2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Vehicle Trajectory Prediction with Gaussian Process Regression in Connected Vehicle Environment*

Sepideh Afkhami Goli[1], Behrouz H. Far[2], Abraham O. Fapojuwo[3]

***Abstract*—This paper addresses the problem of long term location prediction for collision avoidance in Connected Vehicle (CV) environment where more information about the road and traffic data is available through vehicle-to-vehicle and vehicle-to-infrastructure communications. Gaussian Process Regression (GPR) is used to learn motion patterns from historical trajectory data collected with static sensors on the road. Trained models are then shared among the vehicles through connected vehicle cloud. A vehicle receives information, such as Global Positioning System coordinates, about nearby vehicles on the road using inter-vehicular communication. The collected data from vehicles together with GPR models received from infrastructure are then used to predict the future trajectories of vehicles in the scene. The contributions of this work are twofold. First, we propose the use of GPR in CV environment as a framework for long term location prediction. Second, we evaluate the effect of pre-analysis of training data via clustering in improving the trajectory pattern learning performance. Experiments using real-world traffic data collected in Los Angeles, California, US show that our proposed method improves prediction accuracy compared to the baseline kinematic models.**

Keywords— connected vehicles; collision avoidance; pattern recognition; trajectory prediction; Gaussian Process regression

I. INTRODUCTION

Driver's safety is gaining increasing attention in the future Advanced Driver Assistance Systems (ADAS) and autonomous vehicles. To operate safely, it is crucial for a safety system to be able to anticipate what would happen to a vehicle's surrounding environment in the near future and plan ahead in real-rime. If projected trajectories of vehicles are available, the system can make timely decisions to avoid or warn about potential collisions which leads to safer and more efficient driving maneuvers [1]. To that end, this paper focuses on the long term (more than a second) trajectory prediction of nearby vehicles, in uncertain and dynamically changing environment of connected co-operative vehicles.

An intelligent vehicle is provided with different sources of data to obtain information about the nearby road users. On-board sensors such as camera, RADAR (Radio Detection And Ranging) and LIDAR (LIght Detection And Ranging) [2], provide location information about the objects on the road. However, on-board sensors are limited by their Field of View (FOV). For instance, a camera's view can easily be occluded by a building or vegetation. Most of the path prediction methods and Collision Avoidance Systems (CASs) in the literature, such as [3], exploit on-board sensors only and they are not able to observe beyond the obstacles. Therefore, there is always a high probability of crash in situations where the approaching vehicles are not located in the sensor range of visibility [4].

CV technology, on the other hand, enables a vehicle to act as a smart node that can collect and share information with other vehicles and road infrastructures [5]. Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications [5] provide this opportunity to expand the vehicle's perception range well beyond the limits of its on-board sensors [6]. A number of algorithms have been proposed in the literature that fuse the location sensor data with information received from CV environment and create a comprehensive picture of vehicle locations on the road [7]–[9]. However, prediction of vehicle locations via V2V and V2I communications has not been addressed in the literature extensively. In a recent work [6], Qiao *et. al* leveraged CV to collect location data and predicted vehicle trajectories using a prefix-projection-based trajectory prediction method as opposed to the current study, where V2V and V2I communications are utilized both in data collection and real-time path prediction phases.

Conventionally, physics-based parametric models are used for vehicular motion prediction [10]. Bayesian filtering approaches, such as the Kalman filter [11] or its variations [12] are the most commonly used approaches that adopt such models. Although these models are perfect for state estimations, predictions using these models are limited to the short-term as they make several assumptions such as constant velocity or acceleration for a certain period. These models are not capable of considering changes in the motion of a car caused by performing a maneuver, e.g. making a turn at an intersection [10].

In this work, vehicles are assumed to follow typical driving patterns in a given area. These patterns provide clues to the events taking place on the road and allows inference about the interactions among traffic objects. Markov Models and specifically Hidden Markov Models (HMMs) [13] as well as Recurrent Neural Network (RNN) have been used to extract trajectory patterns from collected data [14][15]. However, both of these approaches require state space discretization which needs additional processing and reduce the accuracy [16].

