Situation specific learning for an ego-vehicle behavior prediction scenario

Michael Ortiz, Jens Schmüdderich, Franz Kummert, Alexander Gepperth

2011

Preprint:

This is an accepted article published in IEEE Intelligent Transportation Systems Conference (ITSC). The final authenticated version is available online at: https://doi.org/[DOI not available]

2011 14th International IEEE Conference on
Intelligent Transportation Systems
Washington, DC, USA. October 5-7, 2011

Situation-specific learning for ego-vehicle behavior prediction systems

Michaël Garcia Ortiz, Jens Schmüdderich, Franz Kummert and Alexander Gepperth

***Abstract*— We present a system able to predict the future behavior of the ego-vehicle in an inner-city environment. Our system learns the mapping between the current perceived scene (information about the ego-vehicle and the preceding vehicle, as well as information about the possible traffic lights) and the future driving behavior of the ego-vehicle. We improve the prediction accuracy by estimating the prediction confidence and by discarding unconfident samples. The behavior of the driver is represented as a sequence of elementary states termed *behavior primitives*. These behavior primitives are abstractions from the raw actuator states. Behavior prediction is therefore considered to be a multi-class learning problem.**

In this contribution, we explore the possibilities of situation-specific learning. We show that decomposing the perceived complex situation into a combination of simpler ones, each of them with a dedicated prediction, allows the system to reach a performance equivalent to a system without situation-specificity. We believe that this is advantageous for the scalability of the approach to the number of possible situations that the driver will encounter. The system is tested on a real world scenario, using streams recorded in inner-city scenes. The prediction is evaluated for a prediction horizon of 3s into the future, and the quality of the prediction is measured using established evaluation methods.

I. Introduction

The perception of the environment has improved tremendously in the past few years: sensors are of better quality and advanced sensory processing techniques in inner-city driving can emerge. As systems are now able to perceive (to a certain extent) their environment, we focus our work on how to use the information extracted from the scene in order to predict the future behavior of the driver.

In this contribution, we compare several behavior prediction systems based on learning from experience. The first system is trained using all the information available, the second system trains one subsystem per specific situation, and the third system is an extension of the second system, where the behavior prediction is also applied to other traffic participants in order to improve the quality of the prediction.

We show that learning the prediction for complex scenes is not needed: decomposing these scenes and predicting the behavior for simpler situations is sufficient. We can reach the same quality, given that the system is correctly designed. We believe that situation-specificity can guarantee scalability, because low-dimension situations are easier to learn and the system is faster to converge compared to higher-dimension complex situations. Moreover, it becomes possible to predict a situation that has not been encountered by combining already learned simpler situations. In addition to showing the validity of situation-specific learning, we demonstrate that the knowledge acquired by the ego-vehicle can be applied to predict the future behavior of other traffic participants.

M. Garcia Ortiz is with the CoR-Lab, Bielefeld University, Bielefeld, Germany michael.garcia.ortiz@gmail.com

F. Kummert is with the Faculty of Technology, Bielefeld University, Bielefeld, Germany Franz@techfak.uni-bielefeld.de

A. Gepperth and J. Schmüdderich are with Honda Research Institute Europe GmbH, Offenbach am Main, Germany firstname.lastname@honda-ri.de

II. Related work

Recent developments in the area of behavior prediction for Advanced Driving Assistant Systems (ADAS) show that more and more approaches go in the direction of using learning techniques ([1], [2] and [3]). One reason for this is the achieved reduction of design effort, especially when scaling systems to inherently complex scenarios such as inner-city traffic. The price to pay for this is an increase in initial design effort for setting up of learning methods and collecting training data. In the context of driver behavior prediction, several systems circumvent the learning issue by using designed models to estimate the behavior or trajectory of the ego-vehicle ([4], [5]). We believe that for behavior prediction, learning approaches must be used at some point because the number of situations or behaviors in complex environments will become too big to cope with manually. It is our conviction that learning will cope with the complexity of the task, and also greatly reduce the overall design effort.

Our approach has some similarities to other systems already proposed. In [6], complex behaviors are segmented into a sequence of basic elements, and this abstraction from raw actuator states is presented as a necessary feature for driver behavior prediction. In [7], the traffic situations are decomposed into analyzable subsets called Situation Aspects. The situation-specific learning approach can be compared to this situation-aspect decomposition.

