Data-Driven Probabilistic Modeling and Verification of Human Driver Behavior

D. Sadigh, K. Driggs-Campbell, A. Puggelli,
W. Li, V. Shia, R. Bajcsy, A. L. Sangiovanni-Vincentelli, S. S. Sastry, S. A. Seshia
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley. Berkeley, CA.
email:{dsadigh, krdc, puggelli, wenchaol, vshia, bajcsy, alberto, sastry, sseshia}@eecs.berkeley.edu

Abstract

We address the problem of formally verifying *quantitative* properties of driver models. We first propose a novel stochastic model of the driver behavior based on Convex Markov Chains, i.e., Markov chains in which the transition probabilities are only known to lie in convex uncertainty sets. This formalism captures the intrinsic uncertainty in estimating transition probabilities starting from experimentally-collected data. We then formally verify properties of the model expressed in probabilistic computation tree logic (PCTL). Results show that our approach can correctly predict quantitative information about driver behavior depending on her state, e.g., whether he or she is attentive or distracted.

1 Introduction

The problem of modeling driver behavior in cars has long been studied, due to its relevance to applications ranging from teaching techniques for safer driving and developing more effective driving regulations and norms to, more recently, the development of autonomous and semi-autonomous control techniques to reduce the number of car fatalities (Cacciabue 2007). In particular, one of the major focuses of the research community has been on addressing the problem of driver distraction (Vasudevan and others 2012; Anderson and others 2010; Coelingh and others 2007). This includes distraction caused by cell phone calls, which accounts for $22-50\%$ of all accidents (Klauer 2006). Potential solutions to the driver distraction problem rely on semi-autonomous or "human-in-the-loop" control techniques, some of which try to predict the car trajectory based on estimations of the driver behavior and actively take control of the car. For instance, existing techniques perform a braking maneuvers if a collision is predicted (Coelingh and others 2007). To correctly take human actions into consideration before intervention, modeling the driver behavior is thus of crucial importance.

Since it would be hard to create a single and deterministic model of the driver behavior, due to the unpredictability of human behavior under varying environments and levels of distraction, stochastic techniques using Markov Chain (MC) models have been proposed (Lin and others 2004). As a first contribution, in this paper we develop a novel *probabilistic driver model* that predicts the driver trajectories using a Convex-MC (CMC) model, i.e., a Markov Chain in which the transition probabilities are only known to lie in convex uncertainty sets. The prediction is based on the future environment surrounding the car, the state of the driver (i.e. attentive or distracted), and the history of steering maneuvers for a given individual, which we collected using a car simulator [1]. For each environment and state of the driver, the model predicts a set of trajectories for the subsequent time interval based on empirical observations of past behavior. These predictions are then used as transition probabilities within the driver model. Due to the ambiguity that inherently affects the observed driving data used to estimate the transition probabilities, we allow probabilities to be expressed in terms of uncertainty sets, which can be precisely defined based on statistical techniques. This framework allows a more conservative prediction of the driver behavior and gives guidelines to the model developer to determine when the collected data are statistically relevant to correctly infer properties of the system.

Due to the criticality of the system under consideration, formal techniques to verify properties of the constructed model are required to rigorously assess the validity of the model and give guarantees of its safety and liveness. As a second contribution of this paper, we show how to verify logical properties of the CMC model of the driver expressed in Probabilistic Computation Tree Logic (PCTL), using recently-proposed polynomial-time verification algorithms (Puggelli and others 2013). Our main focus is to quantify the effects of different attention levels on the quality of driving by formally analyzing the driver behaviors while they are either attentive or distracted. PCTL is a suitable choice because it allows to express *quantitative* properties of a system, as opposed to other logics, e.g. Linear Temporal Logic (LTL), which only allow *qualitative* properties. For example, we aim to determine whether "the maximum probability of exiting the road for a distracted driver is higher than 90%", while LTL would only allow to inquire whether "eventually a distracted driver will exit the road", a property that would trivially be always true for some of the executions of the model, without giving insight about how

[1] http://www.carsim.com

likely the event would actually take place.

The structure of the paper is as follows. In Section 2, we summarize the main theoretical results about the verification of PCTL properties of CMCs. In Section 3, we give details on how to construct the proposed stochastic model of the driver behavior, and in Section 4 we present results on the verification of its properties. We conclude and discuss future research directions in Section 5.

