks from the experiments. The summary of GPS error in our dataset is shown in Table I.

TABLE I
GPS Errors

GPS Error	Number of errors	Used in experiments
Missing Points	3	no
Off by 10s of meters	15	no
Track drifting with time	7	no
U turn	3	no
Bad Time	21	yes
Small or medium drift	9	yes
Track crosses intersection	8	yes
Incorrect lane	4	yes
Track Discontinuity	2	yes

B. Feature Extraction for Classification

We extract five features that are used in HMM and SVM learning models for classification:

1) Distance between vehicle and lane center d_{center};

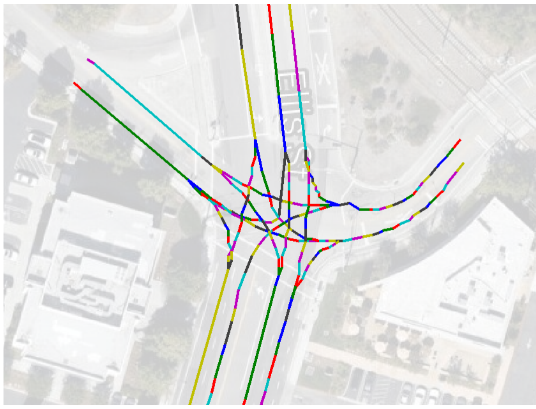


Fig. 5. Generated lane center-line data at the intersection (best seen in color)

2) Distance to the intersection $d_{intersection}$;
3) Angle of vehicle's driving direction relative to the lane direction α.
4) Velocity of vehicle v .
5) Acceleration of vehicle a.

Fig. 6 shows the features of d_{center} and α along with the distance to the intersection $d_{intersection}$ for left/right/straight driving at all intersections. A positive value of d_{center} indicates that the vehicle drives on the right side of lane center, and a negative value of d_{center} indicates the vehicle drives on the left side of lane center. Also a value of α is positive when the vehicle drives to the right of the lane and is negative when it drives to the left of the lane. The point with $d_{intersection} = 0$ is the point that the vehicle drives into the intersection. Both d_{center} and α indicate the turning movement. For example, if turning left, the vehicle is usually more likely located in or towards to the left side of lane, i.e., there is negative d_{center} or negative α. If driving straight, both values of d_{center} and α would be close to zeros. We show the average distance between vehicle and lane center d_{center} approaching and leaving the intersection in Fig. 7, the average yaw rate α in Fig. 8, and the average speed and acceleration in Fig. 9 and 10, respectively. The distinguishing feature characteristic for different turning types leads to high recognition performance using various statistical learning methods.

Our CANBus data is at a much sparser sampling rate than GPS data which is collected at 150 Hz. Hence, we compute velocity and acceleration from GPS positions using Savitzky Golay filtering. We perform smoothing using a 3rd degree polynomial.

C. Results

The prediction performance is measured at various stages as a vehicle approaches an intersection. Results are presented at 2.8 m before intersection ($d_{intersection} = 2.8$ m), at intersection ($d_{intersection} = 0$ m), and 2.8 m and 5.6 m after intersection ($d_{intersection} = -2.8$ m and $d_{intersection} = -5.6$ m). For each stage, we train separate models and obtain performance metrics using the corresponding model only. The DBN approach makes use of the traffic rule for

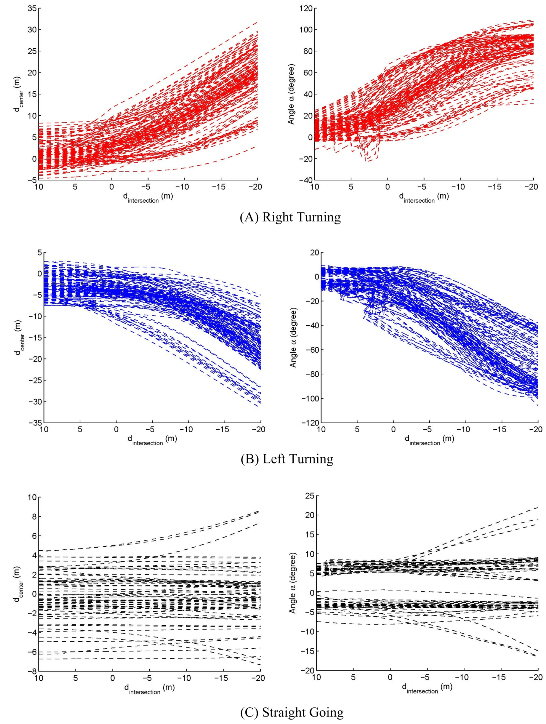


Fig. 6. Features used for classification. On the left are the plots of distance between vehicle and lane center d_{center} vs. distance to the intersection $d_{intersection}$. On right are plots of angle of vehicle α vs. $d_{intersection}$. Zero value on x axis occurs at the intersection with negative values being past the intersection.

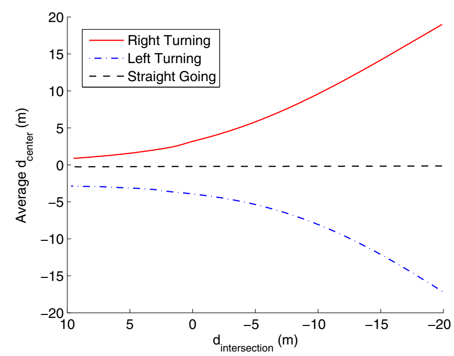


Fig. 7. Average distances to lane center for right turning, left turning and straight going.

inference while both HMM and SVM approaches do not. The prediction performances at different distance level are shown in Table II. The results show that SVM can obtain the best prediction performance 2.8 m before intersection and at intersection and DBN is the best 2.8 m and 5.6 m after intersection. It also shows that one can always obtain better prediction when the vehicle drives closer to the intersection and farther away from the intersection. The average number of choices at intersections is 1.85. So if we guess randomly we get 54% accuracy.