In [1], the authors derive an estimate of driver intent, which amounts to predicting the probability of an imminent lane change. In contrast to our approach, inputs to the learning algorithms are high-dimensional, since they include present *and* past positions and speeds of the ego-vehicle. Furthermore, an explicit measure of driver state using face monitoring is used, which is not done in the present study. Results show that ego-vehicle lane changes on highways can be predicted up to 3s in advance with good accuracy. Our approach differs in the sense that we limit our input space to dimensions that are also observable in other traffic participants (speed, acceleration, distances between traffic participants), in order to apply what has been learned for the ego-vehicle to other vehicles.

III. Scenario

We put our focus on the inner-city traffic, and in this study we restrict the situations to:

- "traffic light situation" : the ego-vehicle is approaching a traffic light
- "preceding vehicle situation": the ego-vehicle is following a vehicle driving in the same direction

As can be seen in Fig. 1, these situations can overlap (the ego-vehicle can follow the preceding vehicle which approaches a traffic light).

Fig. 1. Example of inner-city traffic light approach scene: the ego-vehicle behavior can be a reaction to the traffic light or to the preceding vehicle.

IV. Data used for behavior prediction

A. Segmentation into behavior primitives

We consider that raw actuator states or even trajectories of the ego-vehicle are not easily predictable. They depend on multiple factors that are not always determinable (like, e.g., the characteristics of the car or the stress level of the driver). Two different drivers performing the same behavior can have different trajectories. As an example, two drivers approaching a red traffic light will both brake, but their exact trajectories will differ. Thus, predicting the exact actuator states even considering high error margins is a very difficult task. Therefore, we describe the driver behavior by abstracting from these driver-specific quantities, using a set of standard elementary behaviors that we call *behavior primitives*.

The decomposition of the sequence of vehicle states (e.g., speed, acceleration, pedal status) in a sequence of behavior primitives is done using heuristics, segmenting portions of trajectories over time using data coming from the CAN-bus. Since this contribution focuses on the behavior on straight roads, we limit the behavior primitives to the longitudinal dimension: "accelerating", "decelerating", "keeping speed" and "stopped".

In order to extract the behavior primitives that describe the behavior at time t_0, we measure the mean acceleration $\hat{\mathrm{A}}\mathrm{cc}(t_0)$ using a temporal window ΔT around t_0:

$$\widehat{Acc}_{\Delta T}(t_0) = \frac{1}{\Delta T} \int_{t_0 - \frac{\Delta T}{2}}^{t_0 + \frac{\Delta T}{2}} \mathrm{Acc(t)}\, \mathrm{d}t \qquad (1)$$

where $\mathrm{Acc}(t)$ is the acceleration of the vehicle at time t. We then use this measure in order to determine the Behavior Primitive at time t_0:

- "accelerating" corresponds to $\widehat{Acc}_{\Delta T}(t_0) > \tau_{acc}$
- "decelerating" corresponds to $\widehat{Acc}_{\Delta T}(t_0) < \tau_{dec}$
- "stopped" corresponds to a speed $V(t) < V_{stopped}$
- "keeping speed" corresponds to the default case, when the behavior is none of the three others.

This approach allows us to filter the small variations in acceleration and to consider them as a "keeping speed" behavior. In this study, we use: $\Delta T = 1s$, $\tau_{acc} = 0.03$ $m \cdot s^{-2}$,$\tau_{dec} = -0.05\ m \cdot s^{-2}$ and $V_{stopped} = 1\ m \cdot s^{-1}$.

B. Extraction of the preceding vehicle using laser data

Our experimental vehicle uses two ibeoLUX sensors mounted left and right under the front bumper of the experimental vehicle. Data from the sensors are integrated into a binary, metric "laser image" where filled pixels indicate the presence of a laser target (i.e., an obstacle). We use a simple template-based detection approach for horizontal segments in the metric laser image in order to detect vehicles. Tracking is used to stabilize detections and to determine the relative speed of detected vehicles. Parked vehicles are excluded by computing the absolute speed relative to the road, using the known speed of the ego-vehicle. Vehicles coming from the opposite direction are detected, by extracting objects with a negative speed, but they are not used in this work.

C. Encoding of the situation and behavior representation

The input data for the prediction are restricted to speed and acceleration of the ego-vehicle, distance and status of the possible traffic light, distance and speed of the possible preceding vehicle, and distance between the preceding vehicle and the traffic light, when both are present in the scene.

