Learning Driver Behavior Models from Traffic Observations for Decision Making and Planning

Tobias Gindele, Sebastian Brechtel, and Rüdiger Dillmann
Humanoids and Intelligence Systems Laboratories, Karlsruhe Institute of Technology
Adenauerring 2, D-76131 Karlsruhe, Germany
E-mails: gindele@kit.edu, brechtel@kit.edu, dillmann@kit.edu

***Abstract*—**Estimating and predicting traffic situations over time is an essential capability for sophisticated driver assistance systems and autonomous driving. When longer prediction horizons are needed, e.g., in decision making or motion planning, the uncertainty induced by incomplete environment perception and stochastic situation development over time cannot be neglected without sacrificing robustness and safety. Building consistent probabilistic models of drivers interactions with the environment, the road network and other traffic participants poses a complex problem. In this paper, we model the decision making process of drivers by building a hierarchical Dynamic Bayesian Model that describes physical relationships as well as the driver's behaviors and plans. This way, the uncertainties in the process on all abstraction levels can be handled in a mathematically consistent way. As drivers behaviors are difficult to model, we present an approach for learning continuous, non-linear, context-dependent models for the behavior of traffic participants. We propose an Expectation Maximization (EM) approach for learning the models integrated in the DBN from unlabeled observations. Experiments show a significant improvement in estimation and prediction accuracy over standard models which only consider vehicle dynamics. Finally, a novel approach to tactical decision making for autonomous driving is outlined. It is based on a continuous Partially Observable Markov Decision Process (POMDP) that uses the presented model for prediction.

IMAGE PUBLISHED BY INGRAM PUBLISHING

I. Introduction

Today's assistance systems make driving safer and easier. To develop the next generation of advanced driver assistance systems or even self-driving cars, methods are required for estimating the state of a car's environment and making accurate predictions of the development of traffic situations.

This paper presents an approach for estimating and predicting traffic situations. With this approach, the distribution of possible situation developments over several seconds can be anticipated based on a history of noisy measurements of the pose and velocity of road users. This is a challenging task because traffic situations are complex, continuous, uncertain, highly dynamic and only partially observable.

We aim to resemble the decision-making process of drivers on several abstraction levels in a hierarchical probabilistic model. This allows fine-grained predictions of future vehicle

Digital Object Identifier 10.1109/MITS.2014.2357038
Date of publication: 20 January 2015

1939-1390/15©2015IEEE

poses as well as high-level predictions on the level of intended driving routes. To ensure realistic predictions, context-dependent behavior models are learned from traffic observations and background knowledge. The inherent uncertainties are considered in the whole process in a consistent manner.

The task of anticipating the possible developments of traffic situations is complicated by the partial observability of the environment. A system equipped with various kinds of sensors can only perceive limited aspects of its environment. Often only basic features of a traffic participant can be measured such as his position, orientation, velocity or appearance. Further, even these basic features can only be measured with non-neglectable errors. Other relevant information, like the driver state of mind, what he is planning or which parts of the scene he has seen and considered in his decision making stay hidden. They can only be estimated from the observations to a certain degree of confidence. For example, if a car is quickly approaching a slow driving truck on a highway, it can only be concluded and not directly measured that the driver of the car possibly intends to overtake the truck. The additional possibility that he could also brake and stay behind the truck is only expressible using a probabilistic representation.

The improvement of the sensors or new technologies like Car2X communication will certainly improve the range and quality of information a system can gather about its environment. However, the limited observability of the environment can only be alleviated and never fully eliminated. If the planned route is known, e.g., from a navigation system or because the car drives autonomously, it can be communicated. This information can be directly used by the presented approach to improve the accuracy of the predictions.

Methods that can cope with these limitations to reason about the current state and future development of traffic situations are necessary to implement more sophisticated assistance functions and increase the systems autonomy. These will enable applications like risk assessment, stochastic decision making with Markov Decision Processes [1] and stochastic long-term motion planning. Especially the influence of decisions, which an autonomous vehicle or a driver makes, on the future behavior of other traffic participants is neglected in most approaches. However, the correct anticipation of the reaction of other traffic participants on the own decisions is essential, e.g., for planning merging maneuvers into moving traffic.

Predicting traffic situations is only tractable because the transitions of environment states are not completely random, but instead show a lot of regularities. Cars, e.g., move according to kinematic and dynamic constraints. Also, traffic is structured by regulations and infrastructure like lanes, signs and traffic lights. Most importantly, traffic participants do not act randomly, i.e. they usually try to follow the traffic rules and avoid accidents. All these aspects combined lead to patterns in the behavior of traffic participants that can be exploited by an intelligent system to anticipate how a situation will develop. The key to this ability is to find suitable models that capture these patterns.

Building such a model that describes the behavior of drivers manually is difficult for human experts since there is a large variety of aspects that need to be considered and their meaning can change depending on the current situation. This context dependency and the difficulty of quantifying the influence of the different aspects on the change in state make this approach nearly intractable.

In this paper we propose to learn the behavior models from observations automatically in order to overcome the limitations of deterministic predictions and manually formulated models. The learned models are embedded in a probabilistic framework for context-sensitive state estimation and prediction based on a Dynamic Bayesian Network (DBN). In contrast to previous approaches, the learning process is enriched with background knowledge, like e.g., information about the road topology. As a consequence, it learns not only for static situations, like a single intersection, but is able to extract abstract behavior rules from the observations and thus generalize the learned models to new, unseen traffic situations and road constellations.

After explaining the current state of research in the field of traffic prediction and estimation, we give an overview on the basic concept of the system and the presented ideas. In Section IV the Bayesian model used to predict and estimate traffic situation with arbitrary participants is explained, including the random variables and the conditional models that relate them. In Section V we explain how to learn the policy model that describes the behavior of traffic participants in their context from noisy and incomplete data. We evaluate the approach in a 4-way-intersection scenario in Section VI. Finally, we give a brief overview on the possible uses of the learned Bayesian model in Section VII and explain how it can be embedded into a POMDP-based decision making system.

II. State-of-the-Art

In recent years, the field of state estimation and prediction of traffic scenarios is receiving more and more attention from the research community due to the increasing interest in advanced driver assistance systems (ADAS) and autonomous driving. Other than in the area of multi-target tracking, where objects are often assumed to move independently, the motion patterns of traffic participants are highly coupled and influenced by many aspects like the structure of the road network or traffic signs. In order to predict the behavior of traffic participants successfully contextual information and the interactions between objects need to be considered.