We also analyzed the error cases before and after the intersection. One of the reasons for these errors is the vehicle has not turned when we make the prediction. This may be

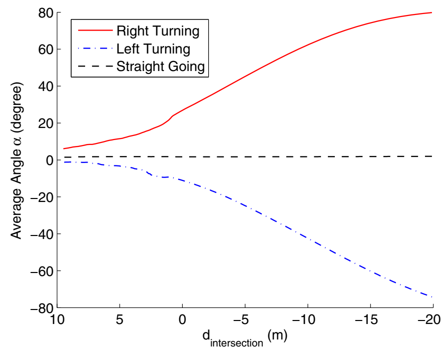


Fig. 8. Average yaw angle for right turning, left turning and straight going.

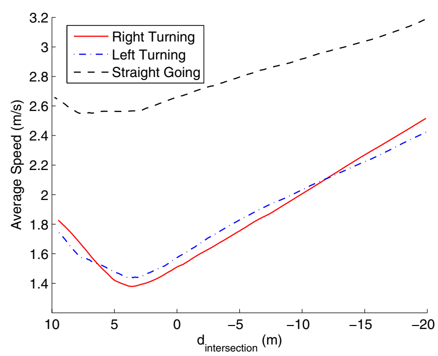


Fig. 9. Average speed for right turning, left turning and straight going.

caused by stop line placement at the intersection, and GPS errors.

IV. Conclusions

In this paper, we have built a general model to predict driver's turning intention at various intersections, which can be used for Intersection Movement Assist (IMA) and Left Turn Assist (LTA) in intelligent vehicle and autonomous vehicle driving in urban area. In our general model, the features that refer to the road layout are extracted by incorporating lane-level road map. Three different statistical learning approaches, including HMM, SVM and DBN, are examined in our collected real-world driving tracks. We get over 90% turn prediction accuracy 1.6 seconds before the intersection. and accuracy of 93.8% at the intersection using SVM.

Our work incorporates lane geometry via high resolution maps and advances the state of the art in ADAS/AD systems by building a turn prediction model that works at general intersections. In the near future, we will examine the prediction performance with the turning indicator and lane index information incorporated into the statistical models.

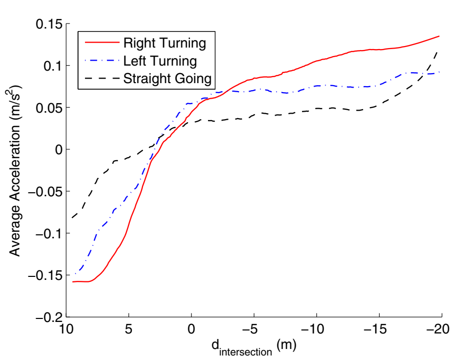


Fig. 10. Average acceleration for right turning, left turning and straight going.

TABLE II
Accuracy in Percentage for Driver's Intention Prediction Using HMM, SVM, and DBN

Distance relative to Intersection	HMM without vel/acc	SVM	Dynamic Bayes Net
2.8 m before	79.1%	**90.6%**	88.4%
at intersection	77.0%	**93.8%**	92.0%
2.8 m after	84.6%	96.5%	**97.0%**
5.6 m after	90.8%	96.8%	**97.4%**

References

[1] G. S. Aoude, B. D. Luders, K. K. Lee, D. S. Levine, and J. P. How, "Threat assessment design for driver assistance system at intersections," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 1855–1862.

[2] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[3] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[4] T. Gindele, S. Brechtel, and R. Dillmann, "Learning context sensitive behavior models from observations for predicting traffic situations," in *IEEE International Conference on Intelligent Transportation Systems*, 2013.

[5] R. Graf, H. Deusch, F. Seeliger, M. Fritzsche, and K. Dietmayer, "A learning concept for behavior prediction at intersections," in *2014 IEEE Intelligent Vehicles Symposium Proceedings (IV)*, June 8-11 2014, pp. 939–945.

[6] R. R. Kumar, S. W. Begum, and M. Manikandan, "Distance based accident prevention in intersection using vanet," in *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, March 2014.

[7] C. Laugier, I. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha, and A. Negre, "Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety," *IEEE Intelligent Transportation Systems*, vol. 3, pp. 4–19, October 2011.

[8] D. N. Lee *et al.*, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976.

[9] S. Lefevre, C. Laugier, and J. Ibanez-Guzman, "Evaluating risk at road intersections by detecting conflicting intentions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7-12 Oct. 2012, pp. 4841–4846.

[10] National Transportation Traffic Safety Administration, "Fatality analysis reporting system encyclopedia," Online, http://www-fars.nhtsa.dot.gov/Crashes/CrashesLocation.aspx, 2010.

[11] B. Rossler, "Situation analysis and adaptive risk assessment for advanced intersection safety systems," Ph.D. dissertation, Universitt Hamburg, 2010.

[12] T. Streubel and K. Hoffmann, "Prediction of driver intended path at intersections," in *2014 IEEE Intelligent Vehicles Symposium Proceedings (IV)*, June 8-11 2014, pp. 134–139.