A Two-Stage-Training Support Vector Machine Approach to Predicting Unintentional Vehicle Lane Departure

Alhadi Ali Albousefi, Hao Ying, Dimitar Filev, Fazal Syed, Kwaku O. Prakah-Asante, Finn Tseng & Hsin-Hsiang Yang


To link to this article: http://dx.doi.org/10.1080/15472450.2016.1196141

Accepted author version posted online: 07 Jun 2016. Published online: 07 Jun 2016.

Article views: 3

View Crossmark data
A Two-Stage-Training Support Vector Machine Approach to Predicting Unintentional Vehicle Lane Departure

Alhadi Ali Albousefi\textsuperscript{a}, Hao Ying\textsuperscript{a,*}, Dimitar Filev\textsuperscript{b}, Fazal Syed\textsuperscript{b}, Kwaku O. Prakah-Asante\textsuperscript{b}, Finn Tseng\textsuperscript{b}, Hsin-Hsiang Yang\textsuperscript{b}

\textsuperscript{a}Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48374, USA

\textsuperscript{b}Ford Motor Company, Dearborn, MI 48121, USA

\textsuperscript{*}Corresponding author; hao.ying@wayne.edu

Abstract

Advanced driver assistance systems, such as unintentional lane departure warning systems, have recently drawn much attention and efforts. In this study, we explored utilizing the nonlinear binary support vector machine (SVM) technique to predict unintentional lane departure, which is innovative as the SVM methodology has not previously been attempted for this purpose in the literature. Furthermore, we developed a two-stage training scheme to improve SVM’s prediction performance in terms of minimization of the number of false positive prediction errors. Experiment data generated by VIRTTEX, a hydraulically powered 6-degrees-of-freedom moving base driving simulator at Ford Motor Company were used. All the vehicle variables were sampled at 50 Hz and there were 16 drowsy drivers (about three-hour driving per subject) and six control drivers (approximately 20 minutes driving each). A total of 3,508 unintentional lane departures occurred for the drowsy drivers and 23 for the control drivers. Our study involving these 22 drivers with a total of over 7.5 million prediction decisions demonstrates that: (1)
excellent SVM prediction performance, measured by numbers of false positives (i.e., falsely predicted lane departures) and false negatives (i.e., lane departures failed to be predicted), were achieved when the prediction horizon was 0.6 s or less, (2) lateral position and lateral velocity worked the best as SVM input variables among the nine variable sets that we explored, and (3) the radial basis function performed the best as the SVM kernel function.

Keywords

Unintentional lane departure, prediction, support vector machines.
1. Introduction

Driver’s inattentiveness is one of the most important causes for unintentional vehicle lane departure. According to a NHTSA report, abnormal lane departure leads to dangerous driving conditions on roadways (National Highway Traffic Safety Administration, 2010). More and more new vehicles are adopting advanced driver assistance systems to help the driver with information, warning, and proactive support (Vahidi and Eskandarian, 2003). Significant R&D efforts worldwide are concentrating on advanced driver assistance systems as forward collision warning, forward crash mitigation and avoidance with active braking, lane departure warning, lane keeping assist, and blind spot monitoring for lane change (Tideman et al., 2007).

Our research focused on unintentional lane departure prediction. The prediction algorithms in the literature can be classified into three categories (Tideman et al., 2007). The first category relies on the time to lane crossing (TLC), first proposed by (Godthelp et al., 1984), which estimates the amount of time that a vehicle will take to deviate out of the ego road boundary. (Lin and Ulsoy, 1996) presented a TLC algorithm that took such factors as vehicle vibration, external disturbances to vehicle (e.g., strong wind), and vehicle state measurement errors into account and studied their impact on the accuracy of TLC estimation. In another effort (Cario et al., 2009), real-time TLC estimation was obtained from road images. The lane lines and TLC were estimated using a data fusion algorithm. Yet another method was developed to reduce the false alarms resulting from the TLC-based lane departure prediction (Angkititrakul et al., 2011). Gaussian mixture models were employed to model stochastic driving behavior such as lane-cross events and driver correction events. Driving signals observed in real-time were utilized to determine a lane departure warning. A distance to line crossing method based on TLC and
vehicle kinematics was developed (Mammar et al., 2006). It estimated the future trajectory of the vehicle based on such vehicle states as the speed and steering angle. Test results involving the real data showed that to use the TLC as a reliable lane departure indicator, the vehicle dynamics needed to be included. In general, the TLC-based techniques are susceptible to false alarms, especially when the driver tends to drive close to the lane markers. This is primarily due to the theoretical assumptions and approximations necessary for the TLC computation. From practice standpoint, it is difficult to accurately determine the TLC partially owing to the nonlinearity, absence of enough vehicle state information, and the difficulty to accurately estimate road geometry.