Such interactions are considered in the related area of micro-simulations, e.g. for traffic flow analysis, safety assessment or infrastructure-based traffic-management [2]. For these tasks it is sufficient, if the planned path of the simulated road users is set by the simulation. For state estimation and prediction this information has to be estimated from indirect measurements and the full distribution of possible outcomes has to be considered. Further, such simulations usually use simplified models that do not allow accurate predictions on the level of continuous poses and velocities.

Several directions have been investigated to solve the state estimation and prediction problem in the traffic domain. The authors of [3] [4] proposed a motivation based approach. Other than relying solely on the system dynamics, they conclude that it is necessary to infer the situation specific motivations and goals of a driver in order to predict his behavior. A fully probabilistic approach is presented in [5]. It uses contextual information to detect the type of situation a vehicle is in. This information is used to predict the behaviors of the drivers and the resulting vehicle trajectories. Another approach along the same line is presented in [6]. They use feature functions that evaluate aspects of the situational context to estimate symbolic context classes that allow to predict the next state of a traffic participant. In [7], an approach is presented that shows improved filtering performance over a Bayesian filter without context information by modeling behaviors as spline functions mapped on the road network.

The subproblem of situation recognition is addressed in [8], where a relational hidden Markov model is used to recognize different classes of traffic situations based on a semantic representation. In [9], a system is presented which recognizes the class of the situation a traffic participant is facing, based on a manually constructed decision tree. A situation specific classifier is then used to predict the behavior type of a traffic participant. The authors of [10] also present a prediction system on the level of behavior primitives. They train a multi-class classifier on the basis of context features. While these approaches can classify and predict situations or behaviors on a symbolic level they cannot be used to predict the quantitative properties of vehicles like their position or heading and their future trajectories. This is essential for example for the task of motion planning.

Some authors focused their work on predicting motion patterns at intersections [11] [12].

III. Overview

We address the problem of state estimation and prediction in the domain of traffic situations by modeling the evolution of traffic situations as a stochastic process. By representing this process with a Dynamic Bayesian Network (DBN) [13] the system is able to reason about current, future or past states of the environment given the evidence obtained from noisy measurements of the pose and velocity of the traffic participants. It does not make a difference whether the measurements come from a sensor-equipped vehicle that is part of the observed traffic situation itself, from a remote observer like fixed mounted traffic cameras or are directly communicated by Car2X. To implement the probabilistic relationship models of the DBN, we use a combination of manually formulated models utilizing background knowledge and models automatically learned with random forests [14] for the complex behaviors of traffic participants.

A temporal slice of the dynamic Bayesian network represents the state of the environment at a specific point in time, i.e. a situation. The network is automatically constructed according to the dynamic objects in a scene. Each object can be e.g. a car, a truck or a bicyclist. Objects are represented by a set of nodes representing their internal states as well as the relationships between the objects and the infrastructure. The nodes are connected with conditional distribution functions representing the models. Depending on the asserted objects the concrete network topology is instantiated from templates that represent partial relationships [15].

The Bayesian network uses a factored representation of the joint distribution to exploit conditional independencies between random variables. This allows to use more efficient probabilistic inference algorithms. Further, the definition of the models is simplified because less dependencies between random variables have to be considered due to the conditional independencies. It also provides the facility to define those relationships where expert or background knowledge is available by hand and to learn those dependencies from data that are hard to formulate. This way, we can combine the use of map data and traffic rules with machine learning.

The basic idea behind the structure of the Bayesian model is the recursive formulation of the reasoning and decision processes of the traffic participants. We assume that each traffic participant has an implicit intention that is the driving force behind his actions. The goals of each traffic participant can be achieved through multiple ways. Some of them are more likely than others depending on the current situation. Depending on his plan and situational context a traffic participant conducts the action that he thinks is best. This choice influences the state of the environment and thereby the overall situation. Since any change of the environmental state can have influence on the decision making of a traffic participant, they influence the decision context and consequently the plans and actions of each other continuously. Examples for this interrelation range from cars that decelerate to not collide with slower cars or yielding at intersections to more subtle actions like extending a gap to provide a car more space for merging safely. To realize this idea of the reasoning process we propose the Bayesian model in Figure 1. It combines different levels of abstraction: High-level plans subsume a great set of possible realizations. Low-level actions directly influence the dynamics of the traffic participant. All layers have mutual dependencies through the temporal coupling.

We use a combination of expert knowledge, background knowledge and machine learning to derive the models of the Bayesian network. The idea is to learn all models from data which are difficult for an human expert to define. This especially applies to the policy model which predicts the action of a traffic participant based on his situational context. Quantifying the stochastic influence of every

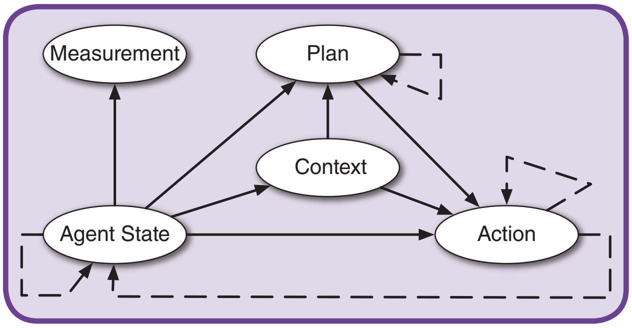

FIG 1 Scheme of the Bayesian model. Solid arcs represent direct dependencies and dashed arcs temporal dependencies.

aspect of a situation is hardly tractable for human experts. Since data of how drivers behave in specific situations can be obtained easily and in big quantities by observing traffic very accurate models can be learned.

IV. Bayesian Model

The Bayesian model defines the complete dynamic process needed to predict and estimate traffic situations. Therefore the state space and the models to predict and measure the traffic participants are defined. The model is decomposed for the different traffic participants and into several aspects.