2 Background

In the area of formal methods, the "model checking" problem can be formalized as a decision problem. Given a structure M and property ϕ, the "model checking" problem answers whether $\mathcal{M}$ satisfies property ϕ starting from state s, written as $\mathcal{M}, s \models \phi$. This formal verification technique is a rigorous mathematical approach to prove the correctness of a system (Clarke and others 1999).

The first model checking techniques targeted deterministic systems, e.g. finite state machines, in which the same input stream always produces the same output stream for a fixed initial state. On the other hand, in many applications, noticeably those involving hybrid systems in which continuous and discrete dynamics coexist, the system model $\mathcal{M}$ is probabilistic (Kwiatkowska and others 2011). Since classic model checking approaches cannot be used to prove the correctness of these systems, probabilistic models and model checking techniques were developed. In this project, we chose $\mathcal{M}$ to be a discrete time CMC, formally defined as follows.

Definition. *A labeled finite CMC, $\mathcal{M}_\mathcal{C}$ is a tuple: $\mathcal{M}_\mathcal{C} = (S, S_0, \Omega, \mathcal{F}, X, L)$, where S is a finite set of states of cardinality $N = |S|$; S_0 is the set of initial states; Ω is a finite set of atomic propositions; $\mathcal{F}$ is a finite set of convex sets of transition probability distributions; $S : S \to \mathcal{F}$ is a function that associates a state, s to probability distributions, denoted $\mathcal{F}_s \in \mathcal{F}$; $L : S \to 2^\Omega$ is a labeling function that associates state, s, to the corresponding atomic propositions.*

Intuitively, given any state $s \in S$, we do not constrain the Discrete Transition Probability Distribution (DTPD) to the next states to be unique, but we allow to specify a convex set $\mathcal{F}_s \in \mathcal{F}$ of DTPDs, to model the uncertainty in estimating the transition probabilities when they are derived from statistical data. Each point in the set represents an *observed* DTPD $\mathbf{f}_s \in \mathcal{F}_s$. We interpret the N-dimensional point $\mathbf{f}_s$ as a vector, with $\mathbf{f}_s \in \mathcal{F}_s \subseteq \mathbb{R}^N$. We then collect the vectors $\mathbf{f}_s, \forall s \in S$ into an *observed transition matrix* $F \in \mathbb{R}^{N \times N}$, where $f_{ss'}$ is the observed probability of transitioning from state s to state s'. A transition between state s to state s' in a CMC occurs in two steps. First, an observed DTPD $\mathbf{f}_s \in \mathcal{F}_s$ is chosen. Second, a successor state s' is chosen randomly, according to the DTPD $\mathbf{f}_s$.

A path π in $\mathcal{M}_\mathcal{C}$ is a finite or infinite sequence of the form $s_0 \xrightarrow{f_{s_0 s_1}} s_1 \xrightarrow{f_{s_1 s_2}}, \cdots$, where $s_i \in S$ and $f_{s_i, s_{i+1}} > 0 \ \forall i \geq 0$. $\pi[i]$ is the i^{th} state along the path and $\Pi_s = \{\pi \mid \pi[0] = s\}$ is the set of paths starting in state s.

The shape of the convex uncertainty set is based on the selected uncertainty model. In our project, we used both the *interval* and the *likelihood* models of uncertainty (Nilim and others 2005). Intervals commonly describe uncertainty in transition matrices:

$$\mathcal{F}_s = \{\mathbf{f}_s \in \mathbb{R}^N \mid \mathbf{0} \leq \underline{\mathbf{f}}_s \leq \mathbf{f}_s \leq \overline{\mathbf{f}}_s \leq \mathbf{1}, \mathbf{1}^T \mathbf{f}_s = 1\}$$

where $\underline{\mathbf{f}}_s, \overline{\mathbf{f}}_s \in \mathbb{R}^N$ are the element-wise lower and upper bounds of $\mathbf{f}_s$. While intuitive to understand, intervals often result in over-conservative predictions of the effects of uncertainty. Thus, we will use the interval model as a baseline for our experiments, and further use the *likelihood* model of uncertainty, which is more statistically accurate when data are derived from empirical measurements. For the likelihood model, we first collect for each state $s \in S$ the vector $\mathbf{h_s}$ of measured transition frequencies to all next states s'. The uncertainty set of DTPD is then defined as:

$$\mathcal{F}_s = \{\mathbf{f}_s \in \mathbb{R}^N \mid \mathbf{f}_s \geq \mathbf{0}, \mathbf{1}^T \mathbf{f}_s = 1, \sum_{s'} h_{ss'} \log(f_{ss'}) \geq \beta_s\} \quad (1)$$

where $\beta_s < \beta_{s,max} = \sum_{s'} h_{ss'} \log(h_{ss'})$ represents how uncertain the estimation is (a smaller value of β represents more uncertainty). In Section 3, the choice of β in the context of modeling the behavior of driver is explained in detail.

After creating the CMC models, we aim to verify their logical properties. We use PCTL to express properties of interest in the probabilistic framework we have defined. PCTL is a probabilistic logic derived from Computation Tree Logic (CTL) which includes a probabilistic operator P. (Hansson and others 1994) The syntax of this logic is as follows:

$$\begin{array}{ll} \phi ::= True \,|\omega\, |\neg\phi\, |\phi_1 \wedge \phi_2\, |P_{\bowtie p}[\psi] & \text{state formulas} \\ \psi ::= \mathcal{X}\phi\, |\phi_1 \mathcal{U}^{\leq k} \phi_2\, |\phi_1 \mathcal{U} \phi_2 & \text{path formulas} \end{array}$$

where $\omega \in \Omega$ is an atomic proposition, $\bowtie \in \{\leq, <, \geq, >\}$, $p \in [0, 1]$, and $k \in \mathbb{N}$. Here $\mathcal{X}$ is the *Next* operator, $\mathcal{U}$ is the *Unbounded Until* operator and $\mathcal{U}^{\leq k}$ is the *Bounded Until* operator.

Probabilistic statements about MCs typically involve computing the probability of taking a path $\pi \in \Pi_s$ that satisfies ψ. With uncertain transition probabilities, we will need to universally quantify across all possible resolutions of uncertainty to compute the worst case condition within the uncertainty set $\mathcal{F}_s$. This choice is suitable in a verification setting to determine the most conservative behavior of the model under analysis. We define $P_s(F)[\psi] \triangleq Prob(\{\pi \in \Pi_s(F) \mid \pi \models \psi\})$ the probability of taking a path $\pi \in \Pi_s$ that satisfies ψ when using the DTPDs collected in the observed transition matrix F. $P_s^{max}[\psi]$ ($P_s^{min}[\psi]$) denote the maximum (minimum) probability $P_s(F)[\psi]$ across all possible observed transition matrices F.

The PCTL semantics for CMC is then defined as:

$$\begin{array}{lll} s \models P_{\triangleleft p}[\psi] & \text{iff} & P_s^{max}(\{\pi \in \Pi_s | \pi \models \psi\}) \triangleleft p \\ s \models P_{\triangleright p}[\psi] & \text{iff} & P_s^{min}(\{\pi \in \Pi_s \models \psi\}) \triangleright p \\ \pi \models \mathcal{X}\phi & \text{iff} & \pi[1] \models \phi \\ \pi \models \phi_1 \mathcal{U}^{\leq k} \phi_2 & \text{iff} & \exists i \leq k | \pi[i] \models \phi_2 \wedge \forall j < i \; \pi[j] \models \phi_1 \\ \pi \models \phi_1 \mathcal{U} \phi_2 & \text{iff} & \exists k \geq 0 | \pi \models \phi_1 \mathcal{U}^{\leq k} \phi_2 \end{array}$$

Details on the polynomial-time algorithms used to verify PCTL properties of CMCs, together with a more rigorous exposition of the material presented in this section, can be found in (Puggelli and others 2013). The source code of the implemented verification tool can be found on our website [2].

[2] http://www.eecs.berkeley.edu/~puggelli/

3 Modeling of the Driver Behavior

In this section, we show how to create a probabilistic model of the human driving behavior. The primary goal is to give an accurate prediction of the future trajectories of the car over a long-time horizon. This also allows the analysis of complex maneuvers, e.g., a sequence of a right and a left turn, and the study of how decisions made early in the maneuver can affect later decisions. Our approach postulates that the driver state, e.g., attentive or distracted, must be considered to increase the accuracy of the prediction. By examining the driver in this way, we are able to carefully assess the threat the driver faces, which is becoming increasingly important in the development of advanced control algorithms (Vasudevan and others 2012).