Detection of the lane boundary using real-time road images is the second category. Lane width and continuity of lane markings were employed to develop a lane detection system in urban roads using non-calibrated cameras (Leng and Chen, 2010). The lane boundary filter was used to identify the lane boundary pixels in the images first and then the Hough transform was used to find lane boundary lines. A vision-based system capable of detecting the lane departure using image processing and the road geometry model was developed (Liu et al., 2008). A gradient method was applied to detect the both lane marks. Furthermore, lane departure detection through identifying the two lane markers in the road images was proposed (Dong et al., 2012). The endpoint of the lane in the images was employed to identify the lane. (Borkar et al., 2011) reported a multiple-camera technique for lane detection. The lane regions were estimated by cross-correlating the images with the predefined templates. The processing results of the front and rear camera images were combined using data fusion to achieve the detection of the lane lines. A method for lane departure prediction without using a lane model was also proposed
To avoid the difficulty in lane recognition, this method attempted to predict the tendency of lane-leaving based on the detection of lane boundary points. In another effort, a lane departure model was developed by monitoring the position of the vehicle with respect to the detected lane boundaries using a video camera (Cualain et al., 2012). The model was developed using a subtractive algorithm and Kalman filtering in the Hough transform domain. Lane marking segmentation was carried out to segment the region of interest from the video images. In (Gaikwad and Lokhande, 2012), the authors proposed to improve the lane departure detection by applying the Hough transform to detect lines in the segmented regions of interest. A lane departure decision was made based on the distance between the lane lines. In summary, good lane departure detection results were often claimed without providing detailed detection performance. Most of the techniques in this category relied on fast CPUs and large memory capabilities to process video data streams in real-time (Lin and Su, 2011). (Hadi and Sinha, 2011) proposed a vision-based lane departure warning system using images of lane markings. The effects of marking’s retroreflectivity on the performance of lane departure warning system were studied based on experiments conducted in the field and in a simulated rain facility. Accuracy of the lane departure detection heavily depended on the accuracy and reliability of the lane marking determination, which by itself is a key research subject at present.

The last category of the prediction algorithms is based on vehicle position estimation with respect to the lane lines using vehicle variables. Relatively fewer reports are available. One such algorithm involved the radial basis probability neural network trained by utilizing lane deviation scenarios characterized by vehicle’s lateral position and speed (Chang et al., 2008). Another algorithm detected lane departure through determining the vehicle’s current position with respect
to the lane lines determined by the video-based lane detection algorithm (Risack et al., 2000). In yet another effort (Dahmani et al., 2015), a lane departure algorithm was developed by using a fuzzy observer to estimate the road curvature and lateral vehicle dynamics. The curvature was compared with the vehicle trajectory to detect lane departures. The algorithms in this category must be able to effectively deal with noisy and sometime inconsistent data in order to produce accurate prediction results. There exist some overlaps in our three categories. Some of the TLC-based lane departure detection techniques and some of the vehicle-variable-based vehicle position estimation methods share a common feature - they use real-time road images. The main difference between these techniques is that the TLC-based methods employ the images to calculate the time that the vehicle is expected to cross the lane line whereas the vehicle position estimation techniques use the images to estimate the vehicle variables and then use these variables to detect lane departures. The real-time road image-based techniques are different from these two categories. They utilize different image processing techniques to process the images to detect the lane lines and find position of the vehicle with respect to the detected lane lines.

Support vector machine (SVM) is a widely used supervised machine learning technique that is based on statistical learning theory (Vapnik, 1995; Cortes and Vapnik, 1995; Burges, 1998). Unlike other techniques that attempt to perform nonlinear classification or regression directly on original features of a data set, SVM first transforms the data via a nonlinear function to a feature space whose dimension may be much higher than that of the original feature space. The transformation is implicitly achieved through a kernel function. Importantly, in the new space, the original nonlinear classification (or regression) problem becomes a linear classification (or regression) problem. SVM then finds a plane/hyperplane that separates two classes of the data.
with a maximum margin through quadratic programming that minimizes the overall classification (or regression) error. This results in a global optimum and hence avoids the common local optimum limitation suffered by most of other machine learning methods (e.g., neural networks). SVM often solves challenging problems more effectively with superior generalization capability than other methods (Shao and Lunetta, 2012; Lin and Wang, 2002; Pradhan, 2013). SVM is characterized by its high accuracy and ability to deal with high-dimensional data (Ben-Hur and Weston, 2010). It reportedly outperformed other machine learning methods when used in time series-based classification (Tay and Cao, 2001). SVM can provide better image classification than other algorithms for real-world remote sensing data as well as simulated data (Mountrakis et al., 2011).

In this paper, we show how to use the SVM approach to predicting an unintentional lane departure, which, to our knowledge, has not been reported in the literature. More specifically, we developed a two-stage SVM training scheme that employed a nonlinear binary SVM to classify vehicle variable time series (e.g., lateral position and lateral velocity) to predict occurrence of a lane departure. We reported the preliminary results on one-stage training and testing of the SVM on lane departure prediction (Albousefi et al., 2014). The present paper significantly extends the scope and depth of our research by introducing the two-stage SVM training and exploring longer prediction horizons. Additionally, more comprehensive evaluation of the one-stage trained SVM is provided and the resulting performances are compared with performances of the two-stage SVM.

The rest of the paper is organized as follows. In the next section, we first provide a brief introduction to the SVM methodology and then present our two-stage SVM training scheme,
which is followed by description of our experiment data and how they were used to train and test the SVM for lane departure prediction. The training and testing results are described in Section 3 and the conclusion is given in Section 4.