A. State Space

The state space of the Bayesian model consists of random variables describing the different aspects of a situation. They are defined as follows:

1) *Traffic participant states* X: Vector of the states of all n traffic participants:
$$X = (X_1 \dots X_n)^{\mathrm{T}}.$$
A state X_i of the i-th traffic participants is described by his position (x_1, x_2), heading ψ and velocity v.
2) *Measurements* Z: Vector of the measurements of the state of all traffic participants:
$$Z = (Z_1 \dots Z_n)^{\mathrm{T}}.$$
We assume that all basic states of a traffic participant such as his position, velocity and heading can be measured directly with noise. Consequently Z and X have the same domain. Other traffic participant states, such as his intentions or plans, are assumed to be non-measurable. We abstract here from specific sensors and their processing of the raw sensor data to keep the model more general.
3) *Relations* R: R describes the relations between all traffic participants. The relation $R_{i,j}$ contains crucial information for the interaction of the i-th and the j-th traffic participant, e.g., their distance and difference in orientation:
$$R_{i,j} \text{ with } i,j \in [1,\dots,n] \text{ and } i \neq j.$$
4) *Lanes* L: Vector of lane segment sequences representing the routes which the traffic participants are planning to use, starting at their current lane segment. The lane segments are indexed and correspond to edges in the road network graph.
$$L = (L_{1,1:m} \dots L_{n,1:m})^{\mathrm{T}}.$$
The routes have a finite horizon of length m and are represented by $L_{i,1:m} = (L_{i,1} \dots L_{i,m})^{\mathrm{T}}$. The routes represent the plans of the traffic participants on an abstract spatial level. This enables high-level reasoning about interactions and potential conflicts between traffic participants on the basis of overlapping and intersecting routes.
5) *Lane relations* LR: Contains vectors of relations between the traffic participants and the lanes.
$$LR = (LR_{1,1:m} \dots LR_{n,1:m})^{\mathrm{T}}.$$
Each lane relation vector $LR_{i,j}$ corresponds to a lane section $L_{i,j}$ in L. The lane relations subsume features that describe the relationship lanes and traffic participants. These contain the difference in orientation between the heading of the vehicle and direction of the lane center line, how far the vehicle is away from the borders of the lane or next lane segment, among others. These relations directly support the learning process of the policy model by providing a preprocessed view on the geometrical properties, thereby eliminating the need to learn some of the non-linear geometrical transformations from data.
6) *Actions* A: Vector of actions of the traffic participants:
$$A = (A_1 \dots A_n)^{\mathrm{T}}; A_i = (\omega \quad a)^{\mathrm{T}}.$$
The actions represent the influence the traffic participants have over their own motion by controlling their yaw rate ω and acceleration a.

B. Joint Distribution and Filter Equations

The joint distribution of all random variables can be decomposed under the assumption of conditional independencies. Since we assume the process to be first-order Markov, only random variables with time index t and $t-1$ must be considered. For the sake of clarity variables marked with a minus (like X^-) correspond to random variables at time index $t-1$, variables that are not marked to time index t. The joint distribution factors as follows:
$$\begin{aligned} p(X, Z, L, LR, R, A, X^-, A^-) = \\ p(X^-, A^-) p(X \mid X^-, A^-) p(Z \mid X) p(L \mid X) \\ p(LR \mid X, L) p(R \mid X) p(A \mid X, R, LR). \quad (1) \end{aligned}$$
Given the joint distribution of all variables (1), the recursive update formulas can be derived using Bayes' rule. The distribution of the posterior state is obtained by marginalizing over the prior state and weighting with the likelihood of the measurement z:
$$p(X \mid z, X^-) \propto p(z \mid X) \int_{X^-, L^-, LR^-, R^-, A^-} p(X, X^-, L^-, LR^-, R^-, A^-). \quad (2)$$

With (2), the current estimate of the situation can be updated with new measurements based on the old estimate. By omitting the weighting with measurements predictions can be derived.

C. Models

The models define the interrelations between the important aspects of a traffic situation. Conditional density functions describe the relations and basically attribute a meaning to the random variables described in Section IV-A. An overview on the interrelations between the random variables in the state space is given in the DBN in Figure 2. Some of the conditional density functions can be efficiently modeled manually, like the lane relations or the physical behavior of traffic participants. The reaction of the traffic participants on their context, however, depends on too many aspects. Hence, in Section V-B we propose a machine learning approach for assigning the *policy model* automatically. By choosing the manually defined relations carefully so that they cover complex and non-linear aspects, e.g., the relation of traffic participants to the road network, the learning can focus on covering the most eminent factor in the process: the human behavior.

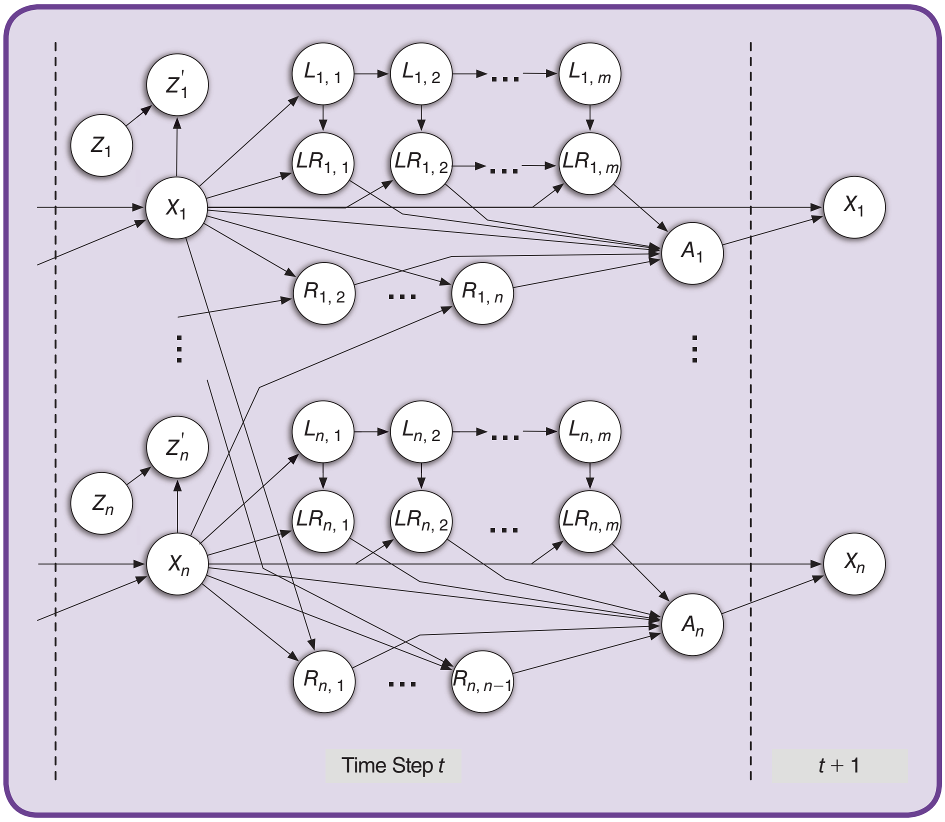

FIG 2 Structure of the dynamic Bayesian network for n traffic participants and a route length of m.