As will be described in the following sections, data is gathered from multiple drivers, in order to create individualized driver models that are generated from empirical observations collected in a simulation environment. Once the predictive model is created, time and space are discretized to form a discrete time Convex Markov Chain (CMC) and to quantitatively analyze the properties of the model. An arbitrary level of accuracy in the model can be obtained by appropriately choosing the discretization step, at the expense of longer runtime of the verification algorithm. The following section describes the experimental setup and methods used to develop the driver model.

3.1 Methods and Experimental Setup

In order to learn the model of an individual driver, we use the approach described in (Vasudevan and others 2012) and (Shia and others 2014). According to this method, the driver behavior is dependent on particular modes or scenarios, which are determined by the future external environment, e.g., a turn in the road, and by the driver state, e.g., attentive or distracted. We start by collecting data representing the observed steering angle input of the driver, sampled every $30ms$. In order to recover these unknown modes without making any unnecessary assumptions, the data is clustered using the k-means algorithm (Hartigan and others 1979), which allows for flexibility in determining the modes in an unsupervised manner. This also allows the driver model to predict the driver behavior in any scenario in which the model is effectively trained. In the setup presented in this paper, these clusters or modes are created from the following data sets:

1. *Driver Pose:* This contains the past two seconds of skeleton data, specifically the positions of the wrist, elbow, and shoulder joints.
2. *Environment Estimation:* This contains a feature vector for the future four seconds of the outside environment, including road bounds and curvature, obstacle locations, and the car's deviation from the lane center.

The model uses a tiered structure by first clustering the environment vector, and then the driver pose. This allows us to compare driver behaviors in a matching environment for different driver states. Empirical considerations suggested us to use 150 clusters, $k_1 = 50$ environment clusters at the first tier and $k_2 = 3$ pose clusters at the second tier, to achieve the desired level of accuracy in the model (Shia and others 2014).

Once the data has been clustered into modes, the future 1.2 seconds (a time frame comparable to the human reaction time) of the driver steering angle inputs associated with each mode become the prediction for that scenario. The predicted steering angle inputs can then be passed to a vehicle model to generate the future vehicle trajectories and thereby infer the driver's intentions. This creates a driver model that is able to continuously predict the future behavior of the driver and, by extension, of the vehicle trajectory, as it moves through a given environment.

For simplicity, a linearized vehicle model is used to generate the trajectory set (Vasudevan and others 2012). In our setup, we assume that the driver is driving at approximately 60 miles per hour. This assumption does not limit our approach and is appropriate in a highway scenario. A sample trajectory set is shown in Fig. 1. This graph shows the change of lateral direction of each trajectory in meters from a given initial condition for the future $1.2s$. As it is apparent from the figure, some of the trajectories stay within the lane, some drift outside of the lane and some others diverge to the left lane.

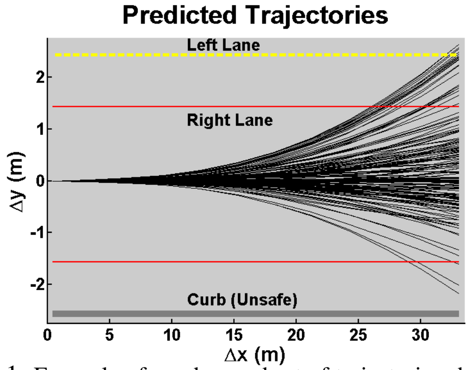


Figure 1: Example of an observed set of trajectories showing the change in lateral and longitudinal directions with respect to the center of the car at $\Delta y = 0$.

To implement this model, we used CarSim, a standard car simulation software used by industry [3], as simulation environment, and the Microsoft Kinect [4] to observe the driver pose in real-time. Multiple subjects were asked to drive through four courses to collect over an hour of training data, to observe individual driving habits in different environments and driver attention levels. To simulate distracted driving, an application was installed on a cell phone to prompt the driver to answer the call or text while continuing to drive. For more details about the formulation and implementation of the driver model, we refer the reader to (Shia and others 2014).

In conclusion, the driver model developed so far answers the question: "Given the mode the driver is currently in, how will he or she drive in the next $1.2s$?" By identifying the modes using observations of the environment and of the

[3] http://www.carsim.com

[4] http://www.xbox.com/en-US/KINECT

driver state, the associated future steering angle inputs can be used as a prediction of the driver behavior.