We compute the behavior primitive for each sample of this dataset in an offline fashion according to the procedure described in Sec. IV-A. It is encoded as a 4-element binary array, one element for each possible behavior primitive.

As we have not yet implemented robust algorithms for detecting traffic lights, we manually annotated the presence and the status (green, yellow, or red) of traffic lights based on the image data. In order to estimate the distance to the traffic light, we extract the moment when the ego-vehicle crosses the traffic light, the distance is then 0. We then calculate past distances to the traffic light by integrating the speed of the ego-vehicle, obtained from the CAN bus. We compute the distance to the traffic light and the status of the traffic light for each sample of the dataset. The distance is encoded in a single real number, whereas the status of the traffic light is encoded into a 3-dimensional binary array, each element corresponding to one possible status of the traffic light (green, yellow, red).

The speed and distance of the preceding vehicle is obtained processing laser data (see Sec. IV-B). They are encoded in two real numbers. When possible, the distance between the traffic light and the preceding vehicle is computed using the distance between the traffic light and the ego-vehicle, and the distance between the preceding vehicle and the ego-vehicle. An estimate of the acceleration of the preceding vehicle is obtained by differentiating its speed.

V. Methods

We predict the future behavior primitives ahead in time depending on the current ego-vehicle status (speed and acceleration), and the scene representation (preceding vehicle and traffic light).

A. Learning and Prediction strategy

The behavior prediction system performs a mapping between the situation representation at time t, and the future behavior primitive at time $t + T_{pred}$. Our mid-term goal is to perform learning and prediction in a running system. This would imply that we train a learning algorithm, for a given time t and a time scale T_{pred}, to represent the relationship between the situation at $t - T_{pred}$ and the behavior primitive at t, since we cannot look into the future. After convergence, the trained algorithm is used to predict the behavior primitive at time $t + T_{pred}$ using the situation representation at time t. This process is illustrated in Fig. 2.

For our current evaluation, we perform the learning and the prediction in an offline fashion where data is stored prior to training and evaluation. As "looking into the future" is thus possible, we do not need to manage a memory of the events, which simplifies the learning task while not influencing the performance of the system. For the online implementation, the system would have to wait for the convergence of the learning before performing behavior prediction.

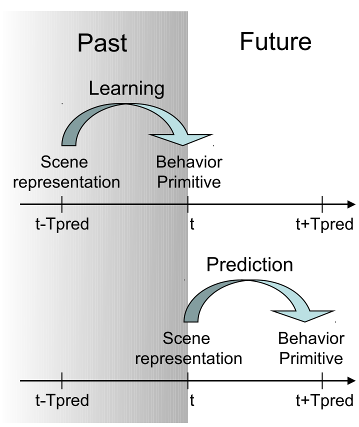

Fig. 2. Learning paradigm: the learning mechanism maps the past situation representation (at time $t - T_{pred}$) to the present behavior primitive (at time t). Then it predicts the future behavior primitive (at time $t + T_{pred}$) using the present situation representation (at time t).

B. Overview of the systems

In order to evaluate the quality and advantages of the situation-specific learning, we created 4 different behavior prediction systems that are trained according to V-A.

1) Baseline: The prediction of the baseline system uses the dynamic characteristics of the ego-vehicle (speed and acceleration) at time t to predict the future driver behavior (behavior primitive) at time $t + T_{pred}$.

2) Full learning: The Full learning system is a naive system which uses, as an input, the speed and distance of the possible preceding vehicle, the status and distance of the possible traffic light, and the speed and acceleration of the ego-vehicle. It learns the mapping between this scene representation at time t and the future behavior primitive at time $t + T_{pred}$.

3) Situation-specific learning: The situation-specific learning system presented in this contribution and illustrated in Fig. 3 is first composed of one *situation prioritization*, which analyzes the scene and triggers *situation-specific learning modules*. The strategy of the situation prioritization has been kept simple in this contribution:

- if there is a preceding vehicle and no traffic light, then the situation-specific learning module "preceding vehicle" is activated.
- if there is a traffic light and no preceding vehicle, then the situation-specific learning module "traffic light approach" is activated.
- if there is a traffic light and a preceding vehicle, then the situation-specific learning module which corresponds to the nearer traffic participant is activated.
- if there is no traffic light and no preceding vehicle, then no situation-specific learning module is activated, and no prediction is performed.