As the DBN can be decomposed, the model descriptions in this section will describe the situation context just for one of the n traffic participants. These n instances are related over time because the state of each traffic participant influences the context of the others, as can be seen in Figure 2.

1) *Measurement Model* $p(Z'|X,Z)$: The measurement model relates the hidden states of the traffic participants with the measurements of the observable quantities. Due to the periodicity of the angular orientation it cannot be modeled as a direct forward model $p(Z|X)$ like in the general Bayes filter case. Instead we use a helper random variable Z' that represents the difference of X and Z. This is a common technique already mention by Pearl [16]. The model is given by $p(Z'|X,Z) = d(x,z) + \varepsilon_z$ with

$$d(x,z) = \begin{bmatrix} x_1^x - x_1^z \\ x_2^x - x_2^z \\ \text{angle}(\psi^x, \psi^z) \\ v^x - v^z \end{bmatrix} \text{and } e_z \sim N(0, \Sigma_z).$$

An observation of the state of a traffic participant then consists of the measured evidence on Z and a zero valued evidence on Z' stating that the difference between the true state X and the measured state Z should be zero. The measurement model contains the assumption that the traffic participants and their corresponding measurements can be uniquely identified. Otherwise methods for solving the data association problem have to be applied. See [17] for further reading.

2) *Traffic Participants Relations Model* $p(R|X)$: The vehicle relations model is a conditional distribution over the state of interrelations between traffic participants. We factorize the traffic model into a set of binary relation models $p(R_{i,o}|X_i, X_o)$, each depending only on the state of two traffic participants. The state space of a relation can be of discrete, like the Boolean relation has_right_of_way, or continuous nature, like the difference in heading or a time to collision (ttc) estimate. Most of these relations can be evaluated deterministically from the dependent states.

The relations can be formalized by a human expert and allow to incorporate prior knowledge about the domain, e.g., traffic rules. The calculation of useful relations extracts valuable knowledge from the given states. Consequently, the relations simplify learning the policy model by providing useful features.

3) *Lane Matching Model* $p(L|X)$: The lane matching model is a conditional distribution over all lanes in the map. It assigns the probability that a lane serves as the main spatial guidance for a traffic participant. The conditional distribution is derived from labeled pairs of states and lanes, yielding a model that assigns higher probabilities to lanes if the heading of the traffic participant is in direction of the lane and if the vehicle is not too close to the lane borders.
4) *Following Lane Model* $p(L_{i+1}|L_i)$: Knowing which lane guides the course of a driver provides information about which lane he is probably going to use next. The conditional distribution over the following lanes is modeled by $p(L_{i+1}|L_i)$. We assume that traffic participants usually move along the road network. Thus, following lanes L_{i+1} are only considered if they can be reached from L_i with normal driving maneuvers including lane changes, overtakes or U-turns.

   The resulting joint distribution over lane sequences can be interpreted as a distribution over possible plans. A lane sequence defines a valid route in the road network graph but does not define how it will be driven. The policy model will use this information to predict possible actions of a traffic participant based on his the current context. Possible interactions of traffic participants can be anticipated with this information, e.g., if their routes cross at an intersection.
5) *Lane Relations Model* $p(LR|X, L)$: The lane relations model is defined depending on the state of the traffic participant and his currently most important lane. The lane relations can be deterministically evaluated using the map knowledge. Like before with the traffic participant relations, the lane relations simplify learning task by reducing the complexity.
6) *Policy Model* $p(A|X, R, LC)$: The policy model is the core of the probabilistic formulation of the stochastic process. It takes all information of the situational context into account to define a density over the actions of a traffic participant. Depending on the individual context, i.e. the subjective view on the environment containing every relevant information to choose an appropriate action, the density predicts the action of each traffic participant. The context includes but is not limited to the state of the other traffic participants and the interrelations between them and the road configuration. While each traffic participant reacts on the current state of the environment individually, the actions of the different road users are coupled over time. Their actions affect the state of the environment and consequently the future decisions of the others. In section V we show how this model can be learned from data.
7) *Motion Model* $p(X^{(t+1)}|X^{(t)}, A^{(t)})$: The motion model defines the changes in state for all n traffic participants depending on their previous states and the actions they executed. For every traffic participant a simplified linear single track vehicle model is used. The following differential equation defines the motion over time:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\psi} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v\cos\psi \\ v\sin\psi \\ \omega \\ a \end{bmatrix}. \tag{3}$$

V. Learning the Context-Dependent Policy Model

Defining the policy model that reflects the decision making of a traffic participant to predict his next action is the most challenging part in the definition of the Bayesian model. Our approach is to learn this model from data and support the learning process by providing all information necessary for the learning algorithm to find the dependencies between specific situational aspects and their implications on the action selection.

In this section we first explain an iterative algorithm based on Expectation Maximization (EM) to overcome the difficulty of learning from noisy measurements and hidden variables. Then we explain how to learn a policy model from complete data, which is one step in the EM algorithm.

A. Learning from Incomplete Data with Monte Carlo Expectation Maximization

The data we can obtain are noisy measurements of the traffic participant states containing their location or velocity. Further, the output of the policy model, the resulting action, is not directly measurable. If we knew the true states of all hidden variables, each model could be learned directly, but this is not the case. For this reasons we have to resort to algorithms that can deal with incomplete data. A general approach to learn the conditional dependencies in a Bayesian network with hidden variables is the well-known Expectation-Maximization (EM) algorithm [18]. EM solves this problem by iterating between an Expectation step (E-step) and a Maximization step (M-step). The E-step estimates expectations resulting in distributions over the hidden variables by using the observations and the current model parameters. The M-step then uses these expectations to find new model parameters that maximize or at least increase the expected complete-data log-likelihood. These two steps are repeated until convergence is reached. For our task, in every iteration the EM algorithm learns a new policy model based on the estimates of the hidden variables which are obtained using the current policy model.

The EM framework distinguishes between the observable nodes Z and the hidden nodes Y. Let θ denote the parameters of the Bayesian network and $\theta^{(i)}$ the parameter estimate of the i-th iteration of the EM. The likelihood of observations z given the model parameters θ is

$$p(z\,|\,\theta) = \int_{y\in Y} p(z, y\,|\,\theta)\,dy = \int_{y\in Y} p(z\,|\,y)p(y\,|\,\theta)\,dy. \tag{4}$$

The E-Step estimates the posterior distribution $p(y|z,\theta^{(i)})$ over all hidden states of the Bayesian network using the current parameter estimate $\theta^{(i)}$. In our application y comprises all the hidden states, like e.g., the traffic participant states, their contexts and actions. Since we are dealing with a mixed state space covering discrete and continuous states in addition with non-linear models, there is no closed-form solution for the distribution over y. Similar to particle filtering we use likelihood weighting for approximate inference, which is an instance of importance sampling [13]. Like in particle filters the true posterior is represented by a finite set of samples S.