3.2 Stochastic Modeling

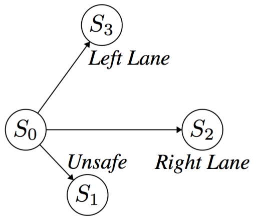
In this section, we show how to create a stochastic model of the driver behavior capable of representing complex maneuvers. We start by collecting the clusters generated as described in Section 3.1 into a library. Each cluster is annotated with *labels* describing the environmental and driver states associated to it. We then instantiate these to form the model of the car driver along the road to be analyzed. Transition between cluster instances are defined in a probabilistic fashion by considering the predicted behavior within the cluster.

To illustrate our methodology, we explain how to convert the trajectory data in Figure 1 to the model in Figure 2. We start by associating to the cluster under analysis the state S_0. We then assume that the trajectory starts in the center of the right lane of a two-lane road. Using standard values for the car and lane widths, we classify the trajectories within the clusters in three subsets, lane changing, lane keeping, or drifting, depending of the final $y-$coordinate. In the example, the trajectories that exit the safe region of the road toward the curb are identified as "*Unsafe*", those that remain in the middle of the road are marked "*Right Lane*", while those that tend towards the left lane are identified as "*Left Lane*". We can now associate a new state to each maneuver, S_1, S_2, S_3, representing the three locations where the car may be in the next time step.

Finally, the empirical probabilities to perform each maneuver are calculated by examining the percentage of trajectories within the cluster that terminate in the corresponding region (see labels in Fig. 1). The table in Fig. 2 illustrates the assigned probabilities for the example under analysis.

Since the computed transition probabilities are based on empirical driving data, we introduce uncertainty sets around the derived probabilities to accommodate for estimation errors. The size of the set depends on the level of confidence in the collected measures. We use both the interval and likelihood uncertainty models. The interval model is intuitive to understand and will be used as a baseline for our quantitative analysis. Given a confidence level $0 \leq C_L \leq 1$ in the measurements and an empirical probability p_0, we compute the transition probability interval $[C_L \times p_0, (2 - C_L) \times p_0]$. We also use the likelihood model of uncertainty, because it is less conservative and it is largely used for representing uncertainties on empirical data. Using classical results from statistics (Nilim and others 2005), we can compute the value of parameter β from set $\mathcal{F}_s$ corresponding to a confidence level C_L. In particular, $C_L = 1 - cdf_{\chi^2_d}(2 * (\beta_{max} - \beta))$, where $cdf_{\chi^2_d}$ is the cumulative density function of the Chi-squared distribution with d degrees of freedom ($d = 3$ in this example because there are three possible next states).

Because the driver model is defined by the driver state, the procedure described here can be applied to both attentive and distracted drivers. This allows us to determine changes in behavior for a specific driver depending on the driver state. The modeling technique described can be iterated to create a stochastic model of the car driver behavior on a road of arbitrary shape-assuming the driver model is appropriately trained. We describe a more elaborated example in Section 3.3.



Transition	Transition Probability Interval
$S_0 \to S_1$	[0.019,0.021]
$S_0 \to S_2$	[0.890,0.980]
$S_0 \to S_1$	[0.048,0.053]

Figure 2: Example of creating transition probability intervals from a trajectory set. We used $C_L = 95\%$.

We are now ready to formally describe the created stochastic model using the formalism of CMCs, as introduced in Section 2.

In the *CMC* corresponding to our models, $\mathcal{M}_C = (S, S_0, \Omega, \mathcal{F}, X, L)$, we let S represent the set of instantiated clusters with associated trajectory sets, and S_0 represent the initial state. We assign a set of atomic labels Ω to each cluster encoding the environmental and driver states. For example, labels can mark clusters on the right or left lane, and clusters used during a right or left turn or during a straight segment of road. Labels are also used to mark *Safe* (*Unsafe*) states if they are within (outside) the road boundaries. We also label a state as *Accelerating* or *Braking*, if the value of acceleration is above or below a chosen threshold, and *Swerving* if the number of swerving trajectories is above a threshold (swerving marks potentially dangerous driving). We label the goal set of states as *Final*, to mark the end of the complex maneuver. Finally, we use the labels *Attentive* and *Distracted* to mark the corresponding data sets.