The situation-specific learning module "preceding vehicle" uses the speed and distance of the preceding vehicle as well as the speed and acceleration of the ego-vehicle as an input. In the same way, the situation-specific learning module "traffic light approach" uses the status and distance of the traffic light and the speed and acceleration of the ego-vehicle as an input. The triggered situation-specific module learns the mapping between the specific scene representation at time t and the future behavior primitive at time $t+T_{pred}$. Once the mapping is learned, these situation-specific modules predict the future behavior of the ego-vehicle.

Finally, one *fusion of predictions* module selects which prediction is relevant depending on the scene. In this contribution, the fusion of prediction is also very simple, since two situation-specific learning modules can not be activated together. We take the output of the activated situation-specific module as the output of the predictions.

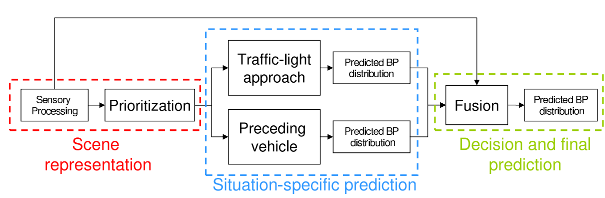

Fig. 3. Overview of the situation-specific learning.

4) Advanced situation-specific learning: The advanced situation-specific learning system (see Fig. 4) follows the same principle as the situation-specific learning presented previously. Additionally, when the preceding vehicle is approaching a traffic light, we apply the already trained "traffic light approach" module to predict the future behavior of the preceding vehicle. We use the distance between the preceding vehicle and the traffic light, the status of the traffic light, and the speed and acceleration of the preceding vehicle to obtain an estimate of the preceding vehicle future behavior at time $t+T_{pred}$. The result of this prediction is used as an additional input for the "preceding vehicle" module, which then uses the predicted behavior of the preceding vehicle, the distance and speed of the preceding vehicle, and the acceleration and speed of the ego-vehicle at time t, in order to predict the behavior of the driver at time $t+T_{pred}$.

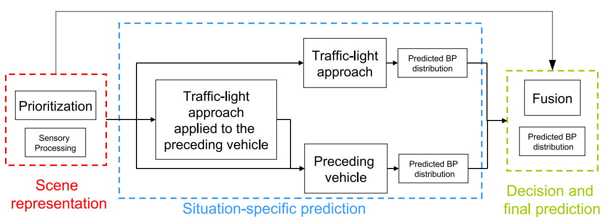

Fig. 4. Overview of the advanced situation-specific learning.

C. Multilayer Perceptron for Behavior Prediction

In order to learn the mapping between the current situation representation and the future behavior primitives, we use a multi-layer perceptron (MLP). MLP is a generic and simple method, which can scale to a wide range of problems, and can be adapted for online learning. The MLP model [8] is a standard nonparametric regression method using gradient-based learning. It is a rather simple neural model, the only free parameters being the number and size of hidden layers. For network training, we employ the back-propagation algorithm with weight-decay and a momentum term (see, e.g., [9]). We configure the MLP to produce four real-valued outputs $A_{stopped}$, $A_{decelerating}$, $A_{accelerating}$ and $A_{keepingspeed}$ corresponding to the predicted behavior primitives. In order to compensate the different amount of training samples for the four behaviors, we normalize these activations over time to have the same mean and same variance for the evaluation of the quality of the prediction. As we are using offline learning and prediction on recorded data, this operation does not violate causality. In an online learning scenario, normalization would have to be performed using a fixed time window.

We used the pyBrain-library [10] for all described MLP experiments. The MLP training algorithm depends on the learning rate parameter ϵ^{MLP} and the momentum parameter ν^{MLP}. The choice of the learning technique is based on a study of different learning techniques in [11].

VI. Evaluation Measures

A. Prediction confidence assessment

As detailed in Sec. V-C, the results of behavior prediction at time t are four normalized activations of neurons A_i. In order to assess the reliability of the prediction, we derive an estimate of the confidence of this prediction C^{conf} by measuring its variance:

$$C^{\text{conf}} = var(A_i) \tag{2}$$

We can now set a confidence threshold τ^{conf} and determine whether the prediction is reliable or not:

$$\begin{aligned} \text{if } C^{\text{conf}} > \tau^{\text{conf}}&: \text{the prediction is confident} \\ \text{else }&: \text{the prediction is not confident} \end{aligned}$$

The variance of the $\{A_i\}$ is highest when there is a single dominant A_{i^*}, which means that the result of the classification is reliable. In contrast, variance is lowest when all activations are similar; as behavior primitives usually are mutually exclusive, this signals high prediction uncertainty.