The inferred posterior distribution over the hidden states is then used to find a new parameter estimate $\theta^{(i+1)}$ in the M-step. The goal is to find the parameters that maximize the expectation of $\log p(z,y|\theta)$ given the distribution over y from the E-step. Since the posterior is represented by a particle distribution, the expectation sums over all particles:

$$\begin{aligned}\theta^{t+1} &= \underset{\theta}{\operatorname{argmax}} \sum_{j=1}^{|S|} \log p\,(z, y^{(j)}\,|\,\theta) \\ &= \underset{\theta}{\operatorname{argmax}} \sum_{j=1}^{|S|} \log p\,(y^{(j)}\,|\,\theta). \end{aligned} \tag{5}$$

This type of EM belongs to the class of Monte Carlo EM due to application of a Monte Carlo method to infer the posterior. More details on Monte Carlo EM methodology can be found in [19]. The M-step that is described in the next section requires that we find a maximum log likelihood estimate given the samples inferred in the E-step. Since (5) has no closed-form solution we resort to the general case, which is also known as generalized EM (GEM) [20]. In this form only new parameters which increase the expected log likelihood have to be found.

B. Learning the Policy Model

In the M-step of the EM a posterior distribution for the hidden states is given. Thus, we can directly learn the policy model, which is a conditional density function. We represent it as a conditional bivariate normal distribution:

$$p\,(A\,|\,x,r,lr) = p\,(A\,|\,c) \sim N(\mu_c,\Sigma_c).$$

X, R and LR are combined to the context C of a traffic participant to simplify the notation. Both the mean μ_c and the covariance Σ_c depend on the context c. That dependency is expressed by two functions $\mu_c = M(c)$ and $\Sigma_c = K(c)$. Since Σ_C is not constant this function is heteroscedastic and can express the varying uncertainty in the choice of action depending on the situation.

We learn these two functions in a two step manner. First we learn the mean function $M(c)$ by minimizing the mean squared error of the data (samples from the E-step). In the second step, we learn the covariance function $K(c)$ by maximizing the likelihood of the transformed data. By subtracting the resulting values of the learned mean function from the target values of the data samples, we obtain the transformed data. The result of this procedure is a new estimate of the model which is then used in the next iteration of the EM.

We use random forests [14] for learning both the mean and the covariance function. We chose random forest for the learning task since they unite several beneficial properties. They perform significantly better than a single decision tree and show good generalization properties which are comparable to other state of the art methods. They are easy to construct and the induction methods have only few parameters to tune. Since the trees are constructed in a non-parametric way, the complexity of the model is data-driven thereby reducing the problem of over- and underfitting. What makes them especially interesting for our application is that they can be constructed sequentially [21]. This enables the potential optimization of the models online when new data becomes available.

To learn a meaningful action policy model another aspect needs to be considered. The Bayesian model as presented so far does not directly couple the routes of the traffic participants with the actions. While the planned route is provided as input to the action policy model, it is up to model to ensure that actions are derived that respect the semantics of the route policy model, i.e. those actions are most likely that lead to trajectories that follow the route. In order to supply this information to the learning algorithm and thereby enabling it to learn the relationship between routes and actions, we introduce an additional feedback loop into the Bayesian model. The feedback loop is realized through an additional binary random variable C_R that indicates if an action at time step t yields a traffic participant state at $t+1$ that follows the route planned at t. Together with a virtual evidence of *true* that is set on the node during the learning phase, it is guaranteed that only state sequences have high probabilities where the actions are realizations of the planned routes. The used constraint model looks as follows:

$$p\,(C_R\,|\,l, l_1^-, l_2^-, \ldots, l_h^-) = \begin{cases} \text{true: } 1-\epsilon & \text{if } \bigvee_{i=1}^{h} l = l_i^- \\ \text{false: } \epsilon & \text{else} \end{cases},$$

where the parameter $0 \le \varepsilon \le 1$ denotes the probability that the traffic participant does not follow its planned route. As a consequence, if actions and planned routes are not consistent, the corresponding particles are down-weighted in the E-step. Since consistent state sequences become more likely, the optimization in the M-step yields an action policy model that will produce actions that are increasingly consistent with each EM iteration. Note, that the feedback loop is only needed to guide the learning process. Once learned, the constraint becomes part of the action policy model.

VI. Evaluation

To evaluate the presented approach we learned a model from traffic observations at an intersection and compared it to a

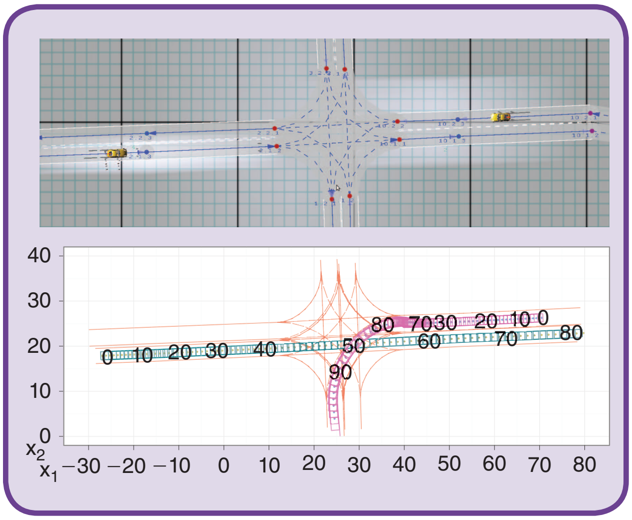

FIG 3 Example of an episode used for training. The trajectories of two cars pass the intersection. The numbers indicate the time indexes. The car coming from the right waits at the intersection until the other car with right of way has passed and then makes a left turn.

standard Bayesian filter that uses no contextual information, but relies solely on the vehicle kinematics and dynamics to predict the future motions of traffic participants.