2. A Two-Stage SVM Training Scheme for Lane Departure Prediction

In a two-class classification application, which is the case of this study, a SVM is trained with input-output examples to find the best decision function to separate the two-class test data with a minimum error. There are two classes of SVMs, linear and nonlinear. A linear SVM can perfectly separate classes of data that are linearly separable by a hyperplane (in the case of binary SVM) or a set of hyperplanes (in the case of multi-class SVM) but may perform poorly when data are not linearly separable, in which case a nonlinear SVM should be utilized. For predicting lane departure events, we used a binary nonlinear SVM with two class labels: 1 (lane departure) and –1 (not lane departure). Before presenting our two-stage SVM training scheme, we will first briefly review how a binary nonlinear SVM works.

2.1 The Nonlinear Binary SVM Used in This Study

Let $X$ and $Y$ be the input set and output set of the data to be classified, respectively. For binary classification, output is $y_i \in Y = \{1, -1\}$. Let’s suppose there are $M$ training examples with $n$ features, $(x_i, y_i)$ where $x_i \in X \in \mathbb{R}^n$, $i=1,2,\ldots,M$. A data point to be classified is denoted $x \in X$. We assume that the data that are not linearly separable. The main idea of training a SVM is to find such a decision function that can best classify the data into one of the two classes. To achieve this goal, a nonlinear binary SVM first maps, via a function $\phi$, the $M$ data points to a higher (can be infinite) dimensional space (i.e., $x_i \rightarrow \phi(x_i)$) so that the mapped data in the new
space (called the feature space) become linearly separable (Vapnik, 1995). The feature space is a vector space where the dot product is applicable. The binary SVM’s decision function is

\[ D(x) = \text{sgn} \left[ \sum_{i=1}^{M} a_i \phi(x_i) \cdot \phi(x) + b \right] = \text{sgn} \left[ \sum_{i=1}^{M} a_i k(x_i, x) + b \right] \quad (1) \]

where \( K(x_i, x) = \phi(x_i) \cdot \phi(x) \) is called the kernel function which implicitly represents the dot product of the mapped functions. Parameters \( a_i \) and \( b \) are to be learned through the training examples, where \( 0 \leq a_i \leq c \) and \( \sum_i a_i y_i = 0, \quad i = 1, \ldots, M \). Parameter \( c \) is a positive constant called the regularization parameter which controls the tradeoff between classification margin maximization and error minimization.

Various kernel functions are studied in the literature. They include the linear kernel, polynomial kernel, exponential kernel, and radial basis function (RBF) kernel, which have been utilized in a wide range of applications. The polynomial kernel is represented by \( K(x_i, x) = (x_i \cdot x + 1)^d \), which leads to the linear kernel when \( d=1 \). The RBF kernel is \( K(x_i, x) = \exp(-\|x_i-x\|^2 / \sigma^2) \) where \( \sigma \) is a design parameter. Even though choosing the most appropriate kernel function for a given application is important, no general method is available. Kernel function selection is application-dependent in nature and hence a certain degree of trial-and-error effort is required. Performance of a SVM also depends highly on its parameters, including the kernel parameters. There is no general theory to determine their values; they must be determined experimentally.

2.2 The Two-Stage SVM Training Scheme

Initially, we explored a set of vehicle variables as inputs of the SVM and trained it in a standard manner to develop a lane departure prediction system. The preliminary results showed
that the SVM was able to predict the majority of the lane departure events. However, there were a quite significant number of falsely predicted lane departures. We carefully analyzed the results and found that most of the SVM classification errors were made when the vehicle was close to the inner edges of either side of the lane boundaries. Because these situations resembled real lane departure patterns, they caused the SVM to misclassify. To better understand these types of situations, let us see two specific cases. Figure 1 shows two different types of driving maneuvers characterized by vehicle lateral position recorded during one of our drowsy driving experiments. The width of the simulated lane was 3.81 m. A lateral position of 0 m means the vehicle is in the center of the lane while a positive or negative lateral position means the vehicle is deviated toward the right or left side of the lane, respectively. A simulated 2000 Volvo S80 was used in our study and its body width is 2.19 m. Hence, the vehicle is out of the lane when its lateral position is either greater than 0.81 m or less than -0.81 m. Fig. 1(a) displays a driving pattern when the driver barely managed to prevent the vehicle to cross the lane boundary (i.e., maintaining the lateral position to be less than 0.81 m). The vehicle was very close to the boundary. Clearly, this is not a lane departure event. Fig. 1(b) shows a similar driving pattern that is a lane departure event – the vehicle is outside of the lane because its lateral position is greater than 0.81 m. Our early investigation showed that the high similarity in cases like these (there were numerous such cases in our experiments) posed a challenge to our SVM that was standardly trained, resulting in many false positive errors (i.e., falsely predicted lane departures).

We thus developed a two-stage SVM training scheme to enable the SVM to better handle these types of situations. “Two-stage” means there are two regular training-testing sessions (or stages) in series. The false positive errors made by the SVM during the testing of the first session
were utilized as part of the training data to train the same SVM in the second session so that the number of false positive errors would be maximally reduced. A potential side effect was slightly decreased detection sensitivity to true lane departure. From a practice standpoint, such a tradeoff is well justified because if the driver is mistakenly alerted too many times, he or she can become annoyed and will likely discontinue the use of the system. Therefore, our system design was focused on false positive error reduction. For convenience, the two-stage trained SVM and the standardly trained (i.e., one-stage trained) SVM are named SVM 2 and SVM 1, respectively. We will show their training and testing performances as well as their comparison results.