This measurement of prediction confidence is important, especially (as we plan to do in the future) when concurrently predicting a large number of behavior primitives. We consider that recognizing uncertain predictions and taking no decisions is preferable to taking wrong decisions.

B. Decision making and error measures

The classification value for any output neuron i is obtained by computing $C_i^{\text{class}} = A_i - \sum_{j \neq i} A_j$. For example:

$$\begin{aligned} C_{\text{decelerating}}^{\text{class}} = A_{decelerating} &- A_{accelerating} \\ - A_{keepingspeed} &- A_{stopped} \end{aligned} \tag{3}$$

We can set a classification threshold τ^{class}, and make a classification decision for each prediction which of course also depends on the prediction confidence measure C^{conf} described in Sec. VI-A:

$$\begin{aligned} &\text{if } C_i^{\text{class}} > \tau^{class} \text{ and } C^{\text{conf}} > \tau^{\text{conf}} : \\ &\quad \text{behavior primitive is predicted} \\ &\text{if } C_i^{\text{class}} \leq \tau^{class} \text{ and } C^{\text{conf}} > \tau^{\text{conf}} : \\ &\quad \text{absence of behavior primitive is predicted} \\ &\text{if } C^{\text{conf}} \leq \tau^{\text{conf}} : \\ &\quad \text{unreliable prediction is rejected} \end{aligned}$$

For each pair of the thresholds τ^{class}, τ^{conf} and for each output neuron i, we compute the detection rate ν_i^{correct}, the false positive rate $\nu_i^{\text{incorrect}}$ and the rejection rate ν_i^{reject}, which are defined as:

$$\begin{aligned} \nu_i^{\text{correct}} &= \frac{\#(\text{reliable correct classifications})}{\#(\text{reliable positive examples})} \\ \nu_i^{\text{incorrect}} &= \frac{\#(\text{reliable incorrect classifications})}{\#(\text{reliable negative examples})} \\ \nu_i^{\text{reject}} &= \frac{\#(\text{rejected examples})}{\#(\text{all examples})} \end{aligned}$$

By varying the classification threshold τ^{class}, a receiver-operator-characteristic (ROC) can be generated. This performance measure is a standard tool in machine learning and has been used to evaluate behavior prediction systems (see, e.g., [12]). In the presented ROCs, we plot the detection rate against the false positive rate, for a fixed value of τ^{conf}.

C. Evaluation procedure

We employ N-fold cross-validation to assess prediction results, splitting the dataset into N subsets, each containing an equal amount of successive samples. We train the system using N-1 subsets and we present the samples from the remaining subset to the trained prediction system. We obtain a sequence of activations for the four output neurons, which we normalize according to Sec. V-C.

We then use the activations from the N evaluation subsets, obtained from the N possible combinations of training and evaluation subsets, in order to evaluate the quality of the prediction over the whole dataset.

VII. Experimental setup

We created a dataset containing 80 scenes, for a total of 50000 samples (image and data) of inner-city driving (see Fig. 1). Approximately 14000 samples are a "preceding vehicle situation", 25000 are a "traffic light situation", and 11000 are a situation including both traffic light and preceding vehicle. As the videos are recorded at 20Hz, this corresponds to 40 minutes of driving. We split this dataset into 6 subsets of roughly 8000 samples, in order to evaluate our systems as described in VI-C, and we train the MLPs configured accordingly to the descriptions in V.

All MLPs have one hidden layer of size 30, and we verified that the results obtained were equivalent from 20 to 50 hidden units. They have 4 output neurons, applying a sigmoid non-linearity for hidden layer and output neurons and a bias neuron for the hidden layer and the output layer. Standard training of the MLP requires 4 *rounds* (gradient steps) before early-stopping [9] occurs (one round is one iteration over the whole dataset). We work with $\epsilon^{\text{MLP}} = 0.01$.

VIII. Experiments and Results

For the following experiments, we set the threshold τ^{conf} to 0. We chose not to discard unconfident prediction, in order to have a fair comparison between the different systems. We verified that the results presented in [13] are still valid: discarding 10% of the most unconfident samples increases the probability of correct detection by 5% on average, for a given probability of false detection of 0.05.

We displayed ROCs for a probability of false detection up to 20%, because probability of false detection higher than 20% is not realistic for real inner-city applications.