A. Settings

The traffic scenario we consider is an unsignaled 4-way intersection where left yields to right. Two vehicles are crossing the intersection and act according to traffic regulations. To acquire the data needed for learning and testing we recorded 100 episodes in a simulation environment[1]. The simulated cars are controlled by the autonomous driving software that was developed by team *AnnieWAY* for the *DARPA Urban Challenge* [22]. A screenshot of the intersection scenario and an example episode is shown in Figure 3. The vehicles show varying motion patterns throughout the episodes depending on their randomized starting states, the routes they take over the intersection and the behavior of the other car. There is also noise in the behavior of the traffic participants, which leads to variations in driving speed, deviations from the centerlines of lanes, variations in where and how they stop at intersections and in seldom cases even to irrational behavior. An underlying detailed vehicle physics simulation assures realistic vehicle dynamics. The example episode depicted in Figure 3 shows a typical situation where the context needs to be considered to predict the behaviors of the drivers correctly. The car coming from the right stops at the intersection ($t = 30$) to let the other car pass before it make a left turn. Approaches that solely rely on the individual motion histories are not able to make accurate predictions in these kinds of situations.

The episodes were recorded with a frame rate of 3:33 resulting in a time step of $\triangle t = 0.3\,\mathrm{s}$. To learn the policy model as presented in section V-B we used 80 of the recorded episodes and used the remaining 20 episodes to evaluate the test performance of the Bayesian model in terms of data log likelihood and prediction accuracy. Before training, we added white noise to the training data with a standard deviation of $(\sigma_{x_1}, \sigma_{x_2}, \sigma_v, \sigma_\psi) = (0.1\,\mathrm{m}, 0.1\,\mathrm{m}, 0.1\,\mathrm{m/s}, 0{,}05\,\mathrm{rad})$ to simulate measurement errors. The statistics of the E-step were infered with likelihood weighting and the random forests trained in the M-step consisted of ensembles of 30 trees. For the random forests we trained ensembles of regression trees that employ more complex basis and split functions to improve the expressiveness and generalization of the model. We used linear oblique splits [23] and linear basis functions [24].

B. Results

We compare our approach using the learned model against a standard Bayesian filter model which is widely used in tracking applications using a linear single track vehicle model for each traffic participant. This model keeps the current velocity and heading with some Gaussian noise. The noise is estimated directly from the control data of the ground truth.

Table 1 compares the root mean squared error (rmse) of the predicted vehicle positions compared to the ground truth for several prediction horizons $\triangle t$. As can be seen, the learned model makes significantly better predictions. The difference in accuracy becomes even more apparent in terms of the data log likelihood (4), which is a more expressive measure for the goodness-of-fit of a generative model. The mean log likelihood is averaged over all time steps of all test episodes. The trained model reaches a mean log likelihood of 2:032 while the single track model only reaches a value of 0:988, i.e. the observations are predicted with an almost 3-times higher likelihood.

The meaning of these values for actual predictions can be seen in visualizations of examples of anticipated situations. The following Figures (4, 5) show predictions for different test episodes at different points in time. The prediction horizon is set to 20 steps in the future, which equals a period of 6 seconds. The densities of the predicted states are depicted in different colors to indicate the time (see Figure 4). The orange dots behind the cars show their smoothed position estimates for past time steps. The predicted routes are also visualized through highlighted lane boarders. The inference computes joint predictions for all cars. For visual clarity, however, the figures do not show which prediction of one car belongs to which prediction of the other car.

Figure 4 illustrates the prediction for a situation where a car (*car 2*) coming from the bottom has right of way over a car coming from the left (*car 1*). It can be seen that the learning algorithm successfully managed to learn the lane

[1]The used simulation environment builds upon the driving simulation *VDrift*, the real-time physics simulation *Bullet* and the 3D graphics toolkit *OpenSceneGraph*.

following behavior of the drivers. The predicted trajectories stay in the lanes and do not diverge over time. The position uncertainty grows mainly in longitudinal direction of the lanes due to the possible changes in speed. Since the map knowledge only serves as source of information in the Bayesian model but not as a hard constraint, a broader and more realistic set of trajectories is derived.

The second observation is that the drivers' different choices at the intersection lead to multiple modes in the predicted distributions. The second car has the right of way in this example. It is realistically predicted that *car 1* stops in front of the intersection while *car 2* takes one of the possible ways over the intersection without stopping. The observed driving behavior and the current heading of *car 2* leads the Bayesian model to the conclusion that not all modes are equally likely. At this point, the driver will rather make a left turn or drive straight than make a right turn.

The prediction that *car 1* will stop at the intersection is only possible through the consideration of the whole situational context. It is necessary to take all other traffic participants and their roles into account to determine the right interpretation and anticipate consequentially that the car will most likely stop at the intersection. In addition the model does not only conclude that *car 1* is going to stop but also provides an estimate of the distribution over where and how it is going to stop.

The influence of the situational context becomes even more evident in the example shown in Figure 5. Again, one can see how the possible decisions of *car 2* lead to different clusters in the density of the predicted states. An interesting aspect is, that the learner implicitly learned the effects of traffic rules on the resulting actions of traffic participants just by observing traffic episodes. For this evaluation they weren't provided through relations in the context model.

VII. Application for Decision Making

Using the presented Bayesian model, the evolution of traffic situations can be accurately predicted several seconds into the future. This information can be used directly, e.g., to evaluate an advanced TTC (Time-To-Collision) estimate or to assess the safety of planned driving paths or trajectories [25]. It could also be the input for hand-build decision rules like '*Cross the intersection, if the way is clear for the next 5 s with a probability greater than* 0:95'.

Table 1. Prediction root mean squared error (rmse) and mean log likelihood (mll).

Model	Pos rmse ($\triangle t$: 0.3 s)	Pos rmse ($\triangle t$: 0.3 s)	mll
Learned model	0.19 m	6.48 m	2.032
Single track model	0.28 m	12.83 m	0.988

The policy model could also be applied to learn reactions to situations directly from observations. The learned policy would then imitate the behavior of human drivers. However, the full potential of this knowledge about the future can only be released when using planning and optimization methods. They can optimize the reactions with regard to specifically defined objectives like safety or economy, which leads to better behaviors and generalization.

On lower abstraction levels, motion planning can be used to find suitable trajectories to target locations. Spatial and temporal constraints ensure that these trajectories are drivable, safe and efficient. Typical tasks are finding optimal trajectories for emergency breaking, evasive maneuvers or merging into moving traffic. On a higher level, decision making chooses the best tactical maneuver in a specific situation. The decision between crossing or waiting at an intersection or the decision if and when a lane change should be performed in highway driving are typical examples for this abstraction level.

The Bayesian model presented in this paper is especially useful for high-level decision making. In this case, the reactions of other traffic participants are of special interest because they substantially influence the long-term situation development and often underlie stochasticity.

