Online Maneuver Recognition and Multimodal Trajectory Prediction for Intersection Assistance using Non-parametric Regression

Quan Tran and Jonas Firl

Abstract—Maneuver recognition and trajectory prediction of moving vehicles are two important and challenging tasks of advanced driver assistance systems (ADAS) at urban intersections. This paper presents a continuing work to handle these two problems in a consistent framework using non-parametric regression models. We provide a feature normalization scheme and present a strategy for constructing three-dimensional Gaussian process regression models from two-dimensional trajectory patterns. These models can capture spatio-temporal characteristics of traffic situations. Given a new, partially observed and unlabeled trajectory, the maneuver can be recognized online by comparing the likelihoods of the observation data for each individual regression model. Furthermore, we take advantage of our representation for trajectory prediction. Because predicting possible trajectories at urban intersection involves obvious multimodalities and non-linearities, we employ the Monte Carlo method to handle these difficulties. This approach allows the incremental prediction of possible trajectories in situations where unimodal estimators such as Kalman Filters would not work well. The proposed framework is evaluated experimentally in urban intersection scenarios using real-world data.

Index Terms—Intersection assistance, maneuver recognition, trajectory prediction, Gaussian process regression, Monte Carlo method, particle filters.

I. INTRODUCTION

Urban intersections are known to be one of the most complex traffic scenarios, which are both confusing and dangerous. Thus, an advanced driver assistance system (ADAS) with an intersection assistance feature is highly desirable and has an enormous potential of increasing the driver’s safety and comfort. An ADAS for such dynamic and sophisticated traffic environments requires a comprehensive understanding of the current situation in order to work intelligently and safely. Research on scene understanding has a long tradition in the artificial intelligence community and has already been demonstrated successfully in some simple indoor environments. In the intelligent vehicle community, many works on this topic have been devoted in recent years. Nevertheless, a full understanding of traffic scenes still remains challenging. In our previous work [1], we introduced a framework using two-dimensional Gaussian process regression to model the spatio-temporal dependencies of urban intersection scenarios. However, the framework still contains two drawbacks:

- It still does not have a wide field of applications.
- The method is not supported to work with different intersections without retraining model parameters.
- Multiple-step ahead trajectory prediction using an unimodal estimator (Unscented Kalman Filter) is not very suitable for intersection scenarios, where multimodalities are significant due to multiple crossing directions.

In this paper, we continue the idea introduced in [1] and provide important extensions to overcome these two drawbacks.

This paper is organized as follows. After reviewing related approaches in the next section, in Section III we present the experimental platform used for implementing and evaluating the proposed approach. In Section IV, we introduce our first extension in form of a feature normalization used to learn the characteristics of motion patterns with three-dimensional Gaussian process regression models. In Section V, we describe how the models are used for online maneuver recognition. We also present our second extension for multimodal trajectory prediction by integrating the Monte Carlo method into the regression models. Finally, Section VI concludes the paper and gives an outlook on future work.

II. RELATED WORK

In the field of scene understanding, approaches can be divided into two main groups: logical and probabilistic based methods. Approaches in the first group describe scenes or situations using mathematical logics as first introduced in [2]. While formal languages are powerful for knowledge representation and logical reasoning, they have crucial difficulties coping with uncertainties from sensor measurements and ambiguities of possible interpretations, which are typical in real traffic scenarios.

To handle the uncertainty and ambiguity issues, approaches in the second group make use of different probabilistic models. In the context of traffic scenarios at urban intersections, Dynamic Bayesian Networks (DBNs) are usually employed. However, learning parameters for DBNs remains challenging due to their complex structures. Hidden Markov Models (HMMs), which are specific structures of DBNs, are used to model uncertain spatio-temporal dependencies as in [3] or [4]. In our previous work [1], we use Gaussian process regression models to describe traffic situations at one urban intersection. Gaussian processes [5] are powerful mathematical tools, which have been extensively used in machine learning, robotic and computer vision such as [6] or [7]. Because the spatio-temporal dependencies of situations are modeled with an infinite number of states, the method offers a consistent framework for solving the both focusing problems: maneuver recognition and trajectory prediction. In
this paper we extend the proposed approach to achieve a greater applicability and feasibility.