The results presented in this section, except for the baseline, were obtained for a prediction horizon of 3s. We verified that the conclusions are also valid for 1s and 2s.

A. Baseline

In order to evaluate the quality of the prediction, we perform a simple prediction from the vehicle state at time t to the behavior primitive at time $t + T_{pred}$. As can be seen in Fig. 5, the quality of the prediction is high for an instantaneous prediction (0s), and it decreases depending on the timescale of prediction.

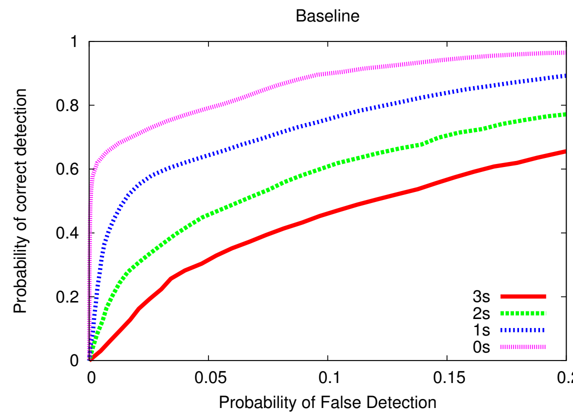

Fig. 5. Results for the baseline.

B. Behavior prediction for the traffic light situation

We first evaluated the situation-specific learning for the traffic light situation. In order to have a fair comparison between the systems, we trained the Full learning system regardless of the situation, and evaluated it only on traffic light situations.

The evaluations of the different systems can be observed in Fig. 6, where ROCs for a timescale of prediction of 3s are displayed. As can be seen, the prediction using all features (Full learning) and the prediction using only features related to the traffic light (situation-specific learning) are equivalent.

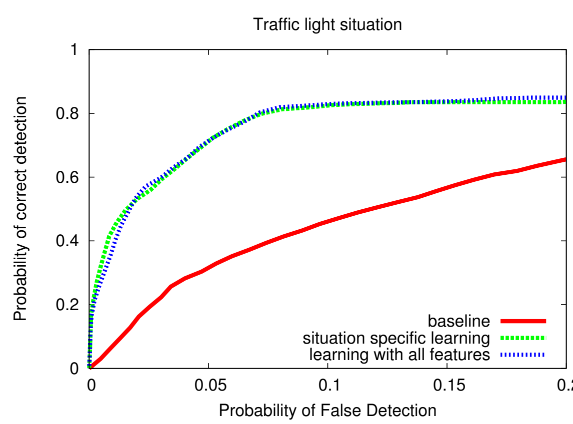

Fig. 6. Result for the "traffic light approach" situation, T_{pred} =3s.

C. Behavior prediction for the "preceding vehicle" situation

We evaluated the situation-specific module for the "preceding vehicle situation", as well as the advanced situation-specific module, which uses the predicted behavior of the preceding vehicle as an additional input. The traffic light module used to predict the future behavior of the preceding vehicle was trained beforehand. We trained the Full learning system regardless of the situation, and evaluated it only on preceding vehicle situation.

The evaluations can be seen in Fig. 7, where ROCs for a timescale of prediction of 3s are displayed. We can observe that the results for the situation-specific module and the baseline are equivalent. We verified that a learning system using only information about the preceding vehicle, without information about the ego-vehicle, reaches the same result. This means that the behavior of the driver in the car-following situation in inner-city is reactive and instantaneous most of the time. We verified this hypothesis by observing the speed and acceleration curves of the preceding and ego-vehicle over time. The comparison between the Full learning and the Advanced situation-specific learning shows that taking into account the prediction of the preceding vehicle behavior (advanced situation-specific learning) improves the prediction quality, which becomes equivalent to the quality of the prediction using all features.

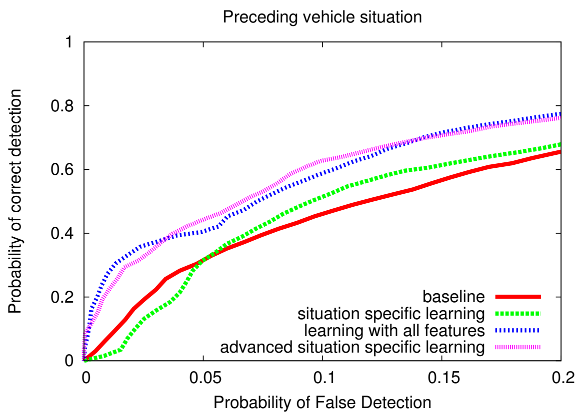

Fig. 7. Result for the "preceding vehicle" situation, T_{pred} =3s.