A. Planning Under Uncertainty

Sequential decision making in domains with uncertain development over time can be modeled as a Markov Decision Process (MDP) [26]. In MDPs the state transition is modeled as a conditional probability distribution in the process model. The agent can influence the process by choosing between actions. Solving an MDP means optimizing the choice of action in order to maximize a reward function over time. The reward function defines the objectives of the agent. In the context of autonomous driving and ADAS, the reward function can be defined to balance safety, efficiency, comfort, compliance and progress.

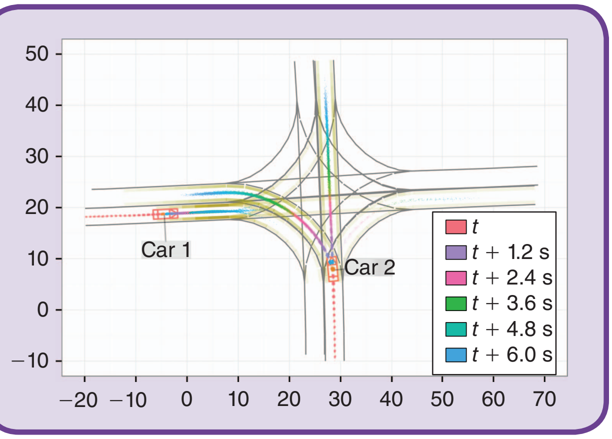

FIG 4 Prediction of a situation at an intersection over a period of 6 s with a policy model learned from traffic observations.

Brechtel et al. show in [1] how a hierarchical Bayesian model of continuous nature can be embedded in an MDP as the transition model. By solving this MDP, optimal decisions for an autonomous car in a highway scenario (including oncoming traffic) with multiple other cars are automatically derived. The actions allow the car to control its speed and change lanes to overtake other vehicles in order to optimize safety and progress. The evaluation showed that sophisticated behavior predictions, e.g. lane change prediction, are essential for good decisions.

While this approach is very general because it uses the natural continuous space, it relies on the assumption that the state is fully observable. In reality, however, cars can only incompletely perceive the environment. Sensors are limited in range, field of view and precision. Even more important, their view is often occluded by other vehicles, houses or other objects. This aspect is highly important for decision making and must not be neglected.

MDPs can be extended to Partially Observable MDPs (POMDP) to fully meet these requirements. In POMDPs a measurement model describes the relationship between states and observations. Ulbrich et al. [27] show how a POMDP with a small set of manually defined symbols can be used to derive lane change decisions in a constrained highway-driving setting. Recent advances in the field of solving continuous state and observation POMDPs allow to directly use learned continuous Bayesian models [28].

B. Deriving a POMDP Model

One of the key factors for safe and efficient driving is the ability to anticipate and assess the consequences of driving decisions. The prediction ability of the Bayesian model presented in this paper is the ideal basis for the process model in (PO)MDPs. An overview of a possible embedding of the Bayesian model into a POMDP is depicted in Figure 6. A situation consists of an agent represented by X_e and the other traffic participants $X_1, \ldots, X_n$ which are subsumed by S.

The process model of the POMDP can be derived from the presented Bayesian Model, by adding a discrete set of actions A_e the ego vehicle can chose from. The influence of actions is described by $p(X_e^{(t+1)}|S^{(t)}, X_e^{(t)}, A_e)$. The other traffic participants are covered by the original Bayesian model $p(S^{(t+1)}|S^{(t)}, X_e^{(t)}, A_e)$. Since the ego vehicle is part of their context, it influences their behaviors as they react to it. This way, cooperative driving of others can be anticipated and considered in the planning process.

The measurement model presented in Section IV-C1 accounts for the sensors' inaccuracies. In addition to this, the POMDP's measurement model needs to account for more complex sensor limitations to obtain policies with information gain strategies. These should include the limited range of the sensors as well as occlusions caused by objects.

VIII. Conclusion and Future Work

In this paper a Bayesian model for estimating and predicting traffic situations is presented. A key element of the Bayesian model is the context-dependent policy model that predicts the behavior of traffic participant based on contextual information. Such a model is hard to define manually. We presented an approach to learn it automatically from unlabeled traffic observations. Additionally, we outlined how the Bayesian model can be used for decision making in the traffic domain. We show how to build a continuous POMDPs using the learned predictive model in order to fully utilize it's capabilities for decision making under uncertainty. This very general POMDP approach can be simplified for decision making with MDPs or motion planning.

Experiments with simulated data showed a significant improvement in estimation and prediction accuracy of the learning approach over a Bayesian filter using a standard single-track model with uncertainty. As a consequence, the presented model can cope better with noisy sensors and uphold a valid estimation, even if traffic participants are occluded for longer periods of time. Additionally it allows more precise long-term predictions without neglecting the uncertainty. We argue that the capabilities of this novel model learning approach are especially useful when combined with a long-term decision making algorithm that is capable of accounting for uncertainties in the process as well as in the measurements. This way, a very powerful and generally applicable decision making approach can be created, which minimizes the need for manually designed models.

An interesting direction to explore is the potential online learning capability. This would enable cars to learn and adapt their models to new situations and changes while they are driving. It is also potentially beneficial to integrate the *intentions* of traffic participants into the Bayesian model, e.g., whether they want to take an exit on a highway. Finally, it would be very interesting to compare the generalization capabilities of other learning algorithms from the field of Deep Learning or Gaussian Processes to the presented random forest approach in the context of the presented framework.

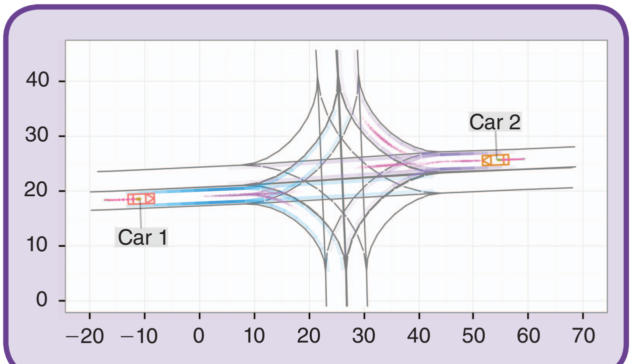

FIG 5 Prediction at $t + 6$ s and without intermediate steps. The model has correctly learned the different modes correctly.