III. DATA COLLECTION AND PREPROCESSING

Traffic data was acquired from two different urban intersections in Karlsruhe, Germany. In the data collection task we employ our research vehicle AnnieWAY (Fig. 1a). The experimental vehicle is equipped with several advanced environment perception and computation platforms [8]. For the acquisition of traffic data at urban intersections, we primarily use the laser scanner sensor Velodyne HDL-64E S2 (Fig. 1b), which is specifically designed for automobile applications. Because this sensor has a field of view of 360° horizontally and 28° vertically and scans continuously with 10 Hz, we can observe the entire area of the intersections. The raw sensor data perceived by the laser scanner are 3D points, which are plotted exemplarily in Fig. 1c. The intersection geometry is extracted from reference aerial images and is parameterized with cubic splines. The parameterization process will be discussed for more details in Section IV. After data has been collected, each moving vehicle is detected and tracked using Kalman filters with a constant velocity model. Some tracked trajectories of moving vehicles are plotted exemplarily in Fig. 2.

IV. MODELING OF TRAFFIC PATTERNS

In this section, we review briefly some important properties and notations of Gaussian process regression as discussed in [1]. A rigorous mathematical treatment of Gaussian process regression can be found in [5]. Afterwards, we provide a feature normalization scheme. The result of this section is normalized three-dimensional Gaussian process regression models of motion flows.

A. Gaussian Process Regression (GPR)

1) Gaussian Processes:

A Gaussian process (GP) is a random process in which the random variables are jointly Gaussian distributed. A GP can be completely described by the mean \( m(x) \) and the covariance function \( k(x, x') \).

\[
f \sim \mathcal{GP}(m(x), k(x, x'))
\]

Additionally, suppose we have a set of training examples \( D = \{ (x^{(i)}, f^{(i)}) \mid i = 1, 2, \ldots, n \} \), or in vector form \( \{ X, f \} \), which are drawn from an unknown noisy process:

\[
f^{(i)} = f(x^{(i)})
\]

According to the definition of GPs, at any particular input \( x_* \) the value of the function \( f(x_*) \) follows the multivariate Gaussian joint distribution:

\[
\begin{bmatrix}
f \\
f(x_*)
\end{bmatrix} \sim \mathcal{N}
\left(0, \begin{bmatrix} K & K_*^T \\ K_* & k_{**} \end{bmatrix}\right)
\]

where the covariance sub-matrices are computed based on the covariance function and sample data as described in [1]. Therefore, the best estimation for the predicted mean \( \mathbb{E}[f(x_*)] \) and the uncertainty \( \text{Cov}[f(x_*)] \) can be directly determined:

\[
\mathbb{E}[f(x_*)] = K_* K^{-1} f \\
\text{Cov}[f(x_*)] = k_{**} - K_* K^{-1} K_*^T
\]

2) Covariance function and hyper-parameter learning:

For the covariance function \( k(x, x') \), we choose the squared exponential (SE) function, because it offers the smoothness in regression models.

\[
k(x, x') = \sigma_f^2 \exp(-\frac{1}{2}(x - x')^T \Lambda (x - x')^T) + \sigma_n^2 \delta(x, x')
\]
Fig. 3: Spatial normalization scheme: (a) spline parameterization of centerlines, (b) data in the Cartesian coordinate system, (c) transformed data in the \((s, n)\) coordinate system.

Moreover, the hyper-parameters \(\Theta = \{\Lambda, \sigma_f, \sigma_n\}\), which characterize a GP, can be estimated from training examples by optimizing the log marginal likelihood:

\[
\Theta_{\text{max}} = \arg \max_{\Theta} \log p(f | X, \Theta)
\]

B. Feature Normalization

For a given set of trajectories, we extract training examples containing sequence of two-dimensional position measurements \(\{(x^{(i)}, y^{(i)}) | i = 1, 2, \cdots, n\}\) and corresponding velocity components \(\{(v_x^{(i)}, v_y^{(i)}) | i = 1, 2, \cdots, n\}\), which can be considered as raw features for constructing regression models. However, we want to develop models that are learned from data captured in various intersections and with different lengths. To achieve this goal we apply a feature normalization scheme to represent raw features in a normalized space.