This paper adopts the Gaussian Process Regression (GPR) models [17] for representing vehicular motion patterns because of their robustness to noise in the observed location data and their ability to represent the variations in the execution of a motion pattern in a consistent and probabilistic manner [18][16]. GPR has been introduced as a Bayesian non-parametric technique for learning regression models from training data [9]. It is a generalization of Gaussian probability distribution as it models a process as a Gaussian distribution

*This work is partially supported by Alberta Innovative Technology Futures (AITF), Calgary, AB, Canada.

All authors are with the department of Electrical and Computer Engineering, University of Calgary, Canada.
[1]sepideh.afkhamigoli@ucalgary.ca,
[2]far@ucalgary.ca,
[3]fapojuwo@ucalgary.ca

978-1-5386-4452-2/18/$31.00 ©2018 IEEE

over functions. GPR models are well-suited for applications with noisy measurements, such as for locational sensor data collected from moving objects. Several works showed that GPR models are suitable for representing motion patterns in the context of road traffic [3], [16], [19].

The contributions of this paper are twofold. First; we suggest a framework for CV environment to learn GPR models from historical location data and share them among nearby vehicles. The GPR models are then leveraged alongside shared location data in the form of Basic Safety Messages (BSMs) from nearby vehicles to perform location prediction. Second; we evaluate the effect of auto-grouping training data into clusters and then using the clustered data for training Gaussian Process trajectory patterns.

The remainder of this paper is organized as follows. After Section II presents the preliminaries, Section III provides the assumptions we made as well as the problem statement. Then in Section IV, the proposed method to solve the long-term trajectory prediction problem is presented. Details of simulation experiments conducted together with the results are provided in section V. Finally, section VI concludes the paper and outlines some possible future works of this research.

II. GAUSSIAN PROCESS REGRESSION

Suppose we have a training dataset $D = \{(X_i, f_i), i = 1,2,\dots,n\}$ where $X = [X_i]$ denotes a multidimensional input vector and f_i denotes a scalar output corresponding to X_i, the i^{th} element in X and n denotes the size of the dataset. f_i values are assumed to be the sample values of a non-linear and noisy process:

$$f_i = f(X_i). \tag{1}$$

GPR is a non-parametric regression approach which finds a distribution over the possible functions $f(X)$ that are consistent with the observed data D [17]. A GPR model is fully described by its mean function $m(X)$ and covariance function $k(X,X')$ as shown in the following equation:

$$f \sim gp\big(m(X), k(X,X')\big). \tag{2}$$

The shape and smoothness of f is determined by the covariance function, as it controls the correlation between all pairs of output values.

The aim in GPR is to predict the value of the function f at any input X_* , $f(X_*)$. By the definition [17], we have:

$$\begin{bmatrix} f \\ f(X_*) \end{bmatrix} \sim N\left(0, \begin{bmatrix} K & K_*^T \\ K_* & k_{**} \end{bmatrix}\right), \tag{3}$$

where covariance sub-matrices, K, K_*^T, K_*, K_{**}, are calculated as in equations (4)-(6) [17] and the mean function is assumed to be 0 i.e. $m(X) = 0$.

$$K = \begin{bmatrix} k(X_1,X_1) & \cdots & k(X_1,X_n) \\ \vdots & \ddots & \vdots \\ k(X_n,X_1) & \cdots & k(X_n,X_n) \end{bmatrix}, \tag{4}$$

$$K_* = [k(X_*,X_1) \quad k(X_*,X_2) \quad \dots \quad k(X_*,X_n)\,], \tag{5}$$

$$K_{**} = k(X_*,X_*), \tag{6}$$

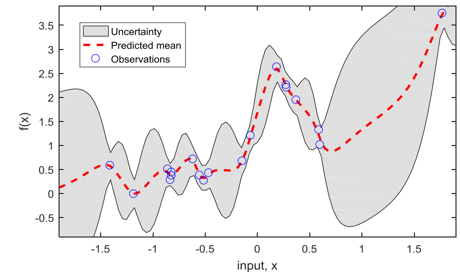


Figure 1. An example of one-dimensional GPR using zero mean and squared exponential covariance function

where $k(X,X')$ denotes the covariance between two input vectors X and X'. The covariance value is determined by the kernel function discussed below.