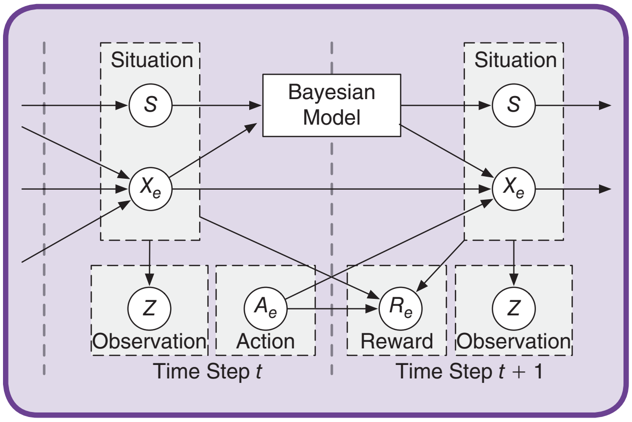

FIG 6 Embedding of the Bayesian model into a Partially Observable Markov Decision Process (POMDP). The ego vehicle X_e can be directly controlled through actions A_e.

Acknowledgment

The authors gratefully acknowledge the support of the *Deutsche Forschungsgemeinschaft*.

About the Authors

Tobias Gindele received his Diploma degree in Computer Science in 2007 at the University of Karlsruhe, Germany. He began his Ph.D. candidature at the Institute of Anthropomatics and Robotics at the Karlsruhe Institute of Technology after participating at the Darpa Urban Challenge as member of Team AnnieWAY. His research interests are machine learning, statistical modeling and decision making.

Sebastian Brechtel received his Diploma degree in Computer Science in 2009 at the University of Karlsruhe, Germany. He then began his Ph.D. candidature at the Institute of Anthropomatics and Robotics at the Karlsruhe Institute of Technology. His research interests are state estimation and decision making under uncertainty as well as machine learning.

Rüdiger Dillmann is full Professor at the Karlsruhe Institute of Technology, Institute for Anthropomatics, Karlsruhe, Germany. His research interests include robot programming by demonstration, machine learning, robot vision, humanoid robotics, cognitive cars, and medical applications of informatics.

References

[1] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic MDP-behavior planning for cars," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, Washington, D.C., 2011, pp. 1537–1542.
[2] S. Algers, E. Bernauer, M. Boero, L. Breheret, C. di Taranto, M. Dougherty, K. Fox, and J.-F. Gabard, "Review of micro-simulation models," Inst. Transport Studies, Univ. Leeds, U.K., 1997.
[3] I. Dagli, M. Brost, and G. Breuel, *Action Recognition and Prediction for Driver Assistance Systems Using Dynamic Belief Networks* (Lecture Notes in Computer Science), 2003, pp. 179–194.
[4] I. Dagli and D. Reichardt, "Motivation-based approach to behavior prediction," in *Proc. IEEE Intelligent Vehicles Symp.*, 2002, pp. 227–233.
[5] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, Madeira, Portugal, 2010, pp. 1625–1631.
[6] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Estimation of multivehicle dynamics by considering contextual information," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 855–870, 2012.
[7] A. Alin, M. V. Butz, and J. Fritsch, "Incorporating environmental knowledge into Bayesian filtering using attractor functions," in *Proc. IEEE Intelligent Vehicles Symp.*, 2012, pp. 476–481.
[8] D. Meyer-Delius, C. Plagemann, G. von Wichert, W. Feiten, G. Lawitzky, and W. Burgard, "A probabilistic relational model for characterizing Situations in dynamic multi-agent systems," in *Proc. 31st Annu. Conf. Data Analysis, Machine Learning Applications*, 2007.
[9] S. Bonnin, F. Kummert, and J. Schmudderich, "A generic concept of a system for predicting driving behaviors," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, 2012, pp. 1803–1808.
[10] M. G. Ortiz, J. Schmudderich, F. Kummert, and A. Gepperth, "Situation-specific learning for ego-vehicle behavior prediction systems," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, 2011, pp. 1237–1242.
[11] J. Zhang and B. Rössler, "Situation analysis and adaptive risk assessment for intersection safety systems in advanced assisted driving," in *Proc. 21st Fachgespräch Autonome Mobile Systeme*, 2009, pp. 249–258.
[12] S. Lefèvre, J. Ibanez-Guzman, and C. Laugier, "Context-based estimation of driver intent at road intersections," in *Proc. IEEE Symp. Computational Intelligence Vehicles Transportation Systems*, 2011, pp. 67–72.
[13] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA: MIT Press, 2009.
[14] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[15] D. Koller and A. Pfeffer, "Object-oriented Bayesian networks," in *Proc. 13th Conf. Uncertainty Artificial Intelligence*, 1997, pp. 302–313.
[16] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Morgan Kaufmann, 1988.
[17] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.
[18] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. Series B (Methodol.)*, vol. 39, no. 1, pp. 1–38, 1977.
[19] R. A. Levine and G. Casella, "Implementations of the Monte Carlo EM algorithm," *J. Comput. Graph. Stat.*, vol. 10, no. 3, pp. 422–439, 2001.
[20] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," *Learn. Graph. Models*, vol. 89, pp. 355–368, Feb. 1998.
[21] E. Ikonomovska, "Algorithms for learning regression trees and ensembles on evolving data streams," Ph.D. dissertation, Jozef Stefan International Postgraduate School, Ljubljana, Slovenia, Oct. 2012.
[22] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schröder, M. Thuy, M. Goebl, F. von Hundelshausen, O. Pink, C. Frese, and C. Stiller, "Team AnnieWAY's autonomous system for the DARPA Urban Challenge 2007," *Int. J. Field Robot. Res.*, vol. 25, no. 9, pp. 615–639, Sept. 2008.
[23] S. Murthy, S. Kasif, and S. Salzberg, "A system for induction of oblique decision trees," *J. Artif. Intell. Res.*, vol. 2, no. 1, pp. 1–32, 1994.
[24] D. Potts, "Incremental learning of linear model trees," in *Proc. 21st Int. Conf. Machine Learning*, 2004, p. 84.
[25] M. Althoff, D. Althoff, D. Wollherr, and M. Buss, "Safety verification of autonomous vehicles for coordinated evasive maneuvers," in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, 2010, pp. 1078–1083.
[26] R. Bellman, "A markovian decision process," DTIC Document, Tech. Rep., 1957.
[27] S. Ulbrich and M. Maurer, "Probabilistic online POMDP decision making for lane changes in fully automated driving," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems*, 2013, pp. 2063–2070.
[28] S. Brechtel, T. Gindele, and R. Dillmann, "Solving continuous POMDPs: Value iteration with incremental learning of an efficient space representation," in *Proc. Int. Conf. Machine Learning*, Atlanta, GA, 2013, pp. 370–378.

ITS