$$\Omega = \{\textit{Right Turn, Left Turn, Straight, Right Lane,}\\ \textit{Left Lane, Safe, Unsafe, Braking, Accelerating,}\\ \textit{Swerving, Final, Attentive, Distracted}\}$$

The set $\mathcal{F}$ collects all the convex sets of transition probability distributions each encoding the chosen confidence level and uncertainty model, while mapping $X : S \to \mathcal{F}$ associates each state with the corresponding convex uncertainty set of probability distributions to the next states. Finally, the labeling function L maps each state to the corresponding set of labels in Ω.

3.3 Model of a Complex Maneuver

As a more elaborated example of our modeling approach, we created a *CMC* model of the road segment shown in Fig. 3. The road consists of the sequence of sections: straight, right turn, left turn and another straight for both lanes. States are assigned to different locations on the road. Each state is a cluster that is chosen from the library of clusters for a specific driver. We let $\{S_0, \cdots, S_{15}\}$ represent the set of states. S_0 is the initial state, and $S_i = \{D_i, A_i\}$ for $i \in \{0, 1, 2, 7, 8, 11, 12\}$ represent a set of two different states, one for distracted driving and one for attentive driving as shown in Fig. 3.

We labeled as *Attentive* the states where the human driver's pose suggests that both hands of the driver are on the steering wheel. States are labeled as *Distracted* if the human pose suggests that the driver is holding a phone or sending a text message.

We created two different models to represent scenarios with (without) an obstacle on the road right before the right turn (Fig. 3 on the left and right, respectively). In the scenario with the obstacle, some trajectories in S_0 terminate on the left lane, represented by state S_1. With no obstacle, trajectories in S_0 instead either keep the right lane or drift outside of the boundaries.

States $\{S_3, S_4, S_5, S_6, S_9, S_{10}, S_{13}\}$ correspond to *Unsafe* states since they are all out of the boundaries of the road; the rest of the states are *Safe* states. States $\{S_{14}, S_{15}\}$ are marked as *Final* states. The *Right Lane* and *Left Lane* labels correspond to states $\{S_0, S_2, S_8, S_{12}, S_{15}\}$ and $\{S_1, S_7, S_{11}, S_{14}\}$, respectively. The rest of the labels in Ω depend on the specific driving style of each subject. Probability distributions for transitions between states are assigned as we discussed in Sec. 3.2. In fact, Fig. 2 shows the transition probability distribution from S_0 to S_1, S_2 and S_3 in Fig. 3. The comparison and analysis of the two models in Fig. 3 are discussed in Sec. 4.

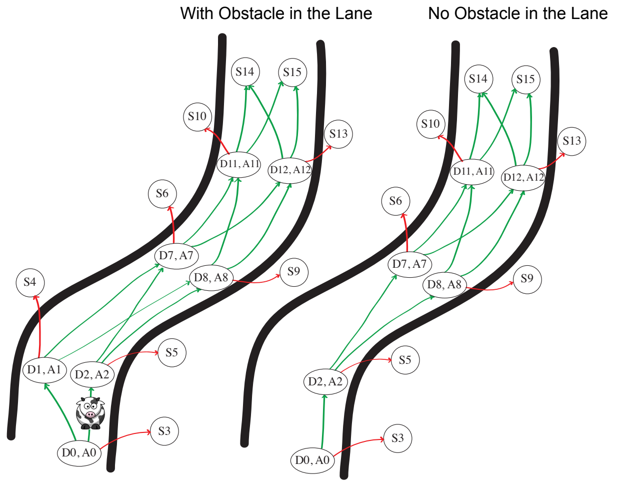


Figure 3: Convex Markov Chain of the road segment. States $\{D_i, A_i\}$ represent a set of two states one labeled *Distracted* and the other *Attentive*.

4 Formal Verification of the Driver Model

In this section, we verify *quantitative* properties of the behavior of car drivers for the two road models introduced in Section 3.3. We use PRISM (Kwiatkowska and others 2011), a probabilistic model checker, as our front-end tool, and the Python back-end verification algorithm developed by Puggelli et al. (Puggelli and others 2013) to incorporate the likelihood and interval uncertainty models.