2.3 Settings for SVM Training and Testing

2.3.1 Data Source for SVM 1 and SVM2

Due to constraints on our resource to produce real-world driving data at this stage of research, we turned to simulated driving data. SVM1 and SVM 2 were trained and tested using the driver experiment data generated by the VIRtual Test Track Experiment (VIRTTEX), a hydraulically powered 6-degrees-of-freedom moving base driving simulator at Ford Motor Company, Dearborn, USA. The experiments were conducted by a group of Ford researchers for evaluating four different human machine interfaces for lane departure warning around 2004 (Kozak et al., 2006). This simulator was designed to accommodate a full-size, interchangeable vehicle cab. Tactile, visual and sound cues were provided so that the driver was fully immersed into the driving task. Realistic sound of the road, wind and engine operation was played over a sound system. The cab had a steering control loader that supplied accurate feedback of road and tire forces to the driver. Projectors were used to form the driving scene on the spherical display surface. Projectors produced a forward field of view and a rear field of view.
The experiment participants, all were licensed adult drivers and signed consent forms, were divided into two groups – a drowsy group and a control group. The drowsy drivers were deprived of sleep for almost one day before starting an approximate three-hour simulated driving of a 2000 Volvo S80 while the control drivers drove the same vehicle for about 20 minutes after having a full night of sleep. The drive was on a simulated 514-km long, dry, four-lane (two for each direction) U.S. interstate under nighttime conditions. The road, divided by a median or concrete barrier, had 118 curves, each of which had a radius of over 900 m, super-elevation of 4%, with entrance and exit spirals over 150 m long. These curves were quite evenly distributed. The road shoulders had increased roughness to emulate realistic road conditions during a lane departure. The driver was asked to drive in the right lane only. Traffic density was very low - less than one overtaking vehicle per minute. A stream of opposing traffic was presented on the other side of the median/barrier, and did not interfere with the driver in any way. More than 100 vehicle variables (e.g., speed, lateral position, and steering angle) were collected. The sampling period T was 0.02 s.

We used the experiment data representing 16 drowsy drivers and six control drivers for the SVM lane departure prediction study. For each driver, the vehicle variable time series at the beginning of the experiment when vehicle speed was below 20 MPH were removed, so were the data at the end of the experiment after the brake was fully applied. The remaining time series were used in the training and testing of the SVMs. There were a total of 3,508 lane departures for the 16 drowsy drivers and only 23 for the six control drivers (two had none).
2.3.2 Preparation of Training Sets

Like any SVM application, selecting appropriate vehicle variables as input variables was critical to the success of our SVM. Guided by knowledge and intuition, we explored the following nine sets of input variables: (1) lateral position, (2) lateral position and lateral velocity [defined as (current lateral position – previous lateral position) / T], (3) lateral position, lateral velocity, and speed, (4) lateral position, steering angle, and speed, (5) lateral position, steering angle, and yaw deviation, (6) lateral position, speed, and yaw deviation, (7) lateral position and yaw deviation, (8) lateral position and steering angle, and (9) lateral position, lateral velocity, and change in steering angle. The speed was normalized to [0, 1] whereas the steering angle, yaw deviation, and change in steering angle were normalized to [-1, 1]. The normalized ranges were computed based on the maximum and minimum values of the variables involved among all the drivers. In a real-world scenario, the system developer could assume reasonable values for the maximum and minimum and treat any values above the maximum as the maximum and below the minimum as the minimum before normalizing. Alternatively, the maximum and minimum may be obtained in real-time based on vehicle sensor measurements.

The prediction horizon for SVM 1 and SVM 2 (i.e., the amount of time in advance to detect a lane departure before it occurs) was set at three different levels: 0.2 s, 0.4 s or 0.6 s (our preliminary testing showed that a horizon larger than 0.6 s resulted in poor SVM prediction performance). Multiple training examples were created from a single lane departure event by using \( P \) data points prior to the lane departure moment. Each example consisted of \( N \)-point time series of the vehicle variables where \( N < P \). Choosing appropriate \( N \) (i.e., time window size) was very important in order for the SVMs to attain the best prediction results. There is no general
method for determining the optimal window size. In this study, it was decided experimentally. Table 1 shows, as an illustrative instance, how 12 training examples are generated, where a lane departure is assumed to occur at time $17T$. In this case, $P=16$, $N=5$, and the prediction horizon is 10 data points or 0.2 s (hence the lane departure class label for $7T$ to $16T$ is set 1 and the rest of times have a label -1). Each row of the five consecutive # symbols below the “Class Label” row represents a time series training example. Note that the class label of a training example is determined by the time instance of the last data point in the example (e.g., –1 for Example 1 and 1 for Example 3). Generalizing from this example, the following relationship is obvious:

$$\text{number of training examples} = P \times \text{number of lane departure events} - N + 1$$

Each data point in the time series was treated as an input variable for the SVMs. Thus, there were $3N$ input variables if three vehicle variables were involved in the $N$-point long time series.