1) Spatial normalization:

The main idea of spatial normalization is to transform the raw features \(\{(v_x^{(i)}, v_y^{(i)})\}\) from the global Cartesian coordinate system into an intersection coordinate system. First, the lane centerlines are parameterized using planar cubic spline curves \(s(u)\) governed by a set of control points \(\{q_1, q_2, \cdots, q_m\}\) with respect to the intersection center \(C\) (Fig. 3a). This modeling is reasonable, because cubic splines fulfill the smoothness requirement of road curvatures. Afterwards, the cubic splines are recomputed in order to yield the arc-length parameterization \(s(l)\) as described in [9]. Next, we introduce a new coordinate system \((s, n)\), which describes tangential and orthogonal components with respect to the centerlines. The spatial normalization is done by projecting the raw features \(\{(v_x^{(i)}, v_y^{(i)}) | i = 1, 2, \cdots, n\}\) into the \((s, n)\) coordinate system to obtain a set of spatial normalized features \(\{(v'_{x}^{(i)}, v'_{y}^{(i)}) | i = 1, 2, \cdots, n\}\). Fig. 3b and 3c illustrate the proposed transformation.

2) Temporal normalization:

Suppose we have a trajectory composed of \(N\) segments or \(N + 1\) samples and we want to represent this trajectory in normalized frames with \(N_0 + 1\) samples while maintaining its main characteristics. This issue can be solved with a two-step procedure:

- Upsample: Each segment of the original trajectory is equidistantly sampled with \(\frac{\text{lcm}(N, N_0)}{N_0}\) samples, where \(\text{lcm}(N, N_0)\) indicates the least common multiple of \(N\) and \(N_0\).
- Downsample: From the upsampled trajectory containing \(\text{lcm}(N, N_0) + 1\) samples, we keep \(N_0 + 1\) equidistant samples and remove other samples in between.

Fig. 4 demonstrates the temporal normalization procedure with \(N = 4\) and \(N_0 = 5\). It is recognizable that the normalized trajectory has the same relative dynamic property as the original trajectory.

C. Motion flow regression with three-dimensional Gaussian processes in the normalized space

After performing feature normalization (both spatial and temporal) we obtain the representation of normalized features \(\tilde{v} = (\tilde{v}_s, \tilde{v}_n)\) in the normalized input space \(\tilde{x} = (\tilde{s}, \tilde{n}, \tilde{t})\). The tilde notations indicate normalized quantities. By considering the feature components as two independent target variables of three-dimensional Gaussian processes, we express the motion flow regression problem as following:

\[
\tilde{v}_s \sim \mathcal{GP}_s(m_s(\tilde{x}), k_s(\tilde{x}, \tilde{x}')) \\
\tilde{v}_n \sim \mathcal{GP}_n(m_n(\tilde{x}), k_n(\tilde{x}, \tilde{x}'))
\]

For the regression purpose, the mean functions of the prior processes are set to zero, and the covariance functions are squared exponential with the hyper-parameters \(\Theta = \{\Lambda, \sigma_f, \sigma_n\}\) as described in Sec. IV-A.2. In this work, \(\Lambda\) is a three-dimensional diagonal matrix containing normalized
Fig. 5: Regression results for three situation: (a) go-straight, (b) turn-left and (c) stop-and-go due to the presence of moving obstacles. Colors encode different confidence levels: red for high confidence, blue for low confidence.

Fig. 6: Rescaling of the observation with respect to normalized frame.

length scales \( \tilde{l}_s, \tilde{l}_n \) and \( \tilde{l}_t \):

\[
\Lambda = \begin{bmatrix}
\tilde{l}_s & 0 & 0 \\
0 & \tilde{l}_n & 0 \\
0 & 0 & \tilde{l}_t \\
\end{bmatrix}
\]

Given normalized training examples \( \mathcal{D} = \{ (\tilde{x}^{(i)}, \tilde{v}^{(i)}) | i = 1, 2, \ldots, n \} \), the hyper-parameters \( \Theta \) are estimated according to Sec. IV-A.

After the hyper-parameters have been learned, for each point \( \tilde{x}_s = (\tilde{s}_s, \tilde{n}_s, \tilde{l}_s) \) in the normalized input space we generate the mean vector associated with the uncertainties as described in Sec. IV-A.

\[
E[\tilde{v}_s] = \text{GP}_\mu(\tilde{x}_s) \\
\text{Cov}[\tilde{v}_s] = \text{GP}_{\Sigma}(\tilde{x}_s)
\]

Fig. 5 shows the result of three possible regression models: go-straight, turn-left and stop-and-go due to the presence of moving obstacles at one T-intersection. The regression is performed in the normalized space divided into a grid of \( 0.5[m] \times 0.2[m] \) in the \((s, n)\) coordinate system and with the normalized frames of 40 samples. Each regression model is learned with about 15 example maneuvers collected from two different intersections. For visualization purposes, the regression models are converted and plotted in the \((x, y, t)\) coordinate system.

V. MANEUVER RECOGNITION AND TRAJECTORY PREDICTION

Our objective is a consistent framework for solving two important tasks:

- Online maneuver recognition of vehicles at the intersection as soon as possible.
- Successive multiple-step ahead trajectory prediction of participating vehicles.

In this section, we present how to use the regression models described in the previous section to handle these two problems effectively. Given the three-dimensional Gaussian process regression models, which describe the characteristic spatio-temporal dependency of possible situations, a new driving maneuver can be online recognized based on the likelihood of successive observations to each individual model. Because trajectories at urban intersections typically refer to multimodalities, we apply the Monte Carlo method for multimodal trajectory prediction.

A. Online maneuver recognition

Given a successive observation as a sequence of position and velocity measurements

\[
\mathcal{O}_{1:K} = \{ (x, y, v_x, v_y)^{(k)} | k = 1, 2, \ldots, K \}
\]

we want to select one regression model from all possible situations \( \mathcal{M} \) which best explains the observation:

\[
\mathcal{M} = \{ \text{turn-left, turn-right, go-straight, stop-and-go} \}
\]

Because the regression models present the spatio-temporal characteristics in the normalized space, the observation data needs to be mapped into the \((s, n)\) space and rescaled with respect to the normalized frames. The rescaling step is done by comparing the observation vector with the corresponding mean value of the regression model in the time axis and then scaling respectively.

Fig. 6 illustrates this rescaling scheme. With the representation of the observation sequence in the normalized space

\[
\tilde{\mathcal{O}}_{1:K} = \{ (\tilde{s}, \tilde{n}, \tilde{v}_s, \tilde{v}_n)^{(k)} | k = 1, 2, \ldots, K \}
\]
the maneuver $\mathcal{M}_s$ is recognized by
\[
\mathcal{M}_s = \arg \max_{\mathcal{M}_i \in \mathcal{M}} P(\tilde{O}_{1:K} \mid \mathcal{M}_i)
\]

Due to the properties of Gaussian processes, the data likelihoods $P(\tilde{O}_{1:K} \mid \mathcal{M}_i)$ can be effectively computed based on $K$-dimensional Gaussian distributions. Fig. 7a and Fig. 7b show the successive developing of the normalized probabilities corresponding to (a) turn-left and (b) go-straight maneuvers at one intersection. For robustness reasons, the decision is only made for the model $\mathcal{M}_s$ if the ratio of the likelihood furthermore is larger than a designed threshold $\gamma$:
\[
P(\tilde{O}_{1:K} \mid \mathcal{M}_s) > \gamma \quad \forall \mathcal{M}_i \neq \mathcal{M}_s
\]

Fig. 7c shows recognition results of different scenarios with the ratio threshold $\gamma = 2.0$ for vehicles approaching from the left at another intersection. It is worth noting that although the regression models are learned with examples taken from different intersections with different lengths, the regression models are representative and the results are reliable.

**B. Multimodal trajectory prediction**

Trajectory prediction concerns with the question where the vehicle will be in the next $T$ time steps. In principle, long term trajectory prediction is obtained by propagating information along the regression fields with respect to the uncertainties and non-linearities. This can be implemented by employing an estimator such as Kalman Filter or its variations as in our previous work [1]. However, the method can only be executed after the maneuver has been recognized correctly. Thus, this approach provides a hard decision and depends highly on the performance of the maneuver recognition stage. Additionally, trajectories at intersections imply the multimodality obviously with diverging possible paths. To achieve a robust and reliable long term trajectory prediction, in this work we make use of the Monte-Carlo method, which is implemented as a modified version of participles filters. The proposed method can cope with multimodal characteristics of urban intersections and non-linearities of GPR in an elegant way.

1) **Construction of mixed regression models:**

Given the regression models in the normalized space $(\tilde{s}, \tilde{n}, \tilde{t})$ and the intersection geometry (intersection center and centerlines), we regenerate the mixed regression model $\mathcal{M}_{\text{mixed}}$ in the Cartesian coordinate system $x = (x, y)$ as depicted in Fig. 8.

\[
\begin{align*}
v_x & \sim \mathcal{GP}_x(m_x(x), k_x(x, x')) \\
v_y & \sim \mathcal{GP}_y(m_y(x), k_y(x, x'))
\end{align*}
\]

These relationships can be formulated as a non-linear dynamical system:

\[
\begin{bmatrix}
v_x \\ v_y
\end{bmatrix}_{k} = \begin{bmatrix}
\mathcal{GP}_x^\mu(x_k) \\
\mathcal{GP}_y^\mu(x_k)
\end{bmatrix} + \varepsilon_k
\]

with $\varepsilon_k$ is zero-mean additive noise

\[
\varepsilon_k \sim \mathcal{N}(0, \begin{bmatrix}
\mathcal{GP}_x^\Sigma(x_k) & 0 \\
0 & \mathcal{GP}_y^\Sigma(x_k)
\end{bmatrix})
\]

2) **Long term trajectory prediction:**

We employ the idea of the particle filter algorithm for multiple-step-ahead trajectory prediction based on the given mixed model. The combination of GPR with various versions of Bayes Filters including particle filter is discussed extensively in [10].

The one-step particle filter algorithm using *importance sampling* scheme for Gaussian processes is summarized in Tab. I. The vehicle position or system state $x_k = (x_k, y_k)$ at time
TABLE I: Particle filter algorithm for Gaussian processes

<table>
<thead>
<tr>
<th>Algorithm: particle_filter4gaussian_process((X_{k-1}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: (X_0 = X_k = \emptyset)</td>
</tr>
<tr>
<td>2: (\triangleright) Prediction step</td>
</tr>
<tr>
<td>3: (\text{for } m = 1 \text{ to } M)</td>
</tr>
<tr>
<td>4: (\text{sample } x_k^m \sim x_k^{m-1} + GPu(x_k^{m-1}))</td>
</tr>
<tr>
<td>5: (w_k^m = \mathcal{N}(z_k; GPu(x_k^{m}), GP\Sigma(x_k^{m})))</td>
</tr>
<tr>
<td>6: (\text{add } (x_k^m, w_k^m) \text{ to } X_k)</td>
</tr>
<tr>
<td>7: endfor</td>
</tr>
<tr>
<td>8: (\triangleright) Resampling step</td>
</tr>
<tr>
<td>9: (\text{for } m = 1 \text{ to } M)</td>
</tr>
<tr>
<td>10: (\text{draw } i \text{ with probability } w_k^i)</td>
</tr>
<tr>
<td>11: (\text{add } (x_k^i, w_k^i) \text{ to } X_k)</td>
</tr>
<tr>
<td>12: endfor</td>
</tr>
<tr>
<td>13: return (X_k)</td>
</tr>
</tbody>
</table>

step \(k\) is represented by a set of \(M\) particles \(x_k^m\) and the corresponding weights \(w_k^m\)
\[X_k = \{ (x_k^m, w_k^m) \mid m = 1, 2, \ldots, M \}\]

At each time step, for each particle, this distribution is predicted based on the previous state \(x_k^{m-1}\) and the provided GPR. Because the GPR implies non-linearities, each particle is weighted by the likelihood of the most recent observation \(z_k = v_k\). Here \(v_k\) is a virtual observation provided by the regression model \(M_{\text{mixed}}\). Other remaining steps are identical to the standard particle filter algorithm [11]. This process is repeated many times to achieve the long term trajectory prediction.

Fig. 9 shows exemplarily the trajectory prediction. The system works with 10 frames/sec and the system state is represented by 250 particles. Due to the learned regression model and the resampling process the uncertainty does not propagate fast so that it provides a reliable prediction result. It is worth noting that the trajectory prediction procedure works independently with the maneuver recognition procedure. Moreover, the results show the reasonable multimodality, which can not be achieved with an unimodal estimator like Kalman Filter.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we present a consistent framework for solving two important problems: maneuver recognition and trajectory prediction of moving vehicles at urban intersections. By introducing the feature normalization scheme it is possible to learn the characteristic spatio-temporal dependencies of situations from data captured in different urban intersections. In the trajectory prediction stage, we incorporate the regression models with particle filters to overcome the non-linearity and multimodality issues. Future work will focus on modeling more complex situations in context of multi-agent scenarios using higher dimensional Gaussian process dynamical models.

VII. ACKNOWLEDGMENTS

The authors would like to thank Dr. Henning Lategahn and Dr. Frank Moosmann for their kind help with the data acquisition.

REFERENCES