We verify the PCTL properties in Table 1. For each property, we consider both attention levels, attentive and distracted, and compute both the maximum and minimum satisfaction probabilities, to give the range of predictions obtainable from the model. Property P1 computes the probability of reaching an *Unsafe* state. Properties P2 - P3 - P4 compute the probabilities of reaching a final state without swerving, by always staying on the right lane, and without braking, respectively. Overall, these properties allow capturing different driving styles among subjects and assess possible threats.

Table 1: Verified Properties

P1	P_{max}/P_{min}	[Attention $\mathcal{U}$ $Unsafe$]
P2	P_{max}/P_{min}	[(Attention $\wedge$ $\neg Swerving$) $\mathcal{U}$ $Final$]
P3	P_{max}/P_{min}	[(Attention $\wedge$ $Right\ Lane$) $\mathcal{U}$ $Final$]
P4	P_{max}/P_{min}	[(Attention $\wedge$ $\neg Braking$) $\mathcal{U}$ $Final$]
Attention is a placeholder for either *Attentive* or *Distracted*		

First, we compare in Fig. 4 verification results when using no uncertainty model, the interval model and the likelihood model of uncertainty. We assume 95% confidence level for both the interval and likelihood models. For ease of comparison, we only report results for one subject and only for maximum probability of P1 (top) and minimum probability of P3 (*bottom*). The results for the other subjects and properties follow a similar trend. As expected, the probability of reaching an unsafe (*safe*) state is higher when the driver is distracted (*attentive*) and in the presence (*absence*) of an obstacle along the road. Further, we note that: 1) probabilities computed for models with no uncertainty are significantly lower (*higher*), which implies that this method is potentially too optimistic for an appropriate threat assessment; 2) even a low value of C_L causes the computed probabilities to increase (*decrease*) substantially for the interval model, which might thus result in overly-conservative estimations; and 3) the likelihood model appears to be a good trade-off between the other two models. In the following analysis, the likelihood model is used, which is often used when probabilities are estimated from experimental data.

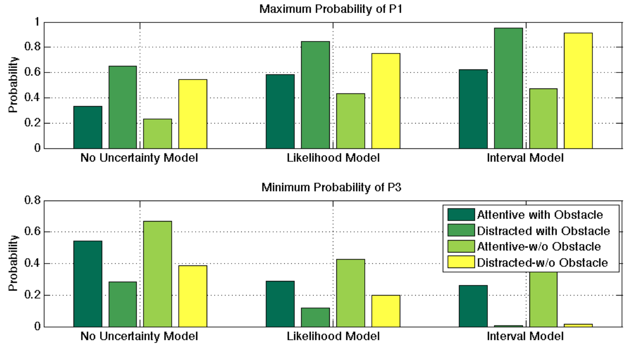


Figure 4: Comparison of the verification results of P1 and P3 for different models of uncertainty.

To examine the effects of different confidence levels on the probabilities, we compare the results from attentive versus distracted driving. Fig. 5 shows the verification results for P1 for values of C_L ranging from 60% to 99% for one driver. The probability of reaching an unsafe state is always lower in the case of attentive driving. The probability also decreases as we increase the confidence level. The model developer can use this plot to determine when the collected measurements are statistically relevant to estimate the driver behavior. The trend shown in Fig. 5 repeats for all the other subjects. However, the disparity between the results for attentive and distracted driving varies for each driver.

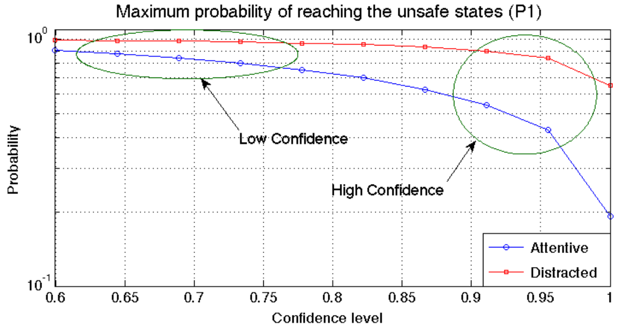


Figure 5: Comparison of distracted and attentive driving for different values of confidence level C_L for Property P1.

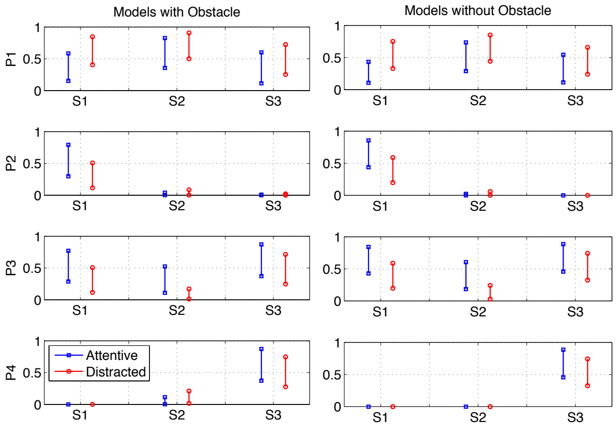


Figure 6: Verification results. The vertical bar marks the range between minimum and maximum satisfaction probability.

Finally, we report the collected verification results for three subjects (S) and for all four properties (P) in Fig. 6. Results capture different driving styles among subjects and help assessing possible threats. For example, results for P3 show that S2 often ended up on the left lane while performing the maneuver, while S1 and S3 managed to keep the right lane in most cases. Further, results for P4 show that S1 and S2 tended to brake often when performing the maneuver, while S3 travelled along the road braking rarely. For all subjects, the presence of the obstacle increased the probability of reaching an unsafe state.

5 Conclusion

In conclusion, we showed how to create a stochastic model of a car driver behavior starting from experimental data. Our modeling approach is suitable to represent complex maneuvers, and to take the uncertainty intrinsic to the measurements into account. We analyzed quantitative properties of the created models. Results show that the developed framework is suitable to discern particular characteristics of the driving pattern of each driver and give insight about the presence of threats while driving. Finally, we note that the modeling and verification techniques introduced in this paper apply also to human-machine systems in a broader set of domains, including avionics and medicine.

As future work, we intend to extend the model to account for choices made by the driver, e.g., an attentive driver during the turn can become distracted on a straight segment of road, and analyze how these choices affect the verification result. We also would like to extend this work to the control domain, where the model is used to give on-line feedback to the driver or even compose with the model of a controller in a real-time semiautonomous framework.

References

Anderson, S., et al. 2010. An Optimal Control-Based Framework for Trajectory Planning, Threat Assessment, and Semi-Autonomous Control of Passenger Vehicles in Hazard Avoidance Scenarios. *International Journal of Vehicle Autonomous Systems* 8(2):190–216.

Cacciabue, C. 2007. *Modelling Driver Behaviour in Automotive Environments*. Crit. Issues in Driver Interactions with Intelligent Transport Systems. Springer.

Clarke, J., et al. 1999. *Model checking*. MIT Press.

Coelingh, E., et al. 2007. Collision Warning with Auto Brake: a Real-Life Safety Perspective. In *Innovations for Safety: Opportunities and Challenges*.

Hansson, H., et al. 1994. A Logic for Reasoning About Time and Reliability. *Formal Aspects of Computing* 6(5):512–535.

Hartigan, J. A., et al. 1979. Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society.* 28(1):pp. 100–108.

Klauer, S. 2006. The Impact of Driver Inattention on Near-Crash/Crash Risk: An Analysis Using the 100-Car Naturalistic Driving Study. *National Highway Traffic Safety Administration*.

Kwiatkowska, M., et al. 2011. PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. 23rd Int. Conf. on Computer Aided Verification (CAV'11)*, LNCS, 585–591. Springer.

Lin, C.-C., et al. 2004. A Stochastic Control Strategy for Hybrid Electric Vehicles. In *Proceedings of the American Control Conference.*, volume 5, 4710–4715.

Nilim, A., et al. 2005. Robust Control of Markov Decision Processes with Uncertain Transition Matrices. *Operations Research* 53(5):780–798.

Puggelli, A., et al. 2013. Polynomial-Time Verification of PCTL Properties of MDPs with Convex Uncertainties. In *Proc. 25th Int. Conf. on Computer Aided Verification (CAV'13)*, LNCS, 527–542. Springer.

Shia, V., et al. 2014. Driver modeling for semi-autonomous vehicular control. *IEEE Transactions on Intelligent Transportation Systems*.

Vasudevan, R., et al. 2012. Safe Semi-Autonomous Control with Enhanced Driver Modeling. In *American Control Conference (ACC)*, 2896–2903. IEEE.