SVM 1’s training dataset was named Training Set A. There can be different ways to create a training dataset. Training examples should be produced without bias. In this spirit, we chose to use all the odd numbers of lane departures (1,756 in total) as the base to create the training examples in Training Set A. The total numbers of training examples for prediction horizon 0.2 s (using $P=36$ and $N=6$), 0.4 s (using $P=74$ and $N=8$), and 0.6 s (using $P=84$ and $N=10$) were 63211, 129937 and 147495, respectively. They represented a total driving time of 21.07 minutes, 43.31 minutes and 49.17 minutes, respectively. Together, they represented less than 5% of the total driving time for the 22 drivers, which was 42 hours and 7 minutes (after the low speed data were removed).
SVM 2’s training dataset for the first-stage training was exactly identical to SVM 1’s (i.e., Training Set A). The training dataset for the second-stage training, termed Training Set B, was created as follows: we employed all the examples in Training Set A plus examples constructed using all the false positive errors made by SVM 2 (59 in total as it turned out) during the first-stage learning in the case of the 0.4 s prediction horizon and 80% of all the false positive errors (150 in total) for the 0.6 s prediction horizon. The number of training examples generated from one false positive event equaled to the number of consecutive non-lane departure examples in the testing dataset for SVM 2 after its first-stage training (defined as Testing Set B1 in Section 2.3.3) that were misclassified by SVM 2 as lane departure. As a result, the total numbers of training examples in Training Set B were 132,301 and 154,194, respectively, for the 0.4 s and 0.6 s prediction horizons. They represented a total driving time of 44.1 minutes and 51.4 minutes, respectively. Comparing to Training Set A, a total of 2,364 and 6,699 false positive examples were added to Training Sets B for the 0.4 s and 0.6 s prediction horizons, respectively, which represent a 2.35% and 4.54% increase in the respective total driving time.

Adding part of the false positive cases produced in the testing of the first-stage-trained SVM 2 to SVM 2’s first-stage training dataset (i.e., Training Set A) generated a slightly larger training set, which was Training Set B. The nature of Training Set B put an emphasis on the classification errors made by SVM 2 during its first-stage training and gave the SVM a second chance to classify these same scenarios correctly in the second-stage training. As we will show, the numbers of false positives after the second-stage training indeed reduced dramatically.

The two-stage training scheme was not needed for the 0.2 s prediction horizon because there were only two false positive cases for all the 22 drivers after SVM 1 was trained.
Preparation of Testing Sets

The testing dataset for SVM 1 was called Testing Set A. To create it, the data in the Training Set A were excluded from the 22 drivers’ original data files (after the low speed data were removed) and the resulting files formed Testing Set A. Unlike SVM 1, SVM 2 was assessed involving only the 16 drowsy drivers and only at the 0.4 s and 0.6 s prediction horizons. This was because for the six control drivers, the testing results of SVM 1 were almost perfect – there was only one false positive each at the 0.4 s and 0.6 s prediction horizons. Also, for the 22 drivers, there were only two false positives at the 0.2 s prediction horizon. SVM 2 had two training stages, which required two testing datasets, and they ought to be mutually exclusive in order to achieve the most objective assessment. To this end, each of the 16 drowsy driver’s data in Testing Set A was divided into two equal-length time series files from the middle point of time to form Testing Set B1 using the first time series and Testing Set B2 using the second time series. Testing Sets B1 and B2 were used to test SVM 2 after its first- and second-stage of training, respectively.

An N-point sliding time window moving one data point at a time was used to generate test cases from Testing Set A and Testing Sets B1 and B2 for each driver. The total numbers of testing examples in Testing Sets B1 and B2 were the same. For the prediction horizon 0.4 s (using \( P=74 \) and \( N=8 \)), and 0.6 s (using \( P=84 \) and \( N=10 \)), they were 3,429,697, and 3,420,918, respectively. They represented a total driving time of 19.054 hours, and 19.000 hours, respectively. Testing Set A = Testing Set B1 + Testing Set B2. Thus, the total number of testing examples in Testing Set A for these two prediction horizons is the sum of those of Testing Sets B1 and B2. As for the prediction horizon 0.2 s (using \( P=36 \) and \( N=6 \)), the total number of testing examples was 3,463,060, which translated to a total driving time of 19.239 hours.
A lane departure prediction was considered correct only if at least four, seven, and nine consecutive windows of a test case were classified by SVM 1 or SVM 2 at each testing stage as lane departure for the 0.2 s, 0.4 s, and 0.6 s prediction horizons, respectively. These were the optimal values based on our experiments.

2.3.4 Other SVM Settings

We experimented with the linear, polynomial and RBF kernels for the SVMs, which are the most popular kernels in the literature. A 15-fold cross validation was first executed using Training Set A to find good estimates for the initial values of the RBF kernel parameter $\sigma$ and the SVM regularization parameter $c$. Then, different combinations of the $c$ and $\sigma$ values were explored experimentally to find the best combination. The same approach was used to find the best combination of the polynomial kernel parameter $d$ and the regularization parameter $c$.