The requirement for GPR is that the kernel function needs to be a positive-definite function. There are several kernel functions used in the literature as covariance function. In [20] an overview of popular kernel functions for GPR is provided. However, the most common one is the Squared Exponential (SE) function, which is used in this work:

$$k(X,X') = \sigma_f^2 exp(-\frac{1}{2}\frac{(X-X')(X-X')^T}{\sigma_l{}^2}). \tag{7}$$

In equation (7), σ_l is the characteristic length scale and σ_f determines the standard deviation of the output. The parameters, σ_l and σ_f, are called hyper parameters and form a set values $\theta = \{\sigma_l, \sigma_f\}$ [16].

The learning process of GPR consists of tuning the values of θ to maximize the posterior probability of $P(f|X,\theta)$. This is done by maximization of the log marginal likelihood with respect to hyper parameters in θ [16].

The conditional probability distribution of $f(X_*)$ given the observed data, D, has a multivariate normal distribution and is defined as:

$$f(X_*)|f \sim N(K_*K^{-1}f, K_{**} - K_*K^{-1}K_*^T) \tag{8}$$

where K, K_* and K_{**} are defined above. The best estimation of $f(x_*)$ is the mean $E[f(x_*)]$ and the uncertainty of prediction is captured in the covariance $Cov\,[f(x_*)]$ which are determined as follows:

$$\mu(X_*, D) = E[f(X_*)|f] = K_*K^{-1}f, \tag{9}$$

$$\sigma^2(X_*, D) = Cov\,[f(X_*)|f] = K_{**} - K_*K^{-1}K_*^T. \tag{10}$$

Fig. 1 shows an example of one-dimensional GPR using SE function. When the number of observations increases, the uncertainty in the model decreases.

III. SYSTEM MODEL

A. Assumptions

1. Vehicles are equipped with Global Positioning System (GPS) receivers and able to receive their coordinates periodically.
2. Vehicles are equipped with (Dedicated Short-Range Communication) DSRC devices. They are able to

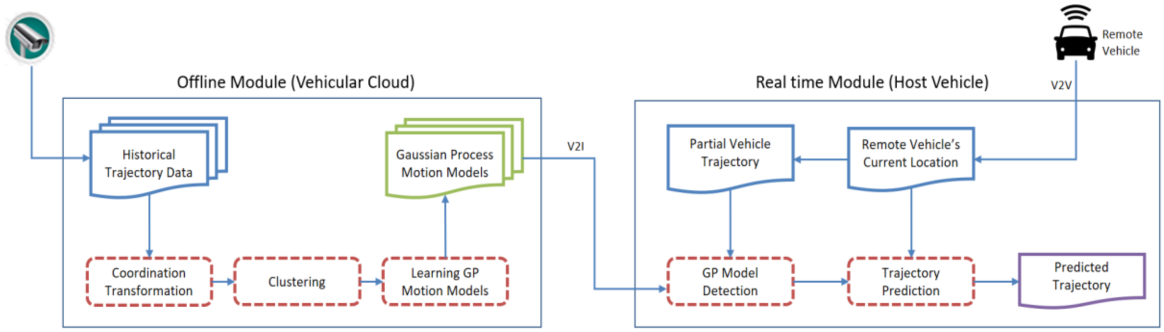


Figure 2. Overview of the proposed approach. The offline part takes place in the road infrastructure while the real time part is executed in the host vehicle.

communicate with other vehicles as well as with Road Side Units (RSUs) in their communication range, R, using the IEEE 802.11p standard [5].

3. Vehicles broadcast their GPS coordinates through BSMs. Since vehicle IDs are included in BSM messages, a receiver can associate the received coordinates to the sender by ID.
4. The coordinates of a surrounding vehicle at time step t is denoted by vector $p^t = (x^t, y^t)$.
5. The trajectory of a vehicle starting from time step l to l+t, denoted by T, is determined by the sequence of its coordinates:

$$T = (x^l, y^l), (x^{l+1}, y^{l+1}), \dots, (x^{l+t}, y^{l+t}). \quad (11)$$

6. A dataset of vehicular position observations, D, measured by static sensors at a certain temporal granularity level Δt (i.e., the sampling interval) is available. This data is provided to the road infrastructure.
7. A vehicle (called host vehicle) receives observations of another vehicle's location (called remote vehicle) in real time and can store the last t positions of a remote vehicle.

B. Problem Statement and Formulation

Given the dataset D, described above, and the last t observations of a remote vehicle's trajectory, as observed by the host vehicle:

$$T_{observed} = (x^1, y^1), (x^2, y^2), \dots, (x^t, y^t), \quad (12)$$

the problem is to use the observed data to predict the position of the remote vehicle for the next l time steps:

$$T_{predicted} = (x^{t+1}, y^{t+1}), (x^{t+2}, y^{t+2}), \dots, (x^{t+l-1}, y^{t+l-1}), (x^{t+l}, y^{t+l}). \quad (13)$$

It is worth mentioning that the remote vehicle must be located in the host vehicle's communication range in order to be able to send its location data to the host vehicle.

IV. Proposed Solution

In this section, a trajectory prediction framework for connected vehicles based on Gaussian Process regression is introduced. In this work, the Vehicle Cloud Model (VCM) presented in [21] is used. The vehicular cloud is created by inter-connecting resources available in the vehicles and RSUs.

The trajectory prediction framework contains 2 essential modules: (1) offline module; and (2) real-time module. The overview of the two modules including their components are depicted in Fig. 2. Offline module, which is implemented in the RSUs and road infrastructure generally, receives collected trajectory data from static sensors mounted on the roads. It performs post processing tasks on the trajectory data and then trains Gaussian Process regression models from the trajectory data. Real-time module, which is implemented in the host vehicle, exploits the trained GPR models generated by the offline module as well as the locational information received from remote vehicle through V2V communication to predict the future trajectory of the remote vehicle. In Fig. 2, the red dashed boxes show the components of the proposed approach that are discussed in the following.

A. Coordination Transformation

Historical trajectory data collected by the static sensors, such as video cameras, is in the road local coordinate system. On the other hand, the location received from the remote vehicle through V2V communication is from the GPS sensor and is in the global geographical coordinate system (latitude and longitude). Since historical trajectory data and the current locations are both the inputs to the trajectory prediction algorithm, they need to be in the same coordinate system. Therefore, historical locational data are transformed to the global coordinates using the closed form solutions proposed in the literature [22].

B. Clustering the training dataset

In this phase, vehicular trajectories are automatically grouped into finite number of clusters. The main motivation of this phase is that vehicles movement and their velocities in each local subset is assumed to have similar properties through space and time [23]. Therefore, training GPR models from the local subgroups have better performance both in the sense of processing time as well as prediction results. K-means clustering algorithm [24] is used here for this purpose and trajectories are clustered according to the first and the last positions in the traffic scene.

C. Learning Gaussian Process Motion Models

The aim of this phase is to learn motion models from the collected trajectory data to be able to predict the future behavior and trajectory of vehicles. Motion model is defined as the mathematical mapping from positions (x, y) to a distribution over vehicle trajectory derivatives $\frac{\Delta x}{\Delta t} = v_x$ and $\frac{\Delta y}{\Delta t} = v_y$ which allows us to predict how the situation will evolve in the future [10]. Given a vehicle's position at time t, (x^t, y^t), and its trajectory derivatives (v_x^t, v_y^t), its predicted next position (x^{t+1}, y^{t+1}) is computed as $(x^t + v_x^t,\ y^t + v_y^t)$. Therefore, modeling the trajectory derivatives would be adequate for modeling trajectories.

Here, motion model is defined by a pair of GPR models mapping coordinate vector of a vehicle, $p = (x, y)$, to a distribution over trajectory derivatives. This motion model has been previously presented in [25]. By considering v_x and v_y as two independent scalar functions defined over input vector $p = (x, y)$, a motion model M is modeled by a pair of two dimensional Gaussian Processes that map position vectors to v_x and v_y as

$$v_x \sim gp_x\big(m(p), k(p, p')\big), \tag{14}$$

$$v_y \sim gp_y\big(m(p), k(p, p')\big). \tag{15}$$

The mean function, $m(p)$, is taken to be zero for the regression purpose which means without any additional knowledge, we expect a vehicle to have zero velocity and stay in the same place. The standard squared exponential covariance function [17] as defined in equation (7) is used for computing covariance values, $k(p, p')$. This function is chosen among different choices since its smoothness assumption is well suited for this problem.

As in every learning process, to train the model a training dataset is needed. Here, historical trajectory data which is already clustered into subsets of training datasets in Section V.B, is used for this purpose. For each cluster of trajectories, the following training datasets are used to train two GPR models:

$$D_x = \{(p_i, v_{x_i}), i = 1,2, \dots, m\} \tag{16}$$

$$D_y = \left\{\left(p_i, v_{y_i}\right), i = 1,2, \dots, m\right\}, \tag{17}$$

where m is the number of data points available in the cluster. The learning phase consists of processing sample trajectories and fitting a Gaussian distribution over them. In other words, two sets of hyper parameters are learned for each cluster of data, θ_x and θ_y.

D. Motion Model Selection

Topredict the future trajectory of a remote vehicle having observations of its last positions, we need to first determine the most similar motion model among the learned models for different clusters. Assume the trajectory of length t has been observed by the host vehicle and denoted by $T_{observed}$. The probability of $T_{observed}$ under the model M_j, is computed by:

$$P\big(\, T_{observed} \big| M_j\big) = \prod_{k=1}^{t} P\big(v_x^k \big| M_j\big).\ P\big(v_y^k \big| M_j\big). \tag{18}$$

Among all models, the model that maximizes the above probability, M_*, is chosen and used for performing the trajectory prediction.

E. Trajectory Prediction

Once the model M_* and the corresponding set of GPR models are selected for a vehicle, we can calculate the Gaussian distribution over trajectory derivatives for every position p_* using the following equations:

$$N\big(\mu_x(p_*, D_x), \sigma_x^2\,(p_*, D_x)\big), \tag{19}$$

$$N\big(\mu_y(p_*, D_y), \sigma_y^2\,(p_*, D_y)\big). \tag{20}$$

To sample a trajectory from the current location of the vehicle (x^t, y^t), first determine a trajectory derivative (v_x^t, v_y^t), to calculate the vehicle's next location. Then starting from (x^{t+1}, y^{t+1}), trajectory derivative is sampled again. This process is repeated until the trajectory of the desired duration is computed. The length of the predicted trajectory depends on the number of times that we repeat the process.

V. Experiments And Results

In order to verify the effectiveness of the proposed method, a real world trajectory dataset was used [26]. The dataset includes 30 minutes detailed vehicle trajectory data on Lankershim Boulevard in the Universal City neighborhood of Los Angeles, CA, collected on June 16, 2005. The area where the data was collected is approximately 500 meters (1,600 feet) in length and consists of bidirectional three to four lane arterial segments and complete coverage of three signalized intersections. The data was obtained using five video cameras mounted on the roof of a 36-story building located adjacent to the U.S. Highway 101 and Lankershim Boulevard.

Table 1. Two sample data in the trajectory dataset.

ID	Frame	Global Time	Local_X	Local_Y	O_Zone	D_Zone
63	3	1118935680200	44.589	67.404	102	205
377	2409	1118935920800	32.755	154.931	102	208

The data includes coordinates of each vehicle, collected every one-tenth of a second in relative space. We made use of data attributes that are shown in Table 1. Each row of the dataset includes the vehicle ID, the local time (Frame) and the global time of collecting that specific row of data. Local_X and Local_Y show the local x coordinate and y coordinate of the vehicle, respectively. Some attributes such as O_Zone or D_Zone, which imply the origin zone and the destination zone of the car passing through the study area respectively, were exploited for the purpose of filtering the data.

Since Δt between two consecutive data samples is very small, 0.1 sec, given a vehicle's two consecutive position vectors, instantaneous velocity vector is calculated as:

$$\left[v_x, v_y\right] \approx \left[\frac{\Delta x}{\Delta t}, \frac{\Delta y}{\Delta t}\right]. \tag{21}$$

Then, the coordinates together with the extracted velocities are inserted into the Gaussian Process training functions (14) and (15).

In order to validate the effectiveness of the method, 5-fold cross-validation [27] was performed on the dataset to derive a more accurate estimate of trajectory prediction performance. The advantage of this method is that all observations are used for both training and validation (but not at the same time), and each observation is used for validation exactly once.

The vehicle trajectories in the training dataset are partitioned into multiple clusters of trajectories using K-means clustering method [24] as discussed in section IV.B. The optimum number of clusters, K, is derived with experiments that will be discussed later in this section. In the clustering phase, Euclidean distance is used as the distance among data points indicating that those trajectories with closer

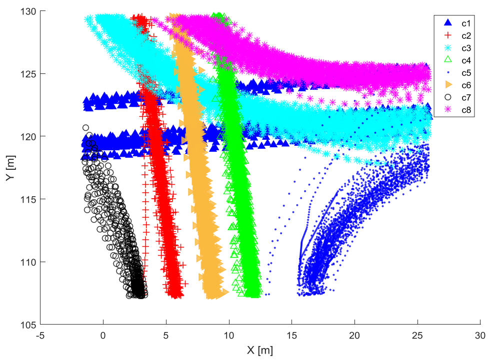


Figure 3. The result of clustering trajectories on the training data using K-means clustering.

start and end positions are clustered in the same cluster while trying to maximize the distance among the start and end positions of the two trajectories in two different clusters. Fig. 3 shows the output of vehicle trajectories for the intersection under experiment. The trajectories in the training dataset are grouped into 8 distinct clusters that clearly express different typical driving patterns at the selected intersection.

After the clustering phase, trajectories in the same cluster are fed into training phase and one pair of GPR models, shown with equations (14) and (15), are extracted for each cluster: gp_x and gp_y. For each trajectory in the test dataset, it is first required to determine the cluster that the trajectory most likely belongs to. The first 10 observations of the trajectory, which is equivalent to 1 second of data, is used for choosing the most appropriate cluster. In other words, t is assumed to be 10 in equation (12). The two trained GPR models for the chosen cluster are used for the prediction of the vehicle's next positions.

Having detected the cluster that the trajectory belongs to, components of the velocity are predicted using the associated GPR models for the next 3 seconds, equivalent to 30 time-steps, with the algorithm explained in section IV.E. Fig. 4 depicts the calculated location based on predicted velocity with both kinematic model and the proposed technique performed at t=10 (after 1 second) together with actual trajectory for 3 different vehicles. As shown, when a vehicle is taking a specific maneuver such as turning right or turning left, its maneuver can be detected successfully using the proposed method. However, kinematic model is not able to adapt to long-term changes in motion pattern and consequently the predictions deviates from the actual trajectory as time passes.

To evaluate the prediction accuracy, the predicted trajectory of the vehicle using our method is compared to that of Constant Acceleration (CA) kinematic model. In CA model, the next location of the vehicle p_2 is computed as follow:

$$p_2 = \frac{1}{2}a\,\Delta t^2 + v\Delta t + p_1. \tag{22}$$

We assume that the true acceleration and velocity vectors of the vehicle is available at the prediction time which is an optimistic assumption.

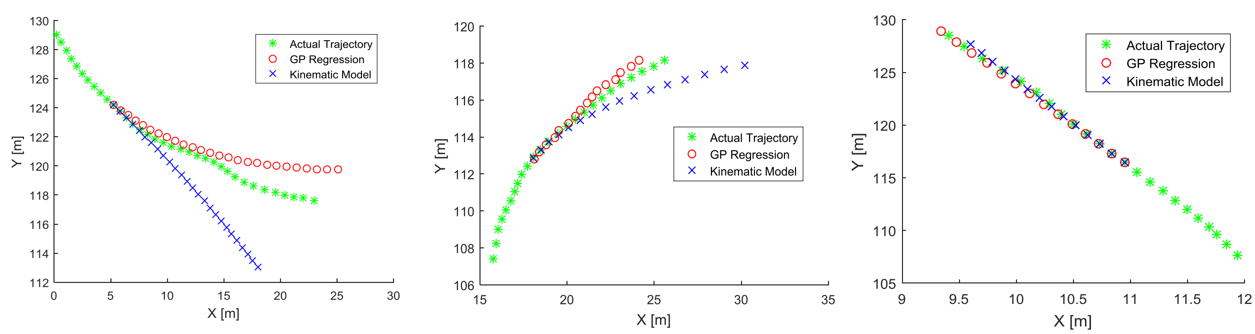
The accuracy of the proposed technique in terms of prediction error is evaluated by the Root-Mean-Square Error (RMSE) metric defined as

$$RMSE = \sqrt{\left(\frac{1}{N}\right)\sum_{i=1}^{N}(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}\,. \tag{23}$$

In (23), N is the number of trajectories in the testing dataset and $\hat{x}_i$ and $\hat{y}_i$ denote the predicted coordinates while x_i and y_i denote the true coordinates of the vehicle. At every prediction step, the RMSE is computed for all trajectories in the testing dataset.

Fig. 5 depicts the location prediction error calculated using the RMSE metric defined above. The error bars represent the standard deviation of the prediction error which was calculated using the Euclidean distance between the predicted location and the true location. As shown, initially the kinematic model has a very low error but as the prediction process continues, the kinematic model prediction deviates from the truth with a higher standard deviation. On the other hand, the proposed GPR technique has much lower error (about 3 meters) with lower standard deviation (about 2 meters) after 4 seconds, t = 40.

Optimum number of clusters in the K-means clustering algorithm is derived by running the GPR method with different number of clusters. Since the dataset contains different types of trajectories related to different maneuvers, GPR could not converge with low number of clusters (less than 5). Therefore, we started the experiment with 5 clusters and increased the number of clusters to 10. As shown in Fig. 6, for this dataset, the optimum number of clusters that has the lowest RMSE during the prediction process is 8.



(a) (b) (c)

Figure 4. Trajectory prediction for 3 different vehicles using GPR method and kinematic model. The first 10 positions in the trajectories are used for detecting the GPR model among the trained models. In (a) vehicle is turning left, (b) vehicle is turning right, (c) vehicle is going straight.

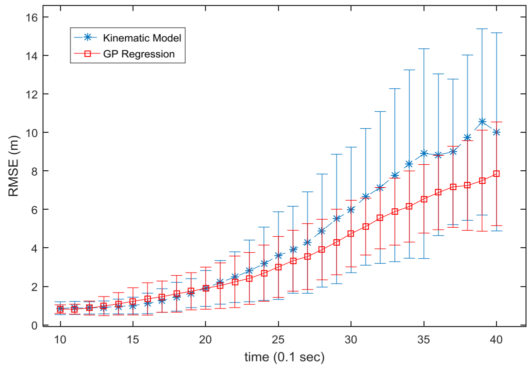


Figure 5. Comparison of the proposed method with kinematic model in terms of RMSE of the location prediction.

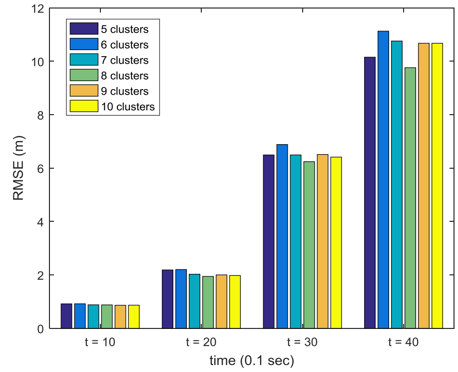


Figure 6. Comparison of the proposed method performance with different number of clusters in terms of RMSE.

VI. Conclusion and Future works

This paper addresses an important problem in traffic safety: real-time location prediction for collision avoidance. To this end, we proposed a framework for learning Gaussian Process Regression (GPR) models from historical trajectory data and share those learned models among vehicles via V2I communication. This framework also provides details on how a vehicle can receive real-time information from another vehicle and incorporate that information together with the received GPR models in location prediction process. In addition, we made use of clustering technique to extract better models from data. The experimental results on a real-world trajectory dataset show the effectiveness of using this framework in terms of accuracy in predicting the location of the vehicle in real time, against the Constant Acceleration kinematic model. More research is required to assess the effectiveness of this framework in avoiding collisions successfully. Future studies will focus on detailed analysis of this approach in collision risk estimation and path planning. Moreover, a separate study is required to delve into various clustering techniques and their effect on GPR performance.

References

[1] K. Golestan, R. Soua, F. Karray, and M. S. Kamel, “Situation awareness within the context of connected cars: A comprehensive review and recent trends,” Information Fusion, vol. 29, pp. 68–83, 2016.

[2] M. S. Shirazi and B. T. Morris, “Looking at Intersections: A Survey of Intersection Monitoring, Behavior and Safety Analysis of Recent Studies,” IEEE Transactions on Intelligent Transportation Systems, vol. PP, no. 99. pp. 1–21, 2016.

[3] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J. D. Yoder, C. Tay, K. Mekhnacha, and A. Negre, “Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety,” IEEE Intelligent Transportation Systems Magazine, vol. 3, no. 4, pp. 4–19, 2011.

[4] S. H. Bayless, A. Guan, P. Son, S. Murphy, and A. J. Shaw, “Connected Vehicle Insights Trends in Roadway Domain Active Sensing,” Intelligent Transportation Society of America, pp. 1–31, 2014.

[5] “IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services,” in IEEE Std. 1609.3 - 2007, 2007.

[6] S. Andrews, “Vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) communications and cooperative driving,” in Handbook of Intelligent Vehicles, 2012, pp. 1121–1144.

[7] R. Parker and S. Valaee, “Vehicle localization in vehicular networks,” IEEE Vehicular Technology Conference, pp. 2713–2717, 2006.

[8] S. A. Goli, B. H. Far, and A. O. Fapojuwo, “An accurate multi-sensor multi-target localization method for cooperating vehicles,” Theoretical Information Reuse and Integration. Advances in Intelligent Systems and Computing, Springer, vol. 446, pp. 197–217, 2016.

[9] M. Efatmaneshnik, A. Kealy, A. T. Balaei, and A. G. Dempster, “Information Fusion for Localization Within Vehicular Networks,” Journal of Navigation, vol. 64, no. 3, pp. 401–416, 2011.

[10] S. Lefèvre, D. Vasquez, and C. Laugier, “A survey on motion prediction and risk assessment for intelligent vehicles,” ROBOMECH Journal, vol. 1, no. 1, p. 1, 2014.

[11] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” Journal of Basic Engineering, vol. 82, no. 1, p. 35, 1960.

[12] S. Sarkka, “Bayesian Filtering and Smoothing,” Cambridge University Press, p. 254, 2013.

[13] B. Schuster-Böckler and A. Bateman, “An Introduction to Hidden Markov Models,” Current Protocols in Bioinformatics, no. January, 2007.

[14] B. Kim, C. M. Kang, S. H. Lee, H. Chae, J. Kim, C. C. Chung, and J. W. Choi, “Probabilistic Vehicle Trajectory Prediction over Occupancy Grid Map via Recurrent Neural Network,” 2017.

[15] D. Vasquez, T. Fraichard, and C. Laugier, “Growing Hidden Markov Models: An Incremental Tool for Learning and Predicting Human and Vehicle Motion,” The International Journal of Robotics Research, vol. 28, no. 11–12, pp. 1486–1506, 2009.

[16] C. Tay, “Analysis of dynamic scenes: Application to driving assistance,” PhD Thesis, 2009.

[17] C. E. Rasmussen; and C. K. I. Williams, Gaussian Processes for Machine Learning. The MIT Press, 2006.

[18] J. Joseph, F. Doshi-Velez, and N. Roy, “A bayesian nonparametric approach to modeling mobility patterns,” Aaai, pp. 1587–1593, 2010.

[19] Q. Tran and J. Firl, “Modelling of traffic situations at urban intersections with probabilistic non-parametric regression,” IEEE Intelligent Vehicles Symposium, Proceedings, no. Iv, pp. 334–339, 2013.

[20] D. K. Duvenaud, “Automatic Model Construction with Gaussian Processes,” University of Cambridge, 2014.

[21] M. Gerla, E. K. Lee, G. Pau, and U. Lee, “Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds,” 2014 IEEE World Forum on Internet of Things, WF-IoT 2014, pp. 241–246, 2014.

[22] B. K. P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” Journal of the Optical Society of America A, vol. 4, no. 4, p. 629, 1987.

[23] T. V. Le, R. Oentaryo, S. Liu, and H. C. Lau, “Local Gaussian Processes for Efficient Fine-Grained Traffic Speed Prediction,” IEEE Transactions on Big Data, vol. 3, no. 2, pp. 194–207, 2017.

[24] J. McQuenn, “Some methods for classification and analysis of multivariate observations,” Computers and Chemistry, vol. 4. pp. 257–272, 1967.

[25] G. S. Aoude, J. Joseph, N. Roy, and J. P. How, “Mobile Agent Trajectory Prediction using Bayesian Nonparametric Reachability Trees,” Proc. of AIAA Infotech@ Aerospace, no. March, pp. 1587–1593, 2011.

[26] U. S. D. of Transportation, “Lankershim Boulevard Dataset,” 2007. [Online]. Available: https://www.fhwa.dot.gov/publications/research/operations/07029/index.cfm. [Accessed: 01-Apr-2017].

[27] R. Kohavi, “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection,” Appears in the International Joint Conference on Articial Intelligence (IJCAI), vol. 5, pp. 1–7, 1995.