Using Support Vector Machines and Bayesian Filtering for Classifying Agent Intentions at Road Intersections

Georges S. Aoude and Jonathan P. How

Technical Report ACL09-02
Aerospace Controls Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology

September 15, 2009

***Abstract*— Classifying other agents' intentions is a very complex task but it can be very essential in assisting (autonomous or human) agents in navigating safely in dynamic and possibly hostile environments. This paper introduces a classification approach based on support vector machines and Bayesian filtering (SVM-BF). It then applies it to a road intersection problem to assist a vehicle in detecting the intention of an approaching suspicious vehicle. The SVM-BF approach achieved very promising results.**

I. INTRODUCTION

Whether driving on highways or navigating in the middle of a battlefield, intelligent vehicles will be required to quickly and robustly compute their motion plans in very uncertain worlds. The sources of uncertainty is typically classified as 1) internal i.e., related to the imperfection in the model of the vehicle, and 2) external i.e., due to incomplete information of the environment the agent lives in. Both types of uncertainty are due to imperfection either in sensing or in predictability [1]. This paper focuses on the problem of external uncertainty in predictability, more specifically on the uncertainty in the intent of the other agents living in our agent's world.

Classifying all other agents as hostile would be overly conservative. A smart agent should be able to gather information from the environment to build models approximating the intentions of the other agents. This information could consist of images captured by onboard cameras, velocity information of surrounding objects obtained from radar measurements, or messages intercepted or shared on some communication channels. Thus, the agent should be able to classify the other agents into different categories, and compute its motion plan around them accordingly.

One motivation of this work is the challenges that were faced by Talos, the MIT autonomous Land Rover LR3 in the 2007 DARPA Grand Challenge (DGC). Talos was one of the six cars that were able to make it to the finish line, out of 35 contestants [2]. The race had a 96 km course located in an urban area, to be completed in less than 6 hours, with the presence of other traffic vehicles and obeying all traffic regulations found on urban roads. One of the main challenges of the DGC race was negotiating the traffic situation at intersections [3]. Talos and several other cars were involved in collisions or near-collisions at intersections. There have been several explanations to this phenomenon, but the one that motivated this work is the inability of the autonomous vehicles to anticipate other vehicles' intents [3]. Figure 1 shows the accident between the vehicles of MIT and Cornell during the 2007 DGC race. It is believed to be the first well documented accident between two full-size autonomous vehicles [4].

Another motivation of this work is related to the high number of car accidents happening each year at road intersections. An estimated 43% of crashes in the United States occur at intersection or are intersection-related [5]. Most of them happen at intersections with stop signs or traffic signals. A main cause of these accidents is the inability of the drivers to accurately perceive the degree of the danger in such situations. These facts suggest that a threat assessment system onboard cars could be very effective in reducing the number of intersection accidents. Such a system would warn or advise the driver, partially control the vehicle, or fully take control of the vehicle depending on the level of the danger it assesses. An integral part of a threat assessment module is a classification algorithm that assigns a category or type for each car driver involved in the intersection. Despite the fact that drivers are not usually hostile, the ability to classify them as "dangerous" when they are not in full control of their cars (*e.g.,* being under the influence of alcohol) is essential in a better performance at predicting accidents. The threat assessment module would then act assuming a worse-case scenario against these dangerous drivers to guarantee the safety of its driver.

Inferring driver intentions has been the subject of several recent studies. Ref. [6] introduced a mind-tracking approach

G. S. Aoude, Ph. D. Candidate, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA 02139, USA, `gaoude@mit.edu`

J. P. How, Professor, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA 02139, USA, `jhow@mit.edu`

Fig. 1. Picture of the collision between MIT's autonomous Land Rover LR3 'Talos' and Team Cornell's autonomous Chevrolet Tahoe 'Skynet' in the Darpa Grand Challenge 2007. It is known as the first well-documented collision between two full-size autonomous vehicles [4].

that extracts the similarity of the driver data to several virtual drivers created probabilistically using a cognitive model. Other work focused on predicting drivers' intention to changing lanes in common driving environments. Ref. [7] used graphical models and HMMs to create and train models of different driver maneuvers using experimental driving data. Ref. [8] introduced a technique to infer driver intentions to changing lanes based on support vector machines (SVMs), a supervised learning technique that is also used in this paper . It was applied on behavioral and environmental data collected from an instrumented vehicle. All of the above approaches tries to model the behavior or intentions of the host driver to support it in performing several driving tasks.

In this paper we describe a method to classify the other agents' intention using a combination of support vector machines and a Bayesian filtering (SVM-BF) method. The goal is to help our agent make better informed decisions. First, we give an overview of SVMs and a description of the chosen sequential Bayesian inference method. Then we introduce the road intersection problem, and explain how we approach it using the SVM-BF method. Finally, simulation results are presented and analyzed.