IX. Conclusion

In this contribution, we proposed several architectures for behavior prediction in the inner-city environment. We showed that it is possible to predict the future behavior of the driver using the current scene information. We presented a system that uses the decomposition of a complex situation into simpler situations: the situation-specific learning.

The complexity of the scene in inner-city traffic can grow very large, because of the number of traffic participants possibly interacting with each other, and influencing the ego-vehicle driver behavior. A system which encounters new situation is not trained to interpret them and thus to predict the future behavior of the driver. However, if we can represent this complex situation with a composition of several simpler situations which have already been encountered, we believe it becomes possible to predict the future behavior of the driver. We showed on a simple scenario that a complex situations can be decomposed into a set of simpler situations without loss of prediction quality. The scalability to more complex situations will have to be demonstrated.

Moreover, we showed that we could use what has been learned from the point of view of the ego-vehicle, and apply it to other traffic participants. This can be applied for scene understanding, and for more advanced predictions. If we can predict what other traffic participants will do, it stands to reason that this will improve the ego-vehicle behavior prediction.

X. Future works

As a future research topic, we want to apply the presented system to highway scenarios. We will investigate whether our approach can be easily transferred to different driving environments. The definition of behavior primitives might have to change, in order to take into account lateral movements for example. We also want to investigate techniques to autonomously extract these behavior primitives.

The benefits of situation-specific learning regarding the scalability will have to be demonstrated by applying this concept to more complex situations.

A further improvement of the current method will be to actively exploit the presence of multiple prediction timescales, which might be used for stabilizing the prediction. We also plan to add to the current system, which predicts behavior primitives (i.e., states), the prediction of *changes* of states, in order to improve the overall prediction quality.

Concerning the possible applications of such a behavior prediction system, knowledge about the future behavior of the driver can be used to detect a dangerous behavior. Another possible application is the use of this prediction to anticipate and start early braking of the car. We expect such a system to help reduce energy consumption.

References

[1] B. Morris, A. Doshi, and M. Trivedi, "Lane change intent prediction for driver assistance: On-road design and evaluation," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2011.

[2] J. Maye, R. Triebel, L. Spinello, and R. Siegwart, "Bayesian on-line learning of driving behaviors," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[3] M. Eilers and C. Mobus, "Learning the human longitudinal control behavior with a modular hierarchical bayesian mixture-of-behaviors model," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2011.

[4] A. Gerdes, "Automatic maneuver recognition in the automobile: the fusion of uncertain sensor values using bayesian models," in *Proc. of the 3rd International Workshop on Intelligent Transportation*, 2006.

[5] I. Dagli, M. Brost, and G. Breuel, "Action recognition and prediction for driver assistance systems using dynamic belief networks," in *Agent Technologies, Infrastructures, Tools, and Applications for E-Services*, ser. Lecture Notes in Computer Science, 2003.

[6] T. Gindele, S. Brechtel, and R. Dillmann, "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments," in *Proc. of the IEEE Intelligent Transportation Systems Conference*, 2010.

[7] M. Reichel, M. Botsch, R. Rauschecker, K. Siedersberger, and M. Maurer, "Situation aspect modelling and classification using the scenario based random forest algorithm for convoy merging situations," in *Proc. of the IEEE Intelligent Transportation Systems Conference*, 2010, pp. 360 – 366.

[8] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall, 1999.

[9] R. D. Reed and R. J. Marks, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press, 1998.

[10] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, "PyBrain," *Journal of Machine Learning Research*, 2010.

[11] M. Garcia Ortiz, B. Dittes, J. Fritsch, and A. Gepperth, "Autonomous generation of internal representations for associative learning," in *Artificial Neural Networks - ICANN 2010*, ser. Lecture Notes in Computer Science, 2010, vol. 6354, pp. 247–256.

[12] J. C. McCall, D. Wipf, M. M. Trivedi, and B. Rao, "Lane change intent analysis using robust operators and sparse bayesian learning," in *IEEE Trans. Intell. Transp. Syst*, 2005, pp. 431–440.

[13] M. Garcia Ortiz, J. Fritsch, F. Kummert, and A. Gepperth, "Behavior prediction at multiple time-scales in inner-city scenarios," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2011.