Inference of Traffic Regulations at Intersections Based on Trajectory Data

Master's Thesis of

Martin Trat

Institute of Measurement and Control Systems
Karlsruhe Institute of Technology

Reviewer:	Prof. Dr.-Ing. Christoph Stiller
Second Reviewer:	Prof. Dr. J. Marius Zöllner

Karlsruhe, June 2020

Erklärung

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht zu haben und die Satzung der Universität Karlsruhe (TH) zur Sicherung guter wissenschaftlicher Praxis beachtet zu haben.

Martin Trat

Karlsruhe, 19.06.2020

Danksagung

An dieser Stelle möchte ich mich bei den Personen bedanken, die mich während der Erstellung dieser Arbeit unterstützt haben.

Dank gilt Herrn Professor Dr.-Ing. Christoph Stiller und Herrn Professor Dr. J. Marius Zöllner dafür, dass sie die Erstellung dieser Arbeit ermöglicht haben.

Auch möchte ich mich bei Frau Annika Meyer für ihre Hilfsbereitschaft, konstruktive Anregungen und zahlreiche Diskussionen, sowie für das entgegengebrachte Vertrauen bei der Betreuung meiner Arbeit.

Ganz besonders möchte ich mich bei meiner Familie, insbesondere bei meiner Frau Sonja Trat, bedanken. Sie ermöglichte mir großartige Freiräume und brachte viel Verständnis für meine Hingabe für diese Arbeit auf. Mir ist bewusst, dass ich mich auch bei der weiteren wissenschaftlichen Arbeit auf ihre Unterstützung verlassen kann.

Abstract

Many state-of-the-art approaches in autonomous driving make use of highly precise maps. Among other annotations, these maps must contain information regarding traffic regulations.

In this thesis an offline approach aiming at the inference of traffic regulations at German intersections on a sub-lane level is suggested. The representation and likelihood-based inference of regulations is realized using Hidden Markov Models. These are parametrized and evaluated on an artificially created set of trajectories crossing several intersections. In a real-world context, the trajectories could be opportunistically collected from a sensor-equipped fleet of vehicles over an extended period of time. In a series of experiments, a suitable trajectory representation is determined and the approach is tested and improved. Classification performance is evaluated in a cross-validation manner. Mean test F_1 scores associated to the best classification results range between 0.809 and 0.832. High performance is achieved in the context of the traffic regulations `priority`, `stop` and `traffic-light`. However, regarding the `yield` and `yield-to-right` regulation, challenges remain. As initial results are promising, the approach is worth being developed and improved further.

Kurzfassung

Zahlreiche moderne Lösungen im Bereich Autonomes Fahren greifen auf hochpräzises Kartenmaterial zurück. Neben anderen Informationen muss das Kartenmaterial solche über Verkehrsregeln enthalten.

In dieser Arbeit wird eine Offline-Lösung für die Inferenz von Verkehrsregeln an Deutschen Kreuzungen entwickelt. Mithilfe dieser Lösung werden für jeden Fahrstreifen einer Kreuzung Klassifikationsentscheidungen für jede mögliche Zielrichtung, welche von diesem Fahrstreifen aus erreichbar ist, getroffen. Verkehrsregeln werden mithilfe von Hidden-Markov-Models repräsentiert und, basierend auf errechneten Likelihood-Werten, bestimmt. Die Modelle werden mithilfe künstlich erzeugter Trajektorien von Kreuzungsüberquerungen parametrisiert und evaluiert. Unter realen Umständen würden solche Daten opportunistisch und sensorgestützt von einer Fahrzeugflotte über einen längeren Zeitraum hinweg gesammelt werden. In einer Reihe von Experimenten wird eine geeignete Trajektorienrepräsentation festgelegt und der Klassifikationsansatz getestet und verfeinert. Die Klassifikationsperformanz des Ansatzes wird mithilfe eines Kreuzvalidierungsverfahren bestimmt. Mittlere F_1-Scores zur Quantifizierung der besten Ergebnisse unter den erzielten Testergebnissen variieren zwischen 0.809 und 0.832. Bezüglich der Verkehrsregeln, welche mithilfe von `Vorfahrts`- und `Stoppschildern`, sowie `Lichtsignalanlagen` kommuniziert werden, werden hohe Klassifikationsleistungen erreicht. Allerdings bestehen Schwierigkeiten bei der Klassifikation im Zusammenhang mit den Verkehrsregeln `Vorfahrt achten` und `Rechts vor Links`. Da die initial erzielten Ergebnisse vielversprechend sind, wird empfohlen diesen Ansatz in zukünftigen Arbeiten weiterzuentwickeln und zu verbessern.

Contents

List of Abbreviations, Symbols and Notations

Abbreviations

Amitran	Assessment Methodologies for Information and Communication Technology in Multimodal Transport from User Behavior to CO2 Reduction
HMM	Hidden Markov Model
IDM	Intelligent Driver Model
OSM	Open Street Maps
SUMO	Simulation of Urban Mobility

Symbols

Units

h	Hour
Hz	Hertz
km	Kilometer
m	Meter
s	Second

Variables

Hidden Markov Models

P	Probability
λ	Hidden Markov Model
$S = \{S_1, S_2, \dots, S_{K^S}\}$	Set of hidden states
S_i	Hidden state
q_t	Hidden state being present at time t
K^S	Number of hidden states
$\boldsymbol{A} = \{a_{ij}\}$	State transition matrix
a_{ij}	Probability of transitioning from state S_i to state S_j
$B = \{b_j(\boldsymbol{o})\}$	Observation model
$b_j(\boldsymbol{o})$	Observation probability density in hidden state S_j
$\boldsymbol{o}$	Arbitrary observation vector sample
D	Placeholder for a probability density function
$O = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \dots\ \boldsymbol{o}_{K^O}\}$	Sequence of observation vector samples
K^O	Number of observations
c_{jm}	Weight of the mth component in the mixture of observation probability densities in hidden state S_j
$\boldsymbol{\mu}_{jm}$	Mean vector of mth component in the mixture of observation probability densities in hidden state S_j
$\boldsymbol{U}_{jm}$	Covariance matrix of mth component in the mixture of observation probability densities in hidden state S_j
M	Number of components in the mixture of observation probability densities
$\boldsymbol{\pi} = \{\pi_i\}$	Initial state distribution vector
π_i	Initial state probability of state S_i
$\alpha_t(i)$	Forward variable
$\beta_t(i)$	Backward variable

$\xi_t(i,j)$	Probability for process transitioning from hidden state S_i, being present at time t, to S_j, being present at time $t+1$
$\gamma_t(i)$	Probability for process being in hidden state S_j at time t

Trajectories

$T = \{\tau_1, \tau_2, \dots, \tau_N\}$	Set of trajectories
τ_i	Trajectory
N	Total number of trajectories
χ_j	Measurement tuple
l_i	Length of trajectory τ_i
$\boldsymbol{x}$	Two-dimensional position
$\boldsymbol{x}_j = (x_j, y_j)$	jth two-dimensional position defined by x and y-coordinate of a trajectory τ_i
v	Speed
v_j	jth speed value of a trajectory τ_i
a	Acceleration
a_j	jth acceleration value of a trajectory τ_i
d	Distance
d^{IntP}	Intersection path distance
t	Time
t_j	jth time value of a trajectory τ_i
t^{Int}	Intersection-relative time

Notations

Vectors	Lower case, bold, italic: $\boldsymbol{x}, \boldsymbol{\pi}$, …
Matrices	Upper case, bold, italic: $\boldsymbol{A}, \boldsymbol{U}$, …
Indices; physical quantities and variables	Lower case, italic: i, j, …; v, a, …
Normalized variables	Lower case, italic, tilde: $\tilde{v}, \tilde{a}$, …
Sets	Upper case, italic: T, S, …
Estimates	Italic, overline: $\overline{\boldsymbol{A}}, \bar{c}_{jm}$, …
Transpose	Italic, prime: $\boldsymbol{A}', \boldsymbol{o}'$, …

1. Introduction

The last two decades brought immense progress with respect to autonomous driving and advanced driver assistance systems. Safely following intersection-free roads without human intervention has become a straightforward task for modern autonomous systems. However, severe difficulties remain in situations involving more complex traffic scenarios. Among others, these include the robust and safe navigation through intersections. Without exact knowledge of the applicable traffic regulation, deducing vehicle behavior for autonomously crossing an intersection is unfeasible. With the aid of optical sensorics, perception-based approaches to detect signaling devices, traffic signs or the absence of any regulator can be employed. However, unsuitable lighting or obstructions may pose issues regarding the detection of such objects. This is one of the reasons, many approaches fall back on map resources annotated with information on traffic regulations. Furthermore, with map information as ground truth, the accuracy of perceived traffic regulations can be validated [1, 2].

A provider for maps annotated this way is Open Street Maps (OSM). This community-driven project is aiming at providing freely available map resources. However, it relies on volunteers manually uploading road-infrastructure-related data. This data is often gathered using ordinary measurement equipment. Thus, in terms of accuracy, it is inferior when compared to data collected with state of the art equipment. Also, intersections have to be manually annotated with information regarding their regulation [3]. The suggested approach could, for instance, contribute to a reduction of manual effort. Traffic regulations could be inferred from trajectory data and used for annotating arbitrary maps. In case data can be regularly collected, changes in traffic regulations could be detected and included in order to keep maps up to date.

1.1. Traffic Regulations

In the context of this thesis, traffic regulations or regulation classes are the set of lawful traffic rules controlling the behavior of traffic participants when entering and crossing intersections. The following traffic regulations are of interest in this work. Those communicated by means of traffic signs are `priority`, `stop` and `yield`. `Traffic lights` control traffic flows by means of signaling devices. The absence of any traffic signs or signaling devices at an intersection requires vehicles to obey the `yield-to-right` rule while crossing. An overview of these traffic regulations, categorized by the way these are communicated, is provided in Figure 1.1. Details of

aforementioned traffic regulations can be found in the German road traffic act *Straßenverkehrsordnung*.

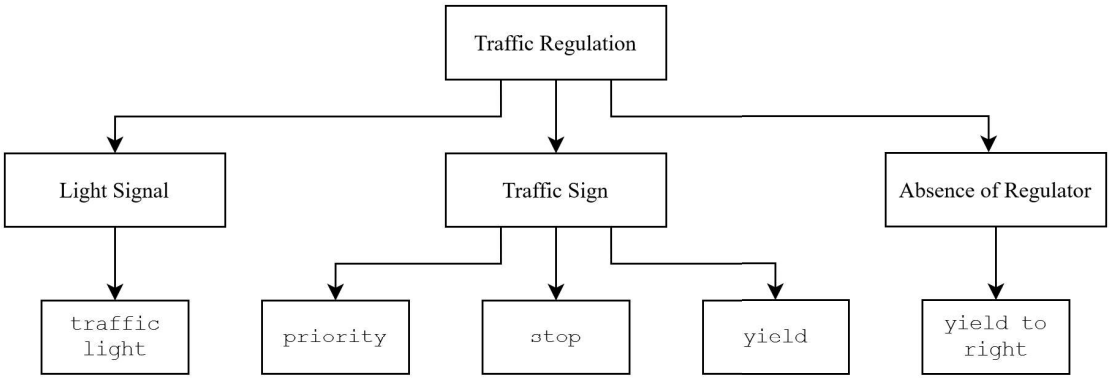


Figure 1.1: Categorization of German traffic regulations. The bottom row of boxes lists the traffic regulations that shall, in this thesis, be classified based on trajectory data. Terms in boxes above categorize regulations by the way these are communicated to traffic participants.

1.2. Intersections and their Representation

Intersections are structures used at places where two or more roads cross. In the context of this thesis, each road segment adjacent to an intersection can also be termed intersection arm. Each arm can have multiple lanes leading into or out of the intersection. In this thesis, intersections are decomposed into and represented by a set of paths that enable to describe their topology on a sub-lane level. The reason for this is illustrated as follows: A traffic participant reaches an intersection using a lane that enables her to either execute a turn or to continue straight. Independently from where she chooses to head, the traffic regulation, for instance `priority`, might uniformly apply for all directions. The crossing behavior, however, might vary with the chosen target lane. For instance, turning left might require her to give way to other traffic participants, whereas driving straight does not. In certain other cases, even different traffic regulations might apply for the different possible target directions of a single lane. For example, on the one hand, a `traffic light` might regulate crossing an intersection straight. On the other hand, turning right from the same lane might be regulated exclusively by a `yield` sign. Hence, elements describing all possible directions of all of an intersection's incoming lanes shall be employed to represent that intersection. These elements are termed intersection paths. The idea of employing intersection paths is inspired by a similar concept, used in Ruhhammer [4].

1.3. Goal

The goal of this thesis is to evaluate whether an approach based on Hidden Markov Models (HMM) can be applied to the problem of classifying intersection paths into different regulation classes. For this, HMMs shall develop representations of regulation-class-specific information. Furthermore, a trajectory representation which is suitable for supporting HMMs in developing stated representations shall be suggested. This is done by experimentally testing a variety of parameter values and design choices that specify certain aspects of the trajectory representation and the way HMMs are applied. The parameter values and design choices that lead to the best classification performance shall then be identified. For this, the classification performance shall be critically analyzed and evaluated using suitable measures.

1.4. Stopping Positions at Intersections

Several traffic regulations require vehicles to come to a halt or to slow down when approaching an intersection. The locations where such actions occur are termed stopping positions. Initially, stopping positions could be analyzed regarding their spatial distribution in proximity of intersections. For instance, it can be analyzed where stopping positions typically accumulate. From this information, it could be attempted to draw conclusions. For instance, lengths of vehicle queues or locations of stop bars could possibly be estimated. Also, lanes not having stop bars might be identifiable this way. Ultimately, it could then be examined whether drawn conclusions have the potential to benefit approaches to infer traffic regulations.

However, it is found that information on stopping positions is implicitly contained within trajectories. In the context of the type of trajectory representation employed in this thesis, trajectory points can be annotated with samples of associated speed or acceleration profiles. Hence, trajectory points with relatively low speed values can be interpreted as occurring during a slow-down. Such with speed values and acceleration magnitudes close to 0 can be interpreted as occurring during a standstill. For this reason, it is decided to refrain from analyzing stopping positions in this thesis.

1.5. Thesis Structure

The structure of this thesis is as follows. Section 2 provides an overview of the state of the art in trajectory-based traffic regulation inference. Theoretical concepts that form the basis for the suggested approach and conducted experiments can be obtained from section 3. Section 4 expands on topics related to the data basis of this thesis. In this section, assumptions and exclusions with respect to traffic situations represented in the data shall be outlined. Also, in order to support the reproducibility of this thesis, it shall be illustrated how trajectories are artificially generated using traffic simulation software. Furthermore, a characterization of the data and a detailed explanation of intersection paths shall be provided. In section 5, details of the suggested approach design are presented. The design and results of experiments conducted in order to scrutinize the proposed approach are shown and discussed in section 6. Section 7 provides a conclusion and recommendations for future work.

2. Related Work

There are a plethora of works aiming at inferring traffic regulations at intersections. Several of these suggest approaches based on trajectory data. From these, the works deemed most relevant are selected and reviewed in detail. In the following paragraphs, these works are summarized and compared to each other.

With regard to the representation of vehicle behavior, all considered works [4–9] make use of motion-related data, such as samples from speed and acceleration signals. Signals could for instance be read from onboard sensors of vehicles. From this data, characteristics of stopping or slow-down events, such as duration, location or count of such occurrences, can be derived. In [4, 6, 7], a combination of motion and stopping or slow-down-related data is used. In all three approaches, statistical measures, such as extrema, means or standard deviations, are computed from stated data. These statistical measures serve as features for a classifying system. Also, in all three works, the Random Forest is found to perform best among compared algorithms. Ruhhammer [4] represents intersections by structures that are termed mean intersection paths. Mean intersection paths represent the set of all pairs of intersection entries and exits that are connected by a direct and feasible path through the intersection. An entry or exit can consist of one or multiple lanes that belong to a specific intersection arm and lead into or out of an intersection. Thus, for an intersection arm, mean intersection paths distinguish between different target directions that are reachable by either turning or driving straight. As traffic regulations can vary among target directions of an arm, the author aims at classifying mean intersection paths separately. For this, Statistical measures of speed and acceleration are calculated and used as features. These are augmented by statistical measures of the distance to the associated intersection center point, duration and number of stopping events occurring on mean intersection paths. Additionally, radar-obtained information on the number of vehicles in front of a vehicle is included as another feature. In a binary distinction between `traffic-light` and not-`traffic-light`-regulated mean intersection paths, a relatively good classification performance is achieved. By also considering `priority`, `stop`, `yield` and `yield-to-right`-regulated mean intersection paths, the classification problem is transformed into a multi-class one. The resulting performance is reported to be lower, but no detailed analysis of associated results is made. Hu et al. [6] classify intersection arms. They also calculate statistical measures of speed and acceleration data and use these as features. Statistical measures of the distance to the respective intersection center and the number of stopping events are also part of the feature set. However, trajectories describing turn maneuvers are excluded from further consideration. In order to reduce data labelling effort, an active learning approach is implemented. Active learning is based on the idea of letting a

classifying system, which is initially trained on a pool of labeled data, request manual labeling decisions from e.g. a human reviser for specific data. More details of the concept of active learning can be obtained from [10]. In this work, intersection arms that are classified with a confidence below a specific threshold are presented for manual labeling. Besides `traffic-light` and `stop`-regulated arms, also `unregulated`[1] arms are classified by their approach. Saremi et al. [7] also classify intersection arms. They make use of statistical measures of speed data and of the number and duration of stopping events. However, a set of static features is additionally included. The static features are various road-length and distance-related features, as well as categorial ones, such as the road type (residential, motorway, etc.). The motivation for using aforementioned static features is that these should reflect the influence of US standards and guidelines applying on the positioning of regulator signs and lights at intersections. In this context, the authors raise concern that the set of static features might not generalize well to other countries. The authors reason that positioning guidelines might vary across countries. The combined use of all features outperforms using either static or dynamic features. In this approach, the traffic regulations `traffic light`, `stop` and `unregulated` are distinguished.

The authors of [5] and [9] employ features solely based on statistical measures of stopping or slow-down-related data. For instance, Pribe and Rogers [5] apply Deep Learning in combination with a set of heuristic rules. Classified are `traffic-light`, `stop` and `unregulated` intersection arms. The heuristic rules eliminate classification outcomes that represent unfeasible traffic regulation combinations determined for the set of arms of an intersection. For instance, if the classifier labels several arms of an intersection as `traffic-light`-regulated but at least one arm of the same intersection as `stop` or `unregulated`, the outcome is considered unfeasible. In that case, all of the affected intersection's arms are rejected without impacting the reported classification performance. Furthermore, only trajectories that do not show turning maneuvers are considered for classifier development and testing. The authors report a high classification accuracy when applying the Neural Network classifier alone. The combined use of the classifier and the heuristic rules leads to the correct classification of all considered intersection arms. Carisi et al. [9] apply threshold-based rules in order to distinguish between `traffic-light` and `stop`-regulated intersection arms. The thresholds are varied until a configuration yielding an optimal classification outcome is achieved.

In contrast, Zourlidou et al. [8] represent trajectories as series of measurements. This way, each trajectory point can be annotated with a sample from a speed signal and the distance to the respective intersection center. The trajectories are either temporally or spatially resampled. For temporal or spatial resampling, each of the resulting trajectories starts and ends at points having the same temporal or spatial distance to the associated intersection center. Using this representation approach, a relatively high number of object instances are available for classifier development. When calculating motion and stopping or slow-down-related features for e.g. arms of the same number of intersections, considerably less object instances are available. The authors show an approach that aims at distinguishing two types of intersections. Such uniformly regulated by `traffic lights` and such where `prioritized` and `yield`-regulated roads meet. As classification algorithm, the Decision Tree is employed. The classification of an intersection is based on the classification of all associated trajectories. More precisely, the traffic regulation that

[1] In areas of low traffic volume in the USA, intersections that are neither controlled by `traffic signs`, nor `light signals` can be found [11]. These are referred to as `unregulated` ones.

is assigned to the majority of trajectories is also assigned to the intersection. The authors report that, in most of the cases, `traffic-light`-regulated intersections are correctly detected, although at the cost of false positives from `priority/yield`-regulated intersections. Based on achieved results, the authors recommend to not aim at classifying intersections but at classifying elements on a higher level of granularity. This way, differences in speed profiles recorded at a certain intersection would be better accounted for. All other reviewed works include this approach in their strategy by classifying intersection arms or mean intersection paths.

Validation of results and classification performance is commonly done by developing and testing classifiers on non-overlapping subsets of the available data. The associated subsets are referred to as development and test data, respectively. In order to evaluate a classifier's performance on more than just one specific test subset of the available data, different test subsets are formed in a cross-validation manner. Readers interested in learning more about cross-validation are referred to [12]. In this context, the work of Saremi and Abdelzaher [7] stands out. Generalization performance is evaluated by developing the classifier on data associated to intersections of one set of cities, whereas testing is solely conducted on data associated to intersections of another set of cities. The different datasets never overlap. Other reviewed works [4–6, 8, 9] do not mention taking into account city or intersection association in the context of their validation strategies.

In the majority of the considered works [4–8], the used trajectory data is proprietary. Data is, for instance, self-collected using probe vehicles or access to datasets, which are not publicly available, is granted. Only Carisi et al. [9] initially use publicly available data from OSM. However, intersections for which enough crossings exist are found to be scarce. For this reason, the authors also resort to collecting data themselves. Therefore, none of the works completely relies on publicly available data alone.

The approach suggested in this thesis aims at holistically distinguishing the German traffic regulations `priority`, `stop`, `traffic light`, `yield` and `yield to right`. Similarly to the approach in [4], each intersection is represented by a set of intersection paths. For these paths, traffic regulations are inferred. Trajectories are represented as time series of samples from motion-related data signals, which is comparable to what is done in [8]. HMMs are used as classification method which, in this context, is novel. Furthermore, it shall be illustrated how trajectories crossing intersections can artificially be generated using traffic simulation software.

3. Theoretical Background

This section introduces several concepts and basics that are applied or serve as a relevant basis for the approach suggested in this thesis. Theoretical basics and algorithms associated to HMMs are introduced in section 3.1. Methods for measuring performance in the context of classification problems are presented in section 3.2.

3.1. Hidden Markov Models

Probabilistic methods can be used to describe and model a great variety of phenomena. For instance, many real-world processes manifest themselves in sequences of events. If the cause for certain observable event patterns is not directly visible, HMMs are a suitable means of modeling these processes. In this context, causes are represented as hidden states. Observable events can be represented as sequences of discrete symbols or continuous signals. Furthermore, an HMM can also be used as generator of observation sequences emitted by hidden state sequences it is intended to model [13, 14].

Parameters of HMMs and how HMMs can be formalized is presented in section 3.1.1. From the three fundamental problems that arise in the context of HMMs [13], two are relevant in this thesis. Firstly, with an observation sequence or signal and a parametrized HMM being given, one aims at computing the probability of the former. This problem can be efficiently solved using the Forward Algorithm, which is outlined in section 3.1.2. Secondly, one aims at parametrizing an HMM with the goal of achieving a maximal probability for a given observation sequence. This problem can be solved using the Baum-Welch Algorithm, which is described in section 3.1.3. The third problem lies in identifying a sequence of hidden states that provides an optimal explanation for a specific observation sequence, given a parametrized HMM. As this problem is not represented in this thesis, approaches to solve it are not shown.

3.1.1. Hidden Markov Model Foundations

An HMM can be formally defined as a tuple $\lambda = (S, \boldsymbol{A}, B, \boldsymbol{\pi})$ with $S = \{S_1, S_2, \dots, S_{K^S}\}$ being the set of K^S individual hidden states that can underlie the modeled process [13]. Based on the concept of Markov Chains, which can be employed to model sequences of states that are directly observable, the Markov assumption also holds for HMMs. This assumption basically states the following: The probability of a state being present at a certain position in a sequence of states depends exclusively on the preceding state [14]. Generally, a state being present at a point in time t is denoted by q_t. $\boldsymbol{A} = \{a_{ij}\}$, where $1 \leq i, j \leq K^S$, is the state transition probability matrix. An

entry a_{ij} of $\boldsymbol{A}$ denotes the probability of transitioning from state S_i to state S_j, which is happening during the step from time t to $t+1$. This can be formalized as $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$. For an ergodic HMM topology, in which any state can be reached from any other state, $a_{ij} > 0$ holds for all pairs i, j. Other HMM topologies allow a_{ij} for specific i, j to take on the value of 0. The observation model is a set $B = \{b_j(\boldsymbol{o})\}$, where b_j is the continuous probability density function in state S_j modeling the emission of observation signals. Observation signals can be described in terms of one or more features. Hence, observation samples $\boldsymbol{o}_t$ are generalized to vectors with feature values as components. These can be emitted at times t, with $1 \leq t \leq K^O$. The notation $\boldsymbol{o}$, without subscript, is used as placeholder for an arbitrary observation vector sample. A set of observation vectors with length K^O is termed observation sequence O. As an observation probability density can be a mixture of M component densities, it can be formalized as $b_j(\boldsymbol{o}) = \sum_{m=1}^{M} c_{jm} D(\boldsymbol{o}, \boldsymbol{\mu}_{jm}, \boldsymbol{U}_{jm})$, with $1 \leq j \leq K^S$. The coefficient c_{jm}, for which $\sum_{m=1}^{M} c_{jm} = 1$ has to hold, determines the weight of the mth component of the mixture in hidden state S_j. D denotes a probability density function with mixture-component-specific mean vector $\boldsymbol{\mu}_{jm}$ and covariance matrix $\boldsymbol{U}_{jm}$. Readers interested in general requirements that a probability density function D needs to fulfill are referred to [13]. The vector $\boldsymbol{\pi} = \{\pi_i\}$ with $1 \leq i \leq K^S$ describes the initial state distribution. Hence, $\pi_i = P(q_1 = S_i)$ denotes the probability that state S_i is the first state in a sequence of states [13]. Figure 3.1 shows an HMM with three hidden states. The states are denoted as circles that are labeled S_i, with $1 \leq i \leq 3$ corresponding to the respective state index.

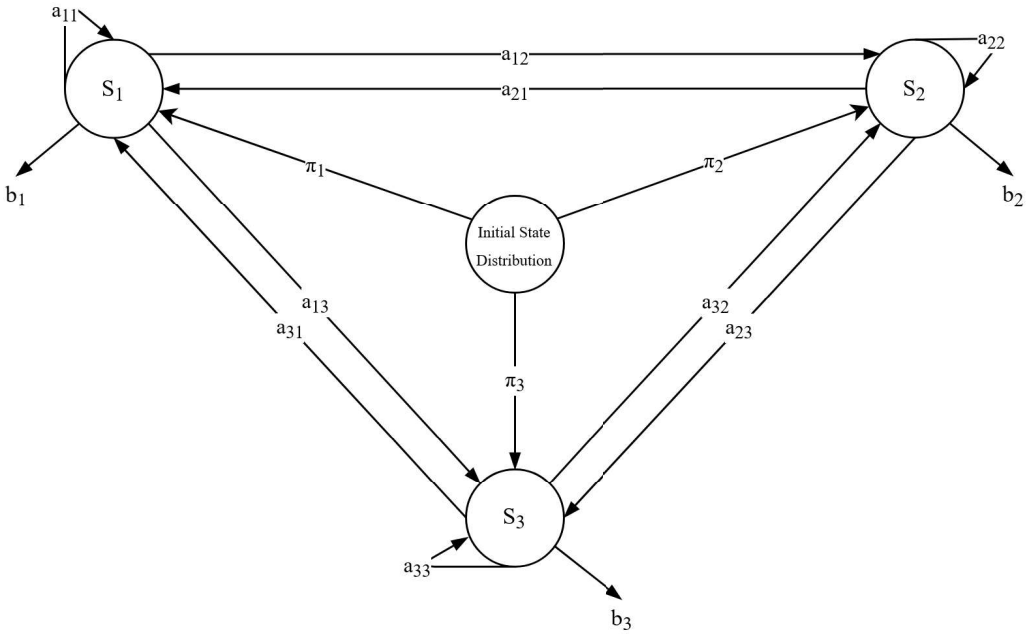


Figure 3.1: Schematic depiction of an exemplary Hidden Markov Model with three hidden states. The a_{ij} mark probabilities of states S_i transitioning to other states S_j, with $1 \leq i, j \leq 3$. The b_i represent the observation probability distributions in the hidden states. The probability of a state S_i being first in a sequence is described by the π_i.

In the center of the figure, the initial state distribution is illustrated as an auxiliary state that is connected to the hidden states of the HMM by three arrows labeled π_i. However, the auxiliary state and connections from it shall only be interpreted as representational elements and do not correspond to an actual state or state transitions, respectively. The entries a_{ij} of the state transition

probability matrix $\boldsymbol{A}$ are represented as labels on the arrows interconnecting the S_i. Worth noting is how each state has an arrow connecting it to itself. This specific arrow is labeled a_{ii}, which denotes the probability for a state not transitioning to another state but staying in the current state until the next step. Furthermore, each state has an arrow pointing to the observation probability distributions b_i. This shall only depict the fact that each hidden state is associated one distribution.

3.1.2. Forward Algorithm

The likelihood $P(O|\boldsymbol{\lambda})$ is the probability of an observation sequence O being emitted, given a parametrized HMM λ. It is irrelevant, whether the observation sequence consists of discrete symbols or samples from a continuous signal [15]. The calculation of this likelihood corresponds to one of the three problems commonly arising in the context of HMM-based approaches. It can be solved in an exact and efficient manner using the Forward Algorithm [13].

The forward variable $\alpha_t(i)$ specifies the probability that the process, modeled by the HMM λ, reaches state S_i when observation $\boldsymbol{o}_t$ is emitted, subsequent to a preceding series of observations. It can be formalized as $\alpha_t(i) = P(\boldsymbol{o}_1, \boldsymbol{o}_2, \dots, \boldsymbol{o}_t, q_t = S_i|\lambda)$. The computation is done by following a procedure consisting of an initialization, an induction and a termination step [13]. The initialization of the forward variable for each state i is defined as

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1) \text{ with } 1 \leq i \leq K^S. \tag{3.1a}$$

In the induction step, $\alpha_t(j)$ is recursively computed using the formula

$$\alpha_t(j) = \sum_{i=1}^{K^S} \alpha_{t-1}(i)\, a_{ij} b_j(\boldsymbol{o}_t) \text{ with } 1 < t \leq K^O,\ 1 \leq j \leq K^S. \tag{3.1b}$$

The algorithm terminates with the calculation of above-stated likelihood which is defined as

$$P(O|\boldsymbol{\lambda}) = \sum_{i=1}^{K^S} \alpha_{K^O}(i). \tag{3.1c}$$

3.1.3. Baum-Welch Algorithm

Parametrizing an HMM "by hand" can be a cumbersome task. Given a set of states S, one wishes to find values to the parameters $\boldsymbol{A}$, B and $\boldsymbol{\pi}$ that render an HMM λ capable of explaining a given observation sequence O well. In other words, one wishes to maximize the observation sequence's likelihood $P(O|\boldsymbol{\lambda})$. The Baum-Welch Algorithm is a popularly employed solution to this problem. Basically, it is an implementation of the Expectation-Maximization concept. It starts out with an initialization of the HMM parameters which can, for instance, be defined at random. The algorithm then iteratively reestimates the parameters which results in gradual improvements of $P(O|\boldsymbol{\lambda})$ until an HMM is determined that locally maximizes it. Thus, in contrast to the previously outlined algorithm, the Baum-Welch Algorithm is not an analytical approach to solve the stated problem [13, 14].

