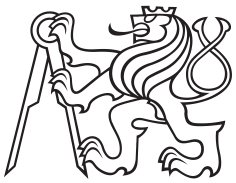
Master Thesis



Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Compute Science

Trajectory prediction of vehicles on highway

Bc. Martin Votruba

Supervisor: Ing. Franc Vojtěch Ph.D.
Field of study: Open Informatics
Subfield: Artificial Intelligence
May 2020

MASTER'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Votruba Martin** Personal ID number: **435575**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Computer Science**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence**

II. Master's thesis details

Master's thesis title in English:

Trajectory prediction of vehicles on highway

Master's thesis title in Czech:

Predikce trajektorií vozidel na dálnici

Guidelines:

This thesis should focus on development of trajectory predictor for highway driving situations. Predicted trajectories of surrounding vehicles should be compared to ground truth trajectories using relevant metric. Currently used trajectory comparison methods usually do not take into account sensor limitations and other physical aspects, it is necessary to develop novel metrics that would be able to weight importance of trajectories of other vehicles with respect to the EGO vehicle. Evaluation of results will be performed on data provided by thesis supervisor.
Tasks:
1. Research path prediction of traffic surrounding EGO vehicle for highway scenarios and identification of driver maneuvers in literature
2. Design and implement metric for comparison of two trajectories that takes into account vehicle sensors limitations and physical limitations of vehicles
3. Design, implement and compare algorithm for trajectory prediction of EGO vehicle surrounding traffic.
4. Evaluate algorithms on provided data

Bibliography / sources:

[1] Deo, Nachiket, et al. 'How Would Surround Vehicles Move? A Unified Framework for Maneuver Classification and Motion Prediction'. IEEE Transactions on Intelligent Vehicles 2018
[2] Adam Houenou, Philippe Bonnifait, Véronique Cherfaoui, Yao Wen. 'Vehicle Trajectory Prediction based on Motion Model and Maneuver Recognition.' IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2013
[3] Lefèvre, S., Vasquez, D., & Laugier, C. 'A survey on motion prediction and risk assessment for intelligent vehicles.' ROBOMECH Journal, 1(1). 2014

Name and workplace of master's thesis supervisor:

Ing. Vojtěch Franc, Ph.D., Machine Learning, FEE

Name and workplace of second master's thesis supervisor or consultant:

Date of master's thesis assignment: **04.02.2020** Deadline for master's thesis submission: **22.05.2020**

Assignment valid until: **30.09.2021**

Ing. Vojtěch Franc, Ph.D. Supervisor's signature	Head of department's signature	prof. Mgr. Petr Páta, Ph.D. Dean's signature

III. Assignment receipt

The student acknowledges that the master's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master's thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student's signature

Acknowledgements

I would like to thank my supervisor Vojtěch Franc, who was extraordinarily supportive and patient throughout writing this thesis. I am also grateful to all of my colleagues from HWPS, who were very supportive, especially Lukáš, Filip and Martin. And last but not least, I would like to express my gratitude to my family, who supported me during my studies.

Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

I hereby declare that I have completed this work on my own.

In Prague, 22. May 2020

Abstract

The topic of this thesis is trajectory prediction of the vehicles on a highway. The task is to predict a 5s future position of a target vehicle, given its current state and history. First, this problem is formally defined, and data on which the proposed models shall be evaluated are described. Subsequently, an overview of currently used techniques is presented, followed by three proposed approaches. The first approach is based on a direct prediction of the future position using Gradient Boosted Trees. The second is focused on improving an existing trajectory generating module in the Frenet frame of reference. The third one is taking inspiration from face recognition, describing the trajectory by Point Distribution Model, and inferring the coefficients of Eigenvectors obtained by a Principal Component Analysis. In the final chapter evaluation and comparison of the proposed methods concludes this thesis.

Keywords: Trajectory prediction, Gradient Boosted Trees, Principal Component Analysis, Eigentrajectories, Point Distribution Model, Manoeuvre Classification

Supervisor: Ing. Franc Vojtěch Ph.D. xfrancv@cmp.felk.cvut.cz

Abstrakt

Tématem této práce je predikce trajektorií vozidel na dálnici. Cílem práce je navrhnout algoritmus, který by na základě současného stavu a krátké historie pohybu předpověděl 5 vteřin budoucí trajektorie cílového vozidla. Tento problém je v práci nejdříve formálně popsán, spolu s představením dat, na kterých se navržené metody budou vyhodnocovat. Následně je zde uveden přehled v současnosti používaných metod k řešení tohoto problému. Poté jsou již prezentovány tři navržené přístupy. První z nich používá Gradient Boosted Trees pro přímou predikci budoucích pozic cílového vozidla. Druhá metoda se zaměřuje na zlepšení výsledků stávajícího modulu generujícího trajektorie vozidel zlepšením přesnosti klasifikace manévrů. Třetí přístup je inspirován rozpoznáváním tváří, trajektorie je zde popsána pomocí Point Distribution Modelu a predikovány jsou koeficienty vlastních vektorů získaných pomocí Analýzy hlavních komponent. V závěru práce je uvedeno vyhodnocení těchto přístupů.

Klíčová slova: Predikce trajektorií, Gradient Boosted Trees, Analýza hlavních komponent, Eigentrajectories, Point Distribution Model, Klasifikace manévrů

Překlad názvu: Predikce trajektorií vozidel na dálnici

Contents

Figures

Tables

Chapter 1

Introduction

In the word of autonomous cars, there is a growing need for a reliable trajectory prediction of vehicles. The challenge is to determine the future position of all traffic participants based on the current situation. One can take many factors into account for such prediction - physical state of the vehicle, road shape and topology, driver's behaviour or interaction between drivers. A precise prediction is a necessary precondition for an autonomous vehicle to be able to engage in road traffic successfully. In recent years the best performing approach is using neural networks. While this approach has impressive results, the big problem with neural networks is the unbounded space of possible predictions. In this thesis, I design novel approaches of trajectory prediction, which focus on producing only a physically feasible trajectory.

I start in Chapter 2 by formally defining the problem and discussing the data that are being used in this thesis. A wide range of methods to solve this problem was proposed in recent years; an overview is presented in Chapter 3. In the following three chapters I successively introduce the novel approaches that I have developed.

Chapter 4 describes a model based on a direct prediction of the future position using Gradient Boosted Trees. This is followed by Chapter 5, which is focused on improving an existing trajectory generating module in Frenet frame of reference. The third approach is described in Chapter 6. This approach is taking inspiration from face recognition, describing the trajectory by Point Distribution Model, and predicting coefficients of Eigenvectors obtained by a Principal Component Analysis.

The thesis concludes with an evaluation of these methods reported in Chapter 7. Finally, in Chapter 8, the thesis conclusion and discussion of possible future work is presented.

Chapter 2

Problem statement

My task is to develop an algorithm that predicts future positions of vehicles on a highway. The prediction time horizon is 5s, and the prediction shall be made for each whole second up to the horizon. A vehicle history of 3s can be assumed to be given with a sampling rate of 3 Hz. I operate on situation representation on the level of detected objects, so position, velocities, and other states of vehicles are known - while the vehicle detection and tracking is not a concern of this thesis.

2.1 Formal definition

I use similar notation as established by Quehl et al. in 2017 [QHT+17]. A trajectory is temporally ordered set of attributes $T_v = (\overrightarrow{p_{v|1}}, \overrightarrow{p_{v|2}}, \overrightarrow{p_{v|3}}, \ldots, \overrightarrow{p_{v^n}})$, where each element of trajectory $T_v(t) = \overrightarrow{p_{v|t}} = (x_{v|t}, y_{v|t})$ describes the position of a vehicle v at a time step t relative to the beginning of the trajectory. I denote the ground truth trajectory (i.e. the real trajectory) as T_v^g. For purpose of this thesis $T_{v|t}^g$ is 5 second long segment of a vehicle trajectory starting at time t (relative to the beginning of the trajectory) with 1Hz frequency. A $T_{v|t}^h$ is 3 second long historical trajectory of a vehicle at time t with 3Hz frequency. $T_{v|t}^p$ refers to the predicted trajectory for vehicle v at time step t. To clarify this, T_v^g is the complete trajectory of a single vehicle with variable length[1] and each 8s long sub-sequence defines one pair of historical $(T_{v|t}^h)$ and future $(T_{v|t}^g)$ trajectory. An example is shown in Figure 2.1.

$$\begin{aligned} T_{v|t}^h &= (T_v(t+\tau) \mid \tau \in \{-3, -2.8, -2.6, \ldots, 0\}) \\ &= (\overrightarrow{p_{v|t+\tau}^g} \mid \tau \in \{-3, -2.8, -2.6, \ldots, 0\}) \\ T_{v|t}^g &= (T_v(t+\tau) \mid \tau \in \{1, 2, 3, 4, 5\}) \\ &= (\overrightarrow{p_{v|t+\tau}^g} \mid \tau \in \{1, 2, 3, 4, 5\}) \\ T_{v|t}^p &= (\overrightarrow{p_{v|t|\tau}^p} \mid \tau \in \{1, 2, 3, 4, 5\}) \end{aligned} \tag{2.1}$$

I omit the subscripts and brackets when I am talking about the whole set of trajectories rather then a single trajectory, so T^g denotes all 5s long trajectories

[1] The length depends on how long was the particular vehicle observed.

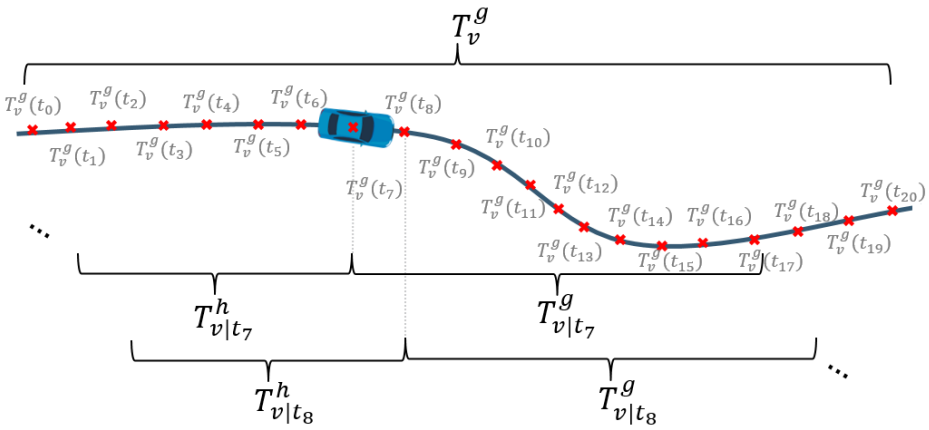


Figure 2.1: Visualisation of a trajectory of a single vehicle, the complete trajectory (top) and its division into sub-sequences (bottom).

of all vehicles. $t_v^1 = 3$ denotes the first point in a vehicle trajectory with 3 second history, t_v^{max} denotes last known point with 5s second observed future ($T_v^g(t_v^{max})$ is defined). Lets define Δ_v as set of all complete measurements - time points t for which both $T_v^{g|t}$ and $T_v^{h|t}$ are defined for a vehicle v, so $\Delta_v = \{t_v^1, t_v^1 + \frac{1}{F}, t_v^1 + \frac{2}{F}, t_v^1 + \frac{3}{F} \dots t_v^{max}\}$ where F is the sampling frequency of the chosen dataset. Set of all vehicles in a dataset shall be denoted as V, M marks number of all complete measurements and $M = \sum_{v \in V} |\Delta_v|$.

The task is for a given history of a vehicle, information about the road (in particular lane marking), and surroundings vehicles to predict the future trajectory of the vehicle.

$$given \quad T_{v|t}^h \quad predict \quad T_{v|t}^p \tag{2.2}$$

2.1.1 Notation of physical states

Throughout the thesis following notation and units of physical states is used, t denotes time.

- $x(t)$ $[m]$ - longitudinal position
- $y(t)$ $[m]$ - latitudinal position
- $v_t(t)$ $[ms^{-1}]$ - tangential velocity
- $v_x(t)$ $[ms^{-1}]$ - longitudinal velocity
- $v_y(t)$ $[ms^{-1}]$ - latitudinal velocity
- $a_t(t)$ $[ms^{-2}]$ - tangential acceleration
- $a_x(t)$ $[ms^{-2}]$ - longitudinal acceleration
- $a_y(t)$ $[ms^{-2}]$ - latitudinal acceleration

- $\theta(t)$ $[rad]$ - yaw, computed from velocities as $\theta = \arctan(\frac{v_y}{v_x})$
- $\omega(t)$ $[rad \cdot s^{-1}]$ - yaw rate, first derivation of yaw

2.2 Data

Methods proposed in this thesis were originally supposed to be evaluated on data delivered by the thesis supervisor (PES data set). However, upon their examination, I have found out that the data exhibits several anomalies (see Figure 2.2 and Figure A.1 as an example) and in general are very noisy. Even after applying a substantial amount of prepossessing and filtering, I was not able to achieve the necessary data quality. For this reason, I searched for a substitute. In this section, I compare two existing datasets, NGSIM, and highD, that are relevant for this thesis and justify our choice for the latter, and also what modifications were needed. There are some other vehicle recordings dataset available, but they are not relevant for me as they are either not recorder on a highway ([CLS+19], [KMB+], [MZZ+19], [BKM+19]) or they provide data before object detection in form of images ([CBL+19]).

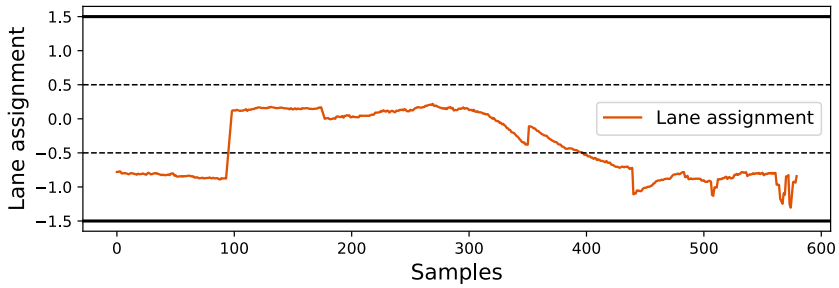


Figure 2.2: Demonstration of poor PES data quality, lane assignment (see Figure 4.2) signal for single vehicle trajectory as function of time (sampled with F=25Hz).