We implemented the SVMs using the MATLAB-based freeware Statistical Pattern Recognition Toolbox, STPRtool for short (Franc and Hlavác, 2004), which contained various pattern recognition methods, including the binary nonlinear SVM. The toolbox was initially developed in 1999 and has been extended to include many pattern recognition algorithms. We used version 2.11 of the software released in August 2011. A PC with an Intel i5 CPU and 6 GB RAM was utilized. The computer execution time for the training ranged from 1 hour to almost 25 hours, depending on the input variables and SVM settings.
3. Lane Departure Prediction Results

3.1 SVM Training Results

Among the three kernel functions with which we experimented, the linear kernel performed the worst - it failed to predict virtually any of the lane departures. Moreover, among the nine input variable combinations, the combination of lateral position and lateral velocity produced the best results for both SVM 1 and SVM 2. Hence, we will report only the training and testing results for the two SVMs that use the RBF kernel (with the experimentally determined optimal values of $c = 10$ and $\sigma = 0.1$) and the second-order polynomial kernel (with the experimentally determined optimal values of $c = 10$ and $d = 2$) in the rest of the paper. We will show only the training results involving lateral position and lateral velocity in the current subsection and will focus on this input combination in Section 3.2 that covers the testing results.

Table 2 shows how the false positives and false negatives (lane departures failed to be predicted) changed with different window sizes at 0.2 s, 0.4 s, and 0.6 s prediction horizons for SVM 1 when the RBF kernel was used and the input variables were lateral position and lateral velocity. Similar trends were observed for SVM 2 and for the other eight input combinations, independent of the kernel types. As the window size increased, the number of false negatives increased and the number of false positives decreased. This was because for a shorter window size, the SVMs became more sensitive to subtle and rapid changes (e.g., an abrupt lane departure) and thus tended to predict lane departures more often. At the same time, however, small changes in the time series could be mistakenly treated as lane departure, leading to more false positives. By the same token, a larger window size would cause fewer false positives but
more false negatives. The tradeoff between the numbers of false positives and false negatives led us to a balanced choice of the window sizes, which were 0.12 s, 0.16 s, and 0.20 s for the 0.2 s, 0.4 s, and 0.6 s prediction horizons, respectively.

### 3.2 SVM Testing Results

To determine which input variable combination yields the best prediction outcome, Table 3 shows the numbers of overall recall (the ratio of the total number of lane departures correctly predicted to the total number of actual lane departures for all the drivers) and overall precision (the ratio of the total number of lane departures correctly predicted to the total number of events classified as lane departures by the SVM for all the drivers) for all the nine input variable combinations at the 0.2 s and 0.4 s prediction horizons when either the RBF kernel or the second-order polynomial kernel was used by SVM 1. There are two numbers in each cell of the table that are separated by the symbol /. The first number represents the RBF kernel result while the second the polynomial kernel result. According to the table, when the prediction horizon became longer, the numbers of false positives and false negatives got worse as indicated by the degradation of the overall recall and overall precision. Moreover, when one input variable combination performed better than the other eight combinations at the 0.2 s prediction horizon, most likely it also did the best at the 0.4 s prediction horizon. These observed patterns extended to the 0.6 s prediction horizon (not shown here). At the 0.2 s prediction horizon with the RBF kernel, Input Variable Combinations 1, 2, and 8 had the highest recall (which were 100%, 99.77465%, and 99.88732%, respectively). Among them, Combination 2 had the largest precision, which was 99.88720%. Because the precision and recall had to be considered
concurrently, Combination 2 stood out as the best combination among the nine. Note that even though the precision of Combination 9, 99.94347\%, was slightly better than 99.88720\% of Combination 2, its recall, 99.60563\%, was appreciably worse than that of Combination 2, 99.77465\%. Analysis along this line led to the conclusions that Combination 2 was the best for RBF kernel SVM 1 at the 0.4 s prediction horizon and second-order polynomial kernel SVM 1 at the 0.2 s prediction horizon. Speaking of second-order polynomial SVM 1, it performed much worse than RBF SVM 1 at the 0.4 s prediction horizon. Based on this fact as well as the desire of having a SVM capable of predicting as far in time as possible, in what follows, we will focus only on the testing results of SVM 1 and SVM 2 involving the RBF kernel and Input Variable Combination 2. It is important to point out that unlike some of the vehicle variables that can be measured in simulated driving only, lateral position and lateral velocity can be obtained in practice during vehicle operation.

Table 4 shows the testing results of RBF kernel SVM 1 for the 22 drivers using Testing Set A with the lateral position and lateral velocity being the input variables. At the 0.2 s prediction horizon, there were merely two false positives and four false negatives for all the 22 drivers, which translated to only one false positive alarm for approximately every 19.5 hours of driving. The prediction accuracy for the 0.4 s prediction horizon was substantially lower – 138 false positives and 10 false negatives, which were reflected by the lower overall recall and precision in the table. To better reflect how SVM 1 performed on the drivers individually, we present the average recalls and precisions as well as their standard deviations at different prediction horizons in Table 4. The average recall was defined as the average of the individual recalls of all the drivers, and the standard deviation of the recalls was the dispersion of the individual recalls from
the average recall. While the average precision was the average of the individual precisions of all the drivers, the standard deviation of the precisions was the dispersion of the individual precisions from the average precision. As expected, the table shows that as the prediction horizon became longer, the recalls and precisions decreased.

For the results listed in Table 4, the overall averages of the actually achieved prediction horizons for the 22 drivers were 0.200s, 0.401s, and 0.594s with the corresponding standard deviations being 0.003s, 0.008s, and 0.019s for the intended 0.2 s, 0.4 s, and 0.6s prediction horizons, respectively. Obviously, the actual averages were very close to the intended prediction horizons. Furthermore, the standard deviations were very small, indicating the intended prediction horizons were achieved quite evenly across the drivers.

SVM 2 was tested using Testing Sets B1 and B2. To show the benefits of the second-stage training, Table 5 compares the testing results of RBF kernel SVM 1 and SVM 2 after its second-stage training with the same kernel function against the 16 drowsy drivers in Testing Set B2. Both SVMs used lateral position and lateral velocity as input variables. According to the table, SVM 2 predicted lane departure more accurately than SVM 1 did and at the same time generated far fewer false positives (18 vs. 78 for the 0.4 s prediction horizon and 63 vs. 181 for the 0.6 s prediction horizon). For these 16 drivers, the total numbers of false negatives at the 0.4 prediction horizon were 7 and 3 for SVM 1 and SVM 2, respectively, and were 11 and 9 at the 0.6 s prediction horizon. Note that when the numbers of true positives (i.e., correctly predicted lane departures) and false positives were the same for SVM 1 and SVM 2, the numbers of true negatives (i.e., correctly predicted non-lane departures) might or might not be the same because
the numbers of examples involved in an affirmative lane departure prediction could be different. Table 6 is the confusion matrix showing the second-stage training results of the SVM 2 model.

Fig. 2 demonstrates visually one specific case that is part of the statistics in Table 5. SVM 1 was tested by this case and classified it as lane deviation – a false positive error. SVM 1 made the mistake because the vehicle maneuver indeed looked very much like a real lane deviation situation. The vehicle was approaching the lane boundary and the driver made a steering correction in the last moment to avoid lane deviation and return the vehicle back to the lane center. Labeling cases like this as non-lane deviation and adding them to the training set for SVM 2 helps SVM 2 substantially reduce its false positive errors.

Table 7 depicts the overall averages and standard deviations of the actual prediction horizons achieved by SVM 1 and SVM 2 presented in Table 5. The achieved horizons were very close to the targeted horizons for both SVMs and the standard deviations were very small.

In summary, the significantly improved precisions for SVM 2 reflect marked reduction in the number of false positive errors. All these comparison data demonstrate the effectiveness of the second-stage training, resulting in a better SVM.

4. Conclusion

We developed a two-stage training scheme to significantly improve the SVM’s prediction performance. To compare the performance of SVM 1, a standard SVM, and SVM 2, the SVM with two-stage training, we optimized them by experimentally finding the most fitting kernel functions, which turned out to be the RBF function, and parameter values. We also experimentally determined the most appropriate vehicle variables as SVMS’ input variables.
They were found to be the lateral position and lateral velocity. The experiment results involving the 22 drivers with a total of over 7.5 million prediction decisions showed that the optimized SVM 2 performed far better than the optimized SVM 1, especially in terms of number of false positives (18 and 63 vs. 78 and 181 at the 0.4 s and 0.6 s prediction horizons, respectively). While the 0.4s and 0.6s prediction horizons may not always give a human driver enough time to avoid a lane departure, the departure warning signal may be early enough for vehicle’s active control systems to proactively control the vehicle and maintain it within the lane or minimize the extent of a lane departure. More research is needed in this regard.

This study is based on the experiment data generated by VIRTTEX. While this simulator is reasonably realistic, there are still a significant gap between the simulator and real vehicles running under much more complex and dynamic traffic conditions in real-world. Also, measurements of various vehicle and road variables produced by the simulator are readily available for the SVMs and the measurements are always accurate with little signal contamination. This is certainly not the case for real vehicles and roads. Computing power for SVM operation in a real vehicle can be another bottleneck. Still another limitation of our study is a relatively small number of drivers.

Future studies should cover all different types of highways and local roads and also include many different traffic conditions. A large number of drivers should be used. Effort must be made to further reduce false positives and negatives for SVM lane departure prediction. Another interesting research direction is how to use driving data of a group of drivers to not only predict a lane departure for these drivers but also assist in predicting a lane departure for drivers unrelated to the group. From supervised learning perspective, this can be a challenging problem.
In summary, the SVM approach appears to be promising for lane departure prediction. The two-stage-training SVM approach tested in this study has the potential to be used in practice after it is fully developed in the future.

Acknowledgement

The authors are thankful to Mike Blommer of Ford’s VIRTTEX Lab for providing the drowsy driving experiment data that were utilized in this study. This work was supported in part by a Ford Motor Company’s University Research Program grant given to Hao Ying via Wayne State University as gift money.
References


Table 1. Illustration of constructing training examples from one lane departure event.

<table>
<thead>
<tr>
<th>Time</th>
<th>$T$</th>
<th>$2T$</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Example</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#</td>
</tr>
<tr>
<td>Example</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#</td>
<td></td>
</tr>
</tbody>
</table>

Downloaded by [University of Arizona] at 11:34 14 June 2016
Table 2. How numbers of false positives and false negatives changed with the window size for RBF kernel SVM 1 using lateral position and lateral velocity as inputs.

<table>
<thead>
<tr>
<th>Window size (s)</th>
<th>Number of false negatives</th>
<th>Number of false positives</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.2 s prediction horizon</td>
<td>0.4 s prediction horizon</td>
</tr>
<tr>
<td>0.08</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>0.12</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>0.16</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>0.20</td>
<td>9</td>
<td>16</td>
</tr>
<tr>
<td>0.24</td>
<td>10</td>
<td>24</td>
</tr>
</tbody>
</table>
Table 3. The overall recall and overall precision generated during the testing of RBF kernel SVM 1 and second-order polynomial kernel SVM 1.

<table>
<thead>
<tr>
<th>Input variable combination</th>
<th>Overall Recall (%)</th>
<th>Overall Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.2 s prediction horizon &amp; RBF/polynomial</td>
<td>0.2 s prediction horizon &amp; RBF/polynomial</td>
</tr>
<tr>
<td></td>
<td>0.4 s prediction horizon &amp; RBF/polynomial</td>
<td>0.4 s prediction horizon &amp; RBF/polynomial</td>
</tr>
<tr>
<td>1 Lateral position</td>
<td>100/100</td>
<td>99.54929/0</td>
</tr>
<tr>
<td>2 Lateral position and lateral velocity</td>
<td>99.77465/100</td>
<td>99.43662/1.29577</td>
</tr>
<tr>
<td>3 Lateral position, lateral velocity, and speed</td>
<td>98.98592/98.92958</td>
<td>98.14085/24.67606</td>
</tr>
<tr>
<td>4 Lateral position, steering angle, and speed</td>
<td>98.87324/98.76056</td>
<td>97.97183/2.14085</td>
</tr>
<tr>
<td>5 Lateral position, steering angle, and yaw deviation</td>
<td>95.77465/99.88732</td>
<td>94.76056/31.32394</td>
</tr>
<tr>
<td>6 Lateral position, speed, and yaw deviation</td>
<td>95.83099/99.26761</td>
<td>94.70422/55.32394</td>
</tr>
</tbody>
</table>
Table 4. Testing results of RBF kernel SVM 1 using Testing Set A and lateral position and lateral velocity. The total number of lane departure in the test set was 1775.

<table>
<thead>
<tr>
<th></th>
<th>Prediction Horizon (s)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td><strong>Overall recall (%)</strong></td>
<td>99.77465</td>
<td>99.43662</td>
<td>98.53521</td>
<td></td>
</tr>
<tr>
<td><strong>Average recall /Standard deviation of recall (%)</strong></td>
<td>99.83356/0.00375</td>
<td>99.61109/0.00786</td>
<td>98.94446/0.01346</td>
<td></td>
</tr>
<tr>
<td><strong>Overall Precision (%)</strong></td>
<td>99.88720</td>
<td>92.74829</td>
<td>82.53893</td>
<td></td>
</tr>
<tr>
<td><strong>Average Precision /Standard deviation of Precision (%)</strong></td>
<td>99.94658/0.00165</td>
<td>91.21587/0.10957</td>
<td>82.31761/0.11470</td>
<td></td>
</tr>
</tbody>
</table>
Table 5. Testing results of RBF SVM 1 and RBF SVM 2 after its second-stage training using the 16 drowsy drivers in Testing Set B2 with lateral position and lateral velocity being the input variables. The total number of lane departure in the test set was 1048.

<table>
<thead>
<tr>
<th></th>
<th>Prediction Horizon (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.4</td>
</tr>
<tr>
<td>Overall recall (%) SVM 1/SVM 2</td>
<td>99.33206/99.71374</td>
</tr>
<tr>
<td>Average recall (%) SVM 1/SVM 2</td>
<td>99.44020/99.78897</td>
</tr>
<tr>
<td>Standard deviation of recall (%) SVM 1/SVM 2</td>
<td>0.01084/0.00457</td>
</tr>
<tr>
<td>Overall Precision (%) SVM 1/SVM 2</td>
<td>93.02940/98.30668</td>
</tr>
<tr>
<td>Average Precision (%) SVM 1/SVM 2</td>
<td>92.82298/98.64437</td>
</tr>
<tr>
<td>Standard deviation of Precision (%) SVM 1/SVM 2</td>
<td>0.041669/0.19469</td>
</tr>
</tbody>
</table>
Table 6. Confusion matrix showing the second-stage training results of SVM 2.

<table>
<thead>
<tr>
<th>Prediction Horizon (s) 0.4 / 0.6</th>
<th>Predicted lane deviation</th>
<th>Predicted non-lane deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lane deviation</td>
<td>1045 / 1039</td>
<td>3 / 9</td>
</tr>
<tr>
<td>Non-Lane-deviation</td>
<td>18 / 63</td>
<td>3241460 / 3229180</td>
</tr>
</tbody>
</table>
Table 7. Overall averages and standard deviations of the actual prediction horizons achieved by SVM 1 and SVM 2 presented in Table 5.

<table>
<thead>
<tr>
<th></th>
<th>Overall average of actually achieved prediction horizons (s)</th>
<th>Overall standard deviation of achieved prediction horizons (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>for intended 0.4 s prediction horizon</td>
<td>for intended 0.6 s prediction horizon</td>
</tr>
<tr>
<td>SVM 1</td>
<td>0.398</td>
<td>0.590</td>
</tr>
<tr>
<td>SVM 2</td>
<td>0.396</td>
<td>0.590</td>
</tr>
</tbody>
</table>
(a) Driving maneuver pattern 1 (not a lane deviation event)

(b) Driving maneuver pattern 2 (a lane deviation event)

Fig. 1. Two different driving maneuvers observed in one of our drowsy driving experiments that appear to be quite similar.
Fig. 2. A non-lane deviation in one of our drowsy driving that classified mistakenly as lane deviation by SVM 1 and correctly classified by SVM 2.