In addition to the above-presented forward variable α_t, the backward variable β_t needs to be introduced. Both variables are necessary for the reestimation of transition probabilities a_{ij}, initial state probabilities π_i and parameters of the observation probability density function $b_j(\boldsymbol{o})$. The backward variable $\beta_t(i)$ represents the probability of an observation sequence, which is starting in $t+1$ and ending with its final observation in time K^O, being emitted, given that the process, which is modeled by HMM λ, is in hidden state S_i at a point in time t. Thus, the probability can

be formalized as $\beta_t(i) = P\left(\boldsymbol{o}_{t+1}, \boldsymbol{o}_{t+2}, \dots, \boldsymbol{o}_{K^O} \middle| q_t = S_i, \lambda\right)$. The procedure for computing $\beta_t(i)$ consists of an initialization and an induction step. The initialization is done by setting

$$\beta_{K^O}(i) = 1 \text{ with } 1 \leq i \leq K^S. \tag{3.2a}$$

Induction is done as follows:

$$\beta_t(i) = \sum_{j=1}^{K^S} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j) \text{ with } t = K^O - 1, K^O - 2, \dots, 1,\ 1 \leq i \leq K^S \tag{3.2b}$$

Additionally, the probability $\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$ needs to be calculated. It can be interpreted as the probability of transitioning from state S_i, being present at a point in time t, to state S_j, being present at $t+1$, with observation sequence O and HMM λ being given. Based on the definitions of forward and backward variable, computed in (3.1a-c) and (3.2a,b) respectively, the calculation is as follows:

$$\begin{aligned} \xi_t(i,j) &= \frac{P(q_t = S_i, q_{t+1} = S_j, O|\lambda)}{P(O|\lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{K^S} \sum_{j=1}^{K^S} \alpha_t(i) a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{3.3}$$

The variable $\gamma_t(i)$ represents the probability that a process is in state S_i at a point in time t, with observation sequence O and HMM λ being given. Hence, a feasible formalization of this probability is $P(q_t = S_i | O, \lambda)$. This variable is used in further calculations and helps writing down formulas more straightforwardly. It is calculated from ξ_t, given by (3.3), as

$$\gamma_t(i) = \sum_{j=1}^{K^S} \xi_t(i,j). \tag{3.4}$$

The reestimates of a_{ij} are calculated by dividing the expected number of times state S_i transitions to state S_j by the expected number of times S_i transitions to arbitrary other states. These are denoted by $\bar{a}_{ij}$. All $\bar{a}_{ij}$, with $1 \leq i,j \leq K^S$, form the reestimated state transition probability matrix $\overline{\boldsymbol{A}}$. Including $\xi_t(i,j)$ from (3.3) and $\gamma_t(i)$ from (3.4), the associated formula is

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{K^O - 1} \xi_t(i,j)}{\sum_{t=1}^{K^O - 1} \gamma_t(i)}. \tag{3.5}$$

The reestimates of parameters of the observation probability density mixture $b_j(\boldsymbol{o})$, namely c_{jm}, $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{U}_{jm}$, are denoted by $\bar{c}_{jm}$, $\overline{\boldsymbol{\mu}}_{jm}$ and $\overline{\boldsymbol{U}}_{jm}$, respectively. The reestimated observation model is denoted as $\bar{B}$. The formulas for calculating these parameters are given as follows:

$$\bar{c}_{jm} = \frac{\sum_{t=1}^{K^O} \gamma_t(j,m)}{\sum_{t=1}^{K^O} \sum_{m=1}^{M} \gamma_t(j,m)} \tag{3.6}$$

$$\overline{\boldsymbol{\mu}}_{jm} = \frac{\sum_{t=1}^{K^O} \gamma_t(j,m) \boldsymbol{o}_t}{\sum_{t=1}^{K^O} \gamma_t(j,m)} \tag{3.7}$$

$$\overline{\boldsymbol{U}}_{jm} = \frac{\sum_{t=1}^{K^O} \gamma_t(j,m) (\boldsymbol{o}_t - \boldsymbol{\mu}_{jm})(\boldsymbol{o}_t - \boldsymbol{\mu}_{jm})'}{\sum_{t=1}^{K^O} \gamma_t(j,m)} \tag{3.8}$$

Modified from (3.4), the variable $\gamma_t(j,m)$ represents the probability of the process being in state S_j at a point in time t, given observation sequence O and HMM λ, while considering solely the mth component of the mixture in state S_j. For simple probability density mixtures or discrete

probability density functions, the variable $\gamma_t(j)$ from (3.4) is used instead of the modified $\gamma_t(j,m)$. Readers interested in the formalization and calculation of the latter, are referred to [13]. The parameter $\bar{c}_{jm}$ is computed by dividing the expected number of cases in which the process is in state S_j by the expected total number of cases in which the process is in state S_j, with the mth mixture component being considered. The reestimated mean vector $\overline{\boldsymbol{\mu}}_{jm}$ is computed by multiplying the modified variable γ_t with the sampled observation vector $\boldsymbol{o}_t$ for each time t and forming the sum over the resulting products. This sum is then divided by the expected number of cases in which the process is in state S_j, when considering the mth mixture component. The parameter $\overline{\boldsymbol{U}}_{jm}$ is computed in a similar way. In the associated formula (3.8), the prime marks the use of a vector's transpose [13].

Using $\gamma_t(j)$ from (3.4), the formula for the reestimation of the probabilities of the initial state distribution vector $\overline{\boldsymbol{\pi}}$ is

$$\bar{\pi}_i = \gamma_1(i). \tag{3.9}$$

Equation (3.9) shows that the $\bar{\pi}_i$ are the expected number of cases in which a sequence is started out with state S_i. This corresponds to the result of evaluating the variable $\gamma_t(i)$ at time $t = 1$ for each state S_i [13].

With all reestimates brought together, a reestimated HMM $\bar{\lambda} = (S, \overline{\boldsymbol{A}}, \bar{B}, \overline{\boldsymbol{\pi}})$ is determined. Generally, with $\bar{\lambda}$ instead of λ, a higher likelihood of O is achieved. Thus, in the context of the Expectation-Maximization concept, the reestimations can be interpreted as the Maximization step. The calculation of the ξ_t and γ_t form the Expectation step [13, 14].

3.2. Performance Measures

Several approaches to quantify the performance of classifiers can be applied. By means of a tabular layout, a confusion matrix, shows both classification errors and successes. Using a generic example, the concept is presented in section 3.2.1. In contrast, the F_1 score is employed to quantify performance using a single value. Its calculation is shown in section 3.2.2.

3.2.1. Confusion Matrix

In many classification problems, classes are imbalanced, hence represented by strongly varying sample counts. For problems of this kind, performance evaluation of classifiers solely based on achieved accuracy might not always be the best solution. Accuracy is computed by dividing the number of correctly classified samples by the total number of samples. Therefore, classifier performance regarding highly represented classes has a greater impact on accuracy than performance regarding classes being represented by much fewer samples. Analyzing confusion matrices is often better suited for evaluating the performance of a classifier on imbalanced classes.

In order to explain the concept of the confusion matrix, an abstract binary classification problem shall be considered. In this problem, samples shall be assigned to the classes *positive* and *negative*. A classifier can correctly assign a *positive* sample to the *positive* class. This outcome is referred to as a true positive. Analogously, the outcome of a *negative* sample being correctly classified as *negative* is referred to as a true negative. However, a *positive* sample can also be falsely classified as *negative*. Such an outcome is referred to as a false negative. The outcome of a *negative* sample being classified as *positive* is a false positive [16]. These four possible outcomes are arranged in

Table 3.1. In this table, rows indicate the actual classes, hence ground truth information. Columns adopt a classifiers perspective, hence the classes as predicted by a classifier.

		Predicted class	
		Positive	*Negative*
True class	*Positive*	True positives	False negatives
	Negative	False positives	True negatives

Table 3.1: Possible outcomes for a binary classification problem in which samples shall be assigned to the classes *positive* and *negative*. Rows indicate the actual classes, whereas columns indicate the classes as predicted by a classifier. (adapted from [16])

A confusion matrix is employed to describe how a specific number of classifications is distributed over the above-explained possible outcomes. Using the same row and column structure as depicted in Table 3.1, the number of occurrences of each of the four outcomes among all classifications can be arranged in the corresponding cells. This way, the confusion matrix associated to the binary classification problem is assembled. It is visualized in Table 3.2. As an example, the information that n_{12} *positive* samples are falsely classified as *negative* can be taken from the upper right cell of the confusion matrix. Relatively high values on the main diagonal are the result of a classifier that assigns a high number of samples to the correct classes. Hence, this can indicate a high performance of the classifier. In contrast, comparatively high values on off-diagonal matrix positions indicate that a classifier labels a high number of samples falsely. Therefore, classifier performance can be visualized on a higher level of granularity. For classification problems with more than two classes, the confusion matrix can be employed analogously [17].

		Predicted class	
		Positive	*Negative*
True class	*Positive*	n_{11}	n_{12}
	Negative	n_{21}	n_{22}

Table 3.2: Confusion matrix for a binary classification problem in which samples shall be assigned to the classes *positive* and *negative*. Rows indicate the actual classes, whereas columns indicate the classes as predicted by a classifier. (adapted from [17])

3.2.2. F_1 Score

In order to define the F_1 score, two other class-specific measures need to be introduced. Firstly, Recall is a measure describing the fraction of the samples that are correctly assigned to a specific class among all samples truly belonging to this class [17]. This measure is also often referred to as True Positive Rate or Sensitivity [18]. The formula for this measure is

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}. \tag{3.10}$$

Secondly, Precision describes the fraction of the samples that are correctly assigned to a specific class among all samples that are assigned to this class by a classifier [17]. Another popular term for this measure is Positive Predictive Value [18]. It can be formalized as follows:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{3.11}$$

The F_1 score is also a class-specific score and based on a combination of both above-introduced scores. It is calculated as the harmonic mean of Recall and Precision, which is formalized as

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \tag{3.12}$$

In equation (3.12), Precision and Recall are weighted equally. However, the F_1 score can also be computed using different weightings of both measures [17]. Values of Recall, Precision and F_1 score range between 0 and 1 [18].

In order to quantify the performance of a classifier with regard to an entire set of classes instead of just a single class, one can use a multiclass variant of the F_1 score. For classifier evaluation in multiclass problems with class imbalances, the macro F_1 score is often employed. It is computed as the average of all class-individual F_1 scores [19]. Readers interested in other variants are referred to [19].

4. Data Basis

The data basis for this thesis is created artificially, using version 1.3.0 of the traffic simulation software Simulation of Urban Mobility (SUMO) [20]. Among other features, this software supports the microscopic modeling of vehicle interactions, provides various implementations of car following models and generates intersection behavior for traffic participants [21]. The use of artificial data for this thesis is primarily motivated by the low availability of publicly available datasets, which is mentioned in section 2. For reasons of simulation reproducibility, all information, configuration files etc. that are necessary for replicating the data with the aid of SUMO are provided in section B of this thesis.

In SUMO, traffic scenarios are represented as networks of directed linear edges interconnecting nodes. Edges represent roads. Roads can have an arbitrary number of lanes with a default width of 3.20 m. In nodes, multiple edges can intersect with each other. Thus, nodes are employed to represent intersections. Section 4.1 expands on the origin and design of SUMO networks representing intersection areas. These serve as infrastructure basis for trajectory simulation. Traffic participants, such as vehicles, motorcycles and pedestrians, are introduced by demand. Demand specifies and schedules routes that traffic participants follow through a road network. Routes can be created in a randomized manner using the SUMO tools RandomTrips and Duarouter. More details on how SUMO is employed to simulate intersection crossings at a high degree of realism are provided in section 4.2. Trajectory data can be created by tracking simulated vehicles that follow routes specified by demands. The trajectory data and the types of features used to describe it are shown in section 4.3. How trajectory data is used for developing and validating the traffic regulation inference method presented later in this thesis is outlined in section 4.4.

4.1. Intersection Scenarios

Initially, a set of 16 intersection scenario networks is made available for simulating intersection crossings. These are created using real-world intersections as reference. There are four intersection scenarios regulated exclusively by `traffic lights` and four by the `yield-to-right` rule. In four intersection scenarios, `priority` and `yield` intersection paths meet and in another four, `priority` and `stop` intersection paths meet. In a later stage of this thesis, it is decided to additionally create four more intersection scenarios. In order to conform to the thesis time frame, these are designed in a simplified way by varying characteristics of selected existing networks. For this, the number of lanes, speed limits and regulation of the selected ones are altered. In these four additional scenarios, solely `priority` and `yield` intersection paths meet.

In subsection 4.1.1, the process of designing a SUMO network from its real-world reference is described. Subsequently, an approach to calculate intersection center points is introduced in subsection 4.1.2.

4.1.1. Representing a Real-World Intersection

The geometrical and topological structure of real-world intersections is obtained from OSM [22]. The SUMO tool osmWebWizard enables the user to select a rectangular region from OSM that can be exported to a SUMO network representation. Apart from the respective intersection node of interest and associated incoming and outgoing roads, the output of osmWebWizard contains several artifacts. These are for instance polygons representing buildings or green spaces, edges not approaching the node of interest directly and nodes representing other contiguous intersections. In order to reduce complexity and restrict each scenario to one specific intersection, the infrastructure-relevant artifacts are removed using the SUMO tool Netedit. Netedit allows the modification of existing networks or their construction from scratch [23]. Additionally, several manual steps have to be taken in order to prepare networks for the simulation of intersection crossings at a degree of realism that is as high as possible. With the real intersection as reference, all edges are checked for the correctness of speed limits and number of lanes. It is also made sure that the intersection-internal crossing lanes establish all of the actual links from lanes approaching the intersection to outgoing ones. In other words, simulated and real-world vehicles shall be able to drive on the same combinations of incoming and outgoing lanes through the SUMO-represented and the real-world intersection, respectively. Furthermore, the network is checked for the correctness of present traffic regulations. In cases of doubt, Google Street View [24], is employed to manually determine the relevant information. This is required especially for OSM extraction results that indicate a node's regulation as "unknown". This occurs for intersections that lack traffic regulation information in the map database. Finally, it is ensured that the network has incoming and outgoing edges start and end, respectively, in points 160-200 m from the intersection center point. The maximum distance of trajectory points that are used as input for the proposed method is considerably lower. The surplus of road length shall carefully account for and avoid possible but unknown effects regarding vehicles driving shortly after or before the point of their insertion or leaving, respectively. More details on the maximum distance of trajectory points are provided later in this thesis.

The above-outlined editing process is costly in terms of manual rework and offers no hint for simple automation. However, the process facilitates a straightforward subsequent simulation of vehicles crossing intersections. For instance, as a result, demand definition can be restricted to routes along intersection-incoming and outgoing edges. This helps to avoid error-raising problems caused by artifacts from OSM extraction. For example, vehicles can no longer enter dead ends, which causes errors, via edges that are removed during the process.

An exemplary depiction of the result of extracting an intersection in Berlin, by means of osmWebWizard, is shown in Figure 4.1. The map cutout in the upper half of the figure shows the intersection, at which the `prioritized` Residenzstraße is approached by Friedrich-Wilhelm-Straße from the east and Deutsche Straße from the west. A `yield` sign obliges both approaching roads to give way. In the lower half of the figure, one can see the white, green and pink polygons representing various building and green-space artifacts. Several edges representing streets not directly approaching the stated intersection are also visible on the right side of the lower half of

the figure. Figure 4.2 shows the network representation of the same scenario, after editing it in above-outlined process.

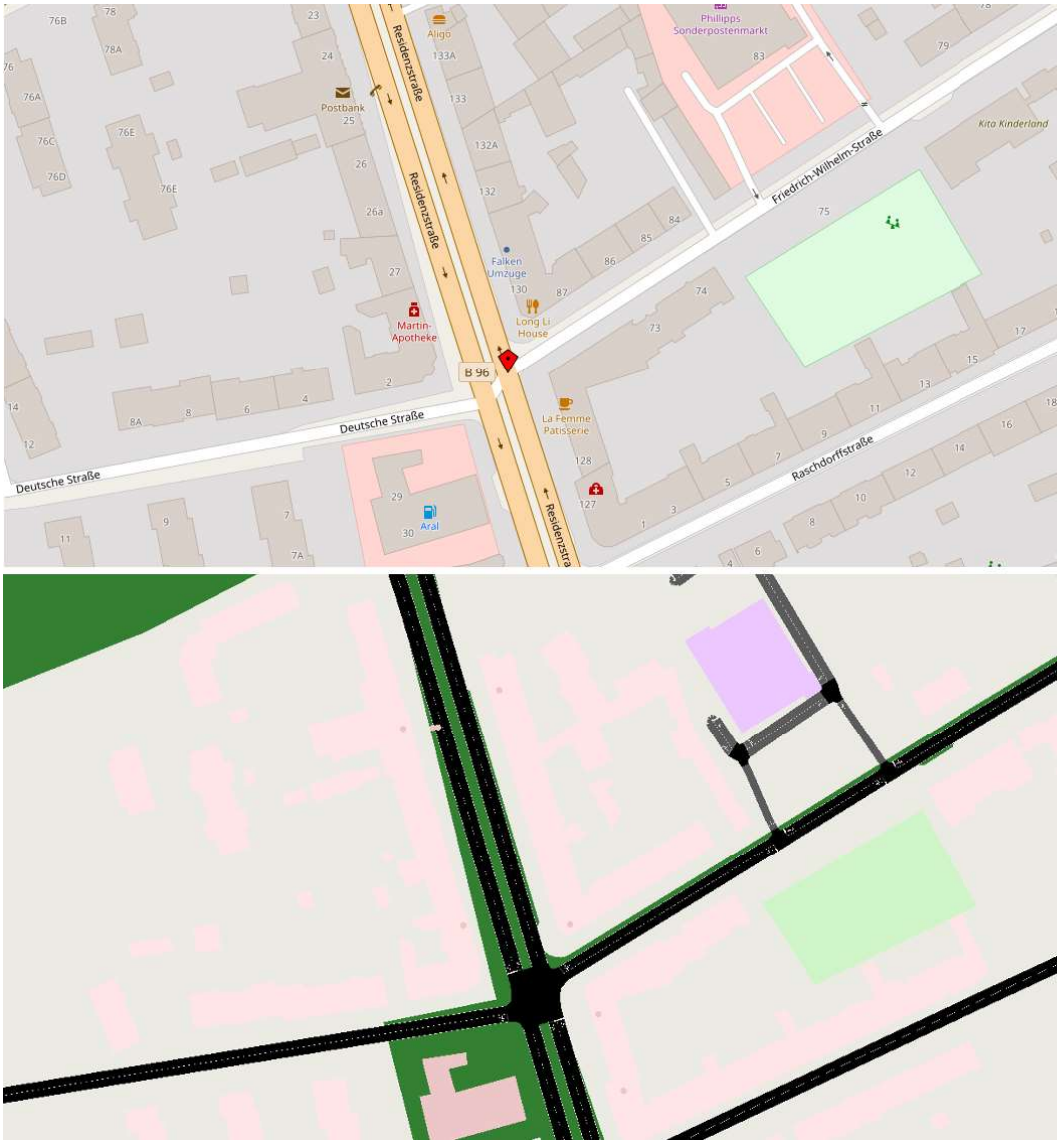


Figure 4.1: Extraction of an Open Street Maps excerpt (above) of the intersection Residenzstraße with Friedrich-Wilhelm-Straße and Deutsche Straße in Berlin by means of osmWebWizard [22]. The unprocessed result of the extraction is visualized via Netedit (below).

4.1.2. Determining the Center Point of an Intersection

Due to the often rotund shape of intersections, several attributes can be used to geometrically describe them. For instance, center points can be defined for intersections. A three or four-arm intersection might be the result of two straight roads crossing each other. In such a case, an intersection center location could simply be defined as the point in which centerlines of these two roads intersect. In this context, a road centerline could be defined as the line that has the same lateral distance to both road boundary lines. However, there are intersections for which center points cannot be defined that straightforwardly. For instance, the four-arm intersection depicted in Figure 4.2 is the result of three roads crossing each other. One road runs nearly vertically through the figure and constitutes two of the four arms. Two other roads approach at different angles from left and right and each correspond to one arm. From viewing the figure, one can see that road centerlines of these two roads intersect each other outside the intersection. Thus, for

intersections like this one and also in general, SUMO calculates intersection center points as follows: Initially, an intersection node is created from the set of the intersection-facing endpoints of all edges representing roads leading into and out of the intersection. The endpoints of edges can be defined either manually by the user or automatically by osmWebWizard. The location of the intersection node, hence the intersection center point, is then calculated as the mean of all endpoints it is created from. This point can be obtained from the associated SUMO network definition. More details can be found in [23]. When evaluated visually for all designed intersections, points calculated this way are well-suited as intersection center locations. Thus, they are used throughout this thesis.

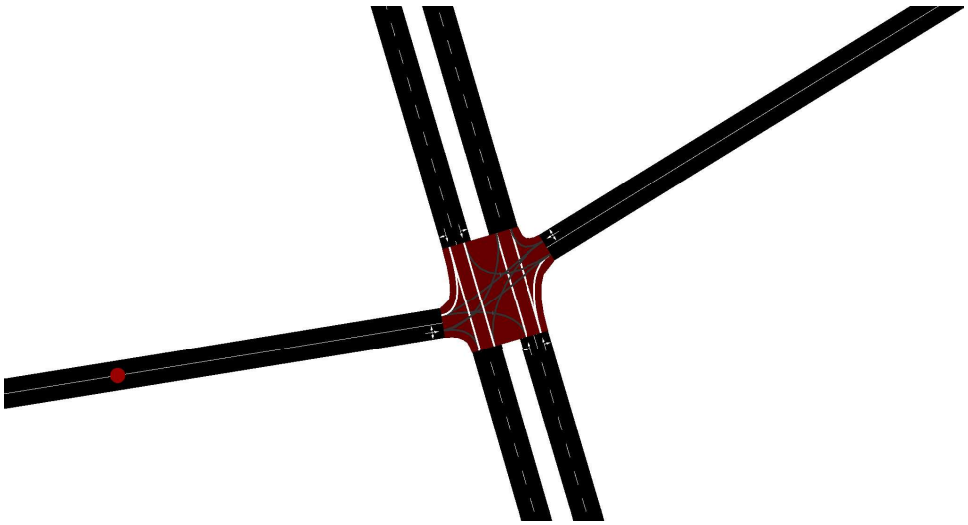


Figure 4.2: Edited SUMO intersection scenario network representing the intersection Residenzstraße with Friedrich-Wilhelm-Straße and Deutsche Straße in Berlin. Visualized via Netedit.

4.2. Simulation of Vehicles Crossing Intersections Using the Software Simulation of Urban Mobility

With intersection scenario networks as basis, vehicles crossing these can be simulated. In this context, several simplifying assumptions regarding the environment of intersections and processes occurring at such are made. This is outlined in subsection 4.2.1. Subsequently, details on how simulation is conducted are presented in subsection 4.2.2.

4.2.1. Reduction of Simulation Complexity

In order to reasonably limit the scope of this thesis, several simplifying assumptions and exclusions with regard to traffic simulation shall be made. By reducing the detail and variety of processes taking place at intersections, the focus can be set on the most relevant ones.

Traffic participants: In this thesis, exclusively passenger cars cross intersections. Larger vehicles, as for instance busses or semi-trucks, and emergency vehicles, such as ambulances or fire engines are excluded. Apart from that, no intersection crossings of two-wheeled vehicles, such as motorcycles or bicycles are simulated. Pedestrians are also not included in simulations.

Traffic-regulation-related specifics: In Germany, `traffic lights` can be found, that allow turning right on red. This is commonly marked by a sign showing a green arrow pointing to the right. This type of regulation is not represented in the data.

Approaching a `stop` sign demands briefly coming to a halt, often at a dedicated line. In case one does not have full sight of all other intersection arms at this point, a subsequent standstill closer to the intersection might be necessary in order to give way to other traffic participants. In the context of this thesis, it is assumed that every stop line is placed at a location where traffic participants have full sight of other intersection arms. Thus, traffic participants stop only once at `stop` signs.

Intersection types: Basis for simulation are intersections with at least three arms. The roundabout can also be classified as a type of intersection. However, roundabouts are excluded.
As already mentioned, pedestrians are excluded from further consideration. Consequently, cross-walks are also not included in simulation scenarios.
Mostly on roads in residential areas, vehicles can be required to adapt their speed to walking speed. Roads of this type are not considered. In this thesis, the lowest speed limit on intersection arms is 30 $\frac{\text{km}}{\text{h}}$ ($\approx 8.33\ \frac{\text{m}}{\text{s}}$).

Specific situations and maneuvers: In simulations of this thesis, no unforeseen events are occurring at intersections. For instance, no congestions on lanes leading out of intersections are represented in the data. Needless to say, this does not apply for intersection-incoming lanes on which vehicles might line up due to, for instance, red `traffic lights`. Also, no violations of traffic regulations occur. Thus, traffic participants, for instance, never run red lights and always yield to others who have the right of way. Furthermore, no accidents occur during simulations.
A U-turn is a maneuver performed by traffic participants at intersections for the purpose of turning around. However, in this thesis, traffic participants are always targeting other intersection arms than the one they are coming from. Also, once an intersection approach is initiated, the intersection is crossed without interruptions, except for such required in the context of traffic regulations. Thus, no traffic participant is reversing or parking in the proximity of an intersection.

4.2.2. Parametrization of the Simulation

The simulation software SUMO provides a vast number of possibilities to control the way vehicle crossings are simulated. With regard to several parameters, deviations from SUMO default settings are made in order to improve the simulation quality and achieve a high degree of realism. For reasons of simulation reproducibility, all modifications of parameter values are explained in this section. Modifications of globally relevant simulation parameters values are presented in subsection 4.2.2.1. Parameter values individually defined for each intersection scenario are shown in subsection 4.2.2.2.

4.2.2.1. Globally Relevant Simulation Parameters

Parameter values which affect all simulations in this thesis, are shown in the following. Table 4.1 additionally provides a joint summary of these parameters. From this table, the context, the SUMO tool involved and the chosen value can be obtained. In order to simplify the reproduction of data generation, the column parameter name provides SUMO-internal keywords and setting paths.

4.2.2.1.1. Network-Related Parameters

The SUMO tool Netedit allows to specify the resolution of intersection-internal lanes, which serve as crossing paths for vehicles. The resolution depends on the number of support points that define

the piecewise linear segments of stated crossing paths [23]. In order to achieve smooth crossing motions, the resolution is increased from 5 to 10 support points.

4.2.2.1.2. Car-Following and Vehicle-Type-Related Parameters

In terms of similarity to real-world vehicle trajectories, the Intelligent Driver Model (IDM) is found to outperform other car following models such as those by Krauss or Wiedemann [25, 26]. For this reason, IDM is chosen instead of the default Krauss model for mathematically modeling longitudinal interactions of vehicles. It takes desired values of acceleration, deceleration, temporal headway to vehicles driving ahead and minimum gap distance to vehicles standing ahead as input. Additionally, a parameter delta serves as acceleration exponent [27].

In this context, the authors of [28] criticize that models generating driving behavior are often not calibrated and used with default parameters. However, with respect to car following model parameters, conducting a calibration based on real trajectory data is far beyond the scope of this thesis. For this reason, several works [25, 29, 30] using SUMO for similar purposes as the one pursued by this thesis are reviewed. From these, reasonable indications for setting car following parameters shall be gathered. The authors of [25] analyze IDM parametrizations and the realism of associated simulation outcomes. They conclude that several car following model parameters should be distributed instead of being set to fixed values. This way, realistic driving behavior manifesting in various driving styles can be simulated. Similarly, [29, 30] show approaches which make use of the IDM to describe and model real-world vehicle behavior and discuss suitable parametrizations.

For this, the SUMO tool CreateVehTypeDistribution is employed. This tool samples a specific number of parameter configurations from value ranges or distribution functions. A parameter configuration defines the characteristics of a simulated vehicle [23]. This way, as mentioned above, car following model parameters are varied among inserted vehicles for the purpose of modeling different driving styles. Based on the reviewed literature, normal distributions of car following parameters are defined. Acceleration, deceleration, temporal headway and minimum gap are distributed with means 1.60 $\frac{\mathrm{m}}{\mathrm{s}^2}$, 2.00 $\frac{\mathrm{m}}{\mathrm{s}^2}$, 0.86 s, and 2.00 m, respectively. The value for the standard deviations of all distributions is 0.20. By choosing these means and standard deviations it is attempted to parametrize distributions from which car following parameters, which fall into the ranges the literature suggests, can be drawn. As done in the reviewed works [25, 29, 30], the delta parameter is kept constant. For this thesis, a value of 4.00 is chosen.

Another parameter, which does not belong to the set of car following model parameters, is also varied by means of CreateVehTypeDistribution. Vehicle lengths shall be sampled uniformly from the interval between 3.90 and 4.90 m, whereas default characteristics such as vehicle width, height and maximum speed capability etc. are kept at default values of the vehicle class “passenger” [31].

Consequently, from a vehicle type distribution of all above-described parameters, a pool of 500 parameter configurations is sampled. In the context of this thesis, the definitions of all vehicles simulated in all scenarios are drawn with replacement from this pool.

4.2.2.1.3. Lane-Change-Related Parameters

In several instances, unrealistic lane change maneuvers are observed during simulation. For example, vehicles are fount to occasionally change lanes back and forth for no obvious reason.

Also, lane changes are observed to occur from standstill, accomplished by purely lateral movement. For this reason, lane changes are suppressed in simulations. This is achieved by adapting the values of two vehicle type parameters that are relevant for lane change decisions. The parameter lcKeepRight controls how strongly drivers feel obliged to drive in the rightmost lane of a road. The parameter lcSpeedGain controls how strongly drivers are motivated to change into lanes that allow driving at higher speeds. With both parameters being set to 0, no lane changes occur [23].

4.2.2.1.4. Vehicle-Insertion-Related Parameters

To further control the way vehicles are inserted on the verge of intersection scenario networks, parameters passed to RandomTrips and Duarouter are modified. In this thesis, the number of inserted vehicles is binomially distributed [23]. The repetition parameter of the binomial distribution, commonly denoted by n, is globally set to 100. The distribution's probability parameter is individually chosen for each intersection scenario, which is described in 4.2.2.2.1 below.

Of all edges of an intersection scenario, an edge has to be selected on which the vehicle is inserted. This is controlled by a setting that varies across intersection scenarios. Section 4.2.2.2.2 expands on this. The selection of an edge's lane for vehicle insertion is controlled by Duarouter. Specifying the option "best" selects lanes that generally require no further lane changes for vehicles on their intersection approach. It also accounts for current lane utilization by other vehicles. This way, the driver's preference for a lane that has the shortest queue and permits to reach the desired target intersection arm is modeled. Concerning the insertion speed, the option "max" avoids vehicles having to start off from standstill where flowing traffic would be expected. Instead, vehicles are inserted at a speed that is adjusted to the edge's speed limit. This complies with the fact that intersection scenario networks are just cutouts of greater traffic networks [23].

4.2.2.1.5. Time-Step-Related Parameters and Numerical Integration Scheme

The frequency at which data is collected is 10.00 Hz. This is modeled by specifying the simulation's time step length, hence the reciprocal of the frequency, as 0.10 s. Another parameter is the action step length. It is the time that passes between two successive decisions made by a driver. The action step length is recommended to be set to values greater than the time step length if the latter is set to a value lower than 1.00 s. This way, the time a driver needs to react on changing situations can be modeled more realistically [23, 27]. The value of 0.50 s is set.

As numerical integration scheme, the ballistic method is selected. The authors of [32] find that it is better suited than SUMO's default Euler method for the integration of car following models in terms of robustness and approximation error. Instead of keeping vehicle speed constant with the Euler method, the ballistic integration method causes the acceleration to be kept constant during a simulation timestep [23].

4.2.2.1.6. Parameters controlling Imperfection and Inaccuracy

Additionally, in order to integrate imperfection in driving, vehicles can be equipped with a driver state model. This way, perception errors are modeled in SUMO. Those manifest in, for instance, deviations of distances kept between vehicles and actually driven speeds from target distances and speeds, respectively [23]. For every intersection scenario, 40 % of all inserted vehicles, which

are randomly selected, are equipped with a driver state model. The parameters associated to the driver state model are kept at default values.

Per default, SUMO calculates vehicle locations exactly. In order to model sensor noise, localization inaccuracy is introduced. It is modeled as additive white Gaussian noise. For this, a standard deviation of 0.20 m is passed to TraceExporter [23].

Description of setting	SUMO tool	Parameter name	Value [unit]
Resolution of intersection-internal paths	Netedit	junctions / internal-link-detail	10
Car following model	CreateVehTypeDistribution	carFollowModel	“IDM”
Acceleration value	CreateVehTypeDistribution	accel	normal(1.60, 0.20) [$\frac{m}{s^2}$]
Deceleration value	CreateVehTypeDistribution	decel	normal(2.00, 0.20) [$\frac{m}{s^2}$]
Temporal headway value	CreateVehTypeDistribution	tau	normal(0.86, 0.20) [s]
Minimum vehicle gap	CreateVehTypeDistribution	minGap	normal(2.00, 0.20) [m]
Delta parameter	CreateVehTypeDistribution	delta	4
Vehicle class	CreateVehTypeDistribution	vClass	“passenger”
Car length	CreateVehTypeDistribution	length	uniform(3.90, 4.90) [m]
Eagerness to use right lane	CreateVehTypeDistribution	lcKeepRight	0
Eagerness to use lanes enabling to drive faster	CreateVehTypeDistribution	lcSpeedGain	0
Binomially distributed vehicle insertion	RandomTrips	--binomial	100
Lane for vehicle insertion	Duarouter	defaults / departlane	“best”
Speed of inserted vehicles	Duarouter	defaults / departspeed	“max”
Simulation time step length	core	time / step-length	0.10 [s]
Simulation action step length	core	processing / default.action-step-length	0.50 [s]
Enabling ballistic numerical integration scheme	core	processing / step-method.ballistic	true
Enabling human imperfection in terms of car following	core	driver_state_device / device.driverstate.probability	0.40
Localization Noise	TraceExporter	--gps-blur	0.20 [m]

Table 4.1: Parametrization of the SUMO simulation. Each row describes the context, the SUMO tool involved, the SUMO-internal name or path and the chosen value of each parameter. A “/” separates different steps in a path leading to a parameter. “--” indicates that a parameter is passed as command line argument in the context of a function invocation.

4.2.2.2. Parameters Specific to Intersection Scenarios

Parameter values that are varied across scenarios are outlined in the following. These are additionally summarized in Table 4.2. In this table, a short description, the SUMO tool involved and SUMO-internal keywords and setting paths are provided for each parameter.

4.2.2.2.1. Parameters Controlling the Frequency of Vehicle Insertion

Per default, by specifying the period parameter, the user can control the frequency at which vehicles are inserted into a network. However, as mentioned in section 4.2.2.1.4, it is decided to randomize the number of inserted vehicles by means of binomial distributions with constant repetition parameter set to 100. In this context, the reciprocal period serves as the probability

parameter for a scenario's binomial distribution. In order to yield no probability value greater than 100 %, only period values equal or greater than 1 are passed to RandomTrips. Increasing the period value results in lower probability values and, thus, in a decrease in insertions [23]. In other words, a certain period value causes a certain traffic volume within a simulation. In this thesis, a period value is determined by visually evaluating the resulting traffic volume. For each intersection scenario, a traffic volume that seems appropriate in the context of the respective regulation is targeted. Each scenario is assigned one period value. Details on traffic volumes that are deemed appropriate and period values that create these are presented in the following paragraph.

When simulating `traffic-light`-regulated scenarios, moderate vehicle queue lengths of frequently more than ten vehicles at red lights and an overall high traffic volume shall be produced. This is achieved by setting rather low period values ranging between 2.20 and 6.00 for the various intersection scenarios. In contrast, traffic volume in `yield-to-right`-regulated scenarios is, on the one hand, attempted to be relatively low. On the other hand, it shall be high enough to occasionally show situations in which vehicles are forced to interact. Situations of this type are occurring if vehicles that approach from different intersection arms reach the intersection at similar points in time. Vehicle queue lengths shall not exceed four vehicles for most of the simulation's runtime. Suitable period values are found to range between 6.00 and 6.60. In the context of scenarios where `priority`-regulated intersection paths meet `stop` or `yield`-regulated ones, traffic volumes that cause a high number of situations with vehicle interactions are targeted. Those are situations in which vehicles following `stop` or `yield`-regulated intersection paths are obliged to give way to vehicles approaching from `prioritized` ones. Consequently, vehicle queue lengths shall be kept below ten vehicles for most of the simulation's runtime. Described traffic volumes are observed for a relative wide range of period values between 3.00 and 8.00.

4.2.2.2.2. Parameters Controlling the Place of Vehicle Insertion

How inserted vehicles are distributed over the edges adjacent to the intersection node of a scenario is defined by source weights. In comparison to edges with lower source weight values, such with higher values are target of insertion more frequently. In contrast, destination weights control which edges are chosen by vehicles for leaving a traffic network. Analogously, edges with higher destination weights are used as exit at a higher frequency than such with lower weights [23].

The `yield-to-right`-regulated intersection scenarios of this thesis have source and destination weights that are balanced across edges. This renders all edges similarly probable to serve as insertion target or as edge vehicles use for leaving the network. However, `traffic-light`-regulated scenarios and such where `priority`-regulated intersection paths meet `stop` or `yield`-regulated ones are often designed to have specific routes being followed by the majority of simulated vehicles. Like in real-world intersection scenarios, such routes serve the purpose of efficiently channeling through-traffic in urban environments. These can, for instance, be termed main roads. For intersections of this thesis, they are represented by combinations of incoming and outgoing edges that are assigned higher source and destination weights than other edges. Therefore, `traffic-light`-regulated scenarios of this thesis often have a higher number of lanes on edges representing main roads than on other edges. For `traffic-sign`-regulated intersections, these edges in some cases also have more lanes than others and are always `priority`-regulated.

4.2.2.2.3. Parameter Controlling the Simulation Duration

The user needs to define the total duration of the simulation of each scenario [23]. Consequently, this controls the number of simulated crossings. The longer the simulation of a scenario, the more trajectories can be gathered from it.

Description of setting	SUMO Tool	Parameter name
Rate of vehicle insertion	RandomTrips	--period
Vehicle insertion and destination distribution over edges	RandomTrips	--weights-prefix
Simulation duration	core	time / end

Table 4.2: Simulation parameters that vary across created intersection scenarios. Each row describes the context, the SUMO tool involved and the SUMO-internal name or path of each parameter. A "/" separates different steps in a path leading to a parameter. "--" indicates that the parameter is passed as command line argument in the context of a function invocation.

4.3. Trajectory Data

During the simulation of intersection crossings, information on the state of each vehicle present in a network can be sampled at constant time intervals. In the context of this thesis, the vehicle state is defined by a two-dimensional position $\boldsymbol{x}$, speed v and acceleration magnitude a. Each sampled state is additionally annotated with a timestamp t. Vehicles can be uniquely identified via a descriptor attribute. The $\boldsymbol{x}$ and v data is exported via the *Floating Car Data*[2] interface. By additionally exporting sampled information to the *Amitran* standard[3], a data is gathered. The SUMO tool TraceExporter can subsequently be used to group data by vehicle association [23].

Trajectories annotated with motion-related information are, as adapted from [4], formalized in the following way. In the context of this thesis, a trajectory $\tau_i \in T$ of length l_i, with $i \in [1,\ N]$ and T being the set of all N simulated trajectories, is represented by a series of measurement tuples $\chi_j^{(i)} = \left(\boldsymbol{x}_j^{(i)}, v_j^{(i)}, a_j^{(i)}\right)$, with $\boldsymbol{x}_j^{(i)} = (x_j^{(i)}, y_j^{(i)})$ and $j^{(i)} \in [1,\ l_i]$. Each tuple has associated a specific point in time t_j beginning at simulation start. Measurement tuples are also referred to as trajectory points. Scalars x_j and y_j are orthogonal coordinates defining $\boldsymbol{x}_j$. A series of $\boldsymbol{x}_j$, v_j or a_j that is associated to a trajectory is also termed as trajectory feature.

With the parameters described in section 4.2, SUMO is used to initially simulate 47583 crossings on 16 intersection scenarios. With four additionally created scenarios, another 16867 trajectories are simulated. On the one hand, for each of the uniformly regulated scenarios, namely `traffic-light` and `yield-to-right`-regulated ones, a simulation output of around 1800-2000 trajectories is targeted. On the other hand, for `traffic-sign`-regulated scenarios, an output of around 3800-4000 trajectories is targeted. This higher number of trajectories is chosen in order to sufficiently represent `stop` and `yield`-regulated intersection paths in the data. As explained in section 4.2.2.2.2, vehicles driving on intersection paths belonging to these regulations are often inserted on edges with rather low source weights. Vehicles following `priority`-regulated intersection

[2] *Floating Car Data* can be defined as data obtainable from localization-based tracking approaches [33].
[3] The *Amitran* standard describes vehicular state data and originates from initiatives aiming at analyzing traffic-induced emissions topics. *Amitran* is an abbreviation for "Assessment Methodologies for Information and Communication Technology in Multimodal Transport from User Behavior to CO2 Reduction" [34].

paths are, however, often inserted on edges with relatively high source weights. Therefore, the former vehicles are much lower in quantity.

Intersections of the initially created set of 16 intersection scenarios can be described by 178 intersection paths. By adding four more scenarios, 230 intersection paths are available in total. More details are provided in Table 4.3. In this table, one can see how the number of intersection paths is distributed over the regulation classes. The first column refers to the set of 16 initially created intersection scenarios and the second column to the augmented set of 20 intersection scenarios. Furthermore, when analyzing all 230 intersection paths, it is found that none is followed by less than 17 or more than 1133 vehicles. On average, an intersection path is followed by approx. 267 vehicles.

	Number of intersection paths	
Traffic regulation	Dataset comprising 16 intersection scenarios	Dataset comprising 20 intersection scenarios
`Priority`	46	74 (+28)
`Stop`	20	=
`Traffic light`	50	=
`Yield`	20	44 (+24)
`Yield to right`	42	=
Total	178	230 (+52)

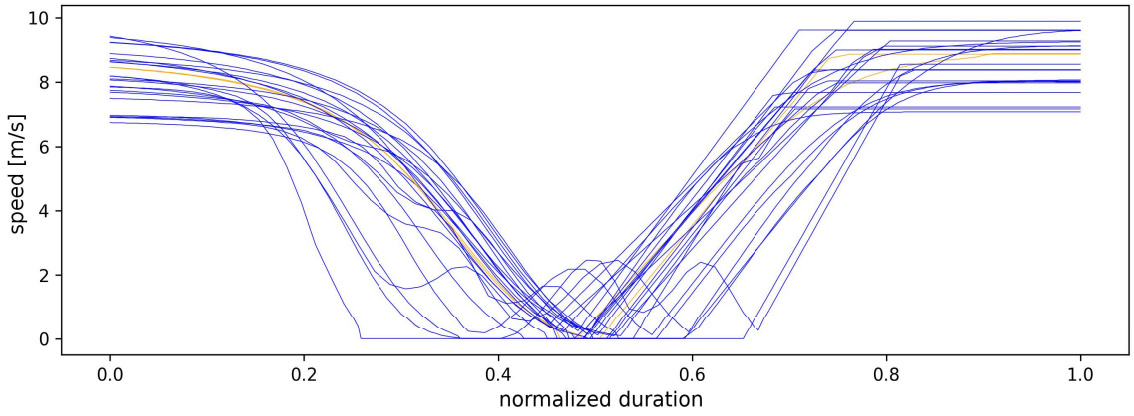
Table 4.3: Quantity of intersection paths in the dataset. The numbers indicate how many paths belong to which traffic regulation. The columns provide a comparison between the dataset comprising the 16 initially created intersection scenarios and the augmented dataset comprising all 20 intersection scenarios. The "=" points out that a certain number of intersection paths in the augmented dataset is the same as in the initial dataset.

Details of the data obtained from simulation are shown in the following subsections. Firstly, motion-related data from the simulation output is focused on in subsection 4.3.1. Secondly, concepts and features for putting this information into an intersection context are introduced in subsection 4.3.2.

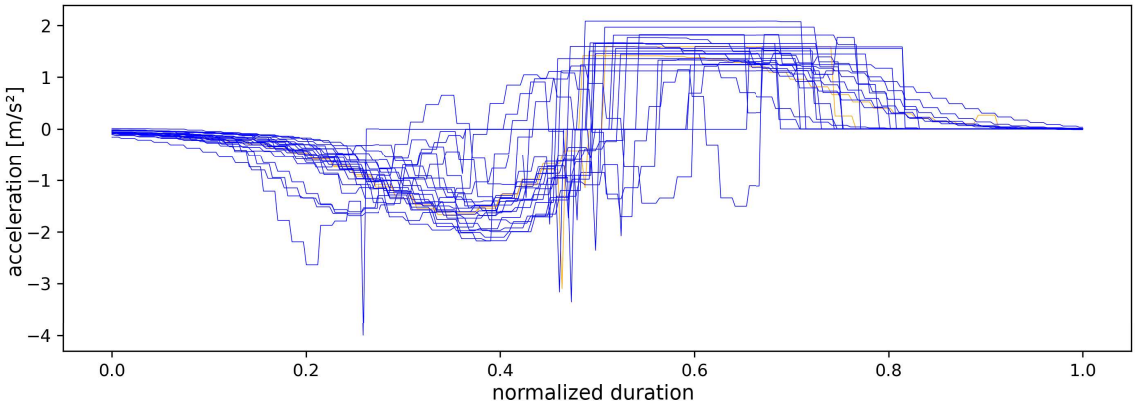
4.3.1. Motion-Related Trajectory Features

From simulations, $\boldsymbol{x}$, v and a data of vehicles crossing intersections is collected. In this thesis, v and a series are employed as motion-related trajectory features. Subfigures (a) and (b) of Figure 4.3 show v and a data, respectively, describing 25 trajectories. The data is plotted against a temporal axis that reflects the individually normalized duration of each trajectory. The trajectories follow various `stop`-regulated intersection paths. These are all limited to a segment that starts 70 m before and ends 70 m after the point closest to the corresponding intersection center. Dark yellow lines describe vehicles going straight, whereas blue lines describe such that execute turning maneuvers. From Figure 4.3 (a), one can see that v data curves are smooth. Vehicles start braking from the beginning which manifests in decreasing v. Approximately after the first fifth of the normalized duration, vehicles brake increasingly harder. This increase in deceleration magnitude can also be seen in Figure 4.3 (b) where the associated a data is shown. The braking continues until, halfway through the depicted range, a brief halt can often be observed. It manifests itself in v values close to $0.00\,\frac{\mathrm{m}}{\mathrm{s}}$. Subsequently, v increases for all trajectories and frequently exhibits knees at the point of transition into constant relatively high values. In contrast to v, a series have a stepped shape. More precisely, small steps separate constant-value segments of short but equal length. A possible explanation might be that these steps are artifacts from employing the ballistic integration scheme. As mentioned in section 4.2.2.1.5, using this scheme

results in vehicle acceleration being kept constant during simulation time steps. Apart from that, in a few instances, large steps in a value can be observed. Trajectories containing a values that fall outside the interval between - 4.00 and 4.00 $\frac{\mathrm{m}}{\mathrm{s}^2}$ are, however, rarely found. These trajectories are completely excluded from further consideration. This is done as a values that fall out of the aforementioned range can be considered unrealistic in the context of urban driving [28, 35].



(a) Speed profiles



(b) Acceleration profiles

Figure 4.3: Speed and acceleration profiles of 25 vehicles crossing intersections via `stop`-regulated intersection paths. The profiles are plotted against the individually normalized crossing duration of trajectories that are described by depicted profiles. Speed data is shown in subfigure (a), acceleration data in subfigure (b). Dark yellow lines indicate vehicles driving straight, whereas blue lines indicate such that execute turning maneuvers.

In a real-world context, the data would be collected by means of sensors. Thus, in order to introduce inaccuracies that one would expect from sensor-based v and a measurements, white Gaussian noise is added to the data. This reduces above-described smoothness of v series. The standard deviation of the density function creating noise for v data is globally set to the value 0.28 $\frac{\mathrm{m}}{\mathrm{s}}$. For the a data, a standard deviation of 0.18 $\frac{\mathrm{m}}{\mathrm{s}^2}$ is chosen. Subsequently, a moving average filter with window size 5 is applied to both v and a measurements of each trajectory. This only slightly reduces the noise of this data. However, the course of the latter measurements does no longer show small steps in value, while greater steps can still be observed. Visualizations of further processed data can be found later in section 5.1.

4.3.2. Intersection Paths and Progress-Related Trajectory Features

Progress-related trajectory features provide a spatial or temporal context for trajectory points. Features of this category are calculated based on intersection paths. A more detailed introduction

to intersection paths and information on how these are defined within this thesis is provided in subsection 4.3.2.1. Subsequently, a progress-related feature specifying the distance of trajectory points to the center point of the intersection the trajectory crosses is introduced in subsection 4.3.2.2. Another progress-related feature describing the temporal difference of trajectory points with respect to such located near the center of the intersection the trajectory crosses is outlined thereafter in subsection 4.3.2.3. As is shown in these subsections, the topology of an intersection, such as its number of arms or lanes, has no influence on the calculation of these two features. This way, they establish a quantified intersection context for trajectories, while simultaneously abstracting from topological intersection characteristics. This enables to compare trajectories independently of which intersection they cross.

4.3.2.1. Intersection Paths

In this thesis, intersections are represented by a set of intersection paths. These also serve as elements that are classified according to the traffic regulation that applies for them. A lane of an intersection arm approaching an intersection often permits to choose between going straight or turning in order to reach certain target arms. Intersection paths individually represent each of the possible paths that can be followed in this context and always start on intersection-incoming lanes. This way, multiple intersection paths coincide on the course of an incoming lane until splitting up in order to describe multiple different paths available to cross and leave the intersection.

An intersection path is represented by a sequence of points spatially describing its course. The coordinates of these points are taken from the SUMO network definition of the associated intersection scenario. The available coordinates are linearly interpolated to consistently represent intersection paths by 250 equally spaced points. The number of intersection paths per scenario spans between six and 14 for the scenarios designed in this thesis. For instance, an intersection with three arms shall be considered. Each arm has one lane per direction. From each incoming lane two outgoing target lanes can be reached. Consequently, this intersection is represented by six intersection paths. Depicted in Figure 4.4, an instance of an intersection that is represented by 14 intersection paths is provided. In this figure, intersection paths are visualized as blue lines running through their points. They overlay the SUMO network extracted from the intersection Residenzstraße with Friedrich-Wilhelm-Straße and Deutsche Straße in Berlin.

By means of intersection paths, trajectories crossing an intersection can be grouped. A trajectory is assigned to the intersection path that best describes the spatial course of it. It is determined as follows: For each point of a trajectory occurring before reaching the point closest to the intersection center, the closest point of the first candidate intersection path is determined. Thus, pairs that each contain a trajectory and an intersection path point are obtained. For each of these two-point pairs, the Euclidean distance between the two contained points is calculated. Subsequently, all resulting distances are summed up. This process is repeated for all candidate intersection paths. The trajectory is then assigned to the intersection path for which the sum of distances is minimal.

4.3.2.2. Intersection Path Distance

The progress-related intersection path distance feature d^{IntP} provides a spatial context for trajectories. It is introduced in this section. Initially, a coordinate system along each intersection path needs to be defined. The point of the intersection path that lies closest to the intersection center point marks the origin of its coordinate system. An intersection path point’s distance value

is calculated by accumulating the point-to-point distances of the sequence of intersection path points starting at the origin point and ending at the point the value is calculated for. Thus, the magnitude of these distance values increases in and against driving direction from the origin along the intersection path. Subsequently, distance values of intersection path points located, in driving direction, before the origin are assigned a negative sign. Distance values of intersection path points located after the origin have a positive sign. Therefore, this one-dimensional intersection path coordinate system describes an axis of continuously increasing distance values.

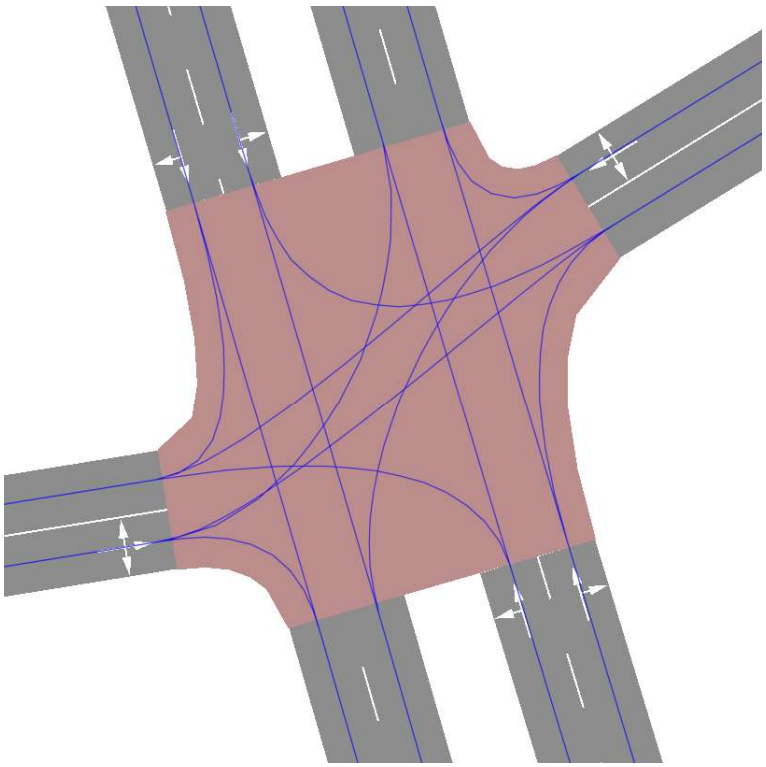


Figure 4.4: A SUMO network representation of an intersection overlaid with the associated set of intersection paths. The network represents the intersection Residenzstraße with Friedrich-Wilhelm-Straße and Deutsche Straße in Berlin. Intersection paths are visualized as blue lines.

All trajectories assigned to the same intersection path share its above-described coordinate system. Before calculating the d^{IntP} feature for each individual trajectory, trajectory $\boldsymbol{x}$ data is processed. All two-dimensional positions $\boldsymbol{x}_j$ of a trajectory together form a series. This series can be divided into separate series of x_j and y_j coordinates. Initially, both coordinate series of each trajectory are individually smoothed using a moving average filter with window size 5. Then, in order to calculate the d^{IntP} feature, each trajectory point is projected onto the intersection path it is assigned to. More precisely, as intersection paths are sequences of piecewise linear segments, each trajectory point is orthogonally projected onto the intersection path's line segment it lies closest to. The respective d^{IntP} value being assigned to the projected trajectory point is then determined via linear interpolation of the distance values of the two intersection path points defining this line segment. Figure 4.5 shows d^{IntP} values of a trajectory segment that starts 70 m before and ends 70 m after the point closest to the center of the intersection the trajectory crosses. The depicted trajectory runs from the top of the depiction and exhibits a right turn at the intersection. The color gradient from blue to red of depicted circles marks increasing d^{IntP} values. The point annotated with the lowest absolute d^{IntP} value is marked by a gray circle. The center point of the associated intersection is marked by a black cross. Both shapes are found close

to the bottom right corner of the figure. The axes describe the geographical coordinates x and y of the represented intersection scenario excerpt.

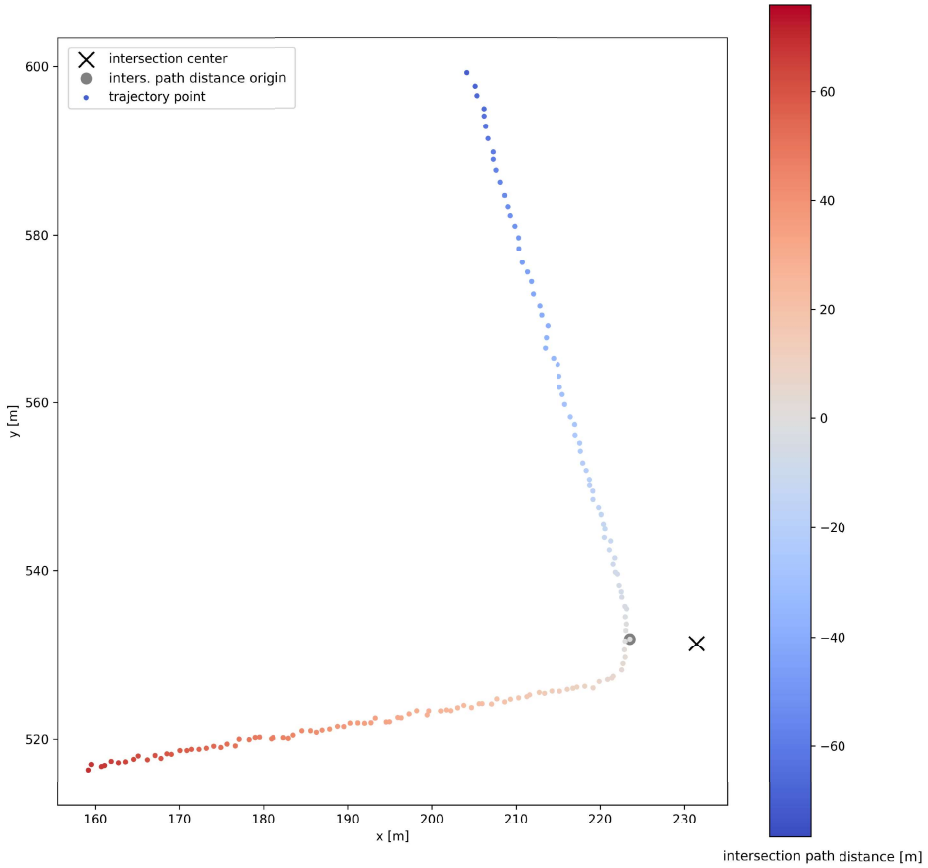


Figure 4.5: Depiction of the intersection path distance feature of a trajectory coming from the top of the figure and exhibiting a right turn at an intersection. Bluish points mark negative pre-intersection-center distance values, whereas points of increasingly red color mark post-intersection-center distance values. The color bar displayed on the right brings the colors into a numerical context. The axes of the figure describe geographical coordinates x and y. The black cross and the grey circle denote the intersection center point and the trajectory point with the lowest absolute intersection path distance, respectively.

4.3.2.3. Intersection-Relative Time Feature

Additionally, trajectory points can be annotated with temporal information with reference to the center of the intersection the trajectory crosses. For this purpose, the progress-related intersection-relative time feature t^{Int} is calculated. The trajectory point lying closest to the origin of the intersection path coordinate system serves as the origin of this feature. Hence, this is a trajectory point with d^{IntP} value close to 0. For each trajectory, the t^{Int} feature is calculated by subtracting the timestamp of the trajectory point closest to the origin of the intersection path coordinate system from the timestamps of all trajectory points. Consequently, negative t^{Int} values describe how long a measurement occurred before the trajectory point closest to the origin of the intersection path coordinate system is reached during a crossing. Measurements gathered after this point are annotated with positive t^{Int} values. Similar to the intersection path coordinate system, values are increasing in driving direction.

4.4. Validation Strategy

In this thesis, simulated data is employed to develop and test classifiers that infer traffic regulations. In order to evaluate the generalization ability of these approaches, development and testing is performed in a cross-intersection manner. This modification of cross-validation is described in the following and additionally depicted in Figure 4.6.

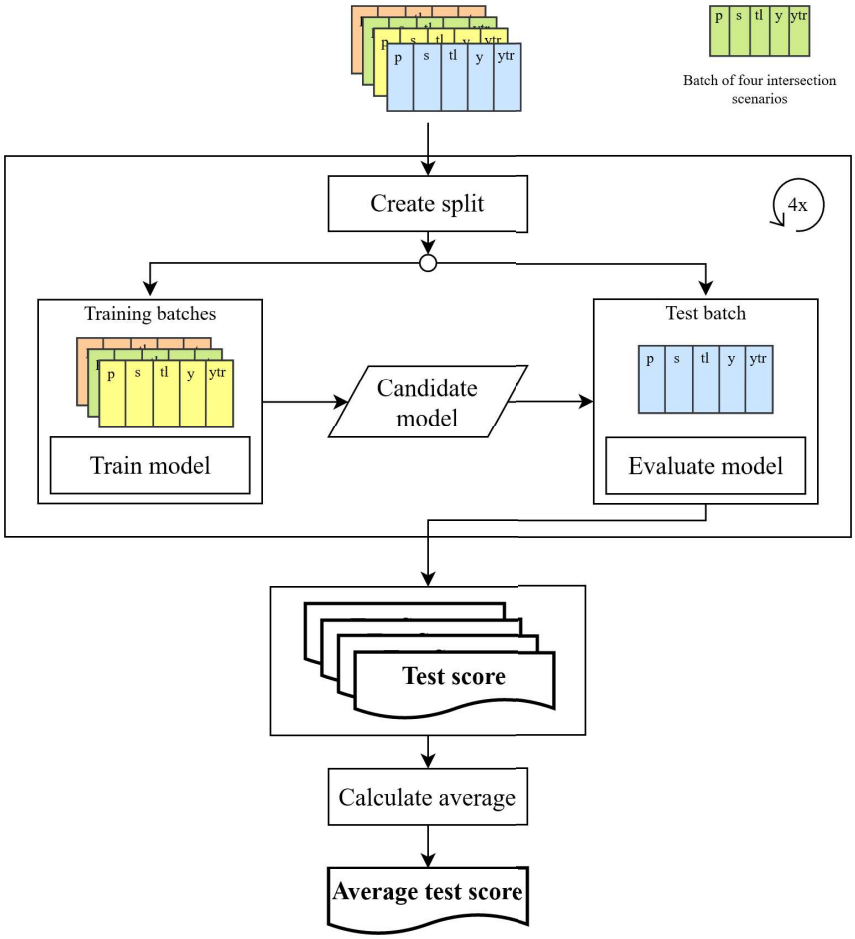


Figure 4.6: Flow chart of the cross-intersection validation strategy employed in this thesis. Trajectory data, associated to the intersection scenarios designed for this thesis, is divided into four batches. In doing so, each intersection scenario's data is entirely contained in solely one batch, hence not distributed over multiple ones. After splitting the four batches into three training and one test batch, a classifying model is developed using the training batches. Its performance is evaluated using the test batch. This process is repeated four times with different test data being used during each round. The average test score is computed by averaging the four individually achieved test performance scores. Differently colored boxes with a slot for each regulation class represent the batches. Regulation classes are abbreviated and appear in the order `priority`, `stop`, `traffic light`, `yield` and `yield to right`. Rectangular boxes and parallelograms denote actions and models, respectively. Scores are represented as boxes with a curved line as bottom boundary.

Validating approaches in a cross-intersection manner means that classifiers are developed on data associated to a specific portion of intersections represented in the data. Data for testing, however, is consistently taken from intersections not comprised in the portion used for classifier development. The latter can also be termed training portion. Furthermore, development and testing data shall be rotated in a way that data from each intersection serves as test data once. For this purpose, the data associated to the 16 initially created intersection scenarios is divided into

four batches of four scenarios each. Each batch then contains one scenario regulated by `traffic lights` and one by `yield-to-right` rule. Also contained in each batch is one scenario in which `priority` and `stop` intersection paths meet and one scenario in which `priority` and `yield` intersection paths meet. Hence, the five regulation classes are represented in each batch. These batches are maintained throughout this thesis. Classifier development is conducted using three of the four batches, whereas the remaining batch serves for testing and performance evaluation. Data associated to the four additionally created intersection scenarios is consistently added to each development portion. This process can be performed four times with the test batch being a different one each time. In other words, a set of these four processes cross-validates a classifier's performance [12]. A score summarizing the individual performances that the classifier achieved on all test sets can be obtained by averaging the four individually calculated test measures.

5. Inference of Traffic Regulations at Intersections

HMMs can be used to effectively model processes that produce visible or measurable output signals [13]. In the context of this thesis, intentions of drivers are interpreted as causes underlying observable signals. A driver might, for instance, find herself in a situation in which she is able to drive without having to give way to other traffic participants or follow slow-moving traffic. In other situations, she might decide to brake and come to a halt or execute a turning maneuver. Motion-related signals, such as v and a profiles, are produced by her behavior. Thus, sequences of trajectory points constitute observable information of Hidden-Markov-modeled traffic processes.

There is a plethora of works in which HMMs are employed to model traffic-related processes. Among these, the works [36–38] are instances of such focusing on processes occurring in the proximity of intersections. In [37] and [38], the authors aim at predicting the direction drivers choose to follow when reaching intersections. They distinguish between turning left, turning right and continuing straight. In [37], it is additionally attempted to predict cases in which drivers intend to stop at an intersection before continuing to cross it. A different goal is pursued in [36]. In this work, drivers approaching intersections shall be categorized according to whether they comply with traffic regulations or not. When sifting through these and more works, a set of frequently recurring design characteristics emerges: Each class that shall be distinguished from others is represented by one HMM. Each of these models is trained using trajectory data that is associated to the class it represents. Also, inference is done by returning the class represented by the HMM that best describes the evidence associated to the trajectory or object to be classified.

In this thesis, an HMM-based approach is applied to the problem of traffic regulation inference. The groundwork of this approach is inspired by above-outlined works [36–38], especially in terms of the above-mentioned recurring design characteristics. An introduction to the theoretical basics of the HMM concept and employed algorithms is provided in section 3.1. The representation of trajectories has to serve as a suitable basis HMMs are parametrized on. The different feasible parametric design choices specifying this representation are outlined in subsection 5.1. The design of the suggested approach, which is developed in this thesis, is described in subsection 5.2.

5.1. Trajectory Data Preprocessing

In this thesis, trajectories are represented as sequences of trajectory points. The following subsections outline preprocessing steps that are employed to specify details and characteristics of this representation. Clipping of trajectories is explained in subsection 5.1.1. The selection and normalization of trajectory features is outlined in subsection 5.1.2. How trajectory segmentation and resampling is done is shown in subsections 5.1.3 and 5.1.4, respectively.

At the end of this section, Figure 5.1 and Figure 5.2 show the normalized v and a feature, respectively, of trajectories following differently regulated intersection paths. Each subplot shows 150 trajectories. Displayed are intersection-incoming segments, plotted against the normalized t^{Int} feature and resampled to a length of 71 points. Dark yellow lines indicate vehicles going straight and blue lines indicate such executing turning maneuvers. Depictions of trajectories resampled to distance-based representations can be found in Figure A.1.1 and Figure A.1.2 in section A.1 of the Appendix. Additionally, Table 5.1 provides a summary of the preprocessing steps described in this section.

5.1.1. Trajectory Clipping

As mentioned in section 4.1.1, intersection scenarios are designed to represent excerpts of real-world intersections. However, simulated trajectories are not used in full length but are symmetrically clipped before being target of further processing steps. More precisely, points of the intersection-incoming and outgoing segment of each trajectory having absolute d^{IntP} values greater than a specified distance threshold are discarded. The clipping distance value of 70.00 m is applied in every experiment of this thesis.

5.1.2. Feature Selection and Normalization

The features employed to describe trajectories in terms of motion are v and a. Those that indicate intersection-relative temporal and spatial progress of trajectories are t^{Int} and d^{IntP}, respectively. Trajectories can be represented using either a non-empty subset of these features or all of them.