2.2.1 NGSIM

The NGSIM dataset is a collection of vehicle trajectories recorded on freeways and interstates in the USA by Federal Highway Administration [oT07]. The data were collected on three different locations, from which only two are relevant for me - Interstate 80 and US Highway 101. The clear advantage of this dataset is that it is well established, and the results of other methods can be easily found in the literature. The pitfall of this dataset is the poor quality of the recorded data. The bounding boxes of observed vehicles oftentimes do not match the real shapes - this is caused by stitching images from different cameras observing a single location. [CL17]. Additionally, vehicles moving side to side are sometimes assigned to the same lane, which causes false-positive collisions instead of overtaking manoeuvres. [oT07] Furthermore,

Attribute	*NGSIM*	*highD*
Total duration [hours]	1.5 hours	16.5 hours
Lanes (per direction)	5-6	2-3
Recorded distance [m]	500-640	400-420
Vehicles	9206	110000
Cars	8860	90000
Trucks	278	20000
Driven distance [km]	5071	45000
Driven time [h]	174	447

Table 2.1: Comparison of NGSIM and highD datasets, taken from [KBKE18]

[TTK08] has shown, that the velocity distribution is unrealistic and some data filtering should be applied before using the dataset. While an updated version released by [MP15] exists, some of the glitches can not be fixed merely by filtering. Tracks would need to be extracted again from the original non-public recordings. Besides the data characteristic does not really fit to what one imagines under highway traffic - the vast majority of tracks has average speed below $60 kmh^{-1}$. Another inconvenience is that as NGSIM was produced in the US, the measurements are provided in the imperial unit system.

2.2.2 highD

HighD dataset [KBKE18] was recorder in 2018 on German highways at six different locations and includes more than 110 500 vehicles. The data quality compared to the NGSIM is superior; the positioning error is typically less than ten centimetres. Also, the data characteristics more closely match the intended use case. As shown by [KBKE18], the vehicle velocities in highD dataset are mostly distributed from 60 to $160 kmh^{-1}$, but also some segments with congestion are present.

2.2.3 Dataset derived from highD

From the above, it is clear that highD is a better fit for this thesis. It is significantly bigger, the quality of the recording is superior, and the observed road traffic is closer to the definition of a highway.

Many published results completely ignore the nature of trajectories present in the dataset. This is not a correct approach from my point of view, as predicting trajectories for vehicles changing lanes is much more challenging than to predict a trajectory for a vehicle going straight. That being said lane changing trajectories represent only about 7.5 % of the trajectories present in the dataset. Evaluation on the whole dataset would hence favour a method precise on straight trajectories. For this reason, I have decided to balance the dataset in the sense of the ratio of trajectories with a manoeuvre and straight

	train	validation	test
STAY IN LANE	4921	1316	1120
CUT LEFT	2224	611	492
CUT RIGHT	2730	745	609
Total	9875	2672	2221

Table 2.2: Number of trajectories per manoeuvre type and data part

	train	validation	test
TTLC >= 10	2 361 934	637 107	541 811
8.0 <= TTLC < 10.0	110 395	30 621	24 785
6.0 <= TTLC < 8.0	180 694	49 717	40 755
4.0 <= TTLC < 6.0	237 598	65 061	52 795
2.0 <= TTLC < 4.0	253 710	69 249	56 500
0.0 <= TTLC < 2.0	262 311	71 722	59 573
Total (M)	3 406 642	923 477	776 219

Table 2.3: Number of measurements per TTLC interval and data part

trajectories. I select all trajectories with a manoeuvre [2] and the same number of randomly selected straight trajectories.

These selected trajectories are then split into training, validation and test sets using ratio 65:20:15. Splitting data into sets is done per vehicle, so all trajectories from one vehicle are in one set. This way, independence of testing data set is ensured. In Table 2.2 I show number of trajectories ($|V|$) for each data part divided by the manoeuvre type. In Table 2.3 a similar data are presented, but in this table number of measurements (M) is shown rather than the number of the trajectories. The data are distributed into bins by time to lane change (TTLC).

[2] I consider only trajectories that are longer than 150 frames, the cut must happen at least 175 meters from the beginning of the observable area and no further than 20 meters from the end.

Chapter 3

Overview of current approaches

In this chapter, I present an overview of prior approaches to vehicle trajectory prediction. I split prior methods based on the information they take into account into the following three categories:

- *Physical model* that takes into account only kinematics of the target vehicle and predicts its future position based on physical laws.
- *Manoeuvre based models* that predict trajectory based on detecting a manoeuvre of a target vehicle.
- *Interaction aware models* that interpret the situation on the road and predicts the trajectory while taking into account interdependencies between manoeuvres of vehicles in the scene.

In this thesis I propose one manoeuvre based model in Chapter 5 and two interaction aware models in Chapters 4 and 6.

3.1 Physical models

Physical models represent each vehicle as a dynamic entity, that is governed by laws of physic. The motion of a vehicle is modelled by a set of mathematical equations describing the kinematics. For simplicity, the friction force is neglected, and the vehicle is modelled as an immaterial point moving in 2D space - the velocity vector is assumed to point precisely in the same direction as the vehicle's wheels.

A survey of the kinematics model has been done by Schubert et al. [SRW08]. Here I present three most commonly used ones.

Constant velocity Model

$$\begin{aligned} x(t) &= x(0) + v_t(0)\ t\ \cos\left(\theta(0)\right) \\ y(t) &= y(0) + v_t(0)\ t\ \sin\left(\theta(0)\right) \end{aligned} \tag{3.1}$$

Constant Acceleration Model

$$\begin{aligned} x(t) &= x(0) + v_t(t)\ t\ \cos\left(\theta(0)\right) \\ y(t) &= x(0) + v_t(t)\ t\ \sin\left(\theta(0)\right) \\ v_t(t) &= v_t(0) + a_t(0)\ t \end{aligned} \tag{3.2}$$

Constant yaw rate and acceleration model

$$\begin{aligned} x(t) &= \frac{a_t(0)}{\omega(0)^2}\cos\left(\theta(t)\right) + \frac{v_t(t)}{\omega(0)}\sin\left(\theta(y)\right) + c_x \\ y(t) &= \frac{a_t(0)}{\omega(0)^2}\sin\left(\theta(t)\right) - \frac{v_t(t)}{\omega(0)}\cos\left(\theta(t)\right) + c_y \\ v_t(t) &= v_t(0) + a_t(0)\ t \\ \theta(t) &= \theta(0) + \omega(0)\ t \\ c_x &= x(0) - \frac{v_t(0)}{\omega(0)}\sin\left(\theta(0)\right) - \frac{a_t(0)}{\omega(0)^2}\cos(\theta(0)) \\ c_y &= y(0) + \frac{v_t(0)}{\omega(0)}\cos\left(\theta(0)\right) - \frac{a_t(0)}{\omega(0)^2}\sin(\theta(0)) \end{aligned} \tag{3.3}$$

3.1.1 Advantages and Limitations

A significant advantage of physical models is that they are computationally efficient and easy to interpret. The biggest pitfall is that, as they do not model drivers as decision-making entities, they tend to be unreliable for longer prediction horizon ($t > 1$s) [LVL14].

3.2 Manoeuvre based models

From our own experience we know, that drivers on a highway do follow two main driving patterns, either they stay in the same lane or engage in changing one. Manoeuvre based models are trying to exploit these and other driving patterns and build them into the prediction model. These models usually have some module that classifies the driver's behaviour and passes this information to another module that predicts the trajectory based on the manoeuvre and other vehicles states.

3.2.1 Manoeuvre classification

Many cues can be used to estimate the driver's intention. A very powerful indicator is the physical state of the vehicle (position, speed, acceleration, heading, yaw rate, turn signals, etc.). Moreover, the manoeuvre can be classified based on the information about the road (shape, topology, speed limit, number of lanes, etc.) and drivers behaviour.

Some approaches for manoeuvre classification rely on using a heuristics [HBBCY13] to determinate which manoeuvres are likely to be performed in

the nearby future. More complex approaches involve training a discriminative model such as Decision Trees [YSH14], LSTMs [KOBT16], SVMs [MS05] or Multilayer Perceptron [YK16]. An alternative and successful approach is to model the driver's decision process by a Hidden Markov Model (HMM) [MGBN13], [SH14]. The transition probabilities between single states can be learned from the data as well as the observation model. For a new sequence of observations, the manoeuvre intention is estimated by comparing the likelihood of the observation for each HMM [LVL14]. Bayesian networks [SWA14] have also been used in some cases.

3.2.2 Trajectory generation

The manoeuvre detected in the previous step is used for generating a trajectory prediction based on a priori knowledge about a typical trajectory associated with a manoeuvre. A common approach is to represent the trajectory as 4th / 5th order polynomial [YIK$^+$18]. This polynomial is chosen to fit the initial vehicle condition and the archetypal trajectory. Another approach is to use Gaussian Processes to represent the trajectory. A GP can be learned for each of the manoeuvre categories from training data, and then a future trajectory can be generated from the initial condition [Tay09]. Alternatively, a Rapidly-exploring random tree can be grown from the initial position. The growth is biased by the detected manoeuvre. This approach has the advantage that only physically feasible trajectories are generated [ALL$^+$10].

3.2.3 Advantages and Limitations

As drivers intention is considered in the Manoeuvre based models, they tend to be more reliable for longer horizon predictions than physical models. The obvious danger of this method is that a wrong manoeuvre classification has a significant impact on the predicted trajectory. It is also self-evident that assuming that other drivers do not influence the future trajectory is wrong. Real-world traffic is very interdependent, which is the next group of models trying to address.

3.3 Interaction aware models

The Interaction aware models model not only a single drivers behaviour but also the interactions between one another. There has been a limited number of such methods proposed in the literature. One method how to capture the inter-dependencies described in [DRT18] is based on minimizing an objective function which expresses the cost of a simultaneous manoeuvre assignment to all vehicles in the scene. The cost is based on the feasibility of the future trajectories given the selected manoeuvres. Another approach is to model the dependencies between multiple moving vehicles by Coupled HMMs [BOP97]. As the complexity rises with the number of entities and tends to get very

large, a simplification to asymmetric CHMMs is usually used, assuming that the surrounding vehicles influence the target but not the opposite.

3.3.1 Advantages and Limitations

As Interaction aware models are more complex than the Manoeuvre based models, they can be more reliable. The problem with them is the computational cost, which comes with higher model complexity.

3.4 Overview of currently used metrics

Many trajectory similarity measures do permit misaligned trajectories in terms of time. While this is desirable in some application, our metric should not be forgivable towards temporally misaligned trajectories. For us, there is no difference if the trajectory is correct but shifted, or just bad. We always compare temporally aligned trajectories of the same length. In the rest of the section, I define three kinds of evaluation metrics.

Euclidean Distance

The most profound trajectory similarity measure is median (MED) or average (AED) Euclidean distance. Each point of the predicted trajectory T_v^p is compared to respective point of the ground truth T_v^g, that is,

$$eucl(p^p, p^g) = \sqrt{(x^g - x^p)^2 + (y^g - y^p)^2}$$

$$m_{AvgED}(T^g_{v|t}, T^p_{v|t}) = \frac{\sum_{\tau=1}^{\tau^{max}} eucl(T^g_{v|t}(\tau), T^p_{v|t}(\tau))}{\tau^{max}}$$

$$m_{MedED}(T^g_{v|t}, T^p_{v|t}) = \underset{\tau=1}{\overset{\tau^{max}}{\text{median}}}(eucl(T^g_{v|t}(\tau), T^p_{v|t}(\tau)))$$

Another use of Euclidean distance is in the $MaxED$, which reports the maximal divergence between the trajectories

$$m_{MaxED}(T^g_{v|t}, T^p_{v|t}) = \underset{\tau=1}{\overset{\tau^{max}}{\max}}(eucl(T^g_{v|t}(\tau), T^p_{v|t}(\tau)))$$

Relative hit count

The Relative Hit Count (RHC) metric is based on counting points of the trajectory, that satisfy some condition formally represented by a predicate $e(p^g, p^p) \rightarrow \{0, 1\}$. A typical point predicate can be an Euclidean distance with a threshold

$$e(p^g, p^p) = \begin{cases} 1 & \text{if} \quad eucl(p^g, p^p) < threshold \\ 0 & \text{otherwise} \end{cases}$$

The metric is than defined as

$$m_{RHC}(T^g_{v|t}, T^p_{v|t}) = 1 - \frac{\sum_{\tau=1}^{\tau^{max}} e(T^g_{v|t}(\tau), T^p_{v|t}(\tau)))}{\tau^{max}} .$$

The apparent limitation of this metric is it's susceptibility to the selection of predicate. With too free choice, it can report all trajectories as identical. On the other hand, having too strict threshold yields no point equality at all.

Longest Common Subsequence

The idea of counting points satisfying some predicate is further expanded by LCS. Where RHC takes the absolute number of equal points, the LCS measure is based on finding the longest common subsequence of following equal points. The LCS is than

$$m_{LCS}(T^g_{v|t}, T^p_{v|t}) = 1 - \frac{\max_{0 \le i < j \le \tau^{max}} (j-i) \prod_{\tau=i}^{j} e(T^g_{v|t}(\tau), T^p_{v|t}(\tau)))}{\tau^{max}} .$$

As with RHC, there is a problem of how to select the point equality relation.

Chapter 4
GBT position prediction

A fairly straight forward approach towards trajectory prediction is to predict points alongside the prediction horizon and then interpolate the areas in between. All the coordinates are predicted independently, and thus neither correlation between neighbouring points nor dependency between the lateral and longitudinal coordinate of a single point is modelled. This might lead to unrealistic predicted trajectories and therefore in Chapter 6 I purpose another method which inherently models also these dependencies.

4.1 Model description

In my model, I predict 5 positions in the future, for the whole seconds and use linear interpolation for the positions in between. The lateral and longitudinal position is predicted independently using Gradient Boosted Trees [Fri00] (GBT), hence in total ten GBT is trained. GBT is an ensemble prediction model where m regression trees are trained using gradient boosting - each new tree added to the ensemble is trained to minimize the pseudo-residuals of the so far obtained ensemble. A regression tree is a decision tree where the target variable is a continuous value and hence in the leafs a real number rather than a discrete class. I have decided to use the XGBoost implementation of GBT as it is memory-efficient implementation and can be parallelized, resulting in shorter learning and evaluation time [CG16].

To formalize this, I am looking for a predictor $h\colon \Re^n \to \Re$, which based on a feature vector $x \in \Re^n$ predicts a hidden state $\hat{y} = h(x)$. The hidden state is either lateral or longitudinal coordinate of one of the 5 predicted points. The features captured in the feature vector are described in the following Section 4.2. The objective during training is to minimize the mean absolute error of the hidden state y

$$L_{MAE}(Y, \hat{Y}) = \frac{\sum_{m=1}^{M} |y_m - \hat{y_m}|}{M}.$$

Each of the trained GBT is composed of 100 regression trees. The maximal depth of a single regression tree is 10.

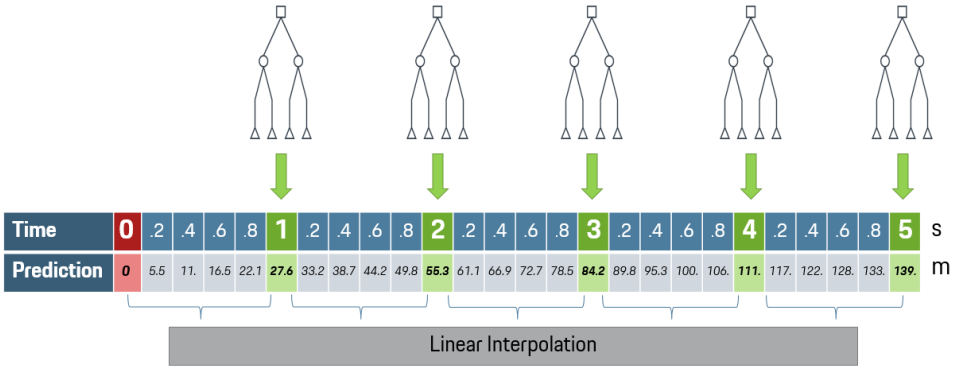


Figure 4.1: Graphical visualisation of the GBT model, example for longitudinal coordinate

4.2 Situation representation

This section describes how a situation on the road is represented by the feature vector. The feature vector is composed of two parts, a description of the target vehicle outlined in the Section 4.2.1 and representation of the surrounding vehicles described in Section 4.2.2.

4.2.1 Target vehicle

The target vehicle is represented by all the physical states describe in Section 2.1.1. Acceleration and velocity are however described only by the longitudinal and latitudinal components, not with the tangential (absolute magnitude). By removing the tangential component, I do not take away any information, as the magnitude of the longitudinal component is for both velocity and acceleration order of magnitude higher and so the tangential velocity is well approximated by this component alone.

In addition to physical states, information about the vehicle type

$$classification \in \{car, truck\}$$

is added. To give the model information about the shape of the road, I add to the representation a state called lane assignment $l(t)$ - dimensionless unit, representing the lateral position of the vehicle on the road with respect to lane markings. Zero represents the centre of reference lane, +1 centre of the lane to left, -1 centre of the lane to the right etc. See Figure 4.2. The centre of the coordinate system is the position of the target vehicle at the current time; the reference lane for the lane assignment is the one the vehicle is driving in at the current time.

Besides these features, a moving average and moving variance over each of the attributes are computed with a time window of 400 ms. As the sampling frequency is $F = 25Hz$ that corresponds to $n_{samples} = 10$ samples.

The feature vector x consists of following components:

- $x(t)$ - longitudinal position
- $y(t)$ - latitudinal position
- $v_x(t)$ - longitudinal velocity
- $v_y(t)$ - latitudinal velocity
- $a_x(t)$ - longitudinal acceleration
- $a_y(t)$ - latitudinal acceleration
- $\theta(t)$ - yaw
- $\omega(t)$ - yaw rate
- $l(t)$ - lane assignment
- $classification \in \{car, truck\}$
- $\mu(x(t), t)$ - moving average of longitudinal position
- $\mu(y(t), t)$ - moving average of latitudinal position
- $\mu(v_x(t), t)$ - moving average of longitudinal velocity
- $\mu(v_y(t), t)$ - moving average of latitudinal velocity
- $\mu(a_x(t), t)$ - moving average of longitudinal acceleration
- $\mu(a_y(t), t)$ - moving average of latitudinal acceleration
- $\mu(\theta(t), t)$ - moving average of yaw
- $\mu(\omega(t), t)$ - moving average of yaw rate
- $\mu(l(t), t)$ - moving average of lane assignment
- $\sigma^2(x(t), t)$ - moving variance of longitudinal position
- $\sigma^2(y(t), t)$ - moving variance of latitudinal position
- $\sigma^2(v_x(t), t)$ - moving variance of longitudinal velocity
- $\sigma^2(v_y(t), t)$ - moving variance of latitudinal velocity
- $\sigma^2(a_x(t), t)$ - moving variance of longitudinal acceleration
- $\sigma^2(a_y(t), t)$ - moving variance of latitudinal acceleration
- $\sigma^2(\theta(t), t)$ - moving variance of yaw
- $\sigma^2(\omega(t), t)$ - moving variance of yaw rate
- $\sigma(l(t), t)$ - moving average of lane assignment

where $t = -3, -2.8, -2.6, \ldots, 0$, $\mu(f(t), t)$ is a functional computing the moving average at time t

$$\mu(f(t), t) = \frac{\sum_{s=0}^{n_{samples}-1} f(t - \frac{s}{F})}{n_{samples}} = \frac{\sum_{s=0}^{9} f(t - \frac{s}{25})}{10}$$

and $\sigma^2(f(t), t)$ is a functional computing the moving variance at time t

$$\sigma^2(f(t),t) = \frac{\sum_{s=0}^{n_{samples}}(f(t-\frac{s}{F}) - \mu(f(t),t))^2}{n_{samples}-1}$$
$$= \frac{\sum_{s=0}^{9}(f(t-\frac{s}{25}) - \mu(f(t),t))^2}{9}$$

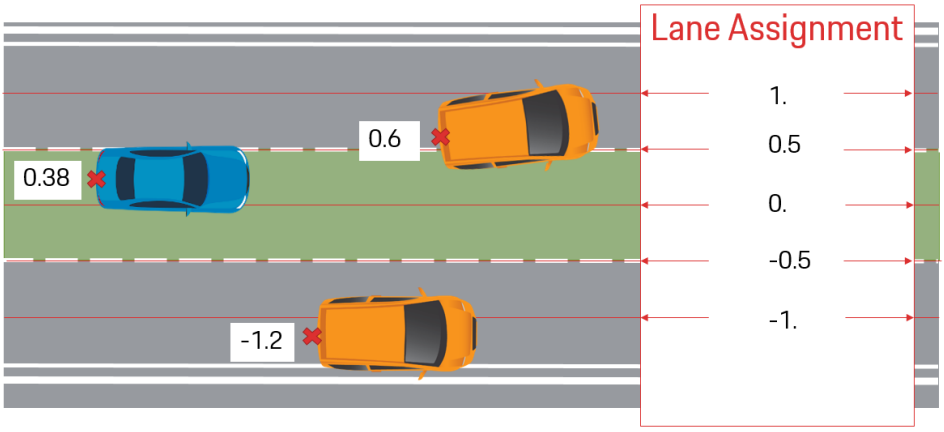


Figure 4.2: Lane assignment visualisation, the green lane is the reference lane.

The representation contains 9 time dependant features, for each of them I also compute moving mean and variance, the history length is 3 seconds with a 5Hz sampling rate. Additionally there is one time non-dependant feature (*classification*). In total the number of features representing the target vehicle is $n_{features} = 9 \cdot 3 \cdot 3 \cdot 5 + 1 = 406$.

4.2.2 Surrounding vehicles

As the driver's behaviour is influenced by other vehicles on the road, the performance of the prediction can be improved if I also represent the surrounding vehicles. I sample the road into 8 segments around the target vehicle, as shown on the Figure 4.3. At most one vehicle is allocated in each segment, if more vehicles are located in the same segment, the closest one in the longitudinal direction of the target vehicle is selected. A vehicle is *alongside* when it overlaps with the target vehicle in the longitudinal direction.

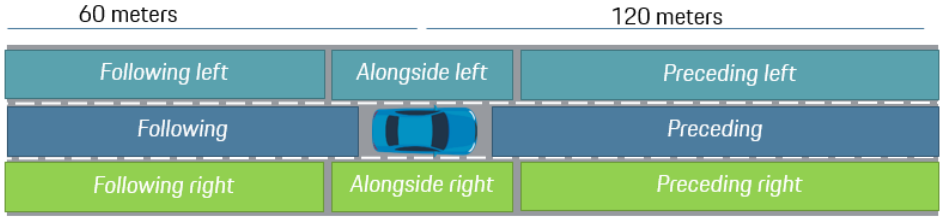


Figure 4.3: Road segments for surrounding vehicles.

For each of the vehicles inside a segment following features are represented:

- x $[m]$ - longitudinal position with respect to the same coordinate center as was used for target vehicle
- v_x $[ms^{-1}]$ - longitudinal speed of the vehicle
- $classification$ of the vehicle

For the time dependent attributes a history is provided with same length and density as with target vehicle. The number of segments is $n_{segments} = 8$, so the total number of features representing the surrounding vehicles is $n_{features} = 8 \cdot (2 \cdot 3 \cdot 5 + 1) = 248$.

It remains to tackle what to do about segments, where no car is present. XGBoost treats all empty records as a missing value, trying to substitute with a value that minimizes the final loss [CG16]. While this can have good results, it might be better to select a special value which represents a missing car. For x a good candidate can be a value that is outside the observable horizon ($x < 60$ for following, $x > 120$ for preceding, see Figure 4.3). For v_x I use some big negative number as there are no cars moving backwards in the data set. A new type is introduced for the $classification$ attribute.

Chapter 5

The FRENET approach

The method described in this chapter is one of the manoeuvre based models, see Section 3.2. I first describe the trajectory generating module in Section 5.1, which produces a predicted trajectory for the vehicle based on a manoeuvre and vehicle state. The trajectory generating module was not developed as part of this thesis and is based on the work of Werling *et al.* [WZKT10]. The main focus here is on the manoeuvre detection block described in Section 5.2.

5.1 Trajectory generating module

The trajectory generating module takes the state of a vehicle as input and produces a prediction of trajectory for vehicle up to the prediction time horizon. It first transforms the initial vehicle state to Frenet reference frame (Section 5.1.1), then identifies the final latitudinal position (Section 5.1.2), generates possible trajectories (Section 5.1.3), selects the one that minimizes a cost functional (Section 5.1.4).

5.1.1 Frenet reference frame

Many methods for trajectory prediction proposed in the literature do not cope with curvy streets, as they are tested only on straight roads. To easily integrate the information about the road shape, the coordinate system is transformed into the Frenet frame of reference. Frenet frame of reference is a Curvilinear Coordinate System which expresses position by the longitudinal position along the curve $s(t)$, and the lateral distance to the curve $d(t)$. Therefore the vehicle state at time step t is defined as $x_t = (s(t); d(t))$ [HST19].

Imagine a scenario of a car keeping lane on a straight road versus car keeping lane in a curve. Even though in both scenarios the driver's behaviour is the same, the first can be reliable predicted only based on the car's history, wheres for precise prediction in the second case, the shape of the road needs to be considered. Transforming the coordinate system to Frenet allows us to treat both cases the same, provided good enough information about the road structure is available.

5.1.2 Selecting final position

To sample possible trajectories, I select a final position of the target vehicle (at the end of the prediction horizon) in the $d(t_f) = d_f^*$ coordinate (latitudinal). It is selected based on the predicted manoeuvre. We assume that if no manoeuvre is in motion, the traffic participants are naturally driving at the centre of a lane. In case a vehicle is changing lanes, it is assumed that the vehicle is aiming for the centre of the target lane.

5.1.3 Generating the trajectory

The longitudinal $d(t)$ and the lateral $s(t)$ components of the trajectory are approximated by 5th and 4th order polynomials, respectively. We assume that the first and second-order derivative of $d(t)$ in the final configuration is zero. For the s coordinate the second-order derivative is assumed to be zero and the first order to be $s'(t) = vel_t(0) + t \cdot acc_t(0)$. Based on the final state, the initial state (zero to second-order derivatives) and the time t_f of the final configuration, the coefficients of the polynomials are computed [BM08].

Finding coefficients for quintic polynomial - latitudinal coordinate

Given time t_f and $d(t_f) = d_f^*$ as estimated in previous step, the coefficient $c_0, c_1, \ldots, c_5$ are found by solving the following set of equations, which has a closed form solution:

$$\begin{aligned} & d(t) = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0 \\ \text{s.t. } & d(0) = d_{line} \\ & \dot{d}(0) = v_t(0) \sin(\theta_{diff}) \\ & \ddot{d}(0) = a_t(0) \sin(\theta_{diff}) + a_n(0) \cos(\theta_{diff}) \\ & d(t_f) = d_f^* \\ & \dot{d}(t_f) = 0 \\ & \ddot{d}(t_f) = 0 \end{aligned} \tag{5.1}$$

where d_{line} is the vehicle distance from line at the initial state, $a_n(0) = \omega(0) v_t(0)$, θ_{line} is yaw of the line at the time zero and $\theta_{diff} = \theta(0) - \theta_{line}$.

Finding coefficients for quartic polynomial - longitudinal coordinate

Given time t_f, the coefficient $p_0, p_1, \ldots, p_4$ are found by solving the following set of equations, which has a closed form solution:

$$\begin{aligned} & s(t) = p_4 t^4 + p_3 t^3 + p_2 t^2 + p_1 t^1 + p_0 \\ \text{s.t. } & s(0) = 0 \\ & \dot{s}(0) = v_t(0) \sin(\theta_{diff}) \\ & \ddot{s}(0) = a_t(0) \cos(\theta_{diff}) - a_n(0) \sin(\theta_{diff}) \\ & \dot{s}(t_f) = v_t(0) + a_t(0) t \\ & \ddot{s}(t_f) = 0 \end{aligned} \tag{5.2}$$

where $a_n(0) = \omega(0)v_t(0)$, θ_{line} is yaw of the line at the time zero and $\theta_{diff} = \theta(0) - \theta_{line}$.

Trajectory sampling

Setting the time of the final configuration t_f to be equal to the prediction horizon would mean, that all vehicles must complete the manoeuvre before the end of the prediction horizon. As this is clearly not the case, several possible trajectories with varying t_f shall be generated. A typical lane change duration is around 4.6 second [TZ07] the range of t_f is therefore set to $t_f \in (0, 4.6 + \tau^{max})$.

5.1.4 Selecting the best trajectory

From trajectories generated in the previous step, the best one is selected based on a cost functional. As human drivers tend to choose trajectories that have low jerk (a second-order derivative of speed) [Hog84], trajectories that minimizes the integral of jerk are preferred. At the same time, I want to penalize big t_f. Otherwise, a trajectory with the biggest t_f will be the one minimizing the jerk. The cost functional in use is a linear combination of those two factors, that is,

$$\begin{aligned} J &= K_S J_s + K_D J_d + K_T t_{end} \\ J_s &= \int_0^{t_{end}} \frac{\dddot{s}(t)^2}{2} dt \\ J_d &= \int_0^{t_{end}} \frac{\dddot{d}(t)^2}{2} dt \end{aligned} \tag{5.3}$$

where $K_S = 0.25, K_D = 0.25, K_T = 0.5$ are weights of particular components of the cost functional.

When the best trajectory is selected, it is transformed back from the Frenet frame to Cartesian coordinates.

5.2 Manoeuvre classification

The trajectory generating module needs information about the manoeuvre as an input. This information needs to be very reliable as misclassification can lead to significant error in the prediction.

An important topic when using manoeuvre detection is how to label data; in other words, how to generate ground truth. A common practice is to base the labelling on time to lane change ($TTLC$) and mark samples with TTLC smaller than a selected threshold thr as a manoeuvre. The direction of the manoeuvre is crucial, so usually samples are classified into three categories - $C = CUTLEFT, CUTRIGHT, STAYINLANE$.

$$man^g(TTLC) = \begin{cases} CUTRIGHT & \text{if } TTLC < thr \text{ and target lane right} \\ CUTLEFT & \text{if } TTLC < thr \text{ and target lane left} \\ STAYINLANE & \text{otherwise} \end{cases}$$

The threshold value is usually set in the range $thr \in (3, 6)$.

This approach however completely ignores the subsequent trajectory predicting module. A classifier trained using this ground truth can be sub-optimal when the approach as whole is evaluated on the predicted trajectories. It can happen that while the classifier predicts the correct label in sense of the determined ground truth, a trajectory generated with some other label could have lower error. For this reason I label the data so that the ground truth is the manoeuvre that minimizes root mean squared latitudinal error, that is, using the following formula:

$$err(T^g_{v|t}, T^{p\_frenet}_{v|t}) = \sqrt{\frac{\sum_{\tau=1}^{\tau^{max}} (y^g_{v|t+\tau} - y^{p\_frenet}_{v|t|\tau})^2}{\tau^{max}}}$$

$$man^g(v,t) = \arg\min_{m \in C} err(T^g_{v|t}, T^{p\_frenet}_{v|t})$$

In the Figure 5.1, I plot the ratio of individual categories with respect to TTLC. In the left plot, there are vehicles targeting lane on the left in the future. The blue colour depicts the ratio of those samples for which the optimal label is CUT LEFT. The right plot is analogous. As we can see the transition from cut to no cut labels is not abrupt, but rather linear in between TTLC 4-5s. Had I used labelling using thresholds, the ground truth for part of those samples would be wrong.

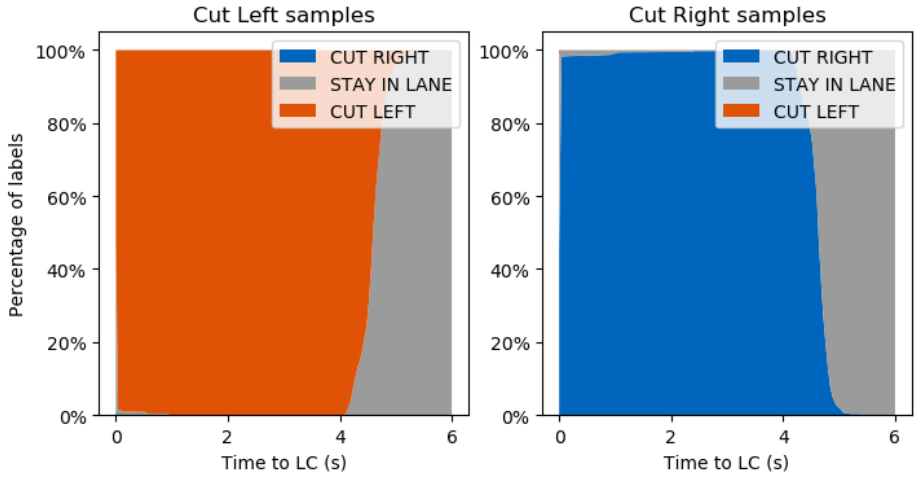


Figure 5.1: Distribution of labels minimizing the lateral error as function of TTLC.

5.2.1 Baseline manoeuvre classifier

Our base line classifier is a simple model based on thresholding the local lane assignment ($local\_la$) and lateral velocity (vel_y). Local lane assignment is the relative position of the target vehicle inside its current lane with zero being the middle of current lane and -0.5, 0.5 being the lateral positions of the right and left lines, respectively. The thresholds are set to $DIST\_THR = 0.2$ and $SPEED\_THR = 0.3$.

Input : $local\_la$, vel_y
Parameter : $DIST\_THR$, $SPEED\_THR$
Output : detected manoeuvre
if $local\_la > DIST\_THR$ *and* $vel\_y > SPEED\_THR$ **then**
  return CUT LEFT
else if $local\_la <$ -$DIST\_THR$ *and* $vel\_y <$ -$SPEED\_THR$ **then**
  return CUT RIGHT
else
  return STAY IN LANE
end

Algorithm 1: Thresholding manoeuvre classifier

5.2.2 Gradient Boosted Trees

I will use GBT classifier to classify the manoeuvres. Again I am using the XGBoost implementation [CG16]. Input features are the same as were described in Section 4.2, so 3 seconds history of target vehicles and also surrounding vehicles is present. During training outputs of the GBT are transformed using softmax defined by (5.4) where s_i is output corresponding to i-th class. The training objective is to minimize cross-entropy loss of the softmax (5.5):

$$\hat{p}_i = \frac{e^{s_i}}{\sum_{k=1}^{N} e^{s_k}} \tag{5.4}$$

$$\begin{aligned} H &= -\sum_{i}^{C} p_i \log(\hat{p}_i) \\ \text{where } p_i &= \begin{cases} 1 & \text{if true class is i} \\ 0 & \text{else} \end{cases} \end{aligned} \tag{5.5}$$

The GBT is composed of 100 trees, and maximal depth is 10. Data are split into training, validation and test sets using ratio 65:20:15. Splitting data into separate sets is done per vehicle, so all trajectories from one vehicle are in one set. This way, independence of testing data set is ensured. Even though the dataset was initially balanced in the sense of trajectories containing cut and straight ones (c.f. Section 2.2.3), the ratio of STAY IN LANE samples is much higher then CUT LEFT and CUT RIGHT, as only fraction of cut trajectory actually has label CUT RIGHT/CUT LEFT. Therefore before training the training and validation sets are balanced, so that the resulting

ratio of CUT LEFT:CUT RIGHT:STAY IN LANE is 1:1:4. This setting proved experimentally to give the best results.

5.2.3 Manoeuvre classification evaluation

In Table 5.1 I report the percentage of correctly predicted samples out of the respective category. We can observe that baseline classifier has slightly better recall on the STAY IN LANE, but in the other two categories, it is significantly worse. Very important is to keep small the number of CUT LEFT left classified as CUT RIGHT and vice versa. Such misclassification is manifested with the highest error in the final trajectory prediction evaluation, as the vehicle is going to the other lane than the model predicts. We can note that fortunately for both classifiers, this type of error is not frequent.

		ground truth		
		CUT LEFT	STAY LANE	CUT RIGHT
GBT	CUT LEFT	82.21%	1.11%	0.05%
	STAY LANE	17.68%	97.40%	19.02%
	CUT RIGHT	0.11%	1.49%	80.93%

Table 5.1: Results of manoeuvre classification using baseline classifier and GBT. Percentages of total number of respective ground truth label presented. The numbers on the diagonal are hence recall.

Based on these results, one can assume that the GBT classifier will lead to better overall results of the prediction. This is truly the case as we will see in the Chapter 7.

For better insight in Figure 5.2 I present the distribution of predicted manoeuvres using GBT with respect to ground truth manoeuvre type and TTLC. Note that this plot is not directly comparable with the one in Figure 5.1, because here on the y-axis the ratio out of samples with the respective label is shown (the blue and orange area), wheres in Figure 5.1 the base is the number of samples with the corresponding future target line. That is also why only TTLC < 4.5s is shown here, as with the higher TTLC the number of samples with the corresponding ground truth label is decreasing fast, and the results become not representative. We can note that the classification is almost flawless for $TTLC < 3$s, at which point the recall starts to decrease dramatically. The correct classification of samples in between $3 < TTLC < 5$s is the most difficult part.

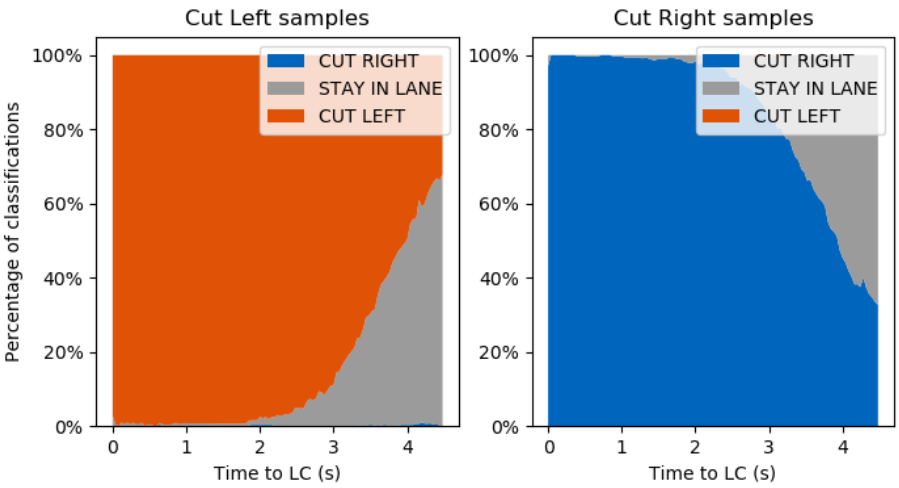


Figure 5.2: Distribution of classified manoeuvres using GBT as function of TTLC.

Chapter 6

Eigentrajectories

In face recognition, a common approach to represent shape of a face is to use Point Distributed Model (PDM) for reducing the number of variables. PDM is a method of describing a geometrical shape using a set of points developed by Cootes and Taylor [CT04]. Model of variation of these points is computed by applying Principal Component Analysis (PCA) to the covariance matrix of the original shapes, stacked into a single matrix. The eigenvectors which PCA yields describing this lower-dimensional space are in the context of computer vision commonly referred to as Eigenfaces. This form of face representation is used in Active Appearance Models or Active Shape Models [CTCG95], which can fit a PDM model into a picture. In [SG07] Saragih *et al.* are predicting coefficients of a previously fitted PDM model based on features from an image. This way, they are able to retrieve an Active Appearance Model from a picture.

A similar approach might be adapted to express and predict vehicle trajectories. As a trajectory is a geometrical shape, it can be described by PDM. In this chapter, I show how to do so by obtaining eigenvectors from a set of trajectories using PCA and observe some properties of the derived eigenvectors. Subsequently, I train models which predict a linear combination of these eigenvectors given a data point, to produce the final trajectory prediction.

6.1 Eigentrajectories

First I arrange all $\tau^{max} = 5$ seconds[1] long segments of each trajectory from the training set into a single matrix $X[M \times O]$ as depicted in Figure 6.1. The sampling frequency is the same as in the original dataset (25Hz), this is how the set of trajectories differs from T^g, where the sampling frequency is 1Hz. The dimensions of this matrix are $M \times O$, where $M = \sum_{v=1}^{N} \Delta_{v_i}$ is number of complete measurement[2] and $O = (\tau^{max} \cdot F + 1) \cdot 2 = (5s \cdot 25Hz + 1) \cdot 2 = 256$ is length of one trajectory flattened into a vector. $v_1, v_1, \ldots, v_N$ references the particular vehicle in the dataset. I am using slightly different notation here

[1] prediction horizon length

[2] As defined in Section 2.1 a complete measurements is such, that both 3s history and 5s future trajectory were observed. Δ_v is number of such measurements belonging to vehicle v.

for sake of better readability, the time index of the point on the trajectory is in brackets instead of being written as lower index so $x_{v_1|t^0_{v_1}} \hat{=} x_{v_1}(t^0_{v_1})$.

Subsequently, PCA is used to find a low-dimensional approximation of the

$$\begin{pmatrix}
x_{v_1}(t^0_{v_1}) & x_{v_1}(t^1_{v_1}) & \dots & x_{v_1}(t^{max}_{v_1}) & \dots & x_{v_2}(t^{max}_{v_2}) & \dots & x_{v_N}(t^{max}_{v_N}) \\
y_{v_1}(t^0_{v_1}) & y_{v_1}(t^1_{v_1}) & \dots & y_{v_1}(t^{max}_{v_1}) & \dots & y_{v_2}(t^{max}_{v_2}) & \dots & y_{v_N}(t^{max}_{v_N}) \\
x_{v_1}(t^1_{v_1}) & x_{v_1}(t^2_{v_1}) & \dots & x_{v_1}(t^{max+1}_{v_1}) & \dots & x_{v_2}(t^{max+1}_{v_2}) & \dots & x_{v_N}(t^{max+1}_{v_N}) \\
y_{v_1}(t^1_{v_1}) & y_{v_1}(t^2_{v_1}) & \dots & y_{v_1}(t^{max+1}_{v_1}) & \dots & y_{v_2}(t^{max+1}_{v_2}) & \dots & y_{v_N}(t^{max+1}_{v_N}) \\
\dots & \dots & & \dots & & \dots & & \dots \\
x_{v_1}(t^0_{v_1} + \tau^{max}) & x_{v_1}(t^1_{v_1} + \tau^{max}) & \dots & x_{v_1}(t^{max}_{v_1} + \tau^{max}) & \dots & x_{v_2}(t^0_{v_2} + \tau^{max}) & \dots & x_{v_N}(t^0_{v_N} + \tau^{max}) \\
y_{v_1}(t^0_{v_1} + \tau^{max}) & y_{v_1}(t^1_{v_1} + \tau^{max}) & \dots & y_{v_1}(t^{max}_{v_1} + \tau^{max}) & \dots & y_{v_2}(t^0_{v_2} + \tau^{max}) & \dots & y_{v_N}(t^0_{v_N} + \tau^{max}) \\
x_{v_1}(t^1_{v_1} + \tau^{max}) & x_{v_1}(t^2_{v_1} + \tau^{max}) & \dots & x_{v_1}(t^{max+1}_{v_1} + \tau^{max}) & \dots & x_{v_2}(t^1_{v_2} + \tau^{max}) & \dots & x_{v_N}(t^1_{v_N} + \tau^{max}) \\
y_{v_1}(t^1_{v_1} + \tau^{max}) & y_{v_1}(t^2_{v_1} + \tau^{max}) & \dots & y_{v_1}(t^{max+1}_{v_1} + \tau^{max}) & \dots & y_{v_2}(t^1_{v_2} + \tau^{max}) & \dots & y_{v_N}(t^1_{v_N} + \tau^{max}) \\
\dots & \dots & & \dots & & \dots & & \dots \\
x_{v_1}(t^{max}_{v_1} + \tau^{max}) & x_{v_1}(t^{max+1}_{v_1} + \tau^{max}) & \dots & x_{v_1}(t^{max}_{v_1} + \tau^{max}) & \dots & x_{v_2}(t^{max}_{v_2} + \tau^{max}) & \dots & x_{v_N}(t^{max}_{v_N} + \tau^{max}) \\
y_{v_1}(t^{max}_{v_1} + \tau^{max}) & y_{v_1}(t^{max+1}_{v_1} + \tau^{max}) & \dots & y_{v_1}(t^{max}_{v_1} + \tau^{max}) & \dots & y_{v_2}(t^{max}_{v_2} + \tau^{max}) & \dots & y_{v_N}(t^{max}_{v_N} + \tau^{max})
\end{pmatrix}$$

Figure 6.1: Trajectories stacked into a single matrix

trajectories in X, i.e. $X \approx \hat{X} = AW$, where $A[M \times n]$ are coefficients and $W[n \times O]$ is a matrix composed of n eigenvectors of a covariance $C = X_c^T X_c$ with largest eigenvalues, where X_c are centered points from X. I shall call these principal components eigentrajectories and use them to express all trajectories in the original dataset as their linear combination.

6.1.1 Feasibility

Before progressing further with the model, I shall validate, that the trajectories can be expressed by a reasonable number of eigentrajectories ($n < 20$). To evaluate this, I show the average pointwise Euclidean distances between the original trajectories in the training set and its projection to the eigentrajectories and back. In the Figure 6.2 dependency of this error on the number of components used is shown. As visible with $n > 10$ the average error is less than 1 cm which is sufficient and hence the trajectory representation using a reasonable number of eigentrajectories is viable. In Appendix A visualisation of few trajectories is attached.

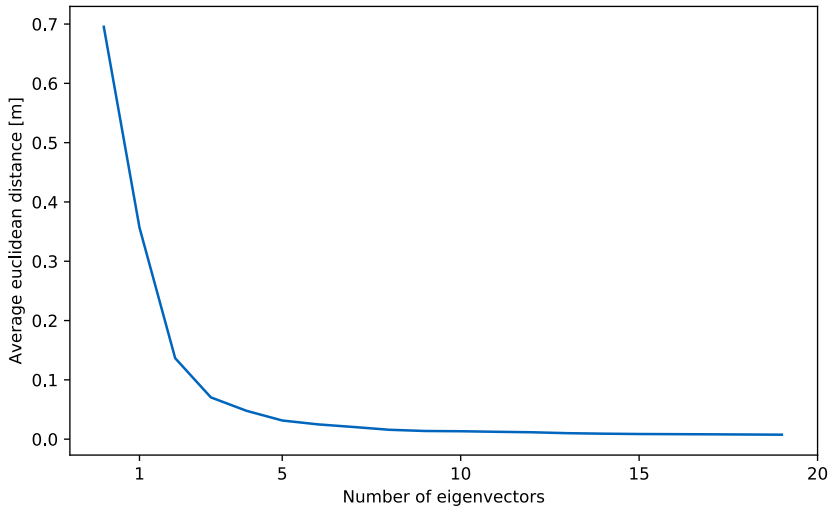


Figure 6.2: Average pointwise Euclidean distance with respect to number of eigentrajectories used to represent the original trajectory.

6.1.2 Interpretation of Eigentrajectories

A nice property of eigenfaces is that they are somewhat interpretable. Figure 6.3 shows exemplar representation of faces found by PCA. From the visualization of eigenfaces (images on the right), one can infer what kind of face type they represent.

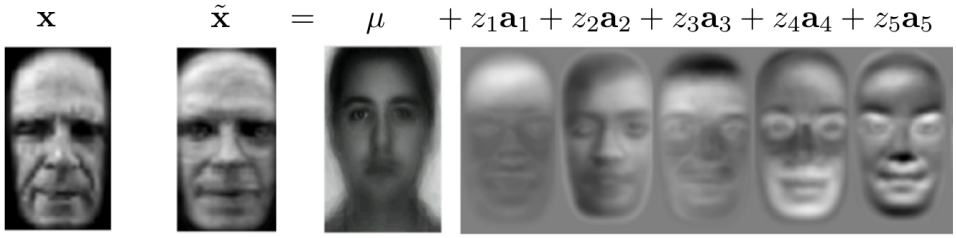


Figure 6.3: Visualisation of eigenfaces extracted by PCA, adopted from [Li11].

It would be interesting to observe, whether similar interpretability is also

captured in eigentrajectories. A reasonable hypothesis is that a eigentrajectories themselves have a shape of some archetypal trajectories, from which all the other trajectories are composed of. In the Figure 6.4 5 eigentrajectories with highest eigen value are shown. There is no straight forward interpretation of what each eigentrajectory might symbolize.

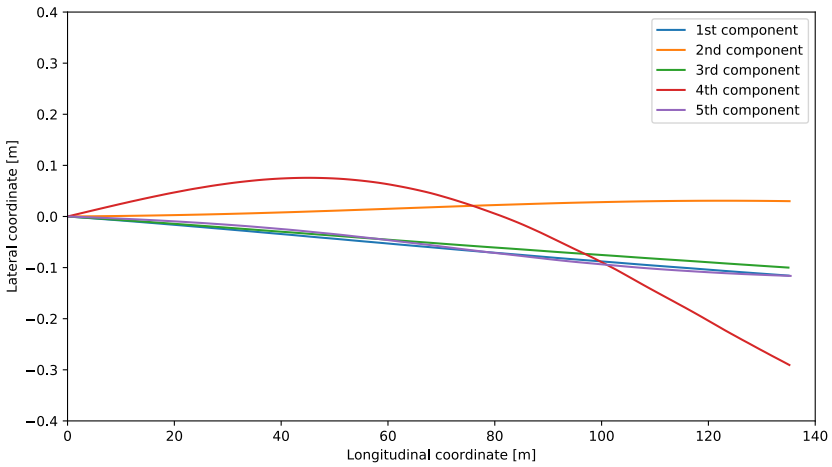


Figure 6.4: Visualisation of 5 eigentrajectories with largest eigen value. There is no clear interpretation of what each of the eigentrajectories should symbolize.

As stated in the Chapter 3, many assume that an essential factor for trajectory prediction are manoeuvres performed by drivers. If this is really the case, one would expect that this vital information would also be captured by eigentrajectories. One might hope that the distribution of a coefficient of some of the relatively important eigentrajectories would be dependent on the manoeuvre. When analyzing the ground truth coefficient, such dependency is truly observable. I show that the coefficient value of 2nd eigentrajectory is dependant on time to lane change (TTLC). Some weak dependency can also be observed between TTLC and other eigentrajectories, but 2nd eigentrajectory was chosen because the dependency manifests the most.

To demonstrate this, I first compute TTLC for each sample. For trajectories where lane change does not occur until the end of the observable road segment, the value is left blank. Subsequently, I group these samples by 1s interval and plot the distribution of the coefficient value for each of the group. In Figure 6.5 this distribution is shown for samples with $TTLC < 7$. The coefficient value distribution is approximated by a histogram with bin size $b_s = 0.1$. The dependency is clearly visible as when the TTLC is approaching zero, the distribution mean is approaching ± 20.

Usually, it is important not only to know whether a manoeuvre is in motion but also the direction. For the trajectory prediction, it is crucial to distinguish between right and left lane changes. In Figure 6.6 I present a comparison of coefficient values distribution for situation when vehicles move from left to right ($CUT\ RIGHT$) and when vehicles move from right to left (CUT

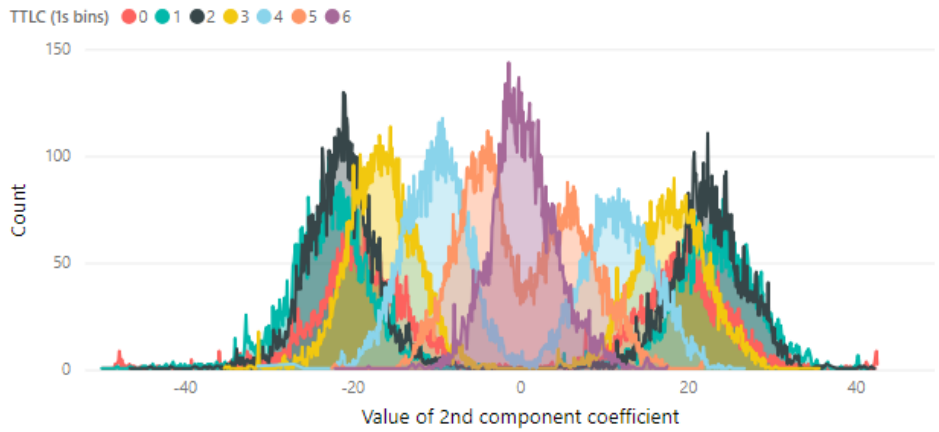


Figure 6.5: Distribution of the coefficient of 2nd component with respect to TTLC. Colors depict data grouped by value of TTLC with 1s long interval. All samples in one group have $t <= TTLC < t+1$, where t is the number in legend. The expected value of coefficient of 2nd component clearly depends on TTLC.

LEFT). Yet again, there is a clear separation of those two classes.

In the Figure 6.5 I have completely ignored samples with $TTLC > 6s$ and samples where the manoeuvre does not occur in the observable future. In Figure 6.6 I also present distribution of coefficients for those samples as *STAY IN LANE*. It can be seen that this class is also separated from the other two with a mean at 0. We can also notice that the distribution of coefficient for samples with $6 < TTLC < 7$ in Figure 6.5 is more similar to the *STAY IN LANE* class distribution.

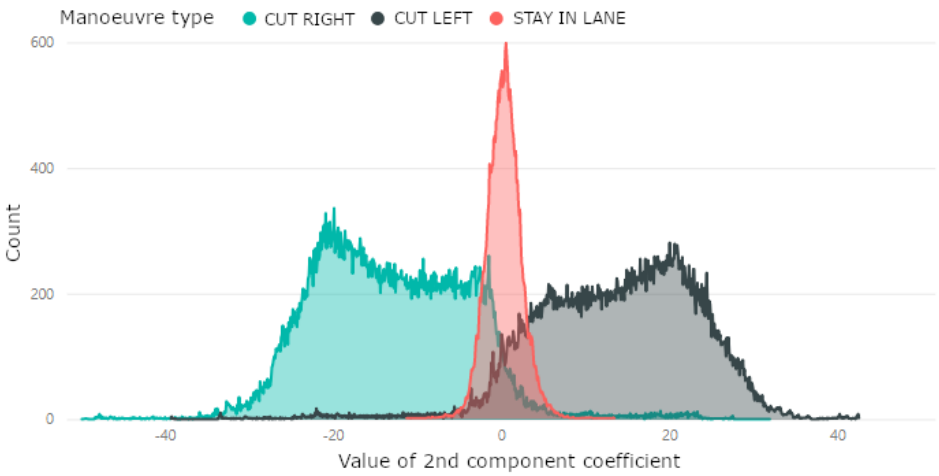


Figure 6.6: Distribution of coefficient of 2nd component with respect to manoeuvre type. The expected value of the coefficient clearly depends on the manoeuvre type.

6.1.3 Properties of Eigentrajectories

As seen in the previous section in Figure 6.4, the inferred Eigentrajectories are smooth. Therefore the trajectories predicted by this method are also guaranteed to be smooth, as the linear combination of the eigentrajectories can not introduce any bumps. This quality will manifest later on during physical feasibility analysis (c.f. Section 7.2).

Another nice property of the linearity is that the resulting trajectory is not so susceptible to bad coefficient prediction. Had I, for example, decided to predict the coefficients of a 5th order polynomial, by which a trajectory is frequently represented, the pressure on the correct prediction of high order coefficient would be significantly higher.

6.2 Model Description

The goal is to create such a model that predicts correctly the coefficients of the linear combination of the eigentrajectories given a vehicle measurement. As I can obtain the optimal values of the coefficients, I can use them as ground truth and train a model that predicts them. A separate GBT is trained for each of the coefficients independently. I am using the XGBoost implementation [CG16]. Feature representation of the measurement remains the same as in Chapter 4. The objective of i-th GBT[3] during training is to minimize mean absolute error

$$L_{MAE}(W^i, \hat{W}^i) = \frac{\sum_{m=1}^{M} |w_m^i - \hat{w_m^i}|}{M},$$

where $W^i[M \times 1]$ are the optimal values of the i-th coefficient and $\hat{W}^i[M \times 1]$ are the predicted values[4]. Each Boosted tree is composed of 100 trees.

The downside of this method is that the objective to minimize MAE, but our true goal is to minimize the Euclidean distance between the projection and the ground truth.

Number of eigentrajectories

As predicting coefficients for all of the eigentrajectories would be impractical, I tune the optimal number of eigentrajectories by looking at the testing error[5] when $n \in \{1 \dots 10\}$ eigentrajectories is used. Dependency of the RMSE (7.1) on n evaluated on the validation set is shown in Figure 6.7.

As one can see with $n = 5$ the curve flattens and adding additional eigentrajectories does not decrease the error anymore. This observation is in alignment with Figure 6.2, where I also observed that more than 5 eigentrajectories do not improve the representation.

[3] GBT predicting the i-th coefficient

[4] M is number of complete measurements in the training set.

[5] average pointwise Euclidean distances between the original trajectory and its projection using the predicted coefficients

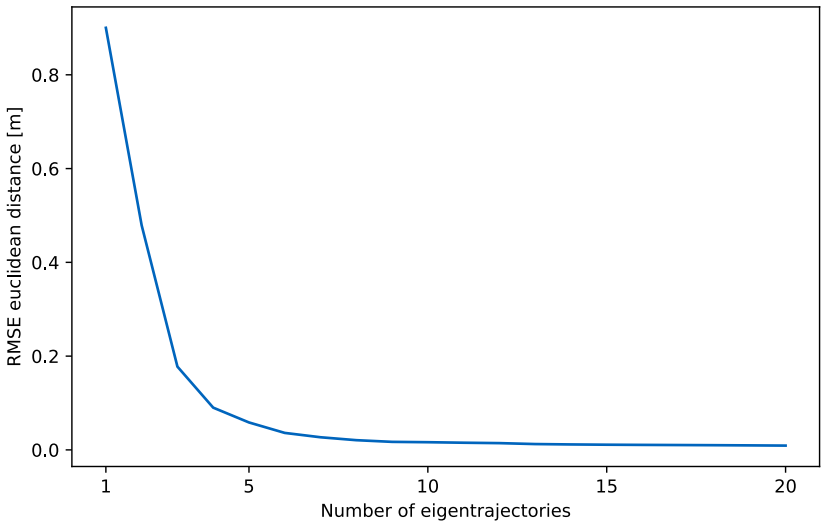


Figure 6.7: RMSE Euclidean distance evaluated on the validation set as function of number of eigentrajectories.

6.2.1 Classifying manoeuvres

I discussed in Section 6.1.2 possible interpretation of the coefficient of 2nd eigentrajectory as a manoeuvre representation. One can wonder whether this dependency could be leveraged and classify a manoeuvre given the coefficient value. First, I study whether this can be done on the optimal coefficient values and then compare this result to the predicted values.

For purpose of this analysis I define $CUT\ LEFT/CUT\ RIGHT$ sample as one where $TTLC < 5s$. I consider a manoeuvre (either $CUT\ LEFT$ or $CUT\ RIGHT$) as a positive class and $STAY\ IN\ LANE$ as negative. The classification is done using two thresholds t_L, t_R:

$$manoeuvre(c) = \begin{cases} CUTRIGHT, & \text{if } c > t_R \\ CUTLEFT, & \text{if } c < t_L \\ STAYINLANE, & \text{otherwise} \end{cases}$$

Based on the analysis done in Section 6.1.2 I will set the thresholds to $t_L = -15$ and $t_R = 15$. One could choose a different threshold which would minimize false positive (FPR) or false negative rate (FNR), but this selection is reasonable choice. The results are presented in Table 6.1 as a confusion matrix. In rows are the ground truth labels, in columns are the predicted categories. Values are presented as total number of samples in the respective cell and theirs percentage as a ratio of the total number of samples. Results are presented for the ground truth coefficients and the predicted ones. Accuracy of the classification is 90.74%, FPR = 5.20% and FNR = 10.65% for the ground truth and acc = 87.71%, FPR = 14.80%, FNR = 5.01% for the predicted coefficients.

In Appendix A figures with distribution of predicted 2nd coefficient values are attached for comparison with the figures presented in Section 6.1.2.

	predicted manoeuvre					
ground truth	CUT RIGHT		CUT LEFT		STAY IN LANE	
	count	perc.	count	perc.	count	perc.
CUT RIGHT	28966	8.90%	616	0.19%	14980	4.60%
CUT LEFT	805	0.25%	27443	8.43%	10787	3.32%
STAY IN LANE	1256	0.39%	1673	0.51%	238830	73.41%

	predicted manoeuvre					
ground truth	CUT RIGHT		CUT LEFT		STAY IN LANE	
	count	perc.	count	perc.	count	perc.
CUT RIGHT	24600	7.56%	368	0.11%	19594	6.02%
CUT LEFT	535	0.16%	22310	6.86%	16190	4.98%
STAY IN LANE	1490	0.46%	1803	0.55%	238466	73.29%

Table 6.1: Results of manoeuvre classification using ground truth (top) and predicted (bottom) coefficients of 2nd eigentrajectory.

The distribution of the predicted values is more or less matched well when $TTLC < 3$s, but behaves differently for $3 \leq TTLC < 6$. This is in alignment with what I noted in Section 5.2.3, that prediction of a manoeuvre and thus the whole trajectory is difficult, especially when $TTLC \in (3, 5)$ seconds.

Chapter 7
Evaluation

In this chapter I will evaluate each of the proposed methods:

- **Frenet-Base** - Frenet model where manoeuvres are predicted using the baseline model described in Section 5.2.1
- **Frenet-GBT** - Frenet model with manouvre prediciton model done by GBT as described in Section 5.2.2
- **GBT-PP** - model described in Chapter 4
- **Eigentrajectories** - model described in Chapter 6

I will also report results for two baseline approaches:

- **CV** - constant velocity model (3.1)
- **CYRA** - constant velocity and yaw rate (3.3)

In Section 7.1 I will quantitatively evaluate how well the predicted trajectory fits the ground truth, while in Section 7.2 I will focus on the predicted trajectory from a physical feasibility point of view. In all the experiments apart from Table 7.4 the data are evaluated on the test set as described in Section 2.2.3.

7.1 Performance

7.1.1 Definition

In Section 3.4 I have presented an overview of currently used metrics. Each of these metrics compares two trajectories, e.g., a ground truth and a predicted trajectory. However it does not solve the problem of characterize prediction errors for all test trajectories by a single value. Here the most straight forward approach is to compute the mean absolute error (MAE). A more common approach used in the literature is root mean squared error (RMSE), which I will use as well. I will report RMSE of $m_{AvgED}(T_v^p, T_v^g)$ (7.1) and also the RMSE of Euclidean error (7.2) computed from errors observed at each whole second $t \in \{1 \dots 5\}$ as this evaluation protocol is usually presented in the

literature. Furthermore I will report the same for longitudinal, equations (7.3) and (7.4), and latitudinal error, equations (7.5) and (7.6) , as this can give us more insight into how is the error distributed for particular model.

$$RMSE\_avgED = \sqrt{\frac{1}{M}\sum_{v\in V}\sum_{t\in\Delta_v}\left(m_{avgED}(T^p_{v|t}, T^g_{v|t})\right)^2} \tag{7.1}$$

$$RMSE\_ED(\tau) = \sqrt{\frac{1}{M}\sum_{v\in V}\sum_{t\in\Delta_v}\left(eucl(T^p_{v|t}(\tau), T^g_{v|t}(\tau))\right)^2} \tag{7.2}$$

$$RMSE\_avgLon = \sqrt{\frac{1}{M}\sum_{v\in V}\sum_{t\in\Delta_v}\left(\frac{1}{\tau^{max}}\sum_{\tau=1}^{\tau^{max}}|x^p_{v|t+\tau} - x^g_{v|t|\tau}|\right)^2} \tag{7.3}$$

$$RMSE\_Lon(\tau) = \sqrt{\frac{1}{M}\sum_{v\in V}\sum_{t\in\Delta_v}\left(x^p_{v|t+\tau} - x^g_{v|t|\tau}\right)^2} \tag{7.4}$$

$$RMSE\_avgLat = \sqrt{\frac{1}{M}\sum_{v\in V}\sum_{t\in\Delta_v}\left(\frac{1}{\tau^{max}}\sum_{\tau=1}^{\tau^{max}}|y^p_{v|t+\tau} - y^g_{v|t|\tau}|\right)^2} \tag{7.5}$$

$$RMSE\_Lat(\tau) = \sqrt{\frac{1}{M}\sum_{v\in V}\sum_{t\in\Delta_v}\left(y^p_{v|t+\tau} - y^g_{v|t|\tau}\right)^2} \tag{7.6}$$

7.1.2 Results

In tables 7.1, 7.2 and 7.3 I report the results as described above. For the Euclidean error I also report the results of the state of the art neural network approach as of May 2018 - CS-LSTM [DT18], as of June 2019 - NLS-LSTM [MYVN19], and finally the current one (as of May 2020) - MHA-LSTM(+f) [MYVN20].

	$RMSE\_Lon(\tau)$					$RMSE\_avgLon$
	1	2	3	4	5	
CV	0.18	0.68	1.48	2.55	3.87	1.60
CYRA	0.19	0.73	1.59	2.74	4.18	1.64
Frenet-Base	0.31	1.20	2.63	4.50	6.67	2.47
Frenet-GBT	0.31	1.20	2.63	4.51	6.69	2.48
GBT-PP	**0.08**	**0.30**	**0.75**	1.41	2.31	0.87
Eigentrajectories	0.22	0.44	0.81	**1.40**	**2.21**	**0.85**

Table 7.1: Results of proposed models in longitudinal direction, $RMSE\_avgLon$ and $RMSE\_Lon(\tau)$.

	$RMSE\_Lat(\tau)$					$RMSE\_avgLat$
	1	2	3	4	5	
CV	0.09	0.32	0.63	0.99	1.36	0.53
CYRA	0.08	0.30	0.67	1.20	1.88	0.66
Frenet-Base	0.11	0.41	0.84	1.23	1.52	0.63
Frenet-GBT	0.07	0.25	0.53	0.80	0.99	0.43
GBT-PP	**0.02**	**0.12**	**0.29**	**0.51**	0.75	0.29
Eigentrajectories	0.07	0.14	0.31	0.52	**0.74**	**0.29**

Table 7.2: Results of proposed models in latitudinal direction, $RMSE\_avgLat$ and $RMSE\_Lat(\tau)$.

	$RMSE\_ED(\tau)$					$RMSE\_avgED$
	1	2	3	4	5	
CV	0.20	0.75	1.61	2.74	4.11	1.80
CYRA	0.21	0.79	1.72	3.00	4.58	1.94
Frenet-Base	0.33	1.27	2.76	4.67	6.84	2.70
Frenet-GBT	0.32	1.23	2.69	4.58	6.76	2.61
CV+Frenet-GBT	0.19	0.72	1.57	2.67	4.00	1.74
GBT-PP	**0.08**	**0.32**	**0.80**	1.50	2.43	0.98
Eigentrajectories	0.23	0.47	0.87	**1.49**	**2.33**	**0.96**
CS-LSTM	0.22	0.61	1.24	2.10	3.27	-
NLS-LSTM	0.20	0.57	0.1.14	1.90	2.91	-
MHA-LSTM(+f)	0.06	0.09	0.24	0.59	1.18	-

Table 7.3: Results of proposed models as Euclidean distance, $RMSE\_avgED$ and $RMSE\_ED(\tau)$.

As we can see the best results has the Eigentrajectories approach followed closely by GBT-PP. Frenet-GBT has a decent latitudinal error, but the improved manoeuvre classification is not contributing to lower the longitudinal error, and hence in the Euclidean distance the model is deficient. A possible way to improve this would be to combine this model with some other model and predict longitudinal and latitudinal coordinate separately. For the Euclidean distance, I have also included results of such a model, CV + Frenet, where the latitudinal coordinate is predicted by the constant velocity model.

Unfortunately, none of the introduced approaches has a lower error than the current state-of-the-art approach. I did not train the model myself, but I have been in contact with the authors of the paper, and my evaluation protocol is comparable. The only difference is that they are not balancing the dataset in the sense of the ratio of trajectories with a manoeuvre and keep lane trajectories[1]. In theory, this could improve my results compared to them. As authors have also included results evaluated only on trajectories that have cut, I can compare those values and get comparable results. In Table 7.4 I present results of the Eigentrajectories model and MHA-LSTM

[1]Details about balancing the dataset are described in Section 2.2.3.

for trajectories containing CUT LEFT. My results are still worse compared to state of the art; however, the results are better compared to evaluation on the whole dataset.

		τ				
		1	2	3	4	5
Eigentrajectories	$RMSE\_Lon(\tau)$	0.25	0.51	0.95	1.68	2.64
	$RMSE\_Lat(\tau)$	0.09	0.18	0.40	0.69	0.98
MHA-LSTM(+f)	$RMSE\_Lon(\tau)$	0.20	0.32	0.42	0.88	1.74
	$RMSE\_Lat(\tau)$	0.03	0.07	0.19	0.45	0.78

Table 7.4: Comparison of Eigentrajectories and MHA-LSTM(+f) on trajectories containing CUT LEFT, $RMSE\_Lon(\tau)$ and $RMSE\_Lat(\tau)$ reported.

7.2 Physical feasibility

An important aspect of the predicted trajectories is that they are physically feasible and can occur in the context of a highway. In this section, I evaluate each of the proposed methods from this point of view. The evaluation is done by computing velocity and acceleration (first and second derivatives of the position) for each of the prediction point. In Figure 7.1 a calculation of longitudinal velocity and acceleration for a single trajectory both predicted (green) and ground truth (red) is depicted. The figure shows a predicted trajectory that would be physically infeasible, as latitudinal velocity from time step $t = 2$ to $t = 3$ is too big, resulting in unrealistic latitudinal acceleration. This analysis should show whether and how often such and other non-feasible trajectories the particular method does produce.

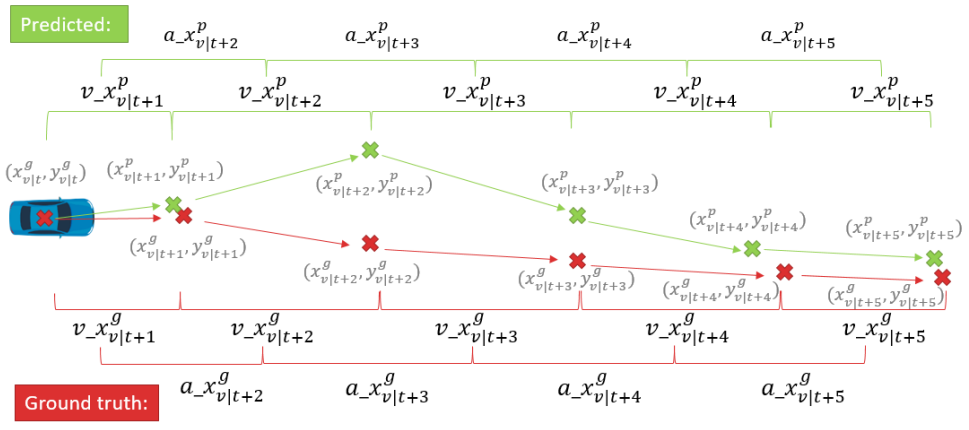


Figure 7.1: Single trajectory visualisation for physical feasibility analysis.

	velocity				acceleration			
	lon		lat		lon		lat	
	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
CV	0.57	0.55	0.11	0.03	0.26	0.08	0.07	0.01
CYRA	0.59	0.70	0.16	0.08	0.27	0.09	0.09	0.02
Frenet-Base	0.84	1.93	0.14	0.04	0.35	0.21	0.10	0.01
Frenet-GBT	0.84	1.93	0.13	0.03	0.35	0.21	0.10	0.01
CYRA	0.59	0.70	0.16	0.08	0.27	0.09	0.09	0.02
GBT-PP	0.46	0.48	0.11	0.03	0.58	0.60	0.14	0.03
Eigentrajectories	0.35	0.22	0.09	0.02	0.20	0.05	0.06	0.01

Table 7.5: Mean with variance of absolute error computed on prediction and ground truth of velocity and acceleration.

7.2.1 Definition

For this analysis, I define an error as the absolute value of the difference between the predicted and ground truth quantities. Subsequently, I compute the mean and standard deviation of the error defined in (7.7). I am also visually comparing the distribution of predicted quantities with the distribution of the same quantity in the original trajectories (distribution of the original values, not the error).

$$
\begin{aligned}
L_{v\_x}(v,t,\tau) &= \left| v\_x^{g}_{v|t+\tau} - v\_x^{p}_{v|t|\tau} \right| \\
\mu_{v\_x} &= \frac{1}{M \cdot \tau^{max}} \sum_{v \in V} \sum_{t \in \Delta_v} \sum_{\tau=1}^{\tau^{max}} L_{v\_x}(v,t,\tau) \\
\sigma^2_{v\_x} &= \frac{1}{M \cdot \tau^{max}} \sum_{v \in V} \sum_{t \in \Delta_v} \sum_{\tau=1}^{\tau^{max}} \left(L_{v\_x}(v,t,\tau) - \mu_{v\_x} \right)^2
\end{aligned}
\tag{7.7}
$$

7.2.2 Results

The results are presented in the Table 7.5. In every respect, the best model matching the ground truth are Eigentrajectories.

I will now have a look at the distribution of predicted values and the ground truth. Data are binned with bin widths $\{1, 0.1\}$ for longitudinal and lateral velocities, respectively and 0.01 for acceleration. Distribution for velocity is quite well matched by all the proposed model, the respective, Figure A.5 and Figure A.6 in appendix. In acceleration we can however observe some interesting tendencies. Looking at the distribution of longitudinal acceleration (c.f. Figure 7.2, *prediction_pca*) the Eigentrajectory model is more conservative with variance $\sigma^2_{eig} = 0.06$. The variance of ground truth is $\sigma^2 g = 0.14$. In contrast the variance of the GBT-PP (c.f. Figure 7.2, *prediction_gbt*) is $\sigma^2_{gbt} = 0.92$. That is in agreement with what we observed in Table 7.5, the error of longitudinal acceleration of GBT-PP was one of

the highest observed. In lateral acceleration presented in Figure 7.3 we can observe similar behavior. Eigentrajectory model is more conservative while GBT-PP deviates more than than the ground truth.

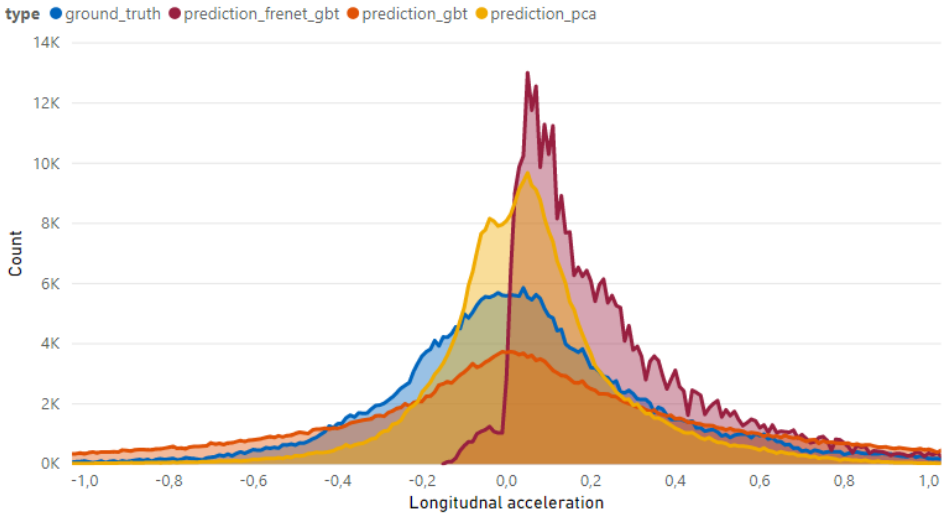


Figure 7.2: Distribution of predicted longitudinal acceleration by model and the ground truth.

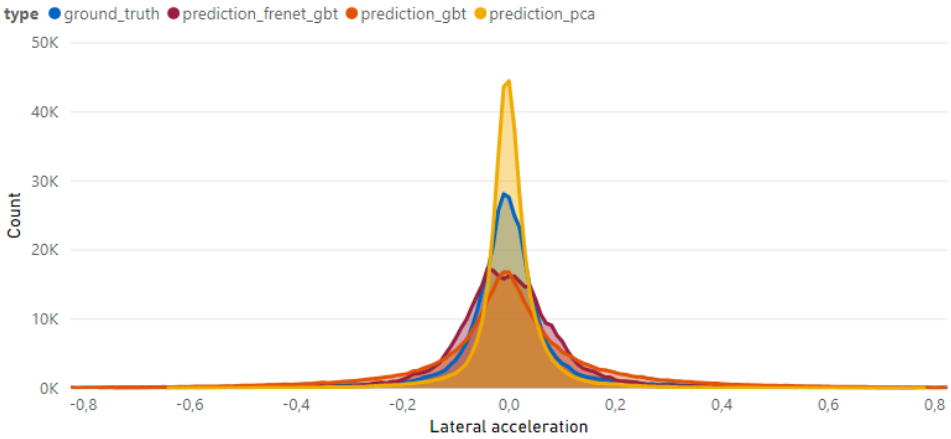


Figure 7.3: Distribution of predicted latitudinal acceleration by model and the ground truth.

Based on this evaluation, the Eigentrajectory model has been proven to be the most conservative and produce the most *naturalistic* trajectories.

Chapter 8

Conclusion

In this thesis, I have introduced three novel methods for trajectory prediction. In Chapter 4 a straight forward approach using Gradient Boosted Trees to directly predict the future position based on the history was presented. In subsequent Chapter 5 I improved results of a trajectory prediction module in the Frenet frame of reference by enhancing manoeuvre detection.

Finally, in Chapter 6 I have described a novel method of how to predict the future trajectory using Eigentrajectories. This techniques first extracts n most important eigenvectors (eigentrajectories) using Principal Component Analysis from the set of trajectories. As the original trajectories can be expressed as a linear combination of these eigentrajectories with sufficient precision, a set of GBT is trained, each to predict a coefficient for one of these eigentrajectories. During inference, the future trajectory is predicted by first obtaining the coefficients from GBT and than computing the linear combination of Eigentrajectories. In contrast with predicting coefficients of a 5th order polynomial, by which a trajectory is often represented, this approach is favourable because the combination of the eigentrajectories is linear and hence there is not such high pressure on correctness. When predicting a coefficient of a high order polynomial, even a slight imprecision can have a detrimental effect on the final prediction. I have also shown that it is possible to interpret the eigentrajectories, particularly I have demonstrated that the distribution of coefficient belonging to 2nd eigentrajectory is dependent on time to lane change.

Eigentrajectories and the direct GBT position prediction proved to have better quantitative results than the Frenet approach. None of the proposed approaches were able to produce better results than the current state of the art methods based on neural networks, however the proposed approaches have better interpretability. In physical feasibility evaluation, the Eigentrajectories dominated the rest, so this model overall proved to be the best of the proposed.

In future work, it would be interesting to change the space of possible eigentrajectories to further improve the interpretability of the model. One possible approach to do this is to use Greedy Principal Component Analysis, where the space of principal components is limited to vectors that are present in the original data set [FH06]. Another approach to increase the interpretability might be to use Non-negative Matrix Factorization. When this method is

used in face recognition instead of PCA, the retrieved basis vectors represent common parts extracted from the original faces, such as eyes, noses and lips [GG12]. A similar phenomenon might be observed with Eigentrajectories.

In this thesis, I have only considered lateral manoeuvres. It might also be interesting to consider longitudinal manoeuvres (e.g. rapid braking and acceleration) and study whether a similar dependency between a coefficient of some eigentrajectory and longitudinal manoeuvre can be observed.

Overall I have accomplished the given task and introduced novel methods with acceptable results and good interpretability.

Bibliography

[ALL+10] G. S. Aoude, B. D. Luders, K. K. H. Lee, D. S. Levine, and J. P. How. Threat assessment design for driver assistance system at intersections. In *13th International IEEE Conference on Intelligent Transportation Systems*, pages 1855–1862, 2010.

[BKM+19] Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. *arXiv preprint arXiv:1911.07602*, 2019.

[BM08] Luigi Biagiotti and Claudio Melchiorri. *Analytic Expressions of Elementary Trajectories*, pages 15–57. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[BOP97] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 994–999, 1997.

[CBL+19] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[CG16] Tianqi Chen and Carlos Guestrin. Xgboost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016.

[CL17] Benjamin Coifman and Lizhe Li. A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset. *Transportation Research Part B: Methodological*, 105:362–377, 11 2017.

[CLS+19] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse:

3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[CT04] T. F. Cootes and C.J. Taylor. Statistical models of appearance for computer vision, 2004.

[CTCG95] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38 – 59, 1995.

[DRT18] Nachiket Deo, Akshay Rangesh, and Mohan M. Trivedi. How would surround vehicles move? A unified framework for maneuver classification and motion prediction. *CoRR*, abs/1801.06523, 2018.

[DT18] Nachiket Deo and Mohan M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. *CoRR*, abs/1805.06771, 2018.

[FH06] Vojtěch Franc and Václav Hlaváč. *Greedy Kernel Principal Component Analysis*, pages 87–105. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[Fri00] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[GG12] Nicolas Gillis and François Glineur. A multilevel approach for nonnegative matrix factorization. *Journal of Computational and Applied Mathematics*, 236(7):1708 – 1723, 2012.

[HBBCY13] Adam Houenou, Philippe Bonnifait, Véronique Berge-Cherfaoui, and Wenbing Yao. Vehicle trajectory prediction based on motion model and maneuver recognition. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4363–4369, 2013.

[Hog84] Neville Hogan. An organizing principle for a class of voluntary movements. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 4 11:2745–54, 1984.

[HST19] Yeping Hu, Liting Sun, and Masayoshi Tomizuka. Generic prediction architecture considering both rational and irrational driving behaviors. *CoRR*, abs/1907.10170, 2019.

[KBKE18] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125, 2018.

[KMB+] Robert Krajewski, Tobias Moers, Julian Bock, Lennart Vater, and Lutz Eckstein. The round dataset: A drone dataset of road user trajectories at roundabouts in germany. submitted.

[KOBT16] Aida Khosroshahi, Eshed Ohn-Bar, and Mohan Trivedi. Surround vehicles trajectory analysis with recurrent neural networks. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2267–2272, 11 2016.

[Li11] Fei-Fei Li. Lecture 2: A case study of computer vision – face recognition, 09 2011.

[LVL14] Stephanie Lefevre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, 1, 07 2014.

[MGBN13] Thomas Michalke, Claudius Gläser, Lutz Bürkle, and Frank Niewels. The narrow road assistant - next generation advanced driver assistance in inner-city. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 10 2013.

[MP15] Marcello Montanino and Vincenzo Punzo. Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. *Transportation Research Part B Methodological*, 80:82–106, 07 2015.

[MS05] Hiren M. Mandalia and Mandalia Dario D. Salvucci. Using support vector machines for lane-change detection. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 49(22):1965–1969, 2005.

[MYVN19] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi. Non-local social pooling for vehicle trajectory prediction. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 975–980, 2019.

[MYVN20] Kaouther Messaoud, Itheri Yahiaoui, Anne Verroust, and Fawzi Nashashibi. Attention based vehicle trajectory prediction. *IEEE Transactions on Intelligent Vehicles*, PP, 04 2020.

[MZZ+19] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6120–6127, 2019.

[oT07] US Department of Transportation. Ngsim – next generation simulation. `https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm`, 2007. Accessed: January 30, 2020.

[QHT+17] Jannik Quehl, Haohao Hu, Omer Sahin Tas, Eike Rehder, and Martin Lauer. How good is my prediction? Finding a similarity measure for trajectory prediction evaluation. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[SG07] Jason Saragih and Roland Goecke. A nonlinear discriminative approach to aam fitting. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 01 2007.

[SH14] Thomas Streubel and Karl Hoffmann. Prediction of driver intended path at intersections. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 134–139, 06 2014.

[SRW08] R. Schubert, E. Richter, and G. Wanielik. Comparison and evaluation of advanced motion models for vehicle tracking. In *2008 11th International Conference on Information Fusion*, pages 1–6, 2008.

[SWA14] M. Schreier, V. Willert, and J. Adamy. Bayesian, maneuver-based, long-term trajectory prediction and criticality assessment for driver assistance systems. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 334–341, 2014.

[Tay09] C Tay. *Analysis of Dynamic Scenes: Application to Driving Assistance*. PhD thesis, Institut polytechnique de Grenoble, 09 2009.

[TTK08] Christian Thiemann, Martin Treiber, and Arne Kesting. Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data. *Transportation Research Record: Journal of the Transportation Research Board*, 2088(1):90–101, Jan 2008.

[TZ07] Tomer Toledo and David Zohar. Modeling duration of lane changes. *Transportation Research Record*, 1999:71–78, 01 2007.

[WZKT10] Moritz Werling, Julius Ziegler, Sören Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 987 – 993, 06 2010.

[YIK+18] Keisuke Yoneda, Toshiki Iida, TaeHyon Kim, Ryo Yanase, Mohammad Aldibaja, and Naoki Suganuma. Trajectory optimization and state selection for urban automated driving. *Artificial Life and Robotics*, 09 2018.

[YK16] S. Yoon and D. Kum. The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 1307–1312, 2016.

[YSH14] Yang Zheng, A. Sathyanarayana, and J. H. L. Hansen. Threshold based decision-tree for automatic driving maneuver recognition using can-bus signal. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2834–2839, 2014.

Appendix A

Additional figures

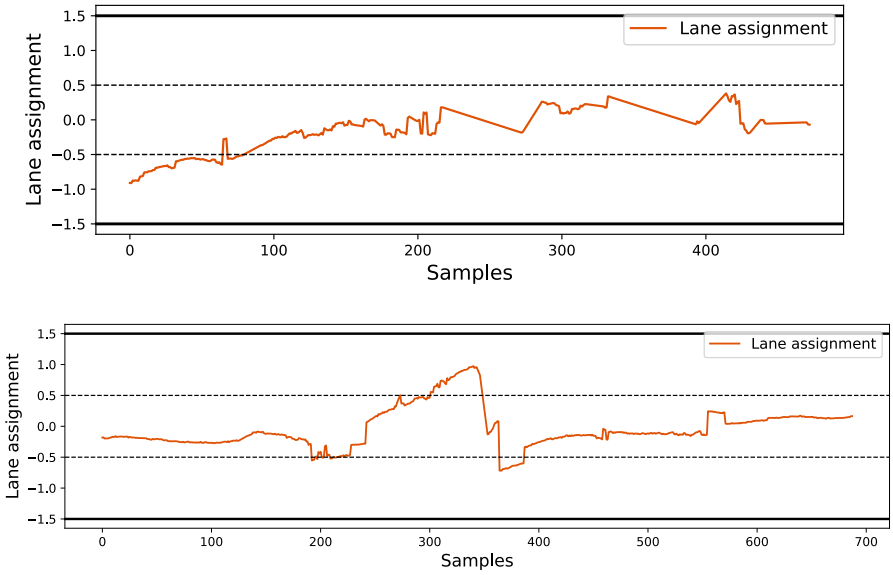
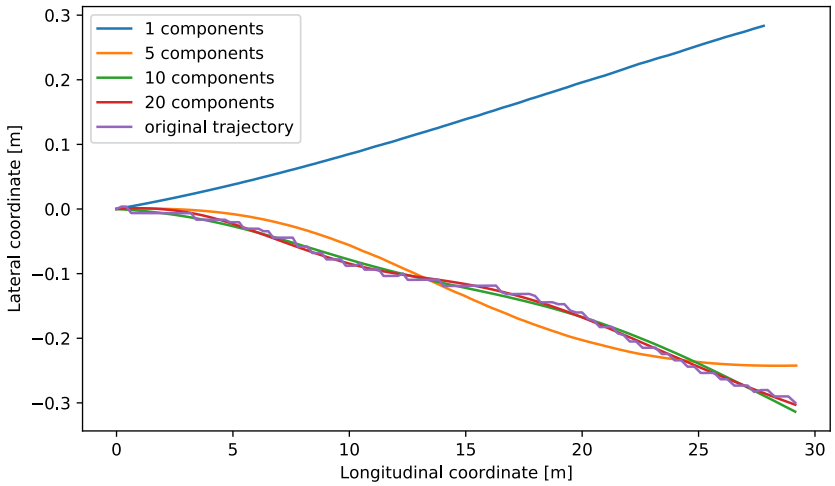
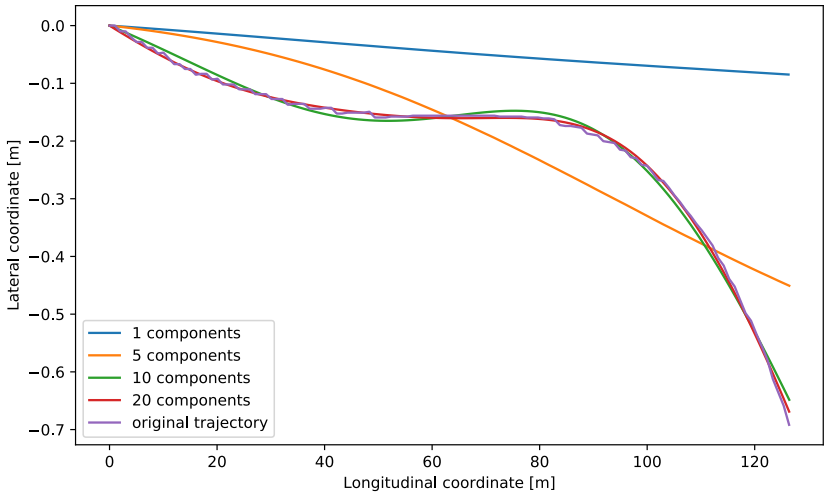


Figure A.1: Demonstration of poor data quality, lane assignment (see Figure 4.2) signal for single vehicle trajectory as function of time (sampled with F=25Hz).


1 components
5 components
10 components
20 components
original trajectory
Lateral coordinate [m]
Longitudinal coordinate [m]
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
5
10
15
20
25
30


1 components
5 components
10 components
20 components
original trajectory
Lateral coordinate [m]
Longitudinal coordinate [m]
0.0
−0.1
−0.2
−0.3
−0.4
−0.5
−0.6
−0.7
0
20
40
60
80
100
120

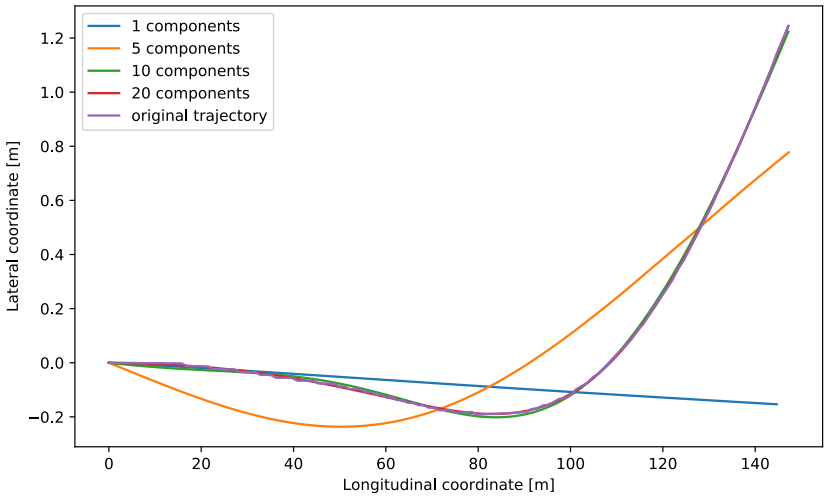


Figure A.2: Trajectory representation using eigentrajectories.

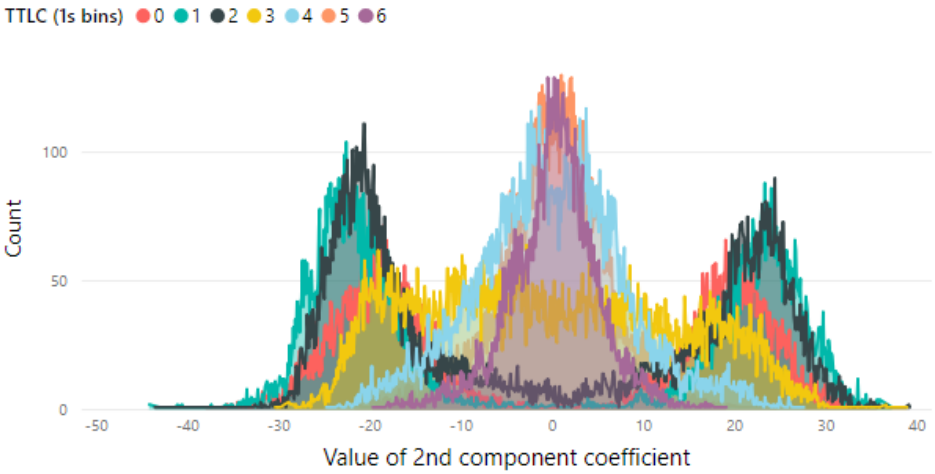


Figure A.3: Distribution of predicted coefficients of 2nd component with respect to TTLC.

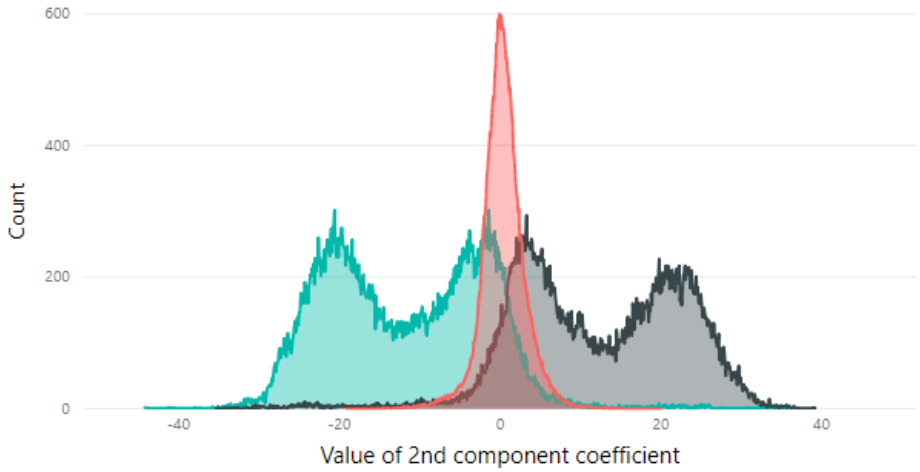


Figure A.4: Distribution of prediceted coefficient of 2nd component with respect to manoeuvre type.

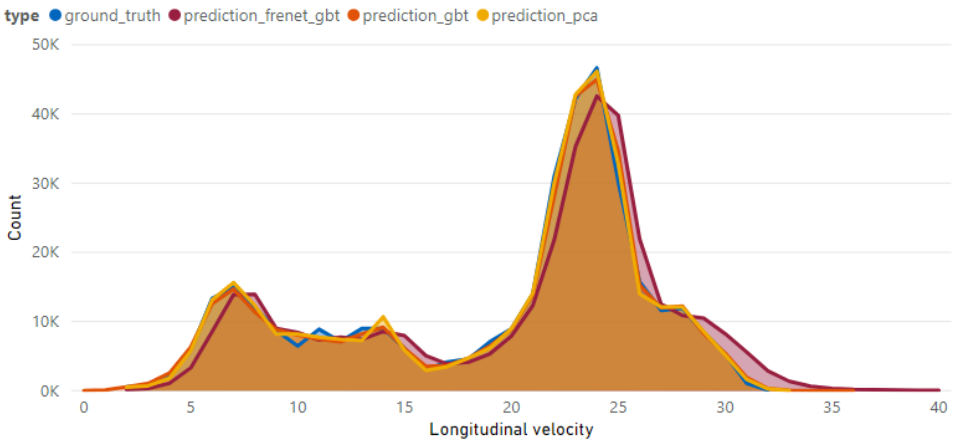


Figure A.5: Distribution of predicted longitudinal velocity by model and the ground truth.

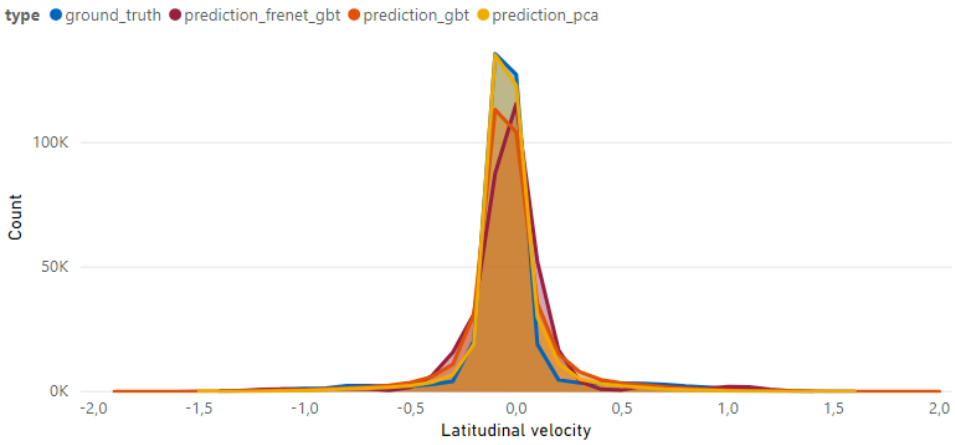


Figure A.6: Distribution of predicted latitudinal velocity by model and the ground truth.