II. Classification

The world our agent interacts with consists of "dangerous" and "harmless" agents. An agent refers to autonomous systems (*e.g.,* robots) or humans. While the dangerous agent's goal is to cause damage to our vehicle or put it in a risky situation, the harmless agent follows a trajectory that minimizes some "peaceful" goal (*e.g.,* time or distance). Next is a brief overview of support vector machines and Bayesian filtering, which will be used in the classification approach.

A. Support Vector Machines

Support vector machines (SVMs) is a relatively new supervised machine learning technique based on the margin-maximization principle [9]. It was originally introduced by Vapnik and Cortes [10], [11], and has been since succesfully applied to several applications including text categorization, bioinformatics and database marketing [12]. The following briefly introduces SVMs, but [10], [13], [14] provide more detailed descriptions.

TABLE I. Typical Kernel Functions [14]

Polynomial	$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = (\langle \boldsymbol{x}_i,\boldsymbol{x}_j\rangle) + 1)^d$
Gaussian Radial Basis	$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}\right)$
Multi-Layer Perceptron	$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \tanh(\rho\langle \boldsymbol{x}_i,\boldsymbol{x}_j\rangle + \varrho)$

Given a set of binary label training data $\{\boldsymbol{x}_i, y_i\}$, where $i = 1,\ldots,N, y_i \in \{-1,1\}, \boldsymbol{x}_i \in \Re^d$, N is the number of training vectors, and d is the size of the input vector, a separating hyperplane between the two classes of data can be written as

$$\boldsymbol{w}\cdot\boldsymbol{x} + b = 0 \tag{1}$$

where b is known as the *bias*, and $\boldsymbol{w}$ as the *weights*.

The training data are typically not linearly separable. So they are mapped into a higher dimensional Hilbert space called *feature space* such that

$$\boldsymbol{x}_i\cdot\boldsymbol{x}_j \rightarrow \phi(\boldsymbol{x}_i)\cdot\phi(\boldsymbol{x}_j) \tag{2}$$

The function $\phi(\boldsymbol{x}_i)\cdot\phi(\boldsymbol{x}_j) = K(\boldsymbol{x}_i,\boldsymbol{x}_j)$ is called the kernel function. It is the inner product between the mapped pairs of points in the feature space. Note that the functional form of the mapping $\phi(\boldsymbol{x})$ needs not to be known since it is implicitly defined in the kernel function. Data which are non-separable in the *input space* can become separable in the *feature space* with the right choice of kernel function. Some typical kernel functions are shown in Table II-A.

To classify a new test vector $\boldsymbol{z}$ into one class ($y = 1$) or the other ($y = -1$), the following decision function is evaluated

$$D(\boldsymbol{z}) = \operatorname{sgn}\left[\sum_{i=1}^{N}\alpha_i y_i K(\boldsymbol{x}_i,\boldsymbol{z}) + b\right] \tag{3}$$

where $\boldsymbol{\alpha}$ is the *argmax* of the following optimization problem

$$\max W(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{4}$$

subject to the constraints

$$\sum_{i=1}^{N}\alpha_i y_i = 0 \tag{5}$$

$$\alpha_i \geq 0 \tag{6}$$

To account for the noise found in most real life datasets, a soft margin support vector machine is usually created. It adds robustness to the classification by reducing the effect of outliers and noise. Typically slack variables are added into the constraints to relax the hard margin constraints.

B. Motivation for using SVMs

Classifying human drivers is a very complex task because of the various nuances and peculiarities of human behaviors [8]. Researchers have shown that the state of a vehicle driver lies in some high dimensional feature space [15]. Classifying autonomous vehicle behaviors could be even more complex mainly because of the difficulty of gathering training data representing the different class of systems that our agent might encounter. SVMs were shown to be a robust and efficient approach for binary classification, and scale well with high dimensional feature spaces [11]. In this paper, the world our agent interacts with consists of "harmless" and "dangerous" agents. An agent refers to autonomous systems (*e.g.,* robots) or human beings. While the dangerous agents' goal is to cause damage to our vehicle or put it in a risky situation, the harmless agents follow trajectories that minimize some "peaceful" goal (*e.g.,* time or distance). Thus SVMs are used to make the binary classification of harmless vs. dangerous agents. In general, SVMs have several theoretical and practical advantages. We highlight few of them: 1) training SVMs involves an optimization problem of a convex function, thus the optimal solution is a global one (*i.e.,* no local optima), 2) the upper bound on the generalization error does not depend on the dimensionality of the space, and 3) there are fewer free parameters to tune in SVMs compared to other methods (*e.g.,* neural networks).

C. Bayesian Filtering

Sequential Bayesian inference deals with the Bayesian estimation of a time-varying dynamic system [16]. Let θ_k denote the state of interest at time k. Then the sequential Bayesian inference estimates the a posteriori probability density function $p(\theta_k|y_{1:N})$ by fusing a sequence of sensor measurements $y_{1:N}$ together.This type of inference makes use of observations either one at a time, or in small groups, and then discards them before the following measurements are considered. They can be used in real-time applications because the whole data set need not to be stored or loaded in memory, so they can be very useful for large amount of data.

In this paper we consider the calculation of the probability density function (pdf) $p(\theta_k|y_{1:N})$ for $k = N$ which is known as (sequential) Bayesian filtering [9]. Renaming θ_N as θ, the pdf is written as $p(\theta|y_{1:N})$. θ is an unknown parameter representing the probability that an agent is "harmless". We choose the prior of θ to be a *beta* distribution, which is a function of some hyperparameters a and b [9].

$$beta(\theta|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)+\Gamma(b)}\theta^{a-1}(1-\theta)^{b-1} \tag{7}$$

where $\Gamma(x)$ is the gamma function [9]. The posterior distribution of θ, $p(\theta|y_{1:N})$, is computed by multiplying the beta prior (See (7)) by the binomial likelihood function given by

$$bin(m|N,\theta) = \binom{N}{m}\theta^{m}(1-\theta)^{N-m} \tag{8}$$

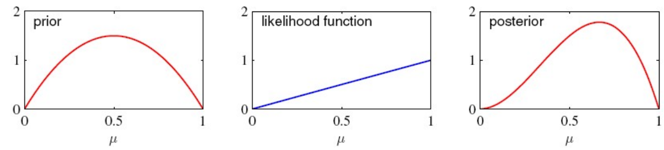


Fig. 2. Illustration of one step of sequential Bayesian inference. The prior is given by a beta distribution with parameters $a = 2$ and $b = 2$. The likelihood function is characterized by $N = m = 1$, which corresponds to a single observation. The posterior is thus given by a beta distribution with parameters $a = 3$ and $b = 2$ [9].

and then normalizing the resulting function to obtain

$$p(\theta|y_{1:N}) = \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)+\Gamma(l+b)}\theta^{m+a-1}(1-\theta)^{l+b-1} \tag{9}$$

Note that m is the number of observations saying that the agent is harmless, and the number of observations saying that the agent is dangerous is $l = N - m$. Note also that choice of the *beta* distribution provides a simple way to represent the effective number of observations for each class of agents through its hyperparameters a and b. In other words, a and b can be interpreted as the initial "confidence" weights for each class, respectively.

The goal is to compute the expected value of the θ parameter which is equivalent to predicting the outcome of the next output of the trial. Therefore we must evaluate the posterior distribution of y_{N+1} given the observed data set $y_{1:N}$

$$\begin{aligned} & p(y_{N+1} = \text{harmless}|y_{1:N}) && (10)\\ & = \int_0^1 p(y_{N+1} = \text{harmless}|\theta)p(\theta|y_{1:N})d\theta && (11)\\ & = \int_0^1 \theta p(\theta|y_{1:N})d\theta = E(\theta|y_{1:N}) && (12)\\ & = \frac{m+a}{m+a+l+b} && (13)\end{aligned}$$

Figure 2 illustrates one step of sequential Bayesian inference, with $a = b = 2$, and $N = m = 1$.

III. Application of SVM-BF on the Road Intersection Problem

A. The Road Intersection Problem

The road intersection problem serves as an illustration of an application where the SVM-BF method could be useful. The problem consists of the following scenario (See Figure 3): our vehicle (denoted as host vehicle) arrives at a four way intersection with four way stop signs. The host vehicle detects another vehicle (denoted as suspicious vehicle) that is almost 190 meters away and coming from its right side. The host vehicle could proceed, but it decides to be cautious. So it assesses the behavior of the suspicious vehicle before moving by using the SVM-BF approach. The host vehicle would typically decide within some time limit to proceed, wait, or possibly follow an escape maneuver if the suspicious vehicle is classified as a dangerous one. The

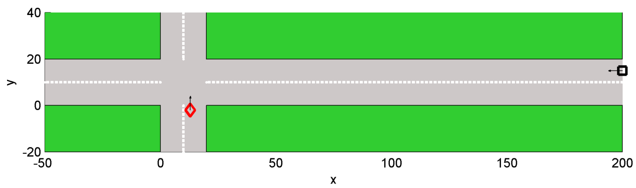


Fig. 3. The Road Intersection Scenario. The red diamond is the host vehicle. The black square is the suspicious vehicle. The arrows show the heading of the vehicles.

Fig. 4. The SVM-BF Architecture.

suspicious vehicle is modeled as a simple car dynamical system (*i.e.,* simple unicycle) with velocity and angular rate as control inputs [17].

B. Implementation of SVM-BF

The SVM-BF architecture is shown in Figure 4. At the beginning of each time period ΔT, the SVM module receives some measurements from the vehicle sensors, it extracts the relevant features (see Section III-C) and outputs a classification (dangerous or harmless). This classification is fed into a Bayesian filter (see Section III-D) which computes the probability that the next SVM output is harmless, which is equivalent to computing the expected value of the θ parameter introduced in Section II-A. Recall θ is the probability of the suspicious vehicle being a harmless agent. This value is then sent into a threshold detector module that outputs the final classification of the SVM-BF system. The threshold used in this problem is 0.8 *i.e.,* SVM-BF declares a suspicious vehicle as dangerous if the expected value of θ is smaller than 0.8. SVM-BF outputs these classifications continuously for each ΔT, but we are mainly interested in the value of the classification when the suspicious vehicle is close to the host vehicle, more specifically when their relative distance is smaller than 20 meters. Note that in the implementation, $\Delta T = 1s$. In real-world applications, ΔT could be much smaller, but the operation of the SVM-BF system would be the same.

C. SVM Parameters

1) Kernel Selection: The choice of a suitable Kernel function in the decision function (3) is essential in obtaining satisfactory results with SVMs. We tested the training data against several types of kernel functions: a) linear, b) quadratic, c) cubic, d) and gaussian radial basis. The best results were observed with both the cubic polynomial and gaussian radial basis kernel functions. The simulation results shown in Section IV are produced using an SVM classifier that uses a gaussian radial basis kernel function.

2) Feature Selection: Choosing an effective set of features is a very challenging design decision. Features that vary randomly or do not show significant predictability should be avoided as they degrade the performance of SVMs [8]. After experimenting with several combinations of features, the best results were obtained when combining the following three features:

- the relative distance Δx between the the host and suspicious vehicles;
- the heading ϕ of the suspicious vehicle relative to the host vehicle;
- the speed v of the suspicious vehicle.

3) Soft Margin: To deal with the noisy measurements, a soft margin vector support machine was used. The parameter representing the penalty to error was set to 1. Please refer to [14] for more details about soft margins.

4) Training the SVM: To train the SVM, a series of trajectories of actual drivers approaching an intersection should be gathered and classified as harmless or dangerous according to some specific rules. For example, a dangerous driver is one whose trajectory ends up colliding with the host vehicle or gets within some specific distance of it. Other factors can be also included in the classification. Relevant features extracted from the trajectories along with their classifications will then be used in the training of the SVM classifier. But for our simulation purposes, we generated 270 combinations of features that cover uniformly different regions of interest, and classified them as dangerous or harmless following some specific rules. To make the classifier robust to noise, 5 of the 270 combinations are intentionally misclassified *i.e.,* do not follow the rules of classification. The goal was to make use of SVM's robustness to faulty data that is typically found in real-world training.

The main rules followed in the training process are the following:

- If Δx is large (*i.e.,* suspicious vehicle at least 50 meters far away from host vehicle) and $v \le 1.25v_{max}$ (*i.e.,* not very fast) then classify as harmless. Otherwise (*i.e.,* same conditions but $v > 1.25v_{max}$) classify as dangerous.
- If Δx is not large (*i.e.,* suspicious vehicle in the vicinity of host vehicle), ϕ is small (*i.e.,* suspicious vehicle pointing towards host vehicle within an angle of 10 degrees) and $v \ge v_{min}$ (*i.e.,* not very slow) then classify as dangerous. Otherwise (*i.e.,* same conditions but $v < v_{min}$) classify as harmless.
- If Δx is not large (*i.e.,* suspicious vehicle in the vicinity of host vehicle), ϕ is not small (*i.e.,* suspicious vehicle not pointing towards host car) but $v > 0.8v_{max}$ then classify as dangerous. Otherwise (*i.e.,* same conditions but $v \le 0.8v_{max}$) classify as harmless.

Note that $v_{max} = 70km/h$ and $v_{min} = 5km/h$.

D. BF Parameters

There are four parameters to choose in Equation (13) of the Bayesian filter (BF): m, N, a and b. In our implementation,

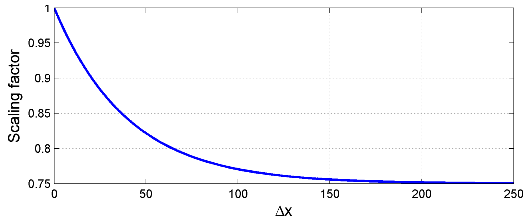


Fig. 5. Plot of a discount function for the BF update.

the BF receives one classification from the SVM at a time, so $N = 1$. Then $m = 1$ if the current observation $y = harmless$ otherwise $l = 1$. a and b represent the initial number of observations (or classifications) of $y = harmless$ and $y = dangerous$ respectively. We set them to $a = b = 1$ at $t = 0$ in our implementation, meaning that we assume a uniform distribution for θ, the parameter representing the probability that the suspicious vehicle is harmless. Then a and b are updated at $t = t + \Delta t$ to $a = a + m$ and $b = b + l$, and so on. Recall $l = N - m$. So the posterior distribution over θ acts as the prior every time the BF receives a new input from the SVM.

E. Discounted BF

To speedup the convergence of the Bayesian filter, a scaling factor can be added to the updating steps of a and b as follows: $a = \gamma(\Delta x) \times a + m$ and $b = \gamma(\Delta x) \times b + l$. The discount factor γ gives more weight to newer SVM outputs by discounting the old posterior of θ (*i.e.,* the new prior). Newer SVM outputs correspond to smaller Δx meaning the suspicious vehicle getting closer to the intersection, which is the desired behavior. For the Bayesian filter to converge, the discount factor should be chosen such that it converges to 1 as the number of iterations goes to infinity [18]. One example of discount function that has been designed for the intersection problem is

$$\gamma(\Delta x) = A + B \exp(C \Delta x) \tag{14}$$

where $A = 0.75$, $B = 0.25$ and $C = -0.025$ (See (Figure 5)).

IV. Simulation Results

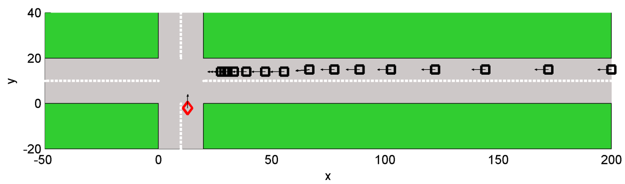
We simulated 60 suspicious vehicles with different velocity and angular rate profiles. 30 of them were dangerous vehicles, *i.e.,* they were designed to either end up colliding with the host vehicle, or/and cross the intersection without stopping at the stop sign. The undiscounted BF parameters (see Section III-D) were used in the simulations shown in this section. The SVM-BF approach detected 39 dangerous drivers (including all the actual 30 dangerous drivers) and 21 harmless drivers. In other words, it resulted in 30 true positives, 9 false positives and 0 false negatives. We will show the cases of 3 suspicious vehicles that were detected correctly by SVM-BF. One of them is a harmless vehicle, and the other two were dangerous vehicles.

Note that in the graphs below, a dangerous classification has value 0 and a harmless one has value 1. The time interval ends when the suspicious vehicle stops at the intersection, crosses the intersection, or hits the host vehicle. Also the snapshots of the suspicious vehicle are taken at $1s$ intervals.

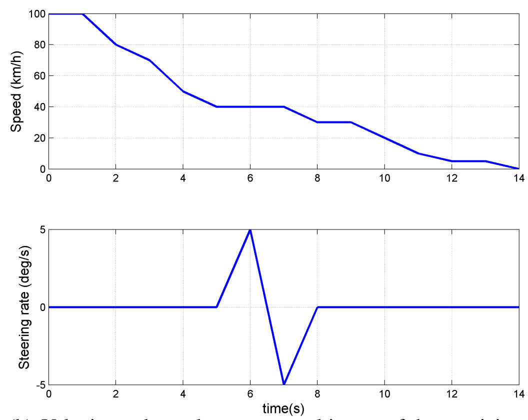
1) Harmless Example 1*:* The first "harmless" example (Figure 6) illustrates the scenario of a harmless driver that is driving at a relatively high speed when far from the intersection. Getting closer to the intersection, the driver changes lanes and starts decelerating until he/she arrives smoothly at the intersection. It is a typical scenario for a driver planing to take a left turn at the intersection. Figure 6(a) and Figure 6(b) show the path of the suspicious vehicle and its control inputs, respectively, as a function of time. Figure 6(c) illustrates the functioning of the SVM-BF approach. $p_{harmless}$ starts at 0.5. It then decreases even though the vehicle is relatively far from the intersection because it is moving at a speed that is larger than $1.25 v_{max}$. But afterwards, the SVM-BF brings $p_{harmless}$ up since the speed of the suspicious vehicle decreases significantly entering the safe region of speeds. Notice also that between $t = 6s$ and $t = 8s$, the driver makes a change of lanes. $p_{harmless}$ ends above the threshold level. So the final classification of the SVM-BF is that the suspicious vehicle is harmless, which is the correct one. This example also stresses that pointing towards the host vehicle did not affect $p_{harmless}$ since it happened relatively far from the intersection *i.e.,* entailing no threat to the host vehicle.

2) Dangerous Example 1*:* The first "dangerous" example (Figure 7) illustrates the scenario where a dangerous driver approaches the intersection with a very high speed in such a way that he/she could not stop at the intersection even after applying a very hard break. This is a typical situation of a driver that missed the stop sign or is not in full control of its car (*e.g.,* being under the influence of alcohol or due to a malfunctioning in the car). Figure 7(a) and Figure 7(b) show the path of the suspicious vehicle and its control inputs, respectively, as a function of time. Figure 7(c) illustrates the functioning of the SVM-BF approach. The $p_{harmless}$ starts at 0.5. Then it continuously decreases since the driver velocity is constantly higher than accepted values that the SVM-BF were trained for. The simulation stops when the vehicle crosses the intersection. So the SVM-BF classified correctly the suspicious agent as dangerous. The $p_{harmless}$ showed that SVM-BF output was gaining more confidence of its classification the more observations it was receiving.

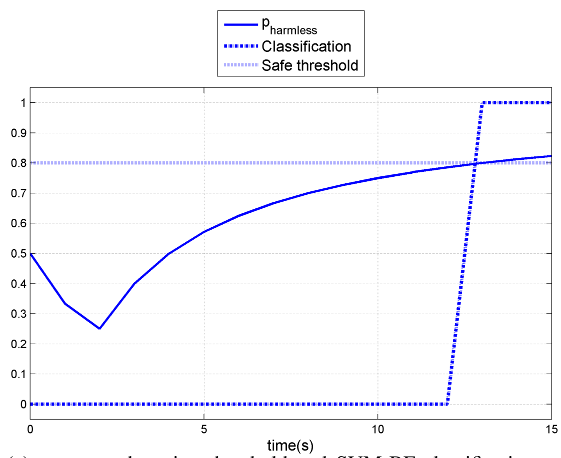
3) Dangerous Example 2*:* The second "dangerous" example (Figure 8) illustrates the scenario of a collision between the suspicious vehicle and the host vehicle. This scenario could be a situation where a driver decides to make a left turn at the intersection but loses control over the vehicle (*e.g.,* due to a slippery road) when it gets to the intersection. Figure 8(a) and Figure 8(b) show the path of the suspicious vehicle and its control inputs, respectively, as a function of time. Figure 8(c) illustrates the functioning of the SVM-BF approach. The $p_{harmless}$ starts at 0.5. First it decreases for a short time since the driver was going faster than $1.25 v_{max}$, but then it increases after the driver slowed down significantly. However, at $t = 5s$, the driver abruptly changes its heading to point towards the host vehicle. This resulted



(a) The intersection scenario showing the suspicious vehicle trajectory.



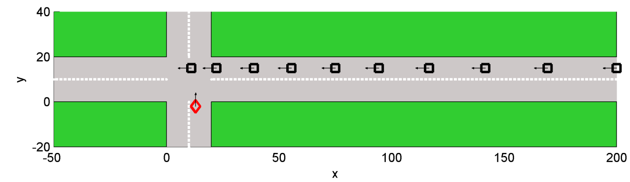
(b) Velocity and angular rate control inputs of the suspicious vehicle.



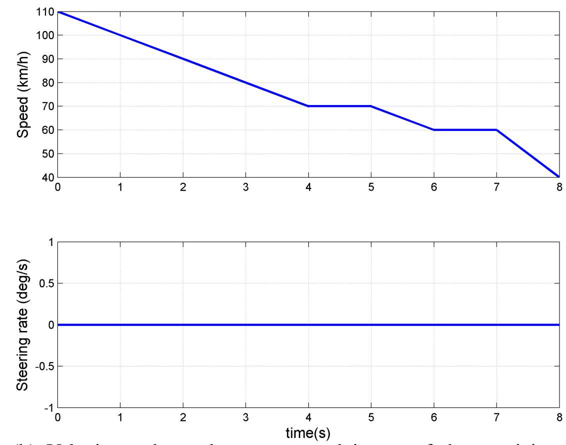
(c) $p_{harmless}$, detection threshold and SVM-BF classification as a function of time.

Fig. 6. Harmless Example

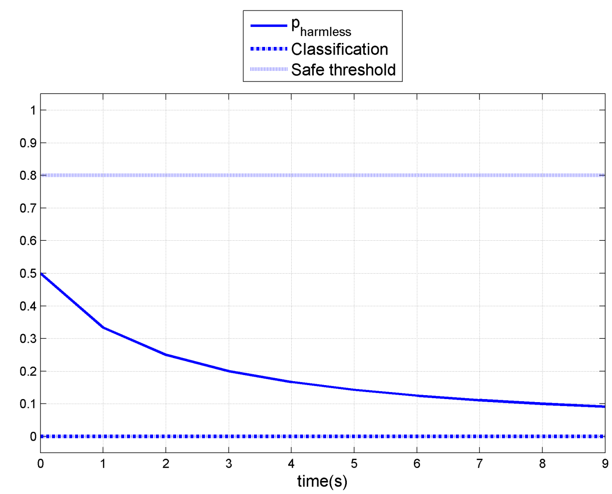
in $p_{harmless}$ decreasing constantly reflecting the danger that this behavior was inducing. An accident finally happens, which also meant that SVM-BF successfully classified the suspicious driver's intention.



(a) The intersection scenario showing the suspicious vehicle trajectory.



(b) Velocity and angular rate control inputs of the suspicious vehicle.



(c) $p_{harmless}$, detection threshold and SVM-BF classification as a function of time.

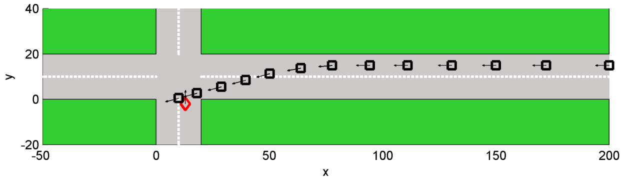
Fig. 7. Dangerous Example 1

V. Discussion

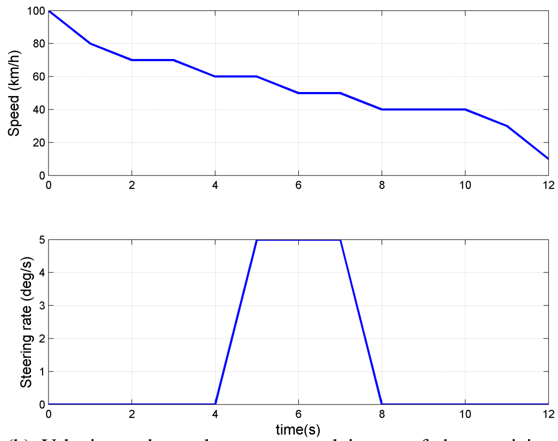
We evaluate the performance of the SVM-BF system based on the following metrics: 1) precision and 2) coverage [19]:

$$\begin{aligned} \text{Precision} &= \frac{\text{valid classifications of dangerous drivers}}{\text{total number of dangerous drivers detected}} \\ &= \frac{\text{true positives}}{(\text{true positives + false positives})} \\ &= \frac{30}{(30+9)} \approx 77\% \end{aligned}$$

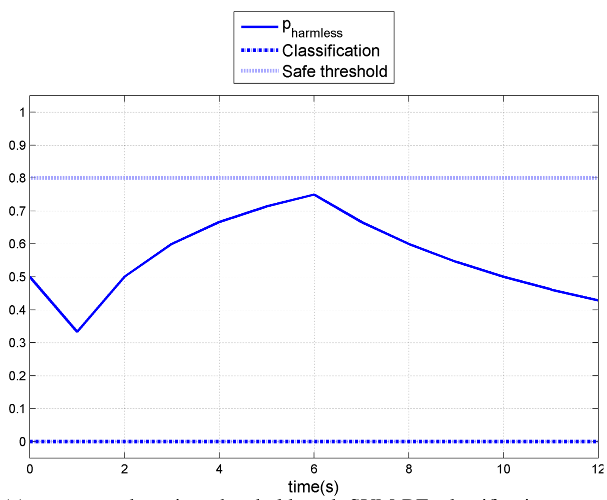
$$\begin{aligned} \text{Coverage} &= \frac{\text{valid classifications of dangerous drivers}}{\text{total number of actual dangerous drivers}} \\ &= \frac{\text{true positives}}{(\text{ true positives + false negatives})} \\ &= \frac{30}{(30+0)} = 100\% \end{aligned}$$



(a) The intersection scenario showing the suspicious vehicle trajectory.



(b) Velocity and angular rate control inputs of the suspicious vehicle.



(c) $p_{harmless}$, detection threshold and SVM-BF classification as a function of time.

Fig. 8. Dangerous Example 2

We achieve 100% coverage which is critical for the safety of the host vehicle. The 77% precision shows that the SVM-BF approach is too conservative. Some presumably harmless drivers are tagged as dangerous. This level of conservatism depends on three factors: 1) the threshold level of the threshold detector module, 2) the SVM training and validation, and 3) the speed of convergence of the Bayesian filter:

1) The threshold level is set to 0.8. Decreasing it would make the system less conservative, but would incur the risk of causing false negative detections. The SVM training follows some set of rules that could be made more detailed to capture a larger and more accurate set of drivers' profiles.
2) The validation also follows simple rules: the suspicious vehicle is flagged as dangerous if it ends up colliding with the host vehicle or/and does not stop at the intersection. These two rules partially explain the existence of a non-negligible number of false positives. For example, a suspicious vehicle that approaches the intersection with a very high speed and then decelerates very sharply succeeding at stopping at the intersection will typically be classified as dangerous by the SVM-BF system. However, it will be flagged as harmless according to the validation rules. Adding more details to the validation rules should make it more reflective of the actual danger of the suspicious vehicles.
3) Another explanation for the conservatism of the SVM-BF is the slow speed of convergence of the BF filter. To improve the response of the filter, more weight can be put on more recent outputs of the SVM as described in Section III-E. To verify the potential of this modification, the simulations were run again with several discount factor functions. The one that achieved the best overall performance is shown in Equation (14). The results of the SVM-BF simulations changed significantly: the number of false positives decreased from 9 to 3, but the number of false negatives increased from 0 to 2. In other words, the discounted SVM-BF approximately achieved a precision of 90% and a coverage of 93%. So the conservatism level of SVM-BF was successfully decreased at the expense of the introduction of some risk of failing to detect a dangerous driver. Figure 9 shows the improvement in the convergence of the discounted SVM-BF in the example introduced in Section IV-.1.

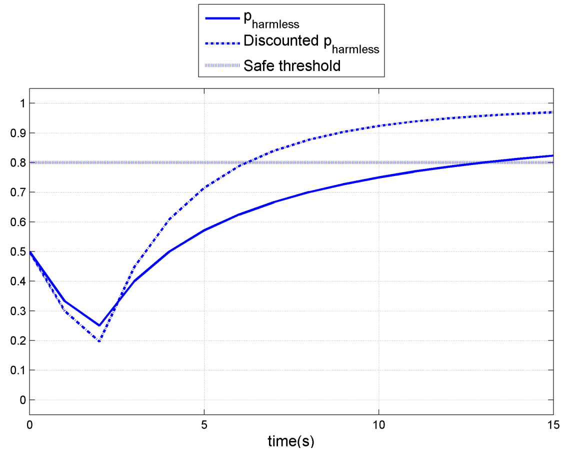


Fig. 9. $p_{harmless}$, discounted $p_{harmless}$ and detection threshold as a function of time. The discounted $p_{harmless}$ reaches the threshold more than twice faster than $p_{harmless}$ for the example described in Section IV-.1.

VI. Conclusion and Future Work

This paper introduced an approach to classify agents' intentions using a combination of support vector machines (SVM) and bayesian filtering (BF). The SVM-BF approach was applied to the road intersection problem that tries to classify the intention of a suspicious vehicle approaching a host car at a four-way intersection. SVM-BF was tested in 60 different scenarios, where in half of them the suspicious vehicle was designed to be a dangerous one. SVM-BF achieved a coverage of 100% and a precision of 77%. Three examples (one with a harmless vehicle and two with dangerous vehicles) were presented to illustrate the SVM-BF approach. A modified version of the SVM-BF that is based on a discounted BF improved the precision to 90% at the expense of decreasing the coverage to 93%. In addition to the threshold level, the discount function in the discounted SVM-BF gives the designer another degree of freedom to optimize the tradeoff between convenience (represented by the precision) and safety (represented by the coverage).

Future work will include comparing the performance of SVM-BF to an approach that consists of feeding a classifier with a history of features instead of point-based set of features. Relevance Vector Machines (RVMs) could handle such a representation. But they are known to have additional computational complexity. We are also interested in training and validating the SVM-BF on real-world traffic datasets, and designing discounting functions that work well on these datasets. Finally, we are looking to assess the value of additional features in the SVM, more specifically investigating the addition of the longitudinal and lateral accelerations of the suspicious vehicle.

ACKNOWLEDGEMENTS

The first author acknowledges Dr. Tom Pilluti, Dr. Luca Bertuccelli and Dr. Han-Lim Choi for their insightful comments. Research funded in part by Ford Motor Company and Le Fonds Québécois de la Recherche sur la Nature et les Technologies (FQRNT) Graduate Award.

REFERENCES

[1] S. M. Lavalle and R. Sharma, "On motion planning in changing, partially-predictable environments," *International Journal of Robotics Research*, vol. 16, pp. 775–805, 1997.

[2] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, vol. 25, no. 10, pp. 727 – 774, October 2008.

[3] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, J. Leonard, I. Miller, M. Campbell, D. Huttenlocher, A. Nathan, and F.-R. Kline, "The mit - cornell collision and why it happened," *Journal of Field Robotics*, vol. 25, no. 10, pp. 775 – 807, October 2008.

[4] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, J. Leonard, I. Miller, M. Campbell, D. Huttenlocher, A. Nathan, and F. R. Kline, "The MIT - Cornell Collision and Why it Happened," *Journal of Field Robotics*, vol. 25, no. 10, pp. 775 – 807, October 2008.

[5] T. O'Connor, "Intelligent Transportation Systems," April 2004. [Online]. Available: http://www.calccit.org/itsdecision/

[6] D. D. Salvucci, "Inferring driver intent: A case study in lane-change detection," in *in Proceedings of the Human Factors Ergonomics Society 48th Annual Meeting*, 2004, pp. 2228–2231.

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

[8] H. M. Mandalia and D. D. Dalvucci, "Using Support Vector Machines for Lane-Change Detection," *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 49, pp. 1965–1969, 2005.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[10] C. Cortes and V. Vapnik, *Support Vector Networks*, 1995.

[11] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[12] "SVM Application List," 2006. [Online]. Available: http://www.clopinet.com/isabelle/Projects/SVM/applist.html

[13] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997, pp. 130–136.

[14] S. Gunn, "Support vector machines for classification and regression," Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, Tech. Rep., May 1998. [Online]. Available: http://citeseer.ist.psu.edu/7854.html

[15] D. Wipf and B. Rao, "Driver Intent Inference Annual Report," University of California, San Diego, Tech. Rep., 2003.

[16] H. B. Mitchell, *Multi-Sensor Data Fusion*. Springer Berlin Heidelberg, July 2007.

[17] S. M. Lavalle, *Planning Algorithms*. Cambridge University Press, 2006.

[18] L. F. Bertuccelli, "Robust Decision-Making with Model Uncertainty in Aerospace Systems," Ph.D. dissertation, Masschusetts Institute of Technology, September 2008.

[19] F. Salim, S. W. Loke, A. Rakotonirainy, B. Srinivasan, and S. Krishnaswamy, "Collision Pattern Modeling and Real-Time Collision Detection at Road Intersections," *IEEE Intelligent Transportation Systems Conference*, pp. 161–166, Sept-Oct 2007.