All trajectory features are normalized for the use in experiments. These are denoted by a tilde. For example, $\tilde{v}$ is the symbol for the normalized v feature, $\tilde{d}^{IntP}$ is the symbol for the normalized d^{IntP} feature etc. This is done in order to initially force the traffic regulation inference method to not focus on or get influenced by absolute values. The following example shall illustrate how this is found to occur: In the dataset, all arms of `yield-to-right`-regulated intersection scenarios have a common speed limit of 30 $\frac{\text{km}}{\text{h}}$ ($\approx 8.33\ \frac{\text{m}}{\text{s}}$). In early stages of this thesis, developed classifiers are found to misuse absolute v data resulting from this speed limit as a characteristic for `yield-to-right`-regulated intersection paths. As a result, many `yield`-regulated intersection paths that also have a 30 $\frac{\text{km}}{\text{h}}$ speed limit are falsely classified as `yield-to-right`-regulated ones.

The v feature of trajectories is normalized depending on their intersection path assignment. From all trajectories assigned to a specific intersection path, the maximum v value is being determined. The v series of all these trajectories are then scaled by dividing all values by the intersection-path-wide maximum v value. Negative v values, which can be caused by measurement noise, are replaced by the value 0. Thus, $\tilde{v}$ values range between 0 and 1.

As already mentioned in section 4.3.1, trajectories with a values falling outside the range between - 4.00 and 4.00 $\frac{\mathrm{m}}{\mathrm{s}^2}$ are excluded. The a feature of all remaining trajectories is scaled by dividing all values by 4.00 $\frac{\mathrm{m}}{\mathrm{s}^2}$. This way, the baseline $a = 0.00\ \frac{\mathrm{m}}{\mathrm{s}^2}$, hence the absence of acceleration and deceleration, is not shifted and the sign of all a values is preserved. Consequently, $\tilde{a}$ values range between - 1 and 1.

After clipping, as described in section 5.1.1, no trajectory point with a d^{IntP} value greater than the clipping distance remains. For all trajectories, the d^{IntP} feature is scaled by dividing all values by the clipping distance. This results in $\tilde{d}^{IntP}$ values that range between - 1 and 1, as well.

The t^{Int} feature is normalized by dividing all values by the t^{Int} value assigned to the first point of the clipped trajectory. This way, the scaled $\tilde{t}^{Int}$ values range between - 1 and a varying upper bound. As explained in section 4.3.2.3, the t^{Int} value assigned to the first point of the clipped trajectory can be interpreted as the duration of its intersection-incoming segment, hence the duration until the point closest to the intersection center point is reached. Thus, for outgoing trajectory segments having lower durations than their corresponding incoming ones, upper bounds take on values lower than 1. Conversely, for outgoing segments having greater durations than their corresponding incoming ones, upper bounds take on values greater than 1.

5.1.3. Trajectory Segmentation

Trajectories can be split into intersection-incoming and outgoing segments. Therefore, it is possible to choose between using entire trajectories or only segments for further processing. The incoming segment is obtained by discarding trajectory points that have d^{IntP} values greater than 0. Discarding only trajectory points having d^{IntP} values lower than 0 yields the outgoing segment. Besides, the splitting can be analogously done for intersection paths. However, instead of d^{IntP} values, distance values of intersection path coordinate systems are used. Both incoming and outgoing segments of intersection paths retain the point associated to the coordinate system's origin.

When evaluating trajectories following differently regulated intersection paths, it is found that outgoing trajectory segments often show patterns that are similar among themselves regarding the v and a feature. After possibly giving way to other traffic participants, drivers often accelerate to a specific speed and leave the intersection. This emerges as v values that mostly increase up to a certain point and stay constant from there on. Associated a values increase and fall back to values close to 0. Thus, it is decided to focus on incoming segments for further use.

In this context, an issue with respect to v feature normalization shall not be neglected. Different speed limits may apply for the incoming and outgoing segment of a single intersection path. As illustration, a `yield`-regulated intersection path shall be considered. On its incoming segment, a speed limit of 30 $\frac{\mathrm{km}}{\mathrm{h}}$ applies. Its outgoing segment coincides with a `prioritized` intersection path. On this segment, a speed limit of 50 $\frac{\mathrm{km}}{\mathrm{h}}$ ($\approx 13.89\ \frac{\mathrm{m}}{\mathrm{s}}$) applies. After normalization, $\tilde{v}$ feature values of trajectories following this intersection path unhindered, hence without giving way or stopping, exhibit a characteristic pattern. The $\tilde{v}$ values of trajectory points on the incoming segment generally are considerably lower than corresponding ones on the outgoing segment. As solely incoming trajectory segments are used in subsequent steps, an alternative normalization of the v feature can be done. It is based and applied solely on incoming segments instead of entire trajectories. Consequently, the search for the maximum v value used for scaling considers only

values of points on incoming trajectory segments. Remaining steps, in the context of v feature normalization, are analogously executed as outlined in section 5.1.2.

5.1.4. Trajectory Resampling

Points of trajectories are annotated with measurements that are sampled at a constant frequency of 10 Hz. Thus, the number of points in a trajectory covering an intersection-incoming segment having a specific length varies depending on the driving speed of the associated vehicle. The difference between t^{Int} values of successive points of this trajectory has a constant value of 0.1 s. In the context of this thesis, a trajectory having a t^{Int} feature with equally spaced values is referred to as being represented with reference to a time-based axis. Self-evidently, differences between d^{IntP} values of successive trajectory points are also varying depending on driving speed. Transforming trajectories to a representation in which successive trajectory points have a constant difference between their d^{IntP} values, creates a distance-based view instead of a time-based one. Consequently, trajectories transformed this way are referred to as being represented with reference to a distance-based axis. Both a specific number of points and different representation axes can be realized by means of trajectory resampling. Whenever incoming trajectory segments are resampled for use in experiments of this thesis, the resampled trajectories constantly have a length of 71 points.

5.1.4.1. Resampling to a Time-Based Representation

In the context of this thesis, trajectories are represented with reference to a time-based axis per default. This reference axis type can be preserved, while altering the number of trajectory points by means of resampling. For this, the t^{Int} feature serves as reference axis. A function mapping arbitrary values of the t^{Int} feature to values of each of the features v, a and d^{IntP} is required. As trajectory points are consistently annotated with values of all four mentioned features, the function can be approximated by means of linear interpolation. This way, for a desired number of values equally spaced along the t^{Int} feature range of each clipped trajectory or trajectory segment, values of the features v, a and d^{IntP} are sampled from the approximated function.

5.1.4.2. Resampling to a Distance-Based Representation

With no moving backwards occurring, values of the d^{IntP} feature consistently increase between each trajectory point and its successor. However, localization noise inevitably causes omnidirectional dispersion of trajectory point positions $\boldsymbol{x}$. Thus, not only increases but also decreases between d^{IntP} values of successive trajectory points are possible while an associated vehicle drives slowly or is in standstill. For this reason, considering measurement noise, a standstill is assumed for trajectory points that have v values equal to or lower than 1.00 $\frac{\mathrm{m}}{\mathrm{s}}$ on a contiguous interval with a duration of at least 0.50 s. In order to simplify resampling to a distance-based representation, the following is done: Separately for each of the features v, a and t^{Int}, all values sampled on a standstill interval are replaced by their mean.

Transforming trajectories to a distance-based representation is done also by means of resampling, which is outlined in the previous section. Instead of t^{Int}, the d^{IntP} feature serves as reference axis. Thus, for a desired number of values equally spaced along the range of this feature of each clipped trajectory or trajectory segment, values of the features v, a and t^{Int} are sampled from the function that is analogously approximated.

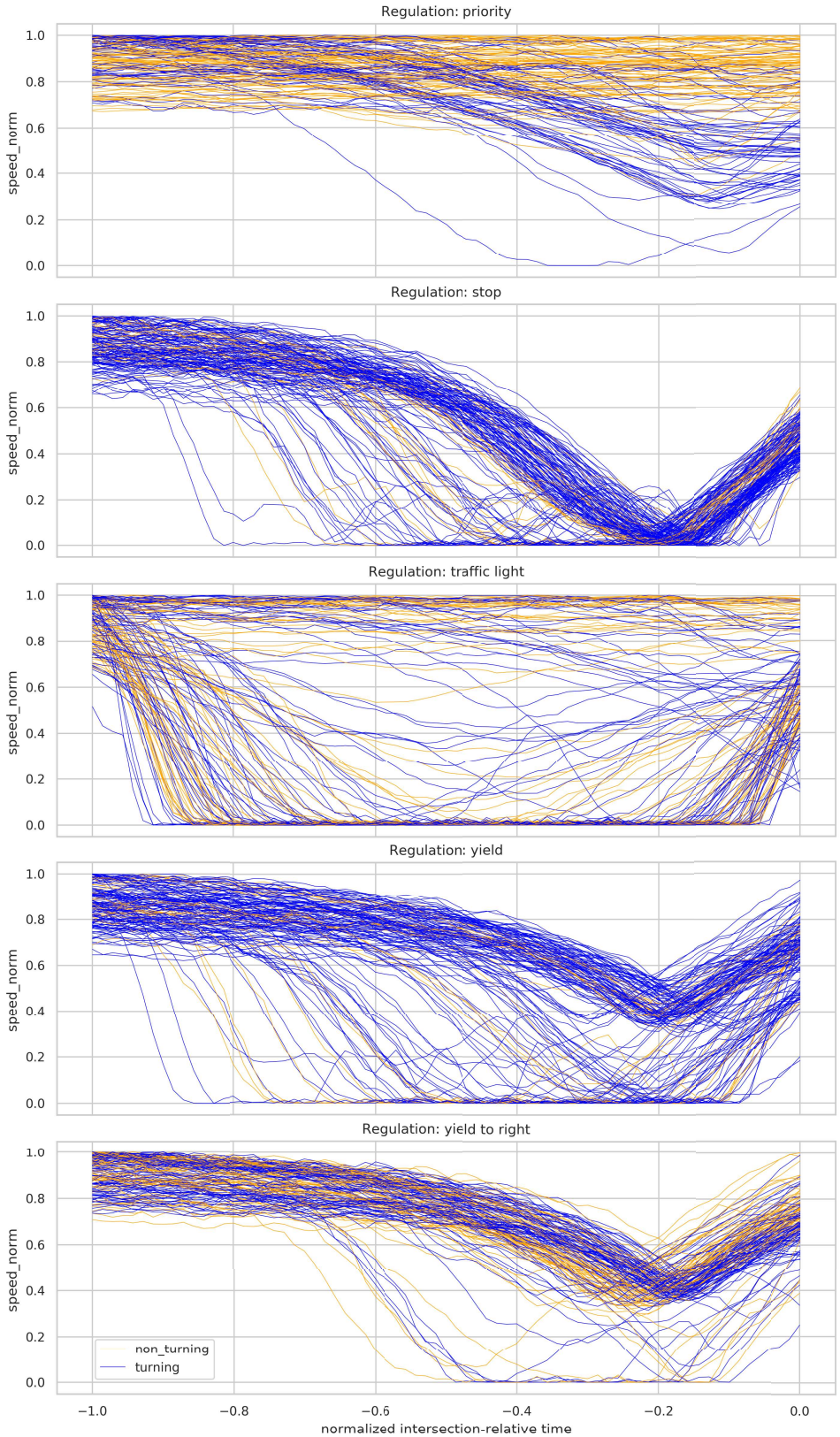


Figure 5.1: Normalized speed of incoming trajectory segments following differently regulated intersection paths, plotted against the normalized intersection-relative time feature. Depicted trajectories are resampled to a time-based representation and have a length of 71 points. Dark yellow lines indicate vehicles going straight, blue lines indicate such executing turning maneuvers. Each subplot shows 150 intersection-incoming trajectory segments.

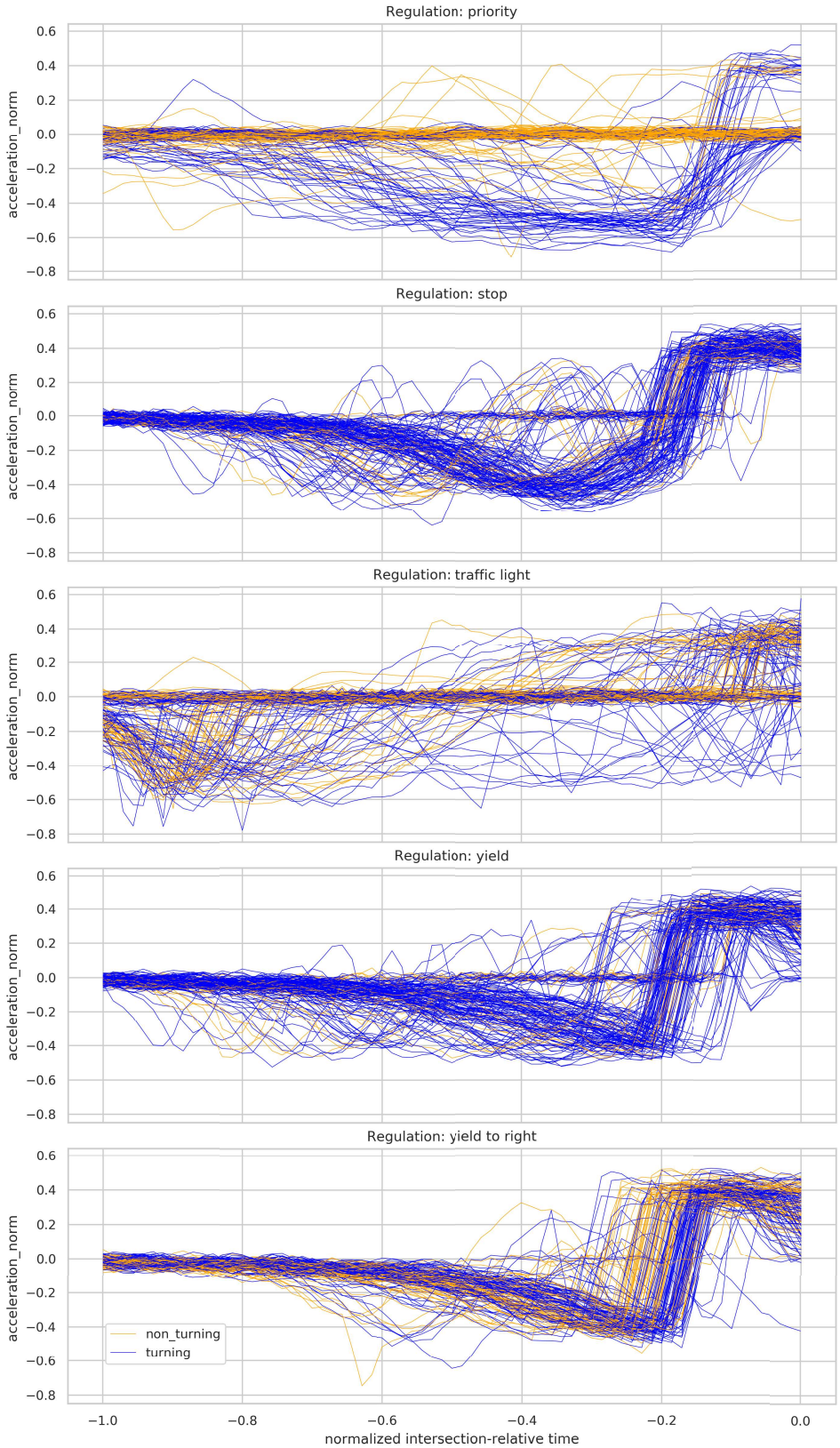


Figure 5.2: Normalized acceleration of incoming trajectory segments following differently regulated intersection paths, plotted against the normalized intersection-relative time feature. Depicted trajectories are resampled to a time-based representation and have a length of 71 points. Dark yellow lines indicate vehicles going straight, blue lines indicate such executing turning maneuvers. Each subplot shows 150 intersection-incoming trajectory segments.

Parameter	Value / value range
Clipping distance	70.00 m
Feature selection	One or more of $\{v, a, t^{Int}, d^{IntP}\}$
Feature normalization	With
Trajectory segment	Incoming
Resampling	One of {with, without}
Reference axis	One of {time-based, distance-based}
Number of points in resampled and segmented trajectory	71
Standstill interval conditions	$v \leq 1.00\,\frac{\mathrm{m}}{\mathrm{s}}$, interval duration ≥ 0.50 s

Table 5.1: Parameters and design choices specifying the series-based trajectory representation employed in this thesis. Each row of the table corresponds to a preprocessing step and the associated parameter value or range that is considered in this thesis.

5.2. Approach Design

The approach designed in this thesis employs HMMs to represent regulation-class-specific information gathered from trajectories. How HMMs are parametrized and which parameters have to be set in this context, is explained in subsection 5.2.1. Subsection 5.2.2 expands on the suggested design of an HMM-based classification of intersection paths.

5.2.1. Parametrizing Hidden Markov Models

HMMs are parametrized on trajectories. In this context, several parameters have to be set. These are explained in the following. Table 5.2 additionally provides an overview of these.

Topology: The topology of HMMs is chosen to be ergodic. Thus, transitions shall initially be possible between all possible pairs of hidden states [13].

Number of hidden states: HMMs are parametrized based on trajectory data. For each parametrization procedure, the number of hidden states has to be set to a fixed value. In experiments of this thesis, this number takes on values ranging between 2 and 10.

Parameter estimation procedure: HMM parametrization is performed using the Baum-Welch Algorithm, which is introduced in section 3.1.3. In order to provide an initialization of HMM parameters for this algorithm, the software employed for experiments uses the K-means++ clustering algorithm. Readers interested in details of this algorithm are referred to [39]. More details of the employed software are provided later in section 6. In the subsequent step, all trajectories that shall be considered as basis for parametrizing HMMs are grouped by the regulation class of the intersection paths they are assigned to. The initialized Baum-Welch Algorithm is then separately applied to all groupings. Consequently, this yields one individual HMM for each class. In order to measure how well an HMM describes trajectories of a certain regulation class, the natural logarithm of the likelihood, commonly referred to as log-likelihood, is used. The maximum number of iterations of the Baum-Welch Algorithm is set to $1.00 \cdot 10^8$. However, the algorithm terminates as soon as convergence is observed. This is the case if the increase in log-likelihood between two iterations is lower than $1.00 \cdot 10^{-9}$ or in case of a decrease. Convergence is observed for all parametrizations performed in the context of experiments in this thesis. Thus, the maximum number of iterations is never exhausted.

Observation probability density functions: As observation probability densities of HMMs, Gaussian distributions are use throughout this thesis. HMMs. During the parameter estimation procedure, the mean vector and covariance matrix, which define a Gaussian distribution, are

estimated from trajectory data. Each state of an HMM is assigned one Gaussian. The use of mixture distributions is not examined in this thesis.

Impact of trajectory feature selection: In the context of trajectory representations, one or more trajectory features are selected for describing trajectories. Details are provided in section 5.1.2. This feature choice also has an impact on characteristics of employed observation models. As mentioned above, these are represented by Gaussian distributions. In case one feature is selected, univariate distributions are employed. If more than one feature is selected, the distributions need to be multivariate. Also, each component of a Gaussian distribution's estimated mean vector can be related to one of the selected features. This becomes apparent when considering that trajectory points, hence vectors, randomly drawn from such a Gaussian frequently have low Euclidean distances to the Gaussian's mean vector. From its estimated covariance matrix, the standard deviations can be computed. Analogously, each standard deviation can be related to one of the selected trajectory features [40]. Thus, in this thesis, features are referred to as being associated to mean vector components and standard deviations. Apart from that, if the t^{Int} feature is part of the feature choice, the order of hidden states in a parametrized HMM is determined by each Gaussian's mean vector component associated to this feature. More precisely, the first hidden state is the one with the estimated Gaussian distribution having the smallest mean vector component that is associated to this feature. The second one has the second-smallest value and so forth. In case, the t^{Int} feature is not part of the feature choice, the d^{IntP} feature is used instead. The hidden-state order is then analogously determined. With none of the progress-related trajectory features being part of the feature choice, hidden states are randomly sorted. After that, this order is conserved.

Parameter	Value / value range
Topology	Ergodic
Number of hidden states	{2, 3, …, 10}
Initialization for parameter estimation procedure	K-means++
Parameter estimation procedure	Baum-Welch Algorithm
Observation probability density functions	Gaussian (univariate or multivariate, depending on trajectory feature selection)
Maximum number of iterations	$1.00 \cdot 10^{8}$
Convergence threshold	$1.00 \cdot 10^{-9}$

Table 5.2: Parameters and design choices relevant in the context of parametrizing Hidden Markov Models. Each row of the table corresponds to one characteristic and the associated parameter value or range that is considered in this thesis.

5.2.2. Hidden-Markov-Model-Based Inference

Classification of traffic regulations is done for intersection paths. More precisely, inference is conducted on all trajectories assigned to a specific intersection path. For this, the Forward Algorithm is used. It is introduced in section 3.1.2. Figure 5.3 is provided to additionally illustrate the design of the classification procedure.

Firstly, the Forward Algorithm returns a likelihood value for each trajectory, assigned to a specific intersection path, given a parametrized HMM. For further processing, the associated log-likelihood is calculated. Secondly, the determined log-likelihood values are summed up. The result can be interpreted as a log-likelihood measure of the intersection path. Outlined steps are repeated on the same set of trajectories with a different HMM being given each time. Thus, each log-likelihood value calculated for this intersection path refers to the class represented by the associated HMM. For the remainder of this thesis, these values are termed class log-likelihood

values. The classification decision is then made by identifying the regulation class that is related to the greatest class log-likelihood value. In principle, explained inference design resembles a majority-vote-based system. A purely majority-vote-based one would, for instance, assign a regulation class to each trajectory. The most common class would then determine the class of the intersection path. However, in this case, the intersection path's class is determined by comparing sums over trajectory-individually calculated log-likelihoods.

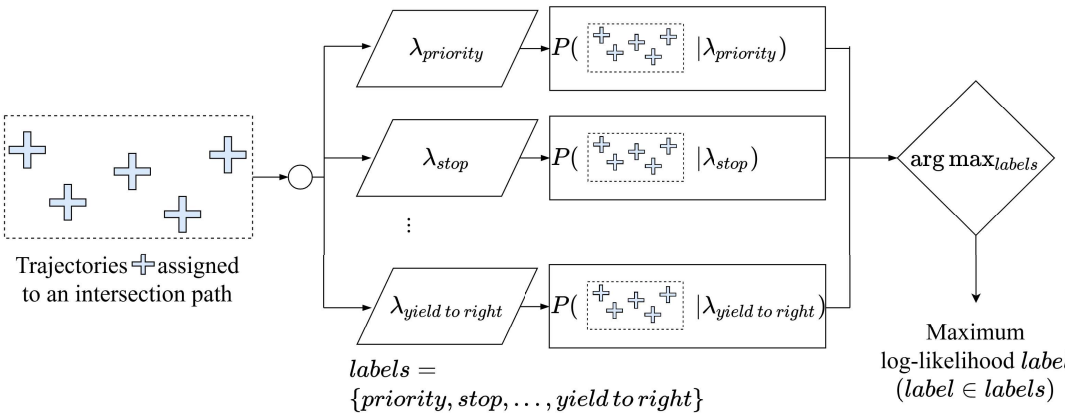


Figure 5.3: Schematic depiction of intersection path classification using multiple Hidden Markov Models. Trajectories are depicted as blue crosses, intersection paths as dashed boxes. Each model λ_{label}, with $label \in labels$, represents one traffic regulation class. The Hidden Markov Model that assigns the highest log-likelihood to an intersection path is the one determining the predicted class. Terms starting with *P*, depicted in the middle of the figure, denote the log-likelihoods. On the right side of the figure, the step of selecting the highest log-likelihood is visualized as a diamond-shaped box, with log-likelihoods incoming from the left and a maximum log-likelihood label being outputted. (Adapted from [37])

In order to compare inference results across different intersection paths, normalization of class log-likelihood values is done. For this, it is decided to only consider class log-likelihoods with positive values for normalization. Normalized values are then calculated by dividing each positive class log-likelihood value by the sum of positive values. For the case in which all log-likelihood values are negative, the reciprocals of all values, multiplied by -1, are used. However, this case is not observed in the context of this thesis.

6. Experimental Evaluation of the Suggested Approach

In experiments, the suitability of the suggested HMM-based approach for the purpose of traffic regulation inference is analyzed. This is done by comparing the classification performance of HMMs that are parametrized in different ways on trajectories having differently specified trajectory representations. For this, various performance measures are employed. The macro F_1 score is used to express classification performance as a single value while accounting for class imbalances. It is introduced in section 3.2.2. However, more detailed result analyses require more insight into performance on the different traffic regulation classes. For this task, confusion matrices are employed. From these, weaknesses of the approach in classifying intersection paths belonging to certain classes can be identified. This information can help to overcome such weaknesses in further experiments. A theoretical introduction to the concept of confusion matrices is provided in section 3.2.1. In this thesis, all presented confusion matrices are normalized. This is done by dividing each entry by the sum of all values belonging to the row associated to it. Apart from that, it is important to mention that classification results and, therefore, also the assessed performance can vary within small ranges. This is primarily owed to the fact that the associated algorithm starts from a random initialization, which is outlined in section 5.2.1.

Also, observation densities and transition probabilities of differently parametrized HMMs are analyzed in every experiment. As HMMs are parametrized and tested in a cross-validation manner, not just one but multiple sets of HMMs are parametrized based on the same parameters. However, each of these sets is parametrized on a different development data portion. Just as test data portions, these development data portions are maintained throughout all experiments. Details of the designed cross-validation strategy are provided in section 4.4. No approach to merge multiple sets of HMMs into one that would represent all sets is designed. Thus, for reasons of consistency, all HMMs that are presented and analyzed in the context of experiments are always parametrized on a particular development portion, unless explicitly indicated otherwise. For the remainder of this thesis, this particular development portion is referred to as analysis development data portion.

Approaches and experiments are implemented using version 3.6 of the Python programming language [41]. Program codes can be obtained from section B of the Appendix. The Python package pomegranate [42] implements HMMs. With the aid of an application programming

interface that is designed to be intuitive and similar to the one used by scikit-learn [43], it enables fast prototyping. Furthermore, multi-threading and graphics-processing-unit-based means for accelerating computations are already implemented. Apart from HMMs, pomegranate provides implementations of various probability distributions and a wide range of probabilistic modeling methods, as for instance Markov Chains and Bayesian Networks. Mathematical and vector or matrix-related operations are performed with the aid of the packages NumPy [44] and Pandas [45]. As tool for managing and evaluating the great numbers of experiments in this thesis, Sacred [46] is employed.

In order to maintain a reasonable scope of this thesis, only a selection of conducted experiments is presented in the following. This selection contains those experiments that contribute the most important insights regarding the scientific questions this thesis addresses. In an extensive grid search experiment, various parameters specifying trajectory representations and HMM training characteristics are scrutinized regarding their impact on classification performance. Design and results of this experiment are shown in subsection 6.1. Subsequent experiments examine ways to improve the classification result achieved in the grid search experiment. In order to achieve this, the impact of using more trajectory data gathered from more intersection scenarios for HMM development is examined. The corresponding experiment is presented in subsection 6.2. A different approach is shown in subsection 6.3. In this experiment, the number of HMMs that represent the set of traffic regulations is increased. Results of aforementioned experiments are discussed in section 6.4.

6.1. Grid Search

In addition to initially gathering valuable insights into the applicability of HMMs, results of this experiment form the basis and inspiration for further experiments. Subsection 6.1.1 presents the varied parameters and the experiment approach. Results are shown in subsection 6.1.2.

6.1.1. Experiment Design

Table 5.1 and Table 5.2 provide overviews of parameters that specify the representation of trajectories and characteristics of HMM parametrization, respectively. Some parameters are not varied in this thesis, whereas others can take on different values within provided ranges. Variation of the latter and comparing their impact on classification performance is target of this experiment. For clarity, these are listed in Table 6.1 together with their feasible ranges. Concerning the trajectory feature selection, four different tuples containing different combinations of trajectory features are chosen. A parameter combination is then sampled by choosing exactly one value for each parameter. Hence, these parameters represent the dimensions of the grid the search for suitable parameter combinations is performed on. However, one parameter combination that can be sampled as described above is infeasible. Namely, without resampling, trajectories cannot be represented with reference to a distance-based axis. The reason for this is that resampling is the necessary method to achieve this type of representation axis for trajectories. Details of trajectory resampling can be found in section 5.1.4. Consequently, all feasible parameter combinations amount to 108.

In what is referred to as an experiment run, one certain parameter combination is examined. Thus, during one run, one set of HMMs, representing the different traffic regulations, is parametrized on trajectories that are represented in a certain way. Both the trajectory representation and the

HMM parametrization are done as specified by this parameter combination. In this experiment, each of the five traffic regulations is represented by one HMM. Making use of the cross-validation strategy, which is introduced in section 4.4, each feasible parameter combination defines four separate runs. In each of these runs, one F_1 score describing the performance of the HMMs on the intersection paths of the test dataset is calculated. These are also referred to as test F_1 scores or, as presented scores describe test results per default, as F_1 scores. The performance associated to a specific parameter combination is then characterized by calculating the mean and standard deviation of these four F_1 scores.

Consequently, 432 individual runs are executed in this experiment. The 16 initially designed intersection scenarios provide the data that is used in the context of this experiment.

Parameter	Considered value range
Trajectory-representation-related	
Resampling	{with, without}
Reference axis	{time-based, distance-based}
Feature selection	$\{(\tilde{v}), (\tilde{v}, \tilde{a}), (\tilde{v}, \tilde{t}^{Int}), (\tilde{v}, \tilde{a}, \tilde{t}^{Int})\}$
Hidden-Markov-Model-parametrization-related	
Number of hidden states	{2, 3, …, 10}

Table 6.1: Parameters varied in the context of the grid search experiment. These parameters specify the representation of trajectories and characteristics of Hidden-Markov-Model parametrization. Each row corresponds to a parameter and the associated feasible range of the values it can take on. Each feasible parameter combination is examined in one experiment run.

The $\tilde{d}^{IntP}$ feature is not included in above-presented grid search. This is motivated by the intention to keep the computation time, which is depending on the number of runs, within reasonable limits regarding the thesis time frame. Instead, this feature is examined solely within the context of the parameter combination that is found to yield the best classification result of the grid search experiment. Stated parameter combination is presented later. In this combination, the $\tilde{t}^{Int}$ feature is substituted for the $\tilde{d}^{IntP}$ feature. However, this yields classification results that are inferior to results achieved by well-performing grid search experiment runs. Thus, it is refrained from showing associated results.

6.1.2. Experiment Results

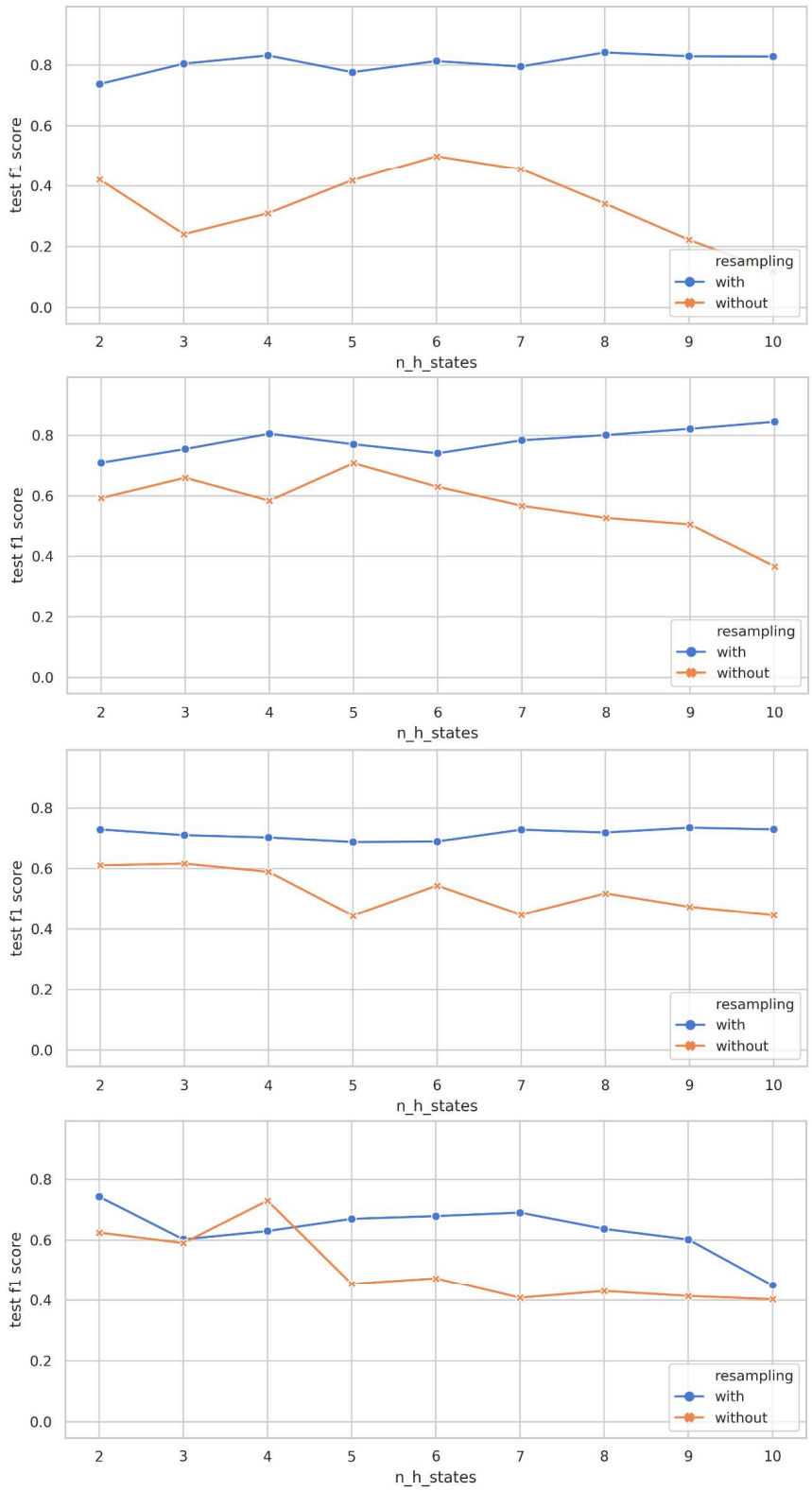
Achieved classification results are presented in subsection 6.1.2.1. A detailed analysis of parametrized HMMs achieving the best results is provided in subsection 6.1.2.2.

6.1.2.1. Evaluation of Varied Parameters

In subsections 6.1.2.1.1 and 6.1.2.1.2, the impact of varying parameters listed in Table 6.1 on achieved performance scores is shown. A detailed overview of all performance score means and standard deviations is provided in Table A.2.1 and Table A.2.2 in section A.2.1 of the Appendix. Subsection 6.1.2.1.3 expands on certain parameter combinations that yield the best performance scores among all examined ones.

6.1.2.1.1. Comparing Results with Regard to Trajectory Resampling

Initially, the 36 parameter combinations determining that trajectories are not resampled shall be considered. With four runs cross-validating each parameter combination, there are 144 runs. In the following, these are compared to parameter combinations determining that trajectories are resampled to a time-based representation. For this, Figure 6.1 provides an overview of test F_1



(a) $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$

(b) $(\tilde{v}, \tilde{a})$

(c) $(\tilde{v}, \tilde{t}^{Int})$

(d) $(\tilde{v})$

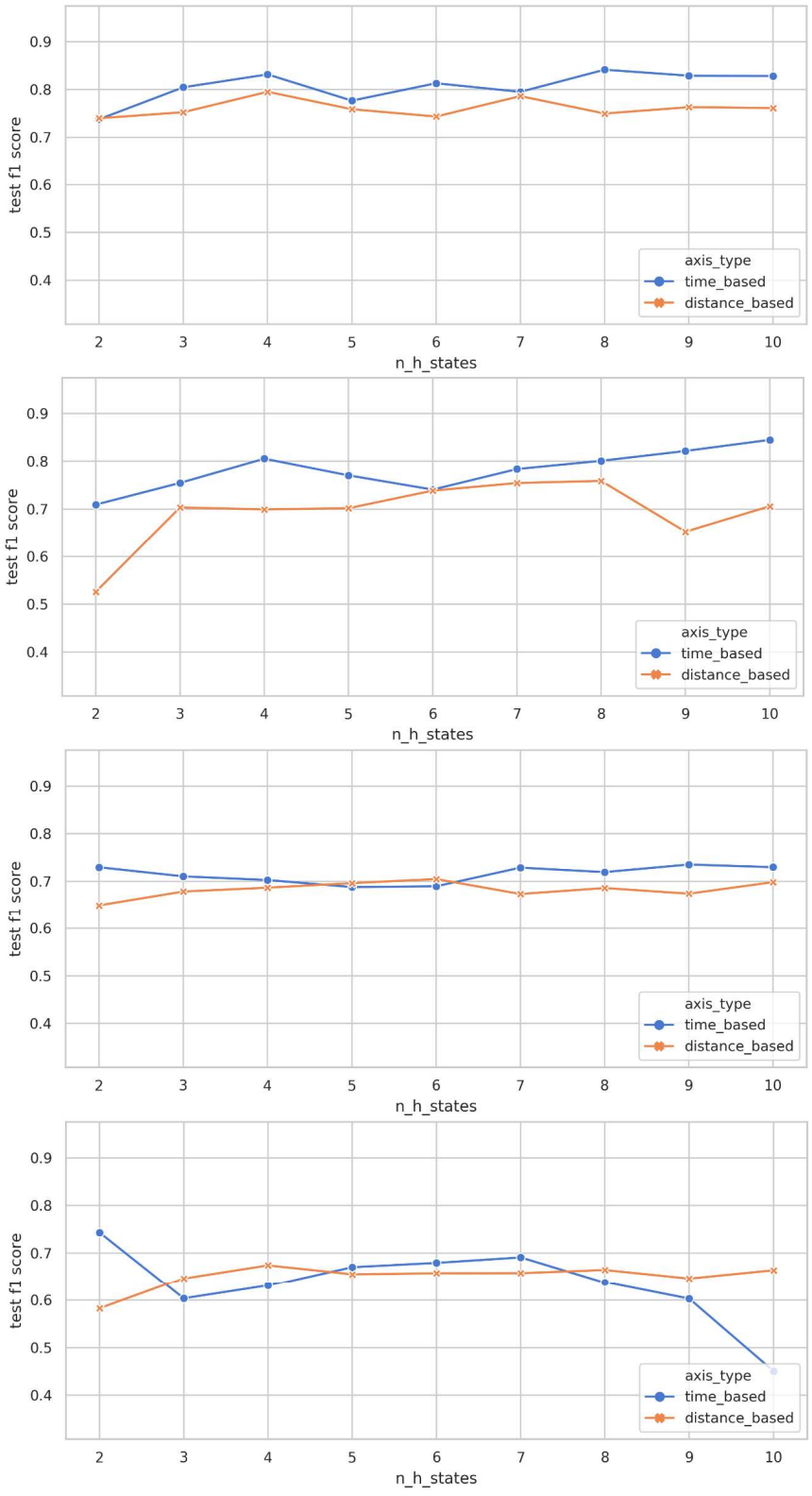
Figure 6.1: Performances associated to parameter combinations determining the use of trajectories resampled to a time-based representation and not resampled trajectories. The performance of each parameter combination is described by averaging the set of F_1 scores determined via cross-validation. Each subplot refers to one feature tuple being employed to describe trajectories in associated experiment runs. Subplot (a) refers to the tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$, (b) to $(\tilde{v}, \tilde{a})$, (c) to $(\tilde{v}, \tilde{t}^{Int})$ and (d) to $(\tilde{v})$. Blue curves denote test F_1 score means associated to the use of resampled trajectories. Analogously, the use of not resampled trajectories is denoted by orange curves. The scores are plotted over hidden-state counts ranging between 2 and 10.

score means associated to parameter combinations of both compared types. Score means associated to parameter combinations determining resampling to a time-based representation are depicted as blue curves. Orange curves mark score means associated to parameter combinations determining that no resampling occurs. These are plotted against the range of examined hidden-state counts. Each of the subfigures (a)-(d) refers to one of the different feature tuples employed to describe trajectories. When viewing subfigures (a)-(c), it is obvious that parameter combinations determining the use of trajectories resampled to a time-based representation are associated greater F_1 score means than those determining the use of not resampled trajectories. Stated subfigures, refer to the feature tuples $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$, $(\tilde{v}, \tilde{a})$ and $(\tilde{v}, \tilde{t}^{Int})$, respectively, to be employed for describing trajectories. F_1 score means associated to parameter combinations determining the use of the feature tuple $(\tilde{v})$ are depicted in subfigure (d). In the context of this feature tuple, for all but one hidden-state count, the use of trajectories resampled to a time-based representation is associated to higher F_1 score means than the use of not resampled trajectories. Stated exception is found for a hidden-state count of four. For this hidden-state count, a test F_1 score mean of approx. 0.731 is associated to the use of not resampled trajectories. The use of trajectories resampled to a time-based representation is associated to a test F_1 score mean of approx. 0.631.

However, as outlined above, a higher F_1 score mean is associated to a parameter combination determining the use of not resampled trajectories only once. Therefore, it is refrained from further analyzing runs using not resampled trajectories in detail. Instead, the main focus of analyses in this thesis is put on parameter combinations determining the use of resampled trajectories.

6.1.2.1.2. Comparing Results with Regard to Representation Axis Type of Trajectories, Trajectory Feature Selection and Hidden-State Count of Hidden Markov Models

The 72 parameter combinations determining the use of resampled trajectories are cross-validated in 288 experiment runs. An overview of test F_1 score means associated to these parameter combinations is provided in Figure 6.2. As in Figure 6.1, each of the subfigures (a)-(d) refers to one of the different feature tuples employed to describe trajectories. The test F_1 score means are plotted against the number of hidden states set for each HMM parametrization. The color of the depicted curves denotes the trajectory representation axis type. Blue marks time and orange marks distance-based representation. When viewing this figure, what stands out is that one of the two trajectory representation axis types is mostly associated to higher performance scores when compared to the other type. In the context of using the feature tuples $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$ and $(\tilde{v}, \tilde{a})$, resampling trajectories to a time-based representation yields greater F_1 score means than resampling to a distance-based representation. This applies independently of which hidden-state count is set. Parameter combinations determining the use of these feature tuples are depicted in subfigure (a) and (b) of Figure 6.2. Concerning parameter combinations that determine the use of the feature tuple $(\tilde{v}, \tilde{t}^{Int})$ to describe trajectories, resampling trajectories to a time-based representation does not yield higher performance scores in all cases. This is only the case for seven out of nine hidden-state counts. For hidden-state counts of five and six, parameter combinations determining resampling to a distance-based representation produce higher test F_1 score means. Their values are approx. 0.697 and 0.705, respectively. The two parameter combinations determining resampling to a time-based representation and the hidden state count to be set to five and six achieve test F_1 score means of only approx. 0.688 and 0.690, respectively. Parameter combinations determining the use of this feature tuple are depicted in subfigure (c).



(a) $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$

(b) $(\tilde{v}, \tilde{a})$

(c) $(\tilde{v}, \tilde{t}^{Int})$

(d) $(\tilde{v})$

Figure 6.2: Performance associated to parameter combinations determining the use of resampled trajectories. The performance of each parameter combination is described by averaging the set of F_1 scores determined via cross-validation. Each subplot refers to one feature tuple being employed to describe trajectories in associated experiment runs. Subplot (a) refers to the tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$, (b) to $(\tilde{v}, \tilde{a})$, (c) to $(\tilde{v}, \tilde{t}^{Int})$ and (d) to $(\tilde{v})$. Blue and orange curves denote test F_1 score means associated to parameter combinations determining the use of trajectories resampled to a time and a distance-based representation, respectively. The scores are plotted over hidden-state counts ranging between 2 and 10.

Concerning parameter combinations that determine the use of trajectories described by the feature tuple $(\tilde{v})$, the performance-score-based dominance of using trajectories resampled to a time-based representation is no longer observed. Parameter combinations determining the use of this trajectory feature are depicted in subfigure (d) of the figure. Only for four out of nine hidden-state counts, the resampling of trajectories to a time-based representation yields higher test F_1 score means. However, for the hidden state counts three, four, eight, nine and ten, parameter combinations determining resampling to a distance-based representation yield better inference performances. The associated test F_1 score means are approx. 0.647, 0.675, 0.665, 0.647 and 0.664, respectively. Resampling to a time-based representation is associated to test F_1 score means of only approx. 0.603, 0.631. 0.638, 0.603 and 0.451, respectively.

In summary, 36 comparisons between parameter combinations determining resampling to time and distance-based representations, respectively, are made above. Out of these, 29, hence roughly 81 %, reveal that the former trajectory representation axis type is associated to higher test F_1 score means. Only in seven comparisons, resampling to a distance-based representation yields higher F_1 score means. As outlined above, these seven comparisons are made in the context of the trajectory feature tuples $(\tilde{v}, \tilde{t}^{Int})$ and $(\tilde{v})$ being used to describe trajectories. Independently of trajectory representation axis type, no parameter combination determining the use of these two feature tuples yields a F_1 score mean greater than approx. 0.744. In contrast, in the context of a great number of parameter combinations determining the use of the feature tuples $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$ and $(\tilde{v}, \tilde{a})$, distinctly greater F_1 score means are observed. This can be seen when comparing the subfigures of Figure 6.2. Furthermore, as also already mentioned above, resampling to a time-based representation yields consistently greater F_1 score means than resampling to a distance-based representation in the context of these two feature tuples. Thus, for presented reasons, and in order to further reduce the dimensionality the search grid, the experiment-parameter-related focus is narrowed down once more. Only parameter combinations determining the use of trajectories resampled to a time-based representation and the use of the feature tuples $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$ and $(\tilde{v}, \tilde{a})$ are considered in further result analyses.

6.1.2.1.3. Parameter Combinations Yielding the Highest Score Results

Finally, among the considered parameter combinations, the three that are associated to the highest test F_1 score means shall be highlighted. Moreover, these score means also constitute the highest of all score means associated to parameter combinations of this experiment. An overview of these parameter combinations and associated test F_1 score means is provided in Table 6.2. The third and second highest test F_1 score mean are associated to two parameter combinations determining the use of the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$ and hidden-state count to be set to four and eight, respectively. The highest test F_1 score mean is associated to the parameter combination determining the use of trajectories described by the feature tuple $(\tilde{v}, \tilde{a})$ and hidden-state count to be set to ten. Standard deviations of scores associated to these parameter combinations can also be obtained from stated table.

In order to further analyze the classification result achieved when conducting inference using a set of HMMs, confusion matrices are used. Each parameter combination is cross-validated in four separate experiment runs. Consequently, four confusion matrices are assembled for the purpose of describing the classification results achieved by the four sets of parametrized HMMs. As performance scores are described by means and standard deviations, also confusion matrices shall be aggregated. Put differently, all classification results associated to a certain parameter

combination shall be described by one instead of four matrices. This is done by forming the elementwise sum of the four individual confusion matrices. The resulting matrix is referred to as a joint confusion matrix. Normalization is done as outlined in the beginning of section 6.

Rank	Test F_1 score mean	Test F_1 score standard deviation	Resampling	Representation	Feature selection	Hidden-state count
1st	0.845	0.052	With	Time-based	$(\tilde{v}, \tilde{a})$	10
2nd	0.841	0.071	With	Time-based	$(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$	8
3rd	0.832	0.057	With	Time-based	$(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$	4

Table 6.2: The three highest test F_1 score means, achieved in the grid search experiment, arranged in descending order. Also presented are associated score standard deviations. The columns Resampling, Representation, Feature selection and Hidden-state count together specify the parameter combinations associated to these test scores. Mean and standard deviation are calculated from individual test F_1 scores achieved during the cross-validation of each parameter combination.

The normalized joint confusion matrix describing the classification result of the parameter combination associated to the third-highest test F_1 score mean is provided on the left side of Figure 6.3. The one depicted on the right side of this figure describes the classification result of the parameter combination associated to the second-highest test F_1 score mean. As apparent from the associated normalized joint confusion matrix, approx. 100 % of both the `traffic-light` and `stop`-regulated test intersection paths are correctly classified by four-hidden-state HMMs. Approx. 89 % of `priority`-regulated intersection paths are classified correctly, whereas remaining ones are falsely classified as `stop`, `yield` or `yield-to-right`-regulated ones. Correctly classified `yield` and `yield-to-right`-regulated test intersection paths amount to approx. 60 % and 74 %, respectively. Apart from stated two classes being confused with each other, `yield`-regulated intersection paths are also falsely classified as `traffic-light`-regulated ones. Eight-

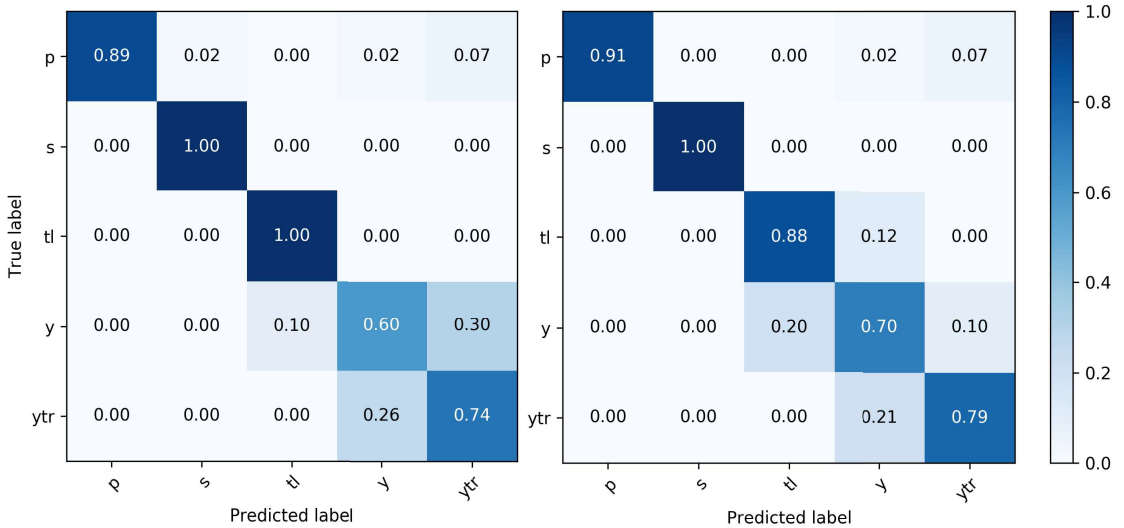


Figure 6.3: Normalized joint confusion matrix describing the classification result associated to two different parameter combinations. One determines the use of trajectories resampled to a time-based representation and the trajectory feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$ for the parametrization of four-state Hidden Markov Models (left). The other one determines the use of the same trajectory representation axis type and feature tuple but aims at parametrizing eight-state Hidden Markov Models (right). A joint confusion matrix is computed by forming the elementwise sum of the individual confusion matrices describing the classification results of experiment runs cross-validating a certain parameter combination. Traffic regulation classes are abbreviated and appear in the order `priority`, `stop`, `traffic light`, `yield` and `yield to right` on rows and columns.

hidden-state models perform slightly better with regard to `priority`, `yield` and `yield-to-right`-regulated test intersection paths. Performance on `traffic-light`-regulated test intersection paths is slightly inferior, whereas both four and eight-hidden-state models classify approx. 100 % of `stop`-regulated ones correctly.

The normalized joint confusion matrix describing classification results related to the parameter combination associated to the highest of all test F_1 score means and the one depicted on the right side of Figure 6.3 share several similarities. However, it is refrained from depicting it here. The reason for this is as follows: Comparing the resulting sets of HMMs shall not solely be based on test performance. Additionally, details of observation-density-related parameters and transition probabilities of HMMs shall be analyzed. This is done in section 6.1.2.2 below. In that section, reasons why the parameter combinations associated to the two highest test F_1 score means are deemed not suitable for efficiently and generically representing traffic regulations are shown. Also, it is explained why the parameter combination associated to the third-highest test F_1 score mean is regarded as the one producing the best-suited HMMs for traffic regulation inference.

Additionally, more information on inference performance of HMM sets parametrized as determined by the two parameter combinations ranked third and second in terms of test F_1 score mean shall be provided. In particular, inference performance on intersection paths of test and development data sets shall be contrasted. Therefore, Figure 6.4 visualizes F_1 score means associated to using trajectories that are resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. The score means are plotted against hidden-state counts ranging between two and ten. The blue curve denotes test F_1 score means, whereas the orange curve denotes training F_1 score means. The latter lies strictly above the former. The pale blue and pale orange band around the respective curves denote the standard deviation of the individual F_1 scores calculated from cross-validating experiment runs. More precisely, the vertical distance between an F_1 score mean point and the local upper or lower bound of the associated band is one standard deviation. Said two parameter combinations are, as already presented above, represented by F_1

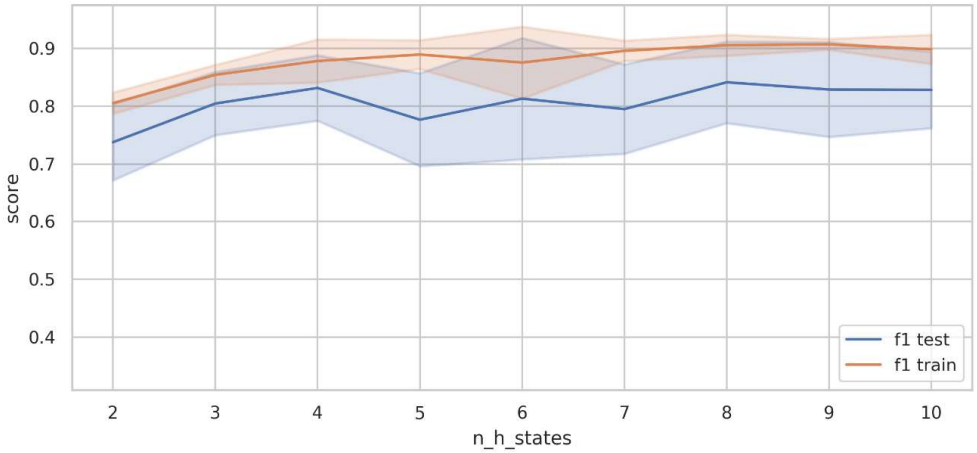


Figure 6.4: Performance associated to specific parameter combinations determining the use of trajectories that are resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. The performance of each parameter combination is described by averaging the set of F_1 scores determined via cross-validation. The blue and the orange curve denote F_1 score means achieved when conducting inference on the test and the training data portions, respectively, used during cross-validation. Analogously colored bands around the curves denote corresponding score standard deviations. The scores are plotted over hidden-state counts ranging between 2 and 10.

score points plotted at hidden-state counts four and eight, respectively. Interestingly, at four hidden states, the test F_1 score mean curve exhibits a local maximum. In contrast, the curve depicting training F_1 score means increases rather steadily up to a hidden-state count of nine.

6.1.2.2. Analysis of Parametrized Hidden Markov Models

In this section, observation-density-related parameters and transition probabilities of parametrized HMMs are analyzed. In subsection 6.1.2.2.1, the analysis of the four-hidden-state HMMs parametrized as determined by the parameter combination ranked third in terms of test F_1 score mean is shown in detail. Furthermore, in order to facilitate the explainability of observation probability densities, a visualization approach is designed for this thesis. With the aid of this approach, multivariate densities can be visualized in order to achieve a better understanding of what kind of observation signals are modeled by these. Besides, it is used throughout experiments in order to visually compare observation probability densities of HMMs. In the context of an example, it is introduced in stated subsection. This visualization approach is inspired by [47] and [48]. Among other goals, the authors of these works also aim at visualizing observation probability densities in the context of biological sequence analysis and speech processing, respectively.

Subsequently, comparisons to other sets of HMMs shall be made. As mentioned at the beginning of section 6, multiple sets of HMMs are parametrized based on one parameter combination in the context of cross-validation. In the context of the above-mentioned four-hidden-state HMMs, different model sets are briefly contrasted in subsection 6.1.2.2.2. In subsection 6.1.2.2.3, also HMMs parametrized as determined by the parameter combinations ranked second and first in terms of test F_1 score means are considered.

6.1.2.2.1. Four-State Hidden Markov Models

In this section, HMMs resulting from examining the parameter combination ranked third in terms of test F_1 score means are considered. The parameter combination can be obtained from the third row of Table 6.2.

Firstly, the set of four-hidden-state HMMs that are parametrized on the analysis development data portion is analyzed regarding associated observation densities. As mentioned above, an approach to visualize the observation probability densities of parametrized HMMs is designed. Gaussians employed as observation densities shall be visualized using combinations of one and two-dimensional figures. As illustration, Figure 6.5 visualizes the observation densities in each of the hidden states of the HMM representing the `priority` regulation. The order of the columns of the plots corresponds to the hidden-state numbering. Hence, the observation density in the first hidden state is visualized in plots of the first column, the second in the second column etc. The dark red points and vertical lines denote the parametrized mean vector components of the Gaussians. In subplots of the upper row, the mean vector components, associated to motion-related features, are visualized. How trajectory features and observation density parameters can be brought into context is explained in section 5.2.1. The component value associated to the $\tilde{v}$ trajectory feature is always provided on the horizontal axis, whereas the vertical axis provides the component value associated to the $\tilde{a}$ feature. One-dimensional subplots of the bottom row visualize the mean component associated to a progress-related trajectory feature. If none of these are included in in the trajectory feature selection this row would be omitted. For the observation densities depicted in Figure 6.5, each bottom row subplot visualizes the mean vector component associated to the

$\tilde{t}^{Int}$ feature. The pale red regions around the dark red points or vertical lines represent the two-standard-deviation regions around the means. The size of this region is chosen, based on the 68–95–99.7 rule [49]. According to this rule, around 95 % of multi-dimensional observations, drawn from such an observation density, lie in the pale red region. The covariances of Gaussian densities are not reflected by the visualization.

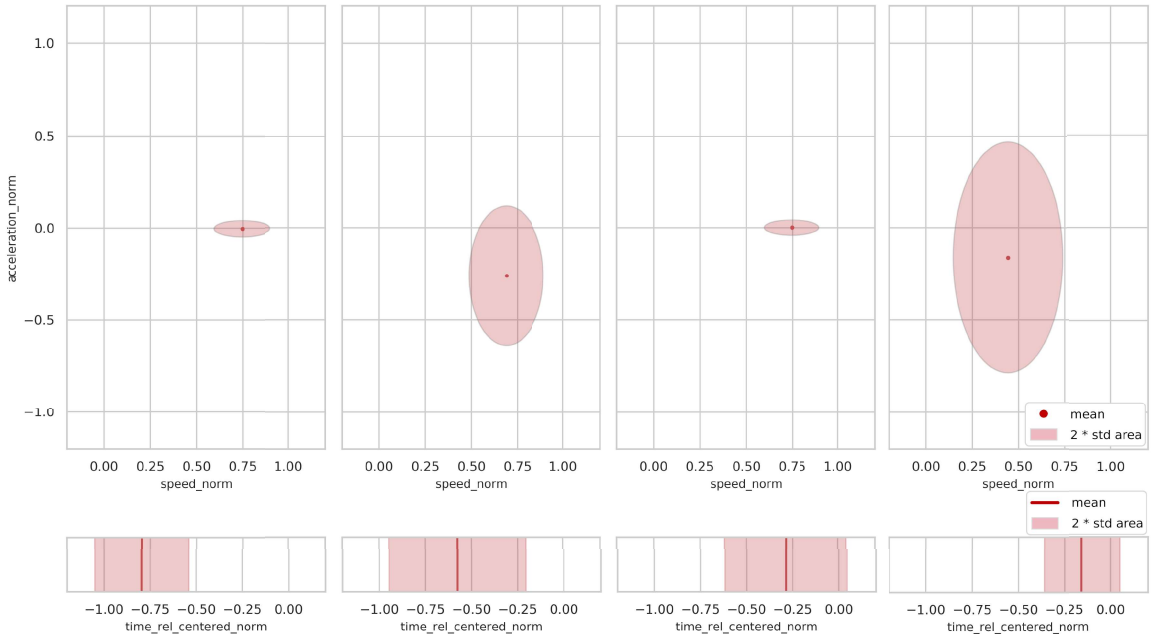


Figure 6.5: Visualization of learned observation densities of a four-hidden-state Hidden Markov Model representing the `priority` class. Each column of plots visualizes a Gaussian in one hidden state. Dark red points and lines denote mean component values and pale red regions denote two-standard-deviation regions around the means. In plots of the upper row, speed and acceleration-feature-related mean component values are always provided on the horizontal and vertical axes, respectively. The horizontal axes of plots of the bottom row provide the mean component values associated to the intersection-relative time feature. The column order reflects the hidden-state numbering of the model. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$.

The observation density in the first hidden state is depicted in the first column of subplots in Figure 6.5. Its mean component value associated to the $\tilde{v}$ feature is approx. 0.747. The learned standard deviation is approx. 0.077. Consequently, the two-standard-deviation range for this feature starts at around 0.593 and ends at around 0.901. The mean component value associated to the $\tilde{a}$ feature is approx. - 0.002 and the standard deviation is approx. 0.022. Hence, the majority of drawn observations have $\tilde{a}$ values rather close to 0. Thus, the state frequently emits observations interpretable as driving at a fairly constant speed. In other words, such a state is interpreted as a free-driving[4] state. The observation density in the third hidden state of this HMM has very similar parameter values. It is depicted in the third column of subplots. Both clearly differ in terms of the $\tilde{t}^{Int}$ feature. The first hidden state's observation density has a mean component value of around - 0.796 associated to the $\tilde{t}^{Int}$ feature. The corresponding standard deviation is approx. 0.127. Thus, the density in the first hidden state models free-driving observations occurring rather early on the approach to the intersection center. In contrast, the density in the third one models the same kind of observations occurring relatively close to the

[4] In the context of this thesis, the term free-driving is used to describe driving at a relatively constant speed. It makes no statement about the presence or absence of other vehicles in front of the ego vehicle.

point in time when the intersection center is reached. The latter has a $\tilde{t}^{Int}$ mean component value of approx. - 0.286. The associated standard deviation is approx. 0.166 which is slightly greater than the one of the former. The second hidden state frequently emits observations that can be interpreted as braking from a rather high speed, with $\tilde{v}$ and $\tilde{a}$-related mean component values of approx. 0.691 and - 0.260, respectively. The standard deviations have values of around 0.102 and 0.190, respectively. With a $\tilde{t}^{Int}$-related mean component value of approx. - 0.576 and standard deviation of approx. 0.186, observations of this type commonly occur roughly at the end of the first half of the intersection approach's duration. The observation density in the fourth hidden state has a $\tilde{v}$-related mean component of around 0.443 and an $\tilde{a}$-related one of approx. - 0.162. The associated standard deviations have rather high values of approx. 0.148 and 0.312, respectively. Therefore, this state emits observations of soft and also hard braking and accelerating from medium speeds. With a $\tilde{t}^{Int}$-related mean component value of approx. - 0.155 and standard deviation of approx. 0.103, observations of this type frequently occur close to the point in time when the intersection center is reached. With a difference of approx. 0.131, the $\tilde{t}^{Int}$ mean component values of the observation densities in the third and fourth hidden state are comparatively close to each other. Furthermore, both associated $\tilde{t}^{Int}$ two-standard-deviation regions exhibit an overlap. This renders the range of the band associated to the fourth hidden state an almost proper subset of the other. An overview of all parameter values of the depicted observation densities can be obtained from Table A.2.3 in section A.2.2 of the Appendix.

In this paragraph, the learned initial state and state transition probabilities of the same HMM are analyzed. A visualization of these is shown in Figure 6.6. The probabilities depicted on arrows illustrate the direction of transitions. Generally, all probabilities of transitions originating in a certain state have to sum up to 1. Due to rounding differences, this is not necessarily the case for the depicted values. Transitions with probabilities rounded to 0.000 are omitted in the figure. The only initial state probabilities greater than 0.000 have values of approx. 0.900 and 0.100 and are assigned to the first and second hidden state, respectively. The two highest state transition probabilities with values of around 0.996 and 0.998 are assigned to the self-loops of state 3 and 4, respectively. The remaining outgoing transition probabilities from these two states are relatively low and do not exceed 0.003. The greatest inter-state transition probability has a value of approx. 0.031. This transition originates in the second state and targets the fourth. The transition connecting the first hidden state to the third has the second-greatest probability with a value of approx. 0.024. Besides, worth noting is that transitions between the first and the second state each occur with a probability of approx. 0.000. Hence, transitions between these two states are, in comparison to other inter-state transitions, improbable.

Learned observation densities and initial state and state transition probabilities of the models representing the `yield` and the `yield-to-right` class are depicted in Figure 6.7 and Figure 6.8, respectively. All parameter values of the observation densities depicted in subfigure (a) of both of these figures are provided in Table A.2.4 and Table A.2.5, respectively, in section A.2.2 of the Appendix. The same visualizations for HMMs representing the regulation classes `stop` and `traffic light` are shown in Figure A.2.1 and Figure A.2.2, respectively, in the same section of the Appendix. In contrast to the `priority` model, both the `yield` and `yield-to-right` model' first hidden state is the initial state with a probability of approx. 1.000. This can be seen in subfigure (b) of both figures. As visible in the first plot column of subfigure (a) of both associated figures, in both models, this state frequently emits observations of free-driving. Also, transitioning back into the first hidden state from any other hidden state has a probability of approx. 0.000 in

both models. In the context of the `yield` model, transitioning into the second hidden state occurs with rather low probabilities having rounded values of maximally 0.006. In contrast, transitions into the third and fourth hidden state from the first and the second, respectively, occur more likely.

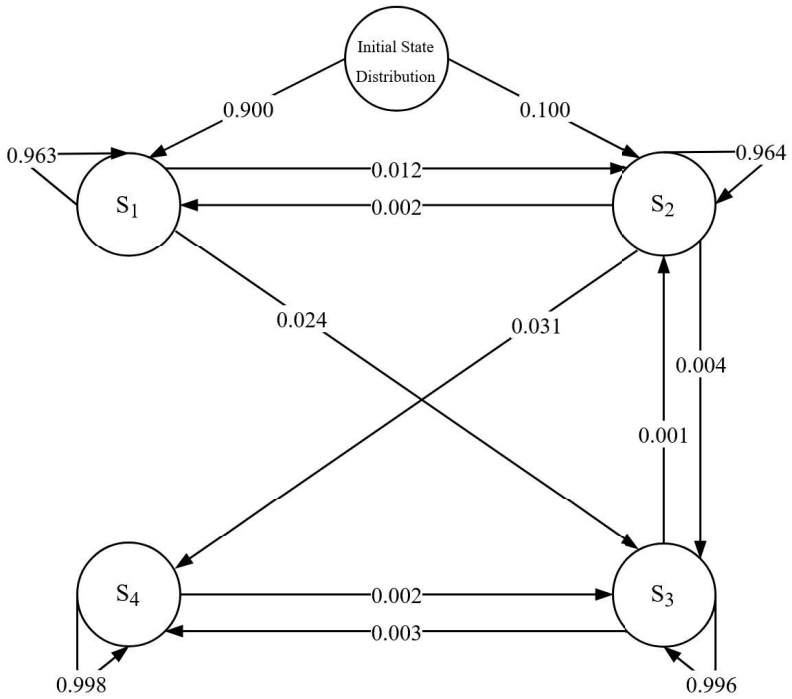
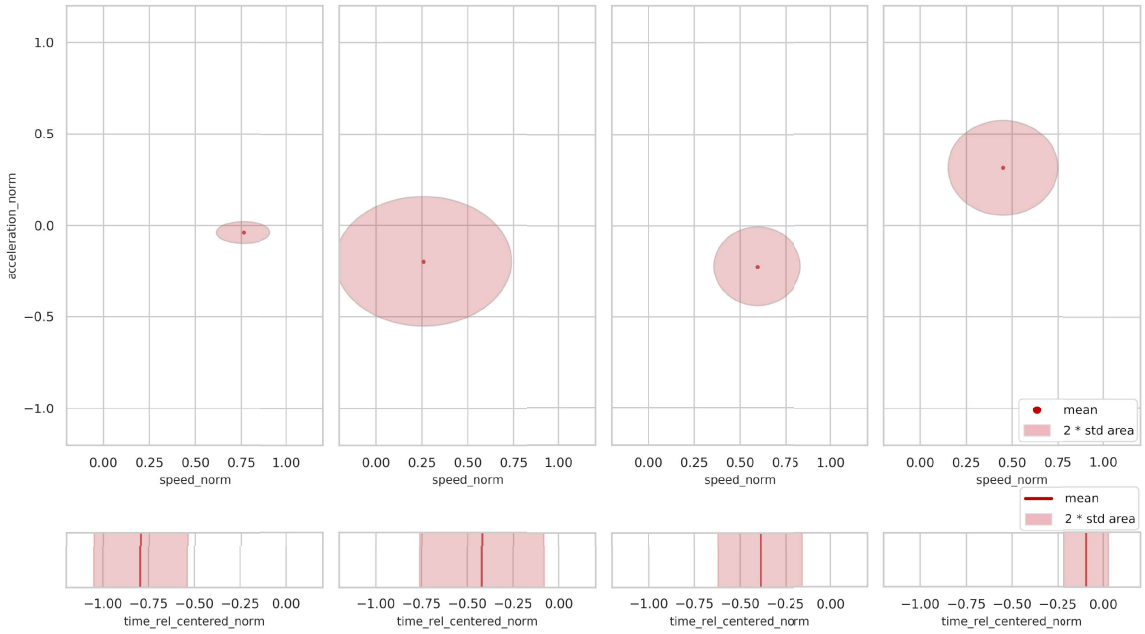


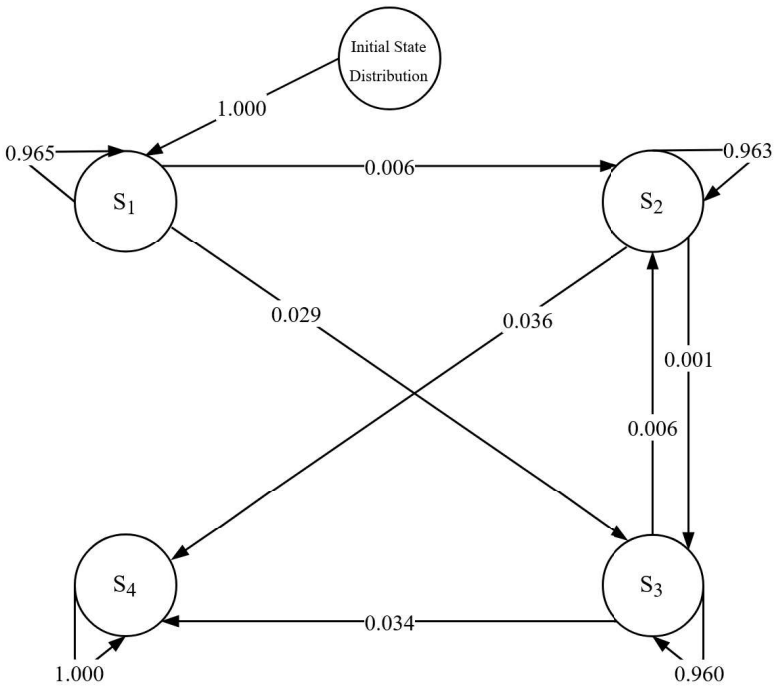
Figure 6.6: Visualization of the learned initial state and state transition probabilities of a four-state Hidden Markov Model representing the `priority` class. Values are rounded, hence all of a state's outgoing transition probabilities do not necessarily sum up to 1. The probabilities are printed on arrows illustrating the direction of transitions. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. The circles labeled S_i, with $1 \leq i \leq 4$, mark the model's hidden states.

Probability values of these transitions are as high as approx. 0.029 and 0.036, respectively. The third hidden state frequently emits observations of braking from higher or medium speeds. Observations emitted by the fourth hidden state frequently show accelerating from medium speeds. The `yield-to-right` model has states with observation densities being similar to those in the states of the `yield` model. In both models, the first and fourth hidden state frequently emit observations of driving freely at high speed and accelerating from medium speed, respectively. Observations of braking from medium or high speeds are frequently emitted by the third hidden state of the `yield` and by the second hidden state of the `yield-to-right` model. The densities in the `yield` model's second and the `yield-to-right` model's third hidden state are visualized in the second and third plot column of subfigure (a) of Figure 6.7 and Figure 6.8, respectively. These densities model observations of braking and accelerating from low or medium speeds by covering slightly similar two-standard-deviation regions regarding the $\tilde{v}$ and $\tilde{a}$ feature. As rather small positive $\tilde{a}$ values can be drawn from the density in second hidden state of the `yield` model, rather gentle accelerating can be observed. In contrast, the third hidden state of the `yield-to-right` model also emits observations of hard accelerating. Also, the point defined by both $\tilde{v}$ and $\tilde{a}$ having the value of approx. 0.000 is comprised in the two-standard-deviation region of both densities. Thus, occasionally even standstill events are emitted by the `yield` model's second and the `yield-to-right` model's third hidden state. What is also worth noting is depicted in subfigure

(b) of Figure 6.8. The three greatest inter-state transition probabilities of the `yield-to-right` model are found on a directed path passing through each of the four states. It starts in the first state and ends in the last one. These probabilities have rounded values greater or equal to 0.123.

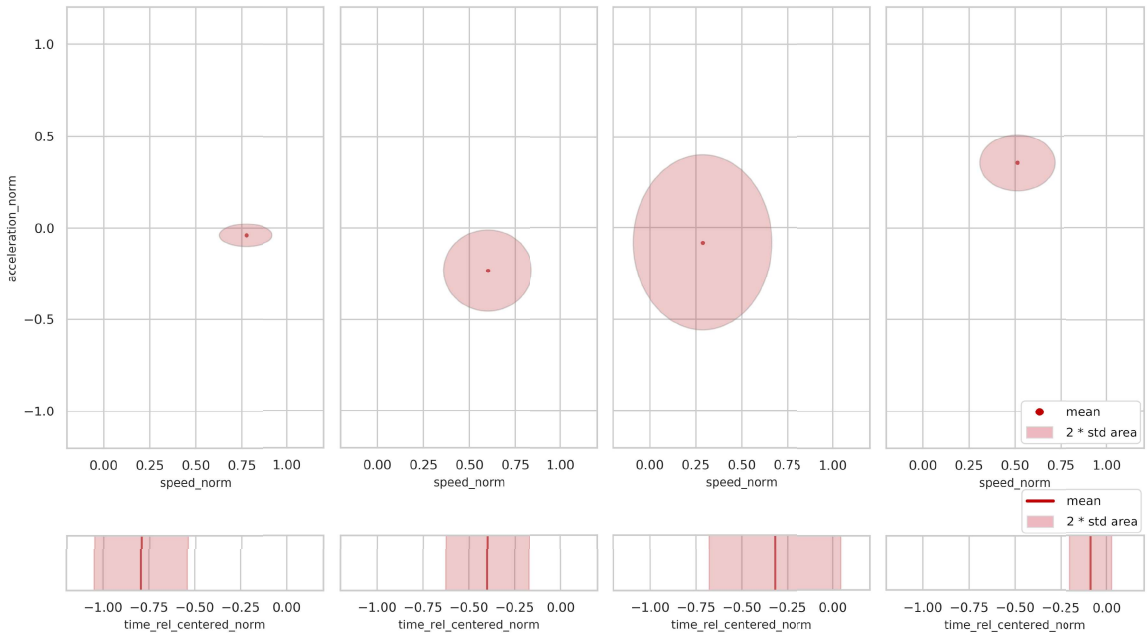


(a) Visualization of learned observation densities.

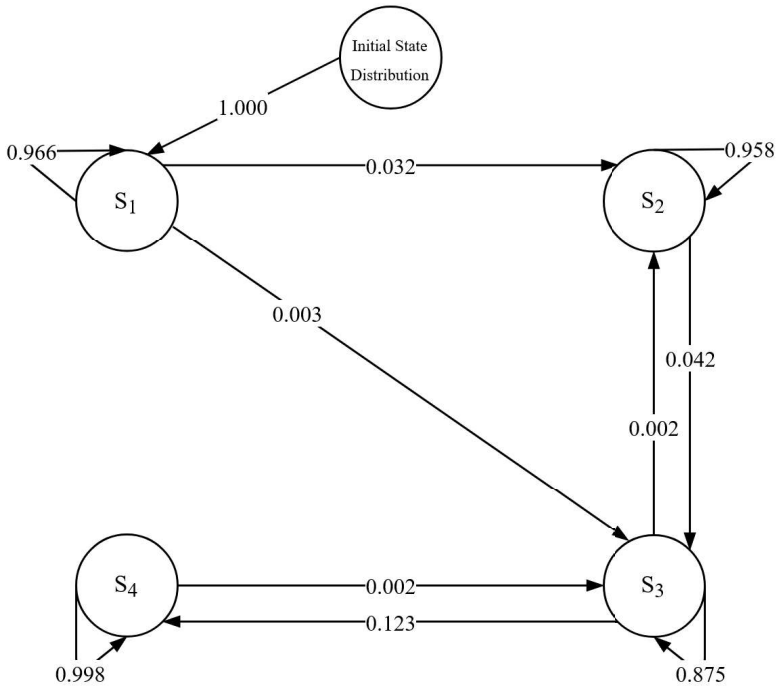


(b) Visualization of learned initial state and state transition probabilities.

Figure 6.7: Visualization of learned observation densities (a) and initial state and state transition probabilities (b) of a four-hidden-state Hidden Markov Model representing the `yield` class. Each column of plots in (a) visualizes the observation density in one hidden state. The column order reflects the hidden-state numbering of the model. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. Values in (b) are rounded, hence all of a state's outgoing transition probabilities do not necessarily sum up to 1. The values are printed on arrows illustrating the direction of transitions. The circles labeled S_i with $1 \leq i \leq 4$ mark the model's hidden states.



(a) Visualization of learned observation densities.



(b) Visualization of learned initial state and state transition probabilities.

Figure 6.8: Visualization of learned observation densities (a) and initial state and state transition probabilities (b) of a four-hidden-state Hidden Markov Model representing the `yield-to-right` class. Each column of plots in (a) visualizes the observation density in one hidden state. The column order reflects the hidden-state numbering of the model. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. Values in (b) are rounded, hence all of a state's outgoing transition probabilities do not necessarily sum up to 1. The values are printed on arrows illustrating the direction of transitions. The circles labeled S_i with $1 \leq i \leq 4$ mark the model's hidden states.

6.1.2.2.2. Comparing Sets of Hidden Markov Models Parametrized as Determined by a Common Parameter Combination on Different Development Datasets

Between HMMs parametrized in different cross-validation runs examining the parameter combination ranked third in terms of test F_1 score means, a comparison shall be made. For each traffic regulation class, associated models are briefly visually compared using the approach visualizing learned observation densities. In doing so, only mean component values and standard deviations associated to motion-related trajectory features are considered.

On the one hand, in terms of observation densities, the models representing the regulation classes `priority` and `stop` and are parametrized highly similar. The models representing the `traffic light` and `yield-to-right` class exhibit only very slight differences in observation density parameters. Thus, these differences shall not be analyzed in detail. On the other hand, the models representing the `yield` class are not parametrized as uniformly as it is the case for above-stated regulation classes. Three of the four cross-validation runs actually produce similar `yield` models. The one analyzed in subsection 6.1.2.2.1 is one of these. Its observation densities are visualized in Figure 6.7 (a). The observation densities of the model parametrized in the remaining cross-validation run exhibit deviations. The visualization of observation densities for this deviating model is shown in Figure 6.9. All parameter values of these densities are provided in Table A.2.8 in section A.2.2 of the Appendix. The differences between this model's observation densities and those of the above-analyzed `yield` model are pointed out in the following. The latter model shall therefore serve as representative of the three similarly parametrized `yield` models. The first hidden state of both compared `yield` models frequently emits free-driving observations. The state's observation densities are relatively similar and are depicted in the first column of Figure 6.7 (a) and Figure 6.9. The third hidden state of both models also clearly emit observations of braking. It is shown in both figure's third column. More precisely, the third hidden state of the representative model emits observations of braking from medium or high speeds. The corresponding hidden state of the deviating model, however, emits observations of braking more often from medium than from high speeds. The density's mean component value associated to the $\tilde{v}$ feature are approx. 0.595 and 0.493, respectively. The associated standard deviations have values that are similar among each other. Besides, the representative model's state frequently emits observations of both gentle braking, characterized by negative $\tilde{a}$ values close to 0.000, and relatively hard braking, characterized by negative values with higher magnitudes. In contrast, the state of the deviating model frequently emits such of relatively hard braking but mostly not such of gentle braking. The $\tilde{a}$-related mean component values of both model's densities are approx. - 0.223 and - 0.311, respectively. The second hidden state of the deviating model is depicted in the second column of Figure 6.9. Its two-standard-deviation region almost exclusively covers negative $\tilde{a}$ values with relatively small magnitudes. Therefore, the state frequently emits observations of rather gentle braking from higher speeds. In contrast, the second hidden state of the representative model frequently emits observations with $\tilde{a}$ values in a range representing everything between hard braking and gentle accelerating. This mostly occurs from low up to medium speeds. This observation density is visualized in the second column of Figure 6.7 (a). The fourth hidden state of both models, each visualized in the last column of the respective density visualization, frequently emit observations of accelerating. In the case of the representative model, the accelerating is mostly quantified by relatively high $\tilde{a}$ values and occurs from mostly medium speeds. The corresponding hidden state of the deviating model also mostly emits observations of accelerating. Additionally, when viewing the comparatively large two-standard-deviation region

of the deviating model's density in this hidden state, observations of accelerating from lower speeds and gentle braking from low up to medium speeds can also be emitted.

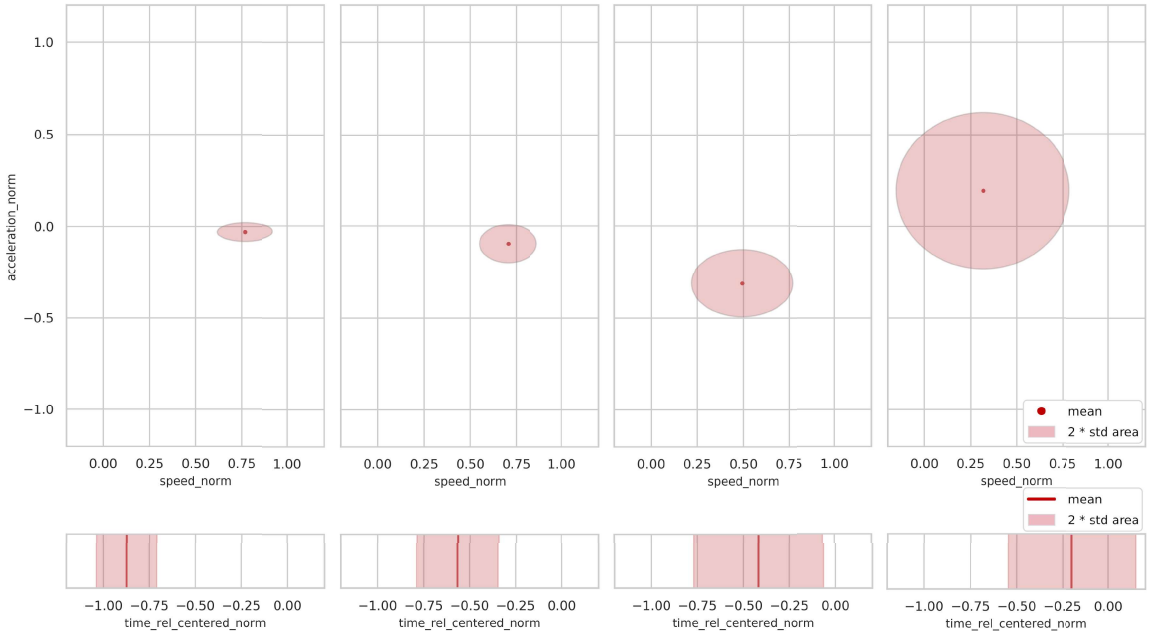


Figure 6.9: Visualization of learned observation densities of a four-hidden-state Hidden Markov Model representing the `yield` class. Each column of plots visualizes a Gaussian in one hidden state. The column order reflects the hidden-state numbering of the model. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. However, the trajectory data portion used for developing this model is a different one than the one used to develop models presented above.

6.1.2.2.3. Hidden Markov Models Parametrized Based on Other Parameter Combinations

In this section, the parameter combinations ranked first and second in terms of test F_1 score means, shall once again be considered. Also, it shall be explained why the HMMs produced by the parameter combination ranked third are regarded as being best suited for classifying traffic regulations. The three parameter combinations can be gathered from Table 6.2.

Parameter combinations ranked first and second in terms of test F_1 score means determine the parametrization of ten and eight-hidden-state models, respectively. HMMs that result from experiment runs using the analysis development data portion are also analyzed. These states of said models also frequently emit observations of, for instance, free-driving, braking, accelerating or standstill. However, in order to maintain a reasonable scope, details of this analysis are not shown. What heavily distinguishes these models from four-hidden-state models that are produced by the parameter combination ranked third in terms of test F_1 score means is the degree of redundancy regarding observation densities. Among four-hidden-state HMMs, only such representing the `priority` class each exhibit this kind of redundancy. More precisely, each of these models has two hidden states that are highly similar in terms of observation density parameters and that frequently emit free-driving observations. In contrast, the considered eight-hidden-state HMMs are found to have frequently more than two states with highly similar observation densities. As an example, five of eight observation densities associated to one of the models representing the `priority` class are visualized in Figure 6.10. These are visualizations of the observation densities in the first, second, fourth, sixth and eighth hidden state of the HMM. The set of all parameter values defining this model's observation densities are provided in Table

A.2.9 in section A.2.2 of the Appendix. One can directly see how mean component values and standard deviations, associated to the features $\tilde{v}$ and $\tilde{a}$, are highly similar across the five individual densities. The $\tilde{t}^{Int}$-feature-related mean component values associated to these densities are spread almost evenly over the feature's value range. The corresponding standard deviations have relatively low values. All of these hidden states frequently emit observations of free-driving.

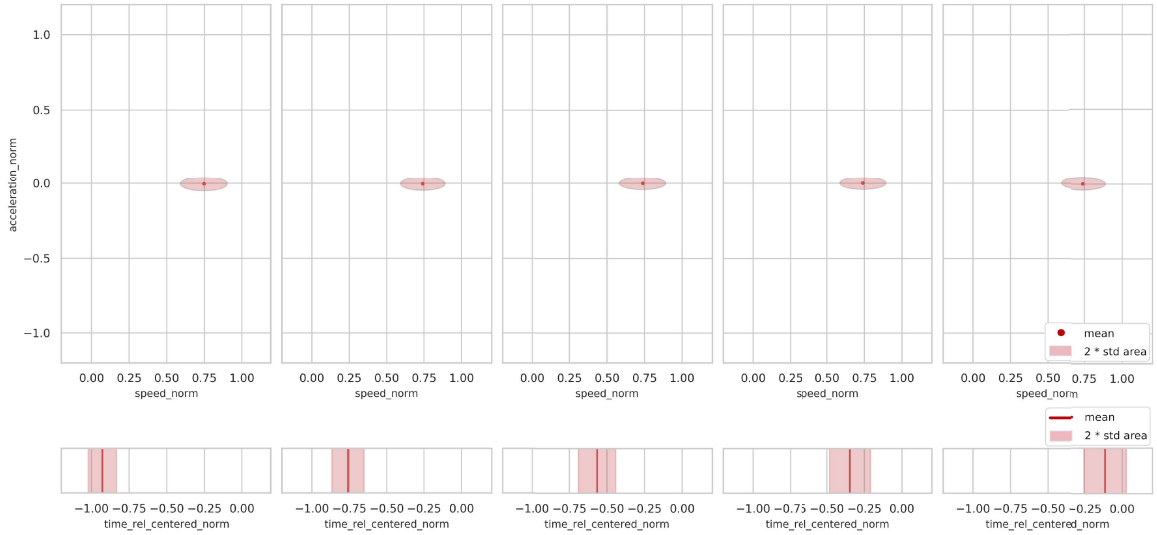


Figure 6.10: Visualization of learned observation densities in, from left to right, the first, second, fourth, sixth and eighth hidden state's observation density of an eight-state Hidden Markov Model representing the `priority` class. Each column of plots visualizes a Gaussian in one hidden state. The model is parametrized during one of the runs validating the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$.

In the case of ten-hidden-state HMMs, redundancies are found to occur even more severely. Additionally, in contrast to the four and eight-hidden-state models, these models are parametrized using trajectories that are only described by the feature tuple $(\tilde{v}, \tilde{a})$. Thus, hidden state densities cannot diversify in terms of mean component values associated to a progress-related feature. One might argue that this renders the observation densities in hidden states of such models even more similar. For reasons of brevity, it is refrained from showing associated visualizations.

In summary, the considered eight and ten-hidden-state HMMs exhibit far greater redundancies regarding observation densities than four-hidden-state ones. Also, it shall be pointed out that he difference between the highest and the third-highest test F_1 score mean is only approx. 0.013. Thus, in consideration of explained redundancies and negligible F_1 score differences, the parameter combination ranked third in terms of test F_1 score means is favored over competing parameter combinations. For the remainder of this thesis, HMMs parametrized as determined by stated parameter combination are referred to as the best-performing HMMs of this experiment.

6.2. Increasing Trajectory Data Volume

The result evaluation of the grid search experiment, shown in section 6.1.2.1, reveals that several misclassifications occur when conducting test inference while using the experiment's best-performing HMMs. Among other error cases, approx. 11 % of `priority`-regulated test intersection paths and approx. 40 % of `yield`-regulated ones are falsely classified by these HMMs. This information can be obtained from the normalized joint confusion matrix depicted on the left side of Figure 6.3. Approx. 7 % of `priority`-regulated test intersection paths are falsely classified as `yield-to-right`, approx. 2 % as `stop` and also roughly 2 % as `yield`-regulated ones. Of all `yield`-regulated test intersection paths, approx. 10 % are confused with `traffic-light`-regulated ones. Roughly 30 % are falsely classified as being `yield-to-right`-regulated.

In the experiment presented in this section, it shall be examined whether parametrizing HMMs on more development data reduces the number of stated misclassifications. For this reason, as mentioned in section 4.1, four intersection scenarios are additionally designed. Details of this experiment's design are outlined in subsection 6.2.1. Results are presented in 6.2.2.

6.2.1. Experiment Design

In order to counter above-stated misclassifications, all additionally designed intersection scenarios are such where `priority` and `yield`-regulated intersection paths meet. It is decided to increase the data volume specifically for these two classes in order to keep the data generation effort within reasonable limits. In the right column of Table 4.3, one can see the number of `priority` and `yield` intersection paths comprised in the augmented dataset. Using this dataset, HMMs representing said two classes are parametrized again. For this, the parameter combination yielding the best-performing HMMs of the grid search experiment is used. This parameter combination determines the use of trajectories that are resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$ for the parametrization of two new four-hidden-state HMMs. The remaining models representing the classes `stop`, `traffic light` and `yield to right` are taken from the best-performing ones of the grid search experiment without modifications. Thus, also in this experiment, each traffic regulation is represented by one HMM. The performance of this approach is cross-validated as explained in section 4.4. However, all intersection paths gathered from additionally designed intersection scenarios are added to each development data portion. None of these intersection paths are added to test data portions.

6.2.2. Experiment Results

The results of using an increased number of `priority` and `yield`-regulated intersection paths for developing HMMs is presented in this section. Achieved scores are presented in subsection 6.2.2.1 and resulting models in subsection 6.2.2.2.

6.2.2.1. Evaluation of the Classification Result

The classification result achieved in this experiment can be obtained from the normalized joint confusion matrix depicted in Figure 6.11. This result is compared to the one achieved by the best-performing HMMs of the grid search experiment, which is described by the normalized joint confusion matrix depicted on the left side of Figure 6.3.

On the one hand, misclassifications of `priority`-regulated test intersection paths slightly increase by approx. 6 percentage points. Approx. 9, 4 and 4 % of these intersection paths are confused with `stop`, `yield` and `yield-to-right`-regulated ones, respectively. Still, none are falsely classified as being `traffic-light`-regulated. Also, a fraction of approx. 29 %, of `yield-to-right`-regulated intersection paths are now falsely classified as being `yield`-regulated. This corresponds to an increase of roughly 3 percentage points. On the other hand, slightly less `yield`-regulated intersection paths are misclassified. Now, only approx. 5 % are confused with `traffic-light`-regulated intersection paths. The confusion with `yield-to-right`-regulated ones remains unchanged. Thus, the number of correctly classified `yield`-regulated intersection paths increased by approx. 5 percentage points. The models achieve a mean test F_1 score of approx. 0.817. It is slightly lower than the corresponding one achieved by the best-performing HMMs in the setting of the grid search experiment. The difference between the scores is approx. 0.015. However, a lower variation of the test F_1 scores is achieved. The associated test F_1 score standard deviation has a value of only approx. 0.035. The associated difference is approx. 0.022.

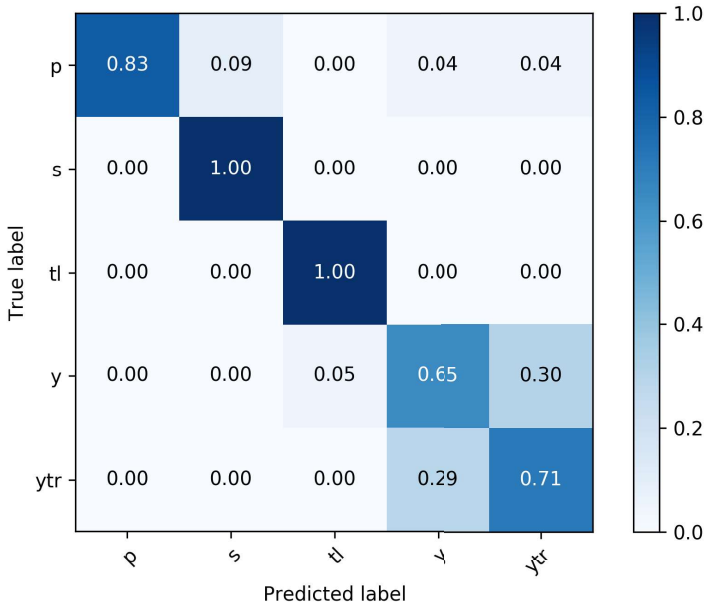
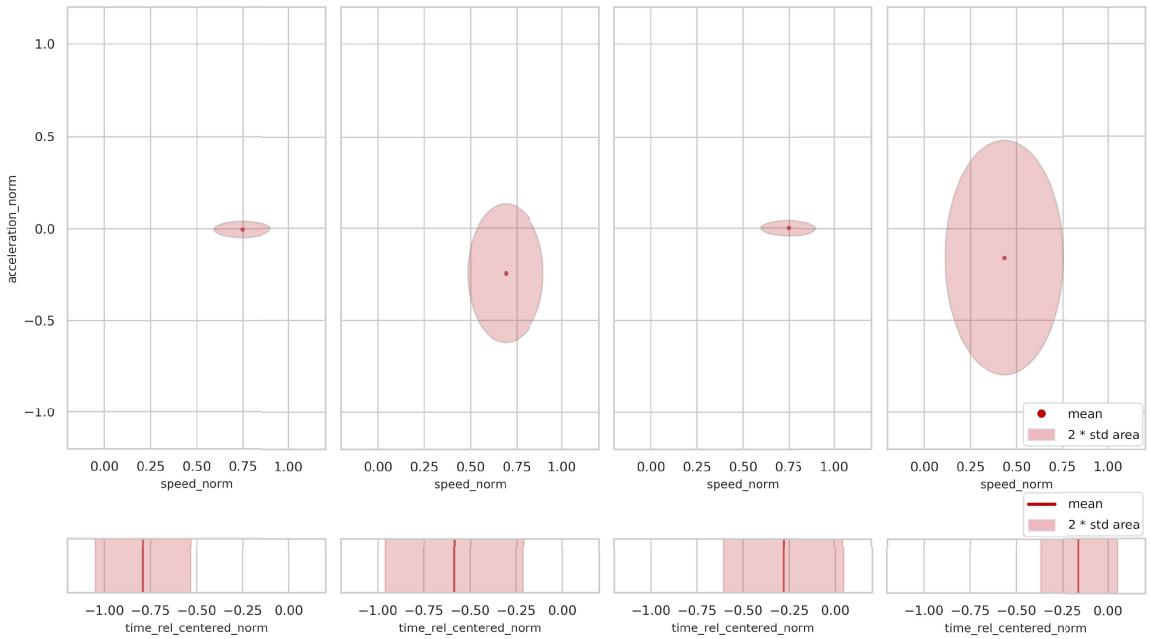


Figure 6.11: Normalized joint confusion matrix describing the classification result achieved in the experiment using an augmented dataset for Hidden Markov Model parametrization. Trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$ are used for the parametrization of four-state Hidden Markov Models. A joint confusion matrix is computed by forming the elementwise sum of the individual confusion matrices describing the classification results of experiment runs cross-validating a certain parameter combination. Traffic regulation classes are abbreviated and appear in the order `priority`, `stop`, `traffic light`, `yield` and `yield to right` on rows and columns.

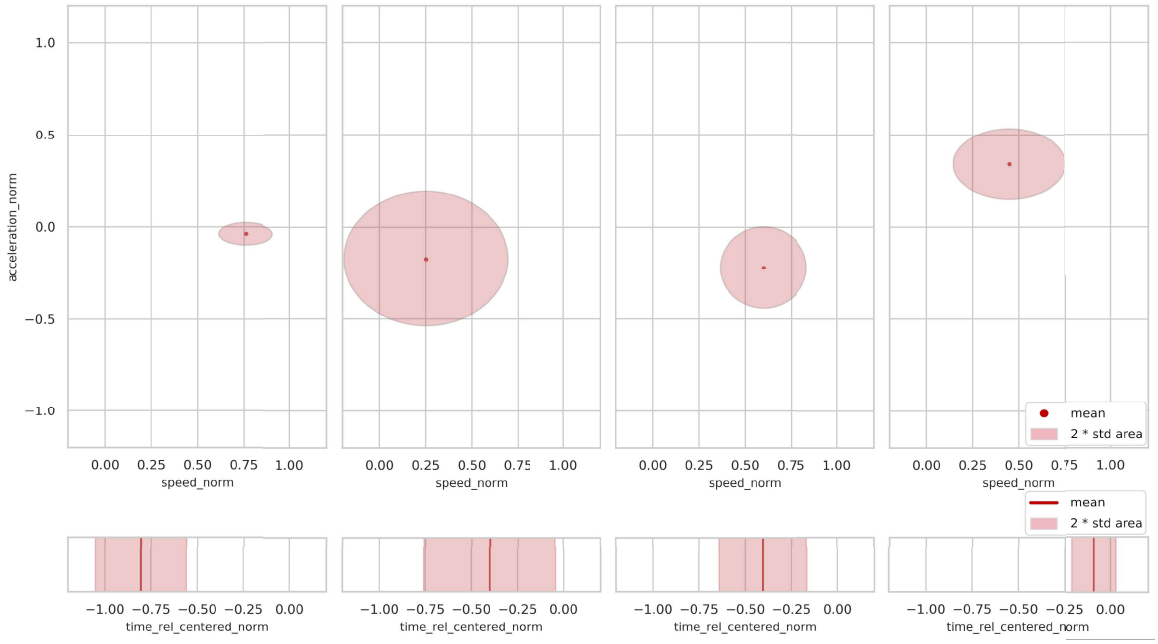
6.2.2.2. Analysis of Parametrized Hidden Markov Models

HMMs representing the classes `priority` and `yield` are reparametrized in this experiment. In the following, the observation densities associated to the models that are reparametrized on the analysis development data portion are shown.

Subfigure (a) of Figure 6.12 shows a visualization of the observation densities associated to the `priority` HMM. When comparing it to the visualization of the corresponding best-performing HMM of the grid search experiment, almost no differences can be found. The visualization of the



(a) Visualization of learned observation densities of a Hidden Markov Model representing the `priority` class.



(b) Visualization of learned observation densities of a Hidden Markov Model representing the `yield` class.

Figure 6.12: Visualization of learned observation densities of two four-hidden-state Hidden Markov Models. The densities in subfigure (a) are associated to a model representing the `priority` class. Subfigure (b) visualizes densities that are associated to a model representing the `yield` class. In both subfigures, each column of plots visualizes a Gaussian in one hidden state. The column order reflects the hidden-state numbering of the models. Both Hidden Markov Models are parametrized using trajectories that are resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$.

observation densities of the compared HMM can be found in Figure 6.5. Also, a strong similarity is found between the observation densities associated to the reparametrized HMM representing the `yield` class and those associated to the corresponding best-performing model of the grid search experiment. Associated visualizations are depicted in subfigure (b) of Figure 6.12 and in subfigure (a) of Figure 6.7, respectively. All parameter values of the depicted observation densities associated to the reparametrized `priority` and `yield` model are provided in Table A.3.1 and Table A.3.2, respectively, in section A.3 of the Appendix. Detailed descriptions of observation densities are omitted at this point. Those for the corresponding models in section 6.1.2.2.1 can be seen as almost equally suitable for the observation density visualizations associated to the HMMs parametrized in this experiment. Visualizations of transition probabilities associated to the two reparametrized HMMs are depicted in Figure A.3.1 and Figure A.3.2, respectively, of the Appendix.

As described in section 6.1.2.2.2 of the grid search experiment, one of the four best-performing `yield` HMMs is found to deviate in terms of learned observation densities. The densities of the deviating model are visualized in Figure 6.9. However, among the four reparametrized `yield` HMMs, none deviates strongly. For this, Figure 6.13 visualizes the learned observation densities of another reparametrized `yield` HMM. Table A.3.3 of the Appendix provides all parameter values of the observation densities visualized in this figure. Above-stated deviating `yield` HMM that is analyzed in the context of the grid search experiment is produced using a particular portion of the development data. The HMM of which densities are visualized in Figure 6.13 is reparametrized on basically the same but augmented data portion. Put differently, the former portion is a subset of the latter. By visually comparing observation densities visualized in subfigure (b) of Figure 6.12 and Figure 6.13, one can see that these are parametrized highly

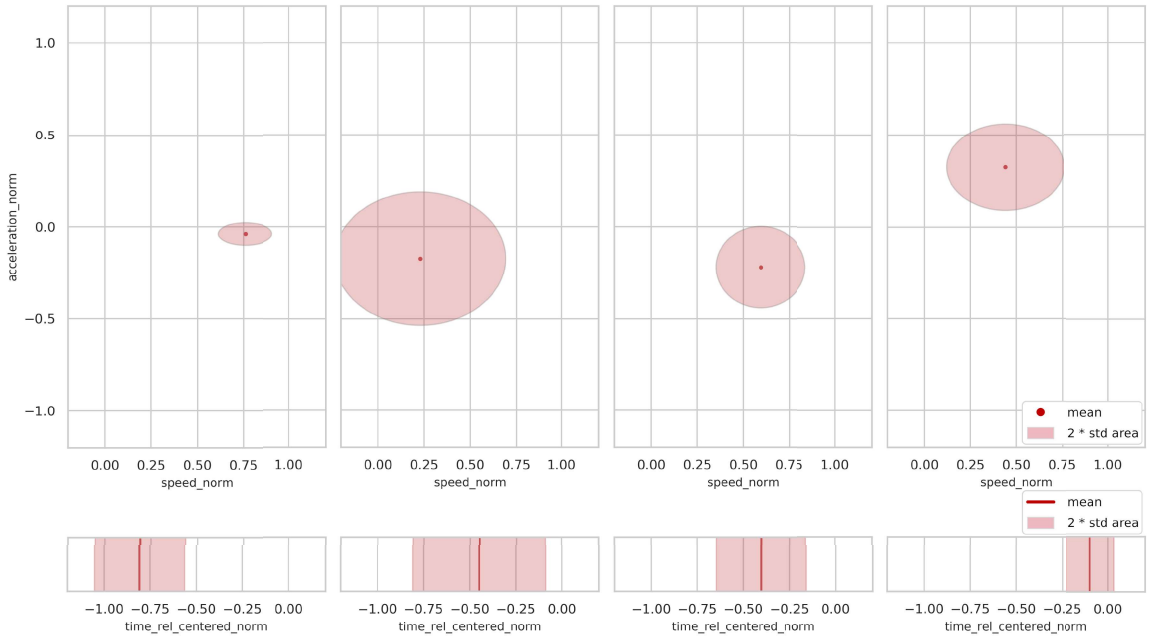


Figure 6.13: Visualization of learned observation densities of a four-hidden-state Hidden Markov Model representing the `yield` class. Each column of plots visualizes a Gaussian in one hidden state. The column order reflects the hidden-state numbering of the model. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. However, the trajectory data portion used for developing this model is a different one than the one used to develop models presented above.

similar. Observation densities of the two remaining `yield` HMMs are also highly similar to those depicted here. However, in order to keep the length of this section within reasonable limits, those are not visualized.

6.3. Counteracting False Classifications of Priority Intersection Paths

The experiment presented in section 6.2 aims at reducing the number of misclassifications of `priority` and `yield`-regulated intersection paths. The measure taken to achieve this is employing an augmented set of intersection paths and associated trajectory data regulated by stated two classes. While this experiment achieves a reduction of falsely classified `yield`-regulated intersection paths, no improvement regarding `prioritized` ones can be shown. Therefore, several experiments are conducted in order to identify possible solutions to this problem. Approaches of these experiments are based on a thorough problem analysis, which is outlined in subsection 6.3.1. Subsection 6.3.2 expands on the design of stated experiments. Results are presented in subsection 6.3.3.

6.3.1. Problem Analysis

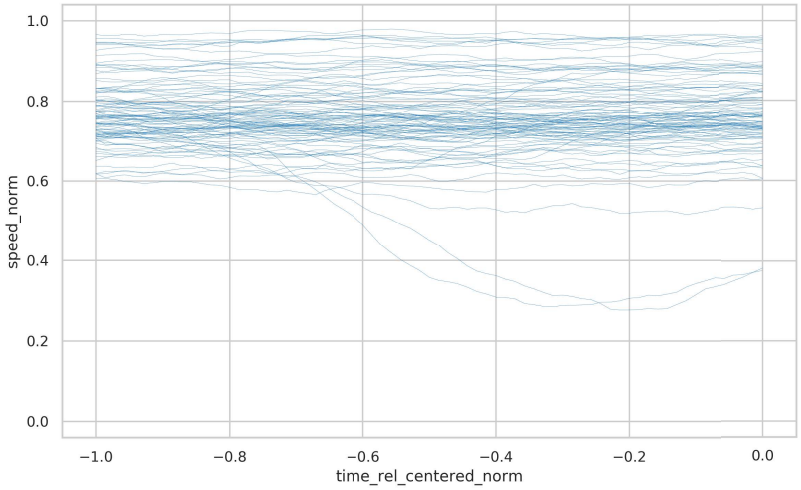
Under the `priority` class, diverse vehicle behavior is aggregated. Vehicles approaching intersections on `prioritized` intersection paths generally have the right of way over vehicles that approach on differently regulated ones. For instance, such approaching from `yield` or `stop`-regulated intersection paths have to give way. However, vehicles approaching from `prioritized` intersection paths that cross oncoming intersection paths which are also `prioritized`, might have to give way, too. Such situations commonly occur for left-turn maneuvers or for going straight at intersections for which the `prioritized` road is bent. Regarding the latter case, going straight only leads to the crossing of oncoming `prioritized` intersection paths, if following the `prioritized` road demands making a right turn. In contrast, vehicles turning left in order to stay on the bent `prioritized` road do so without crossing other `prioritized` intersection paths. For this reason, those `prioritized` intersection paths that involve giving way to other `prioritized` ones shall be termed as yielding `priority` intersection paths for the remainder of this thesis. Interestingly, it is found that intersection paths that allow passing through or describe right turns at intersections without having to give way are classified correctly in all cases. In previously shown experiments, exclusively yielding `priority` intersection paths are getting misclassified.

Before presenting a suggested solution approach, the most important findings of a thorough analysis of described misclassification problem are outlined in the following. Firstly, trajectories following intersection paths regulated by various traffic regulations are analyzed in subsection 6.3.1.1. Also, the viewpoint of inferencing HMMs representing the `priority` class is considered. Subsection 6.3.1.2 expands on this. Among other means, plots visualizing state sequences that are most probable given trajectories are also analyzed. However, these are found to not provide helpful insights and, therefore, are omitted in the following.

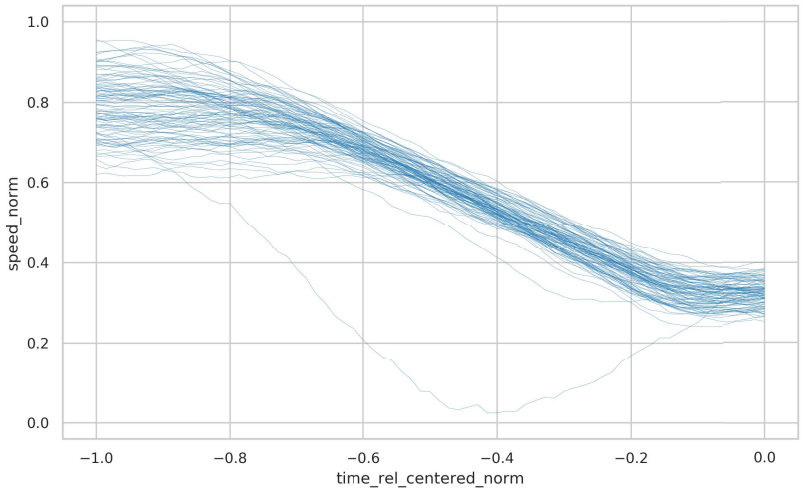
6.3.1.1. Trajectory-Data-Based Analysis

The trajectory $\tilde{v}$ data, depicted as blue curves in Figure 6.14 and Figure 6.15, is associated to different `priority`-regulated intersection paths. Instances of those intersection paths that are followed by vehicles not having to yield to other vehicles are depicted in Figure 6.14. In subfigure (a) of Figure 6.14, $\tilde{v}$ data of 100 trajectories following an intersection path passing straight

through an intersection is shown. The data is plotted over the $\tilde{t}^{Int}$ feature. In subfigure (b), $\tilde{v}$ data of another 100 trajectories following an intersection path describing a right turn is shown. Concerning the former, the great majority of trajectories exhibit no decreases in $\tilde{v}$. Almost all values are spread over the range between approx. 0.600 and 1.000. However, concerning the latter, almost all trajectories initially also show roughly constant $\tilde{v}$ values that are spread over the same range and, subsequently, exhibit almost linear decreases. Decreases begin at $\tilde{t}^{Int}$ values of around - 0.800. Subsequently, associated vehicles continue to drive at almost constant $\tilde{v}$ values primarily spread over the range between 0.250 and 0.400 until reaching the point closest to the intersection center.



(a) Trajectories of vehicles passing straight through an intersection.

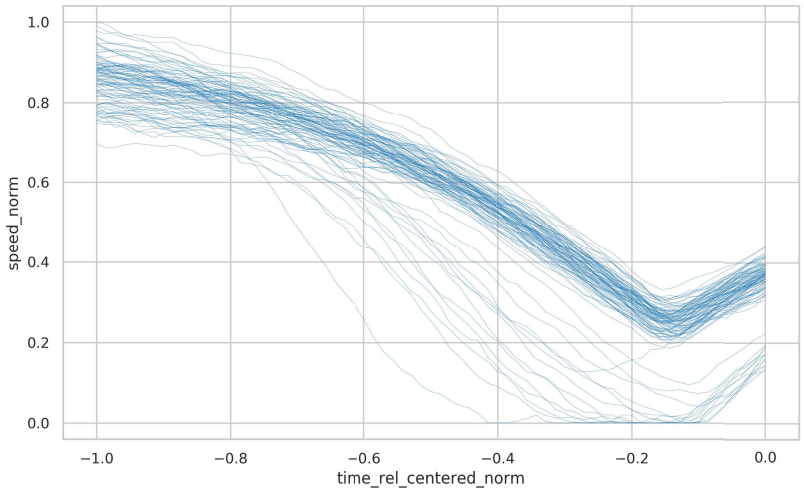


(b) Trajectories of vehicles turning right at an intersection.

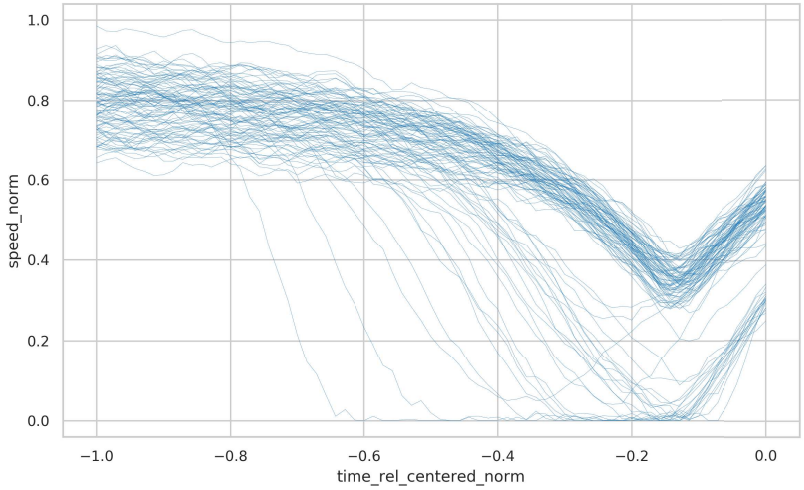
Figure 6.14: Normalized speed data of randomly sampled trajectories associated to two different `priority`-regulated intersection paths, plotted over the intersection-relative time feature. Each blue curve denotes data of one trajectory. In subfigure (a), 100 trajectories associated to an intersection path, followed by vehicles passing straight through the intersection without having to yield to other vehicles, are shown. Subfigure (b) shows 100 trajectories associated to an intersection path, followed by vehicles executing a right turn.

In Figure 6.15, yielding-`priority`-regulated intersection paths and associated trajectory $\tilde{v}$ data is shown. Hence, in order to follow these intersection paths, crossing of other `prioritized` ones is necessary. As mentioned above, several yielding `priority` intersection paths are misclassified. In the context of this thesis, a speed-level-based distinction of these intersection paths is made.

Trajectories depicted in the two subfigures of Figure 6.15 differ in terms of initial v value levels corresponding vehicles drive at when approaching intersections. Firstly, there are intersection paths that are followed by vehicles that, when driving freely, drive at comparatively high speeds on associated intersection-incoming segments. No other speed limits than 50 $\frac{\text{km}}{\text{h}}$ ($\approx$ 13.89 $\frac{\text{m}}{\text{s}}$) and 30 $\frac{\text{km}}{\text{h}}$ ($\approx$ 8.33 $\frac{\text{m}}{\text{s}}$) are represented in intersection scenarios considered in this thesis. Stated vehicles mostly comprise those that adapt their driving speed to the 50 $\frac{\text{km}}{\text{h}}$ limit. For reasons of brevity, described intersection paths are referred to as yielding `priority` intersection paths of type I for the remainder of this thesis. Secondly, other intersection paths are followed by vehicles free-driving at comparatively low speeds on associated intersection-incoming segments. These vehicles commonly adapt their driving speed to the 30 $\frac{\text{km}}{\text{h}}$ limit and frequently drive at speeds close to this limit during intersection approaches. For the remainder of this thesis, associated intersection paths are referred to as yielding `priority` intersection paths of type II.



(a) Trajectories of vehicles approaching at higher speeds.



(b) Trajectories of vehicles approaching at lower speeds.

Figure 6.15: Normalized speed data of randomly sampled trajectories associated to two different `priority`-regulated intersection paths describing left turns, plotted over the intersection-relative time feature. Each blue curve denotes data of one trajectory. In subfigure (a), 100 trajectories associated to an intersection path, followed by vehicles approaching an intersection at comparatively high speeds are shown. Subfigure (b) shows 100 trajectories associated to an intersection path followed by vehicles approaching at comparatively low speeds. All those vehicles potentially have to give way to vehicles following other `prioritized` intersection paths.

In subfigure (a) of Figure 6.15, $\tilde{v}$ data of trajectories assigned to yielding `priority` intersection paths of type I are shown. Subfigure (b) depicts $\tilde{v}$ data of trajectories assigned to yielding `priority` intersection paths of type II. In both cases, $\tilde{v}$ values are initially spread over the range between 0.600 and 1.000 as is the case for trajectories in Figure 6.14. Most of the trajectories depicted in both subfigures of Figure 6.15 show braking behavior already from the left side of the plot, hence at $\tilde{t}^{Int}$ values around - 1.000. In subfigure (a), values decrease strongly from the beginning. In subfigure (b), however, values decrease relatively slowly in the beginning. From $\tilde{t}^{Int}$ values around - 0.6 onwards, $\tilde{v}$ values decrease progressively. Most trajectories of both figures have their $\tilde{v}$ minima at $\tilde{t}^{Int}$ feature values around - 0.150. The values of these minima are spread over the range between 0.200 and 0.350 and between 0.300 and 0.450 in subfigure (a) and (b), respectively. This is followed by accelerating until the point closest to the intersection center, hence at $\tilde{t}^{Int}$ values around 0.000, is reached. Apart from that, a fraction of trajectories depicted in both subfigures show braking behavior earlier than other trajectories. These have $\tilde{v}$ minima around 0.000. In several cases, episodes of waiting in standstill can be interpreted from trajectory points exhibiting $\tilde{v}$ values of approx. 0.000 for extended ranges along the $\tilde{t}^{Int}$ feature. This is followed by accelerating behavior until the point closest to the intersection center is reached.

The difference between trajectories depicted in both subfigures of Figure 6.15 and those depicted in Figure 6.14 (a) is obvious. Almost no trajectories depicted in the latter exhibit speed reductions, whereas such are clearly visible in both subfigures of the former. Trajectories depicted in Figure 6.14 (b) are similar to such depicted in both subfigures of Figure 6.15 solely regarding decreases in $\tilde{v}$ values. However, no minima followed by increases of $\tilde{v}$ values can be found in the former. This way, a clear distinction can be made between trajectories assigned to `priority`-regulated intersection paths followed by vehicles not having to give way to others and yielding `priority` ones.

In contrast, more obvious similarities can be determined when comparing trajectories assigned to yielding `priority` intersection paths to such that are assigned to `yield`, `yield-to-right` and `stop`-regulated intersection paths. This might be one of the reasons why many of the former intersection paths are often confused with such of the three latter ones.

Figure 6.16 and Figure 6.17 show $\tilde{v}$ data of trajectories assigned to a `yield` and a `yield-to-right`-regulated intersection path, respectively. Analogously to previous figures, data is plotted over the $\tilde{t}^{Int}$ feature also in these two figures. Trajectories depicted in both figures exhibit $\tilde{v}$ curve shapes that are highly similar among themselves. Initial $\tilde{v}$ values are spread over a common range between approx. 0.600 and 1.000. After that, values decrease progressively. These characteristics can also be found for trajectories assigned to both types of yielding `priority` intersection paths. However, trajectories assigned to `yield` and `yield-to-right`-regulated intersection paths exhibit $\tilde{v}$ curve minima comparatively early at $\tilde{t}^{Int}$ values of around - 0.200. These minima take on $\tilde{v}$ values between approx. 0.300 and 0.450 which is more similar to trajectories assigned to yielding `priority` intersection paths of type II than to trajectories assigned to such of type I. Some trajectories assigned to `yield` and `yield-to-right`-regulated intersection paths not only show standstills but also episodes of stop-and-go driving. The latter manifest in slight increases of $\tilde{v}$ values followed by decreases, which are occasionally repeated several times. As observed for trajectories assigned to both types of yielding `priority` intersection paths, steep $\tilde{v}$ value increases follow after minima or standstills episodes until the point closest to the intersection center is reached.

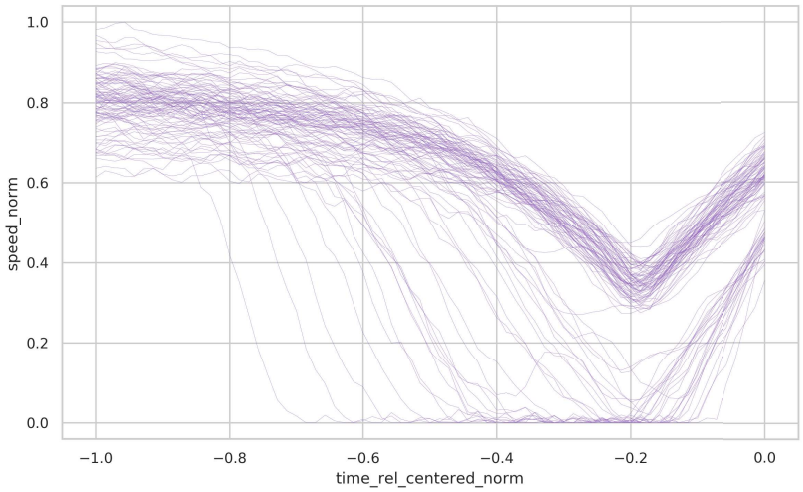


Figure 6.16: Normalized speed data of 100 randomly sampled trajectories associated to a `yield`-regulated intersection path, plotted over the intersection-relative time feature. Each purple curve denotes data of one trajectory. Vehicles following this intersection path execute a left turn.

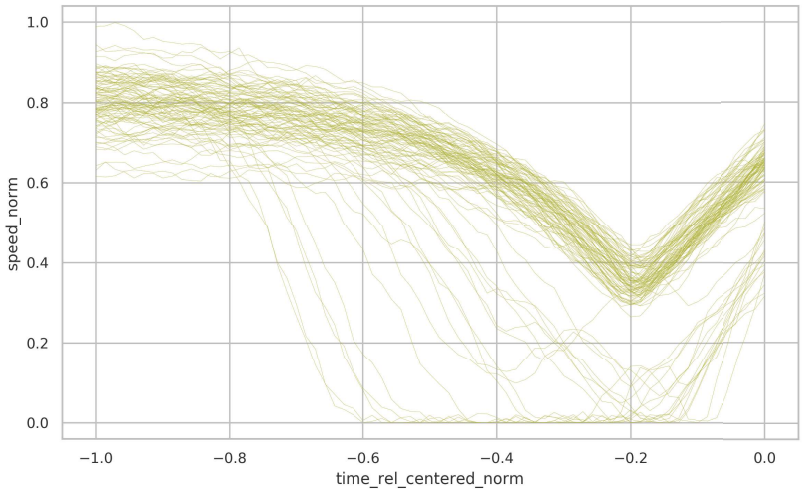


Figure 6.17: Normalized speed data of 100 randomly sampled trajectories associated to a `yield-to-right`-regulated intersection path, plotted over the intersection-relative time feature. Each dark yellow curve denotes data of one trajectory. Vehicles following this intersection path execute a left turn.

Apart from that, also trajectories assigned to a `stop`-regulated intersection path share similarities with trajectories assigned to yielding-`priority`-regulated ones. In Figure 6.18, $\tilde{v}$ data of such trajectories is depicted. Analogously, the data is plotted over the $\tilde{t}^{Int}$ feature. Initial $\tilde{v}$ values are also spread over the range between 0.600 and 1.000. What follows are steady $\tilde{v}$ decreases until minima are reached. In contrast to trajectories associated to yielding `priority` intersection paths, most trajectories exhibit one $\tilde{v}$ minimum at values around 0.000. From this depiction, one can also see that some trajectories exhibit episodes of stop-and-go driving. Finally, all trajectories show accelerating until the point closest to the intersection center is reached.

Conditions of standstill events are defined in section 5.1.4.2. When comparing estimates of the frequency of such with regard to trajectories depicted above, the following is noted. Of 100 randomly selected trajectories assigned to each of both a yielding `priority` intersection path of type I and one of type II, 11 and 15, respectively, exhibit at least one standstill. In the context of `yield`, `yield-to-right` and `stop`-regulated intersection paths, however, 24, 13 and 99, respectively, of 100 trajectories exhibit at least one standstill. Thus, only trajectories assigned to

stop-regulated intersection paths are distinctly different to yielding priority ones concerning this characteristic.

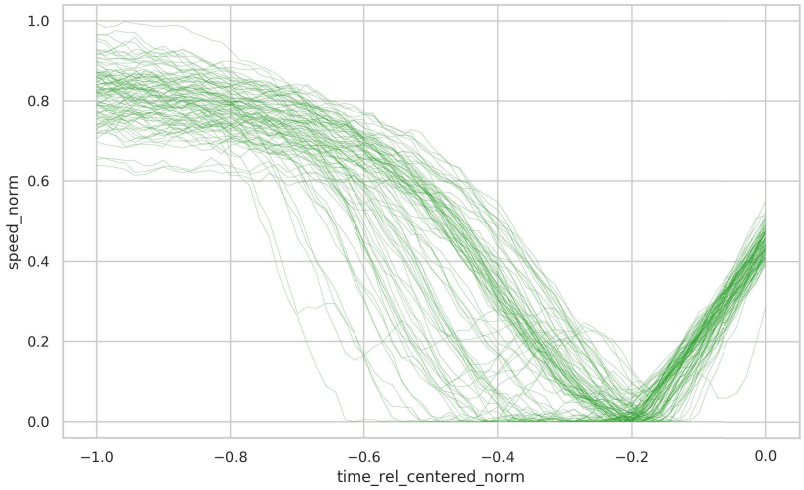


Figure 6.18: Normalized speed data of 100 randomly sampled trajectories associated to a stop-regulated intersection path, plotted over the intersection-relative time feature. Each green curve denotes data of one trajectory. Vehicles following this intersection path cross the associated intersection straight.

In this section, trajectory $\tilde{v}$ data is shown for trajectories assigned to only some specific intersection paths. However, obtained insights can also be confirmed in the context of other intersection paths. Also, the target direction described by yield, yield-to-right and stop-regulated intersection paths, hence turning left, right or going straight, is found to have no impact on this. While insights are shown with respect to the $\tilde{v}$ feature, similar ones can also be gathered when viewing plots of $\tilde{a}$ data. In order to reasonably limit the scope of this analysis, more plots are not shown. Apart from that, important to mention is that all intersection-incoming segments of stop, yield and yield-to-right-regulated intersection paths considered in this thesis are subject to 30 $\frac{\text{km}}{\text{h}}$ speed limits only. Thus, those are mostly followed by vehicles approaching intersections at comparatively low speeds.

6.3.1.2. Analysis of Likelihood-Based Inference Results

Additionally, the view of HMMs on intersection paths shall be considered. More precisely, the normalized class log-likelihoods of prioritized intersection paths, for which trajectory data is depicted above, are analyzed. These log-likelihoods are gathered from the experiment using the dataset which is augmented by data gathered from four additionally designed intersection scenarios. It is presented in section 6.2. The calculation of class log-likelihoods and how these are normalized is explained in section 5.2.2.

Figure 6.19 shows normalized class log-likelihoods that are obtained when conducting inference on two priority-regulated intersection paths that are followed by vehicles not having to give way to others. These are classified correctly. In each subplot, the normalized class log-likelihoods are depicted as blue bars, labeled with the respective value. The classes normalized log-likelihoods are associated to are abbreviated. The classes appear in the order priority, stop, traffic light, yield and yield to right on the horizontal axis below the bars. The two intersection paths depicted on the left and the right side of the figure describe going straight through and turning right at an intersection, respectively. Associated trajectory data is depicted in the subfigures (a) and (b) of Figure 6.14, respectively. In the case of the intersection path that

describes going straight through an intersection, the normalized class log-likelihoods are approx. 0.566 and 0.434 in favor of the `priority` and the `traffic light` class, respectively. The difference between these values is approx. 0.132. Concerning the intersection path that describes a right turn at an intersection, normalized class log-likelihoods are distributed over 4 of the 5 traffic regulation classes. The highest two values among these are approx. 0.444 and 0.214 also in favor of the `priority` and the `traffic light` class, respectively. Their difference amounts to approx. 0.230. Thus, the classification of this intersection path is made more distinctly than the one of the previously described intersection path.

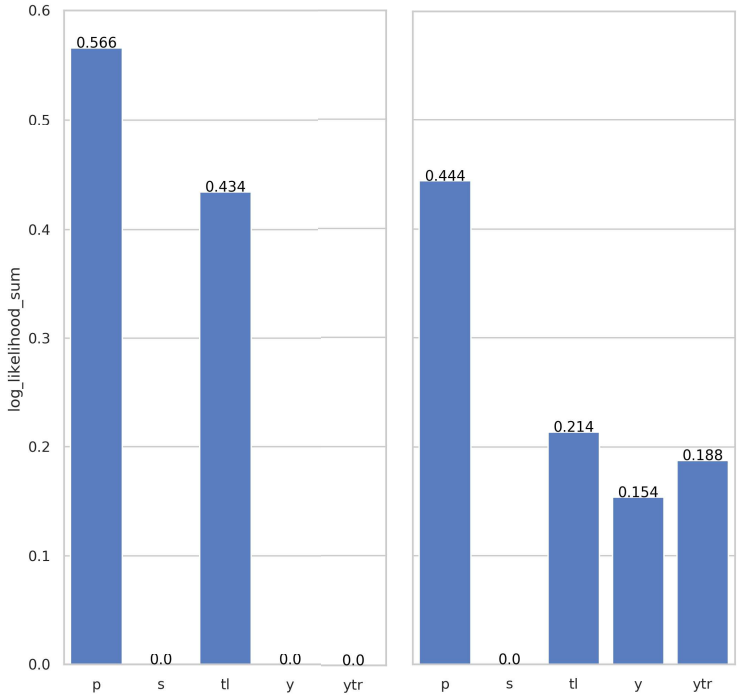


Figure 6.19: Normalized class log-likelihoods associated to two `priority`-regulated intersection paths. Magnitudes of these are visualized as blue bars. Those depicted in the left subplot are associated to an intersection path that describes passing straight through an intersection. Those depicted in the right subplot are associated to an intersection path that describes a right turn at an intersection. The regulation classes values are associated to are abbreviated and appear in the order `priority`, `stop`, `traffic light`, `yield` and `yield to right` below the respective bars.

Normalized class log-likelihoods for the two yielding `priority` intersection paths are depicted in Figure 6.20. As both are misclassified, values associated to other classes than the `priority` class are the highest among all intersection-path-associated values. The normalized class log-likelihoods depicted in the left subplot are associated to the yielding `priority` intersection path of type I. Figure 6.15 (a) shows $\tilde{v}$ data of trajectories assigned to this intersection path. The value of approx. 0.176 marks the normalized class log-likelihood in favor of the `priority` class and is only the third-highest. The highest and second-highest values are approx. 0.475 and 0.324. These are associated to the `stop` and `traffic light` class, respectively. The difference between the normalized log-likelihood in favor of the `priority` class and the highest one is approx. 0.299. The right subplot of Figure 6.20 shows the normalized class log-likelihoods associated to the yielding `priority` intersection path of type II. Figure 6.15 (b) shows $\tilde{v}$ data of trajectories assigned to this intersection path. The normalized class log-likelihood in favor of the `priority` class has a value of approx. 0.194 and is also only the third-highest one. The two highest values of approx. 0.280 and 0.277 are associated to the classes `yield` and `yield to right`, respectively.

The highest normalized class log-likelihood and the one associated to the `priority` class have a difference of approx. 0.086.

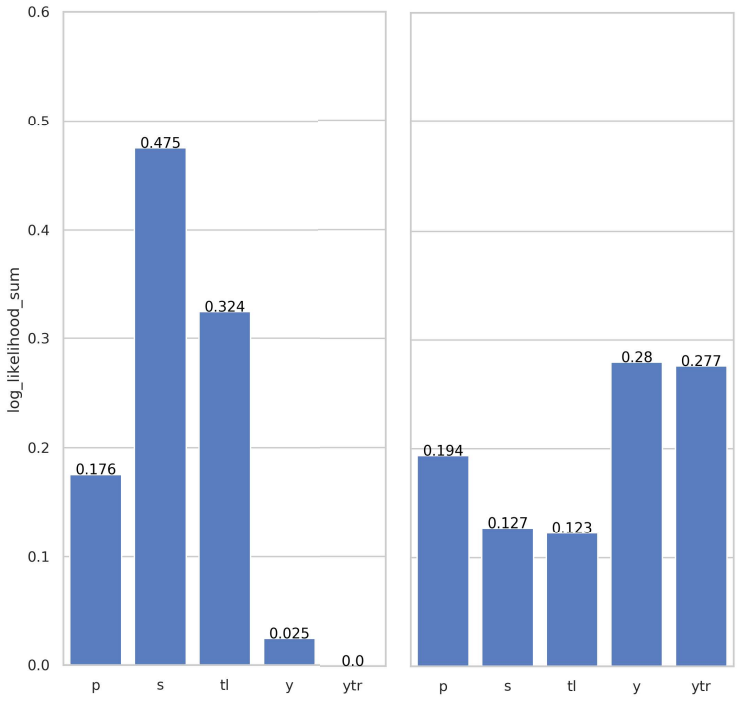


Figure 6.20: Normalized class log-likelihoods associated to two falsely classified `prioritized` intersection paths describing left turns. Magnitudes of these are visualized as blue bars. Those depicted in the left subplot are associated to an intersection path that is followed by vehicles approaching an intersection at comparatively high speeds. Those depicted in the right subplot are associated to an intersection path that is followed by vehicles approaching at comparatively low speeds. All of those vehicles potentially have to give way to vehicles following other `prioritized` intersection paths. The regulation classes values are associated to are abbreviated and appear in the order `priority`, `stop`, `traffic light`, `yield` and `yield to right` below the respective bars.

Above-presented test inference results provide insights into the classification of several `priority`-regulated intersection paths. Similar results are observed for intersection paths other than the shown `priority`-regulated ones that are followed by vehicles not having to give way to other vehicles and yielding-`priority`-regulated ones. However, in order to maintain a reasonable scope for this section, these are not analyzed.

6.3.2. Experiment Design

Insights gained from the problem analysis which is briefly outlined in the preceding section shall help comprehending why misclassifications of `priority`-regulated intersection paths occur. Based on this, a solution for stated misclassifications shall be developed. It shall account for the different behavior characteristics, assumed to be aggregated under the `priority` class. Before explaining the experimental setting of a suggested problem solution in detail in subsection 6.3.2.2, the design of several unsuccessful attempts are briefly outlined in subsection 6.3.2.1.

6.3.2.1. Unsuccessful Problem Approaches

Initially, all experiments conducted up to this point are scoured for already existing `priority` HMMs that potentially provide better performance regarding the misclassification of `priority`

intersection paths. Results of the grid search experiment comprise a great number of HMMs with different numbers of hidden states, using different representations of trajectories etc. This experiment is presented in section 6.1. For each set of runs validating a certain parameter combination, the test F_1 score mean is calculated for the binary classification problem comprising the `priority` class as one class and the remaining classes combined as a second class. However, the parameter combination producing the best-performing HMMs of the grid search experiment turns out to also provide the best solution for the stated binary classification problem.

In another solution approach, it is tried to substitute the `priority` class with two new ones. Technically, these should be referred to as auxiliary classes. However, for reasons of brevity and the fact that an intersection path which is assigned to either one of these auxiliary classes still gets labeled as `prioritized`, these are also referred to as classes. The main reason for creating these classes is deduced from the preceding problem analysis. Namely, two classes are employed in order to better account for the variety detected for trajectories following different `priority`-regulated intersection paths. One of these classes represents all yielding-`priority`-regulated intersection paths. The other class comprises solely the `prioritized` intersection paths that are followed by vehicles not having to give way to others. The remaining four classes remain unaltered. The dataset used for this approach is the augmented dataset that is also used in the experiment presented in section 6.2. All yielding-`priority` intersection paths are manually assigned to the corresponding new class. The remaining ones are assigned to the other new class. Subsequently, two HMMs representing the new classes, are parametrized on the corresponding data. The HMMs representing the remaining four classes, are the same ones that are parametrized in above-mentioned experiment which is conducted on an augmented trajectory dataset. During test inference, results are also validated using the strategy introduced in section 4.4. However, when examining the averaged results, the described modification does not lead to considerable improvements regarding mispredictions of `priority`-regulated intersection paths. Thus, it is not expanded on this six-class approach.

6.3.2.2. Design of the Suggested Solution

In the following, it is shown which further modifications of the six-class approach are performed in order to provide a more successful solution. Analogously to this approach, four of the models that are parametrized in the course of the experiment conducted on an augmented trajectory dataset are employed. These models represent the classes `yield`, `stop`, `traffic light` and `yield to right`. However, instead of substituting the `priority` class with two classes, three classes are created to represent this traffic regulation. Hence, the suggested approach is a seven-class approach. Each of the three new classes is represented by one HMM. As in the six-class approach, one of the three new classes represents `prioritized` intersection paths which are followed by vehicles not having to give way to others. The other two classes represent yielding-`priority`-regulated intersection paths. For this, the manually made distinction between the former class of intersection paths and the latter two, which is mentioned in section 6.3.2.1, is reused. The two classes representing yielding-`priority`-regulated intersection paths differentiate these based on the speed-level-based distinction which is introduced in section 6.3.1.1 of the problem analysis. Therefore, one of these two classes represents yielding-`priority` intersection paths of type I, whereas the other class represents those of type II. In this experiment, the speed-level-based distinction of intersection paths is implemented as follows: For each intersection path, the maximum v value of all points, of trajectories assigned to this intersection path, located on its

incoming segment is determined. In the case of this maximum being equal to or greater than a specific threshold, the associated intersection path is assigned to the class representing yielding-`priority` intersection paths of type I. In the complementary case, hence for v value maxima below stated threshold, the intersection path is assigned to the class representing those of type II. In this context, this simple implementation is used as it achieves the desired speed-level-based distinction well with the value of 12.50 $\frac{\mathrm{m}}{\mathrm{s}}$ being set for the stated threshold. However, for trajectory datasets in which v value outliers might be present, it is recommended to determine the intersection-path-representative v value that is compared to the threshold differently. This could, for instance, be done by also gathering the v maximum from each trajectory which is assigned to a specific intersection path. Instead of determining the greatest maximum value, an average could then be computed from these maxima. Also, the suitability of stated threshold value would need to be reexamined.

The three new HMMs are parametrized as determined by the parameter combination that produces the best-performing models of the grid search experiment. Thus, trajectories that are resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$ are used to parametrize four-hidden-state HMMs. Again, the dataset used for this approach is the augmented dataset that is also used in the experiment presented in section 6.2. As in other experiments, the validation strategy introduced in section 4.4 is applied.

6.3.3. Experiment Results

In the following, results of this experiment are shown. The classification result is presented and evaluated in subsection 6.3.3.1. Parametrized models representing the `priority` class are analyzed in subsection 6.3.3.2.

6.3.3.1. Evaluation of the Classification Result

By viewing the normalized joint confusion matrix depicted in Figure 6.21, one can see that all `prioritized` intersection paths are correctly assigned to their respective classes. However, two additional error cases emerge. Firstly, approx. 5 % of the `yield` intersection paths are falsely classified as yielding-`priority`-regulated ones of type II. Secondly, this label is falsely assigned also to approx. 10 % of the `yield-to-right`-regulated intersection paths. Calculated from the test F_1 scores of all four cross-validating experiment runs, the mean test F_1 score is approx. 0.809. The associated standard deviation is approx. 0.053.

Additionally, class log-likelihoods shall be presented. For this, the normalized class log-likelihoods of those `prioritized` intersection paths that are also analyzed in section 6.3.1 of the problem analysis are considered. Normalized class log-likelihoods of `prioritized` intersection paths followed by vehicles not having to give way to others and yielding-`priority`-regulated ones are shown in Figure 6.22 and Figure 6.23, respectively.

The `prioritized` intersection path that allows passing straight through the associated intersection has its associated normalized class log-likelihoods depicted in the left subplot of Figure 6.22. It is correctly classified with an associated normalized log-likelihood of approx. 0.566. The only other normalized log-likelihood greater 0.000 is associated to the traffic light class. The difference between both values is approx. 0.132. In the right subplot of the figure, one can see that the `prioritized` intersection path describing a right turn at the associated intersection is also

correctly classified. The associated normalized log-likelihood is approx. 0.325. The second-highest value is associated to the class representing yielding-`priority` intersection paths of type I. The value difference of both normalized log-likelihoods is approx. 0.126.

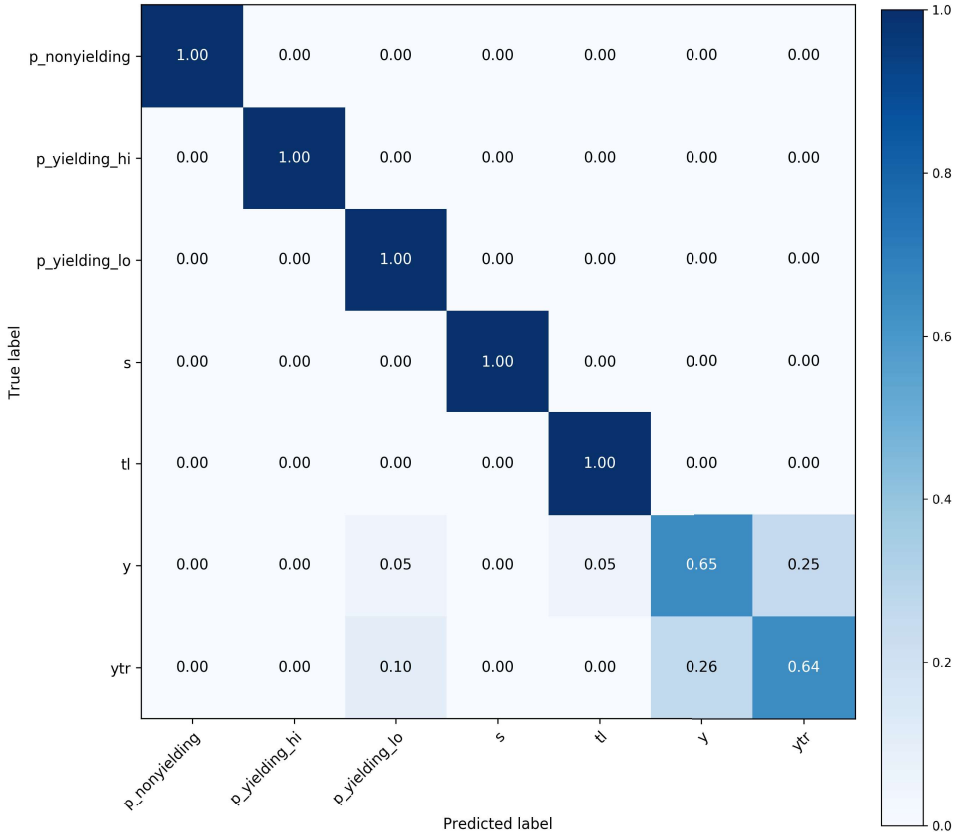


Figure 6.21: Normalized joint confusion matrix describing the classification result achieved in the context of a seven-class approach. Each class is represented by one Hidden Markov Model. Trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$ are used for the parametrization of these seven four-state Hidden Markov Models. A joint confusion matrix is computed by forming the elementwise sum of the individual confusion matrices describing the classification results of experiment runs cross-validating a certain parameter combination. Traffic regulation classes are abbreviated and appear on rows and columns. The first three abbreviations stand for classes jointly representing the `priority` class. The first covers `prioritized` intersection paths that are followed by vehicles which are not obliged to give way to others. The latter two classes cover `prioritized` intersection paths followed by vehicles that potentially have to give way to vehicles following other `prioritized` ones. On intersection paths of these two classes, vehicles approach intersections at comparatively high and low speeds, respectively. The remaining abbreviations stand for the classes `stop`, `traffic light`, `yield` and `yield to right`.

The yielding-`priority`-regulated intersection path of type I is now also correctly classified. The associated normalized log-likelihood is approx. 0.690. This is shown in the left subplot of Figure 6.23. The difference to the second-highest normalized log-likelihood, which is associated to the `stop` class, is approx. 0.512. The magnitude of this difference stands out among evaluated ones. Thus, the classification of this intersection path can be interpreted as a highly distinct one. The majority of other yielding-`priority` intersection paths of type I are also classified similarly distinctly. Normalized class log-likelihoods of the intersection path of type II are depicted in the right subplot of the same figure. This intersection path is also classified correctly with a normalized log-likelihood of approx. 0.238. In contrast to the previous intersection path, the

difference between stated normalized log-likelihood and the second-highest one is only approx. 0.006. In other words, the normalized log-likelihood associated to the correct class is only marginally above the one associated to the `yield` class. Similar results are also observed for other yielding `priority` intersection paths of type II.

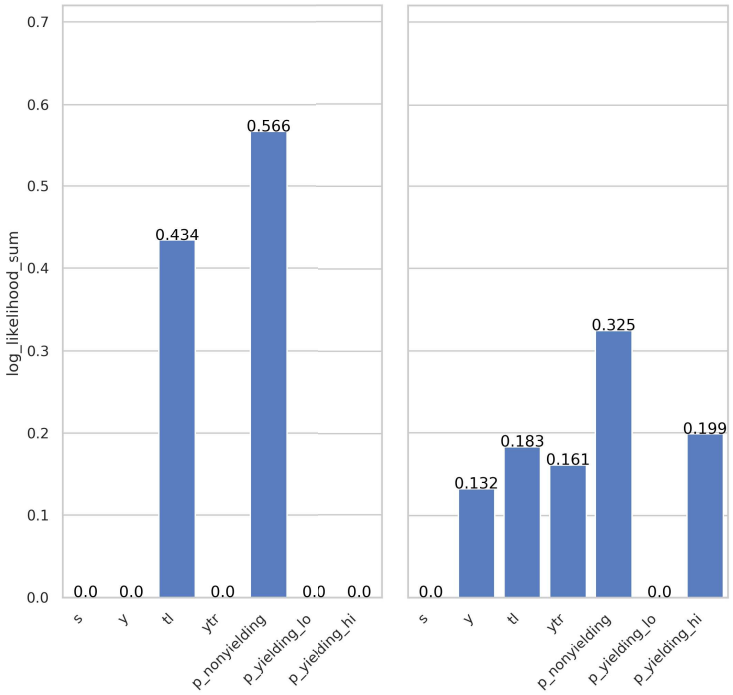


Figure 6.22: Normalized class log-likelihoods associated to two `prioritized` intersection paths. Magnitudes of these are visualized as blue bars. Those depicted in the left subplot are associated to an intersection path that describes passing straight through an intersection. Those depicted in the right subplot are associated to an intersection path that describes a right turn at an intersection. The regulation classes log-likelihoods are associated to are abbreviated and appear below the respective bars. The first four appear in the order `stop`, `yield`, `traffic light`, and `yield to right`. The last three abbreviations stand for classes jointly representing the `priority` class. The first covers `prioritized` intersection paths that are followed by vehicles which are not obliged to give way to others. The latter two classes cover `prioritized` intersection paths followed by vehicles that potentially have to give way to vehicles following other `prioritized` ones. On intersection paths of these two classes, vehicles approach intersections at comparatively low and high speeds, respectively.

With regard to the two additional error cases mentioned above, an associated finding is outlined in this paragraph. Certain `yield` and `yield-to-right` intersection paths are falsely classified as yielding-`priority`-regulated ones of type II. However, a common property shared by these intersection paths is that all describe right turns at intersections. In the context of intersections in Germany, it can be assumed that right-turning `prioritized` intersection paths generally do not cross other `prioritized` ones. Thus, assigning these intersection paths to classes representing any yielding-`priority`-regulated ones goes against the definition of these classes.

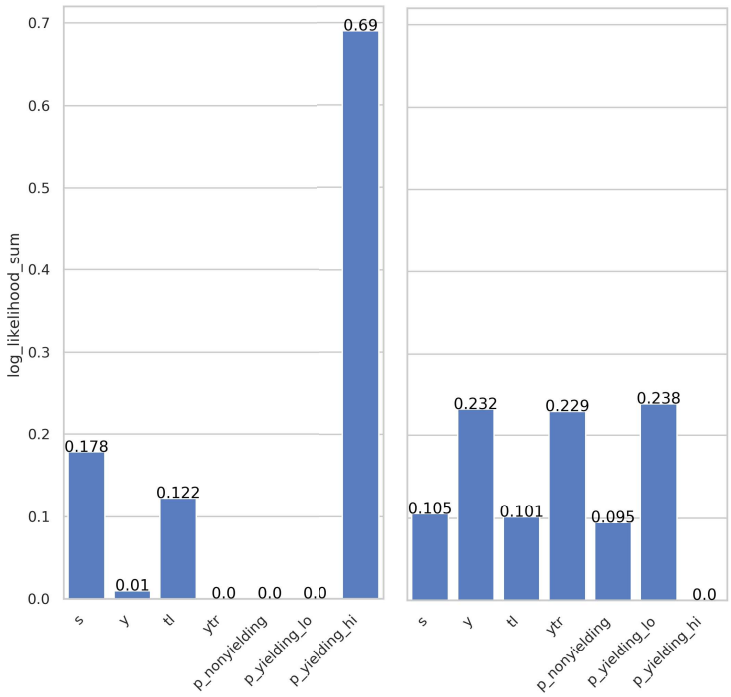


Figure 6.23: Normalized class log-likelihoods associated to two `prioritized` intersection paths describing left turns. Magnitudes of these are visualized as blue bars. Those depicted in the left subplot are associated to an intersection path that is followed by vehicles approaching an intersection at comparatively high speeds. Those depicted in the right subplot are associated to an intersection path that is followed by vehicles approaching at comparatively low speeds. All of those vehicles potentially have to give way to vehicles following other `prioritized` ones. The regulation classes log-likelihoods are associated to are abbreviated and appear below the respective bars. The first four appear in the order `stop`, `yield`, `traffic light`, and `yield to right`. The last three abbreviations stand for classes jointly representing the `priority` class. The first covers `prioritized` intersection paths that are followed by vehicles which are not obliged to give way to others. The latter two classes cover `prioritized` intersection paths followed by vehicles that potentially have to give way to vehicles following other `prioritized` ones. On intersection paths of these two classes, vehicles approach intersections at comparatively low and high speeds, respectively.

6.3.3.2. Analysis of Parametrized Hidden Markov Models

In the following, the observation densities of the three new HMMs jointly representing the `priority` class shall be visualized. The visualization approach designed for this purpose is introduced in section 6.1.2.2.1. All analyzed HMMs are parametrized on the analysis development data portion.

Figure 6.24 visualizes observation densities of the HMM representing the class which covers `prioritized` intersection paths that are followed by vehicles not having to give way to others. All parameter values of depicted observation densities can be gathered from Table A.4.1 in section A.4 of the Appendix. A depiction of transition probabilities associated to this model is found in Figure A.4.1 in the same section in the Appendix. The model shares similarities with `priority` HMMs parametrized in other experiments. With regard to the mean component values and two-standard-deviation regions associated to the features $\tilde{v}$ and $\tilde{a}$, the first and second hidden state frequently emit observations of free-driving. This can be seen in the first and second subplot, respectively, of the upper row of plots in Figure 6.24. The third hidden state frequently emits

observations of braking from medium or high speed and rarely also of gentle accelerating. However, the model does not have a state that frequently emits observations of soft as well as hard braking and accelerating from medium speeds. Instead, the fourth state of this HMM is one of three states that frequently emit observations of free-driving.

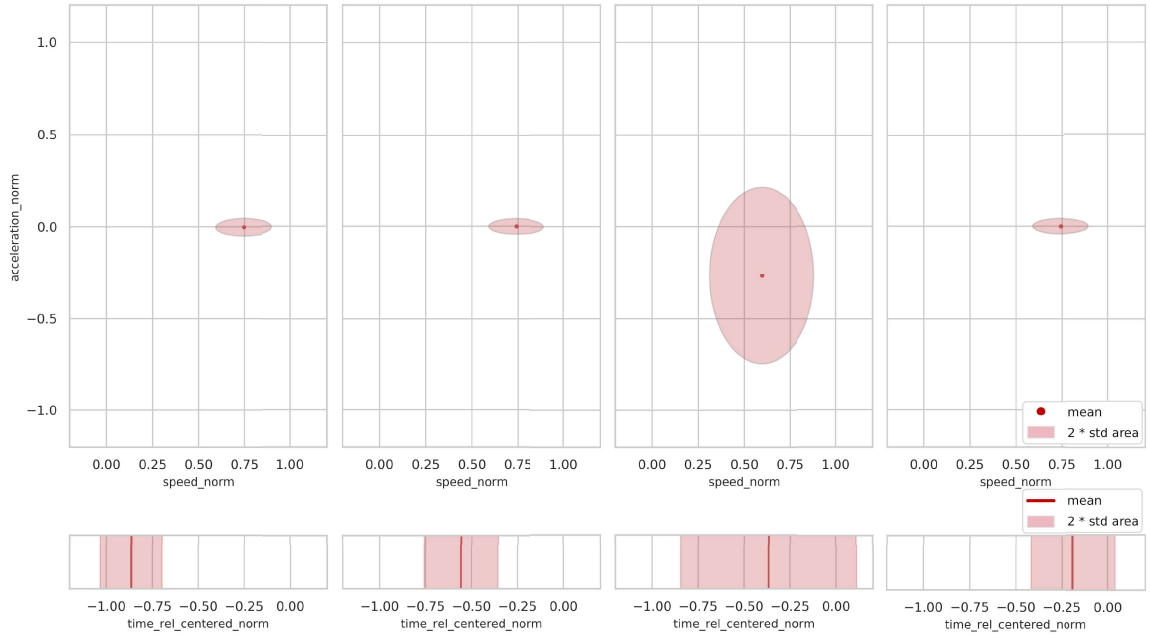
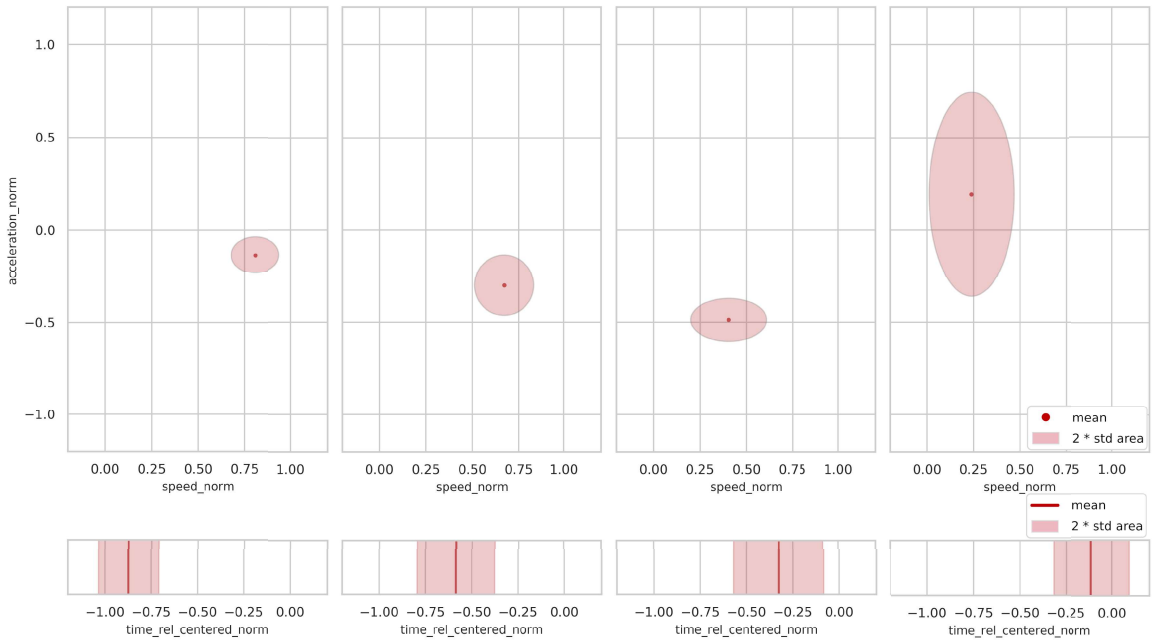
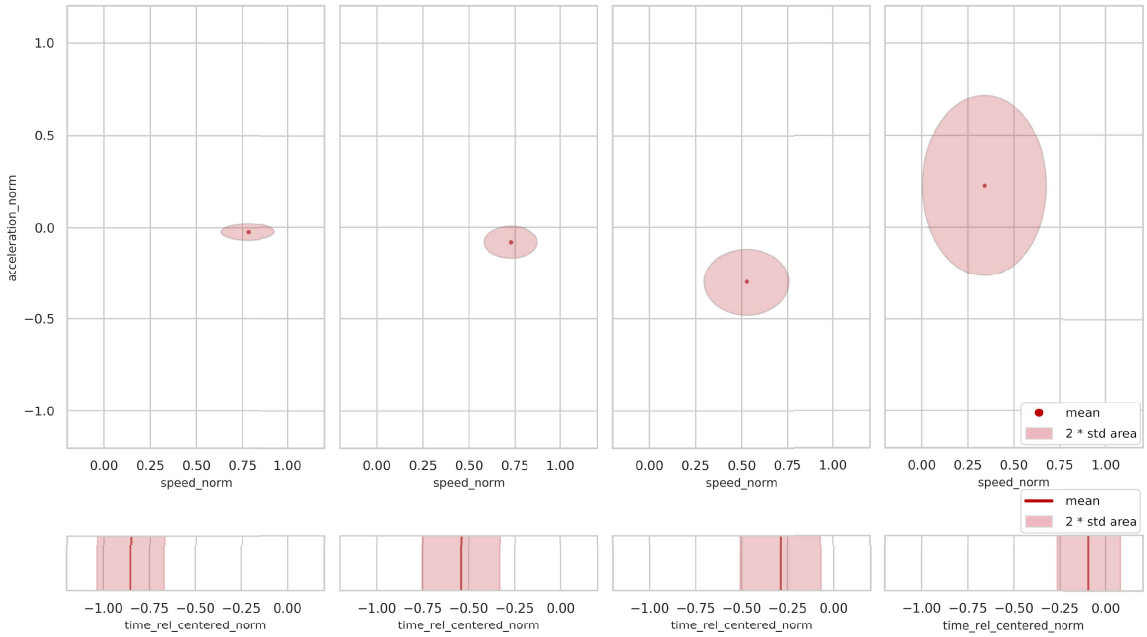


Figure 6.24: Visualization of learned observation densities of a four-hidden-state Hidden Markov Model representing the `priority` class for intersection paths followed by vehicles that are not obliged to yield to others. Each column of plots visualizes a Gaussian in one hidden state. The column order reflects the hidden-state numbering of the model. The model is parametrized using trajectories that are resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$.

The observation densities visualized in Figure 6.25 (a) and (b) are associated to the models representing the classes that cover yielding-`priority` intersection paths of type I and II, respectively. All parameter values of depicted densities are found in Table A.4.2 and Table A.4.3, respectively in section A.4 of the Appendix. Visualizations of transition probabilities are depicted in Figure A.4.2 and Figure A.4.3 in the same section of the Appendix. At first glance, the observation densities of both models seem similar. The first three hidden states of the model representing the classes that cover yielding-`priority` intersection paths of type I frequently emit observations of braking. The individual mean component values associated to the $\tilde{a}$ feature decreases over these three states. Hence, observations of increasingly hard braking are frequently emitted. The densities in the first two hidden states model braking from high speeds and the third from medium speed. The model representing the classes that cover yielding-`priority` intersection paths of type II also has multiple hidden states frequently emitting observations of breaking from high or medium speeds. More precisely, the second and third hidden state have observation densities with negative mean component values associated to the $\tilde{a}$ feature. Again, these values decreases from the second to the third state. The observation density in the fourth hidden state is similar for both HMMs. Observations of accelerating or soft braking from low or medium speed are frequently emitted by these states. However, both models differ in the observation density in the first hidden state. Instead of solely emitting observations of braking, the density in the first hidden state of the model depicted in subfigure (b) frequently emits observations of very gentle braking and also free-driving.



(a) Densities of a model representing the class covering intersection paths followed by vehicles that approach intersections at comparatively high speeds.



(b) Densities of a model representing the class covering intersection paths followed by vehicles that approach intersections at comparatively low speeds.

Figure 6.25: Visualization of learned observation densities of two four-hidden-state Hidden Markov Models representing the `priority` classes for intersection paths followed by vehicles that potentially have to give way to vehicles following other `prioritized` ones. These vehicles approach intersections at comparatively high (a) and low speeds (b), respectively. In both subfigures, each column of plots visualizes a Gaussian in one hidden state. The column order reflects the hidden-state numbering of the models. The models are parametrized using trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$.

6.4. Discussion of Results

In the following, several findings of the conducted experiments are revisited and critically discussed or brought into context. Subsection 6.4.1 expands on gained insights into observation densities and transition probabilities. Characteristics of the trajectory data basis are discussed in subsection 6.4.2.

6.4.1. Insights into Hidden Markov Model Characteristics in the Context of Traffic Regulation Inference

In a series of different experiments, HMMs are parametrized with the purpose of representing traffic regulations. Resulting observation probability densities are discussed in subsection 6.4.1.1, transition probabilities in subsection 6.4.1.2.

6.4.1.1. Hidden States and Observation Densities

The HMMs that are termed the best-performing ones of the grid search experiment are produced by a parameter combination determining the hidden-state count to be set to four. This experiment is presented in section 6.1. As shown in section 6.1.2.1.3, other parameter combinations determine this parameter to be set to higher values which yields higher test F_1 score means. Based on this finding, one might argue that higher hidden-state counts are more suitable to model and describe trajectories of differently regulated intersection paths, given that other parameters are set to suitable values. However, as outlined in section 6.1.2.2.3, analyzed models with higher hidden-state counts and test scores are found to exhibit high redundancies regarding learned observation densities in different hidden states. Almost no redundancies are detected with respect to the best-performing HMMs. Additionally, in the context of certain values of other parameters, evidence is found that overfitting might occur when choosing higher hidden-state counts. In Figure 6.4, one can see how training F_1 score means steadily increase up to a hidden-state count of nine. In contrast, test F_1 score means have a local maximum at the hidden-state count of four. After that, test F_1 score means decrease. A test F_1 score mean higher than stated local maximum is not achieved for state counts lower than eight. This phenomenon of increasing training and decreasing test score can be a sign of overfitting [12]. Furthermore, one of the competing HMM sets with a hidden-state count of eight achieve test F_1 scores with a standard deviation of approx. 0.071. In contrast, the standard deviation associated to the best-performing model set is approx. 0.057, hence lower. Put differently, test F_1 scores associated to the competing HMM set vary stronger, which indicates good performance on specific test data portions and comparatively poor performance on other portions. For described reasons, it is decided to focus on above-mentioned four-hidden-state HMMs.

HMMs developed within the experiment aiming at counteracting false `priority` classifications shall also be critically evaluated. This experiment is presented in section 6.3. One of these is parametrized on `priority` intersection paths that allow passing straight through or turning right at an intersection and are followed by vehicles that are not obliged to yield to others. Associated observation densities are visualized in Figure 6.24. In this figure, one can see how observation densities in three of the four hidden states seem redundant with respect to mean vector components associated to the $\tilde{v}$ and $\tilde{a}$ feature. These frequently emit free-driving observations and diversify in terms of components associated to the $\tilde{t}^{Int}$ feature. Basically, this is what is expected. Vehicles following such intersection paths generally do not stop during intersection

crossings which produces comparatively invariable observation signals. Thus, considering that four-hidden-state HMMs are suitable to model more complex observation signals exhibiting e.g. braking or standstills, aforementioned signals are relatively simple to model by such HMMs. It can be reasoned that two of these three hidden states could be merged into one in order to model observations in a less redundant way. Another instance of this finding might be the model that is parametrized on intersection paths belonging to the yielding `priority` class of type I in the context of the same experiment. From the observation densities in four hidden states, three model observations of braking. These are visualized in the first three plot columns of Figure 6.25 (a). Based on the same reasoning, one might try to merge some of these states.

In the context of the same experiment's analysis another remark regarding yielding-`priority`-regulated intersection paths shall be made. As outlined in section 6.3.3.2, the two models representing the class covering yielding `priority` intersection paths of type I and II, respectively, differ regarding learned observation density in the first hidden state. The densities associated to the model representing the class of stated intersection paths of type I are visualized in Figure 6.25 (a). Figure 6.25 (b) visualizes the densities associated to the model which represents the class of yielding-`priority`-regulated intersection paths of type II. As one can see, the first hidden state of the former model frequently emits observations of gentle braking. The corresponding state of the latter model, however, frequently emits observations of very gentle braking and also free-driving. This difference goes well with what is observed regarding trajectory data in section 6.3.1.1 of the problem analysis of that experiment. The $\tilde{v}$ data of trajectories associated to yielding-`priority` intersection paths is depicted in Figure 6.15. On the one hand, trajectories assigned to yielding-`priority` intersection paths of type I show noticeably decreasing $\tilde{v}$ values which can be seen on the left side of subfigure (a) of Figure 6.15. On the other hand, trajectories assigned to yielding-`priority` intersection paths of type II show only slightly decreasing or almost constant $\tilde{v}$ values at the beginning of intersection approaches. These are depicted in subfigure (b) of this figure. Thus, one might interpret the first hidden state of both aforementioned HMMs as being one element that enables the distinction of both types of yielding-`priority`-regulated intersection paths.

6.4.1.2. Transition Probabilities

HMMs are suitable to model alternative sequences of frequently emitted observations. This can, for instance, be shown for the best-performing `priority` model of the grid search experiment. Associated visualizations of observation densities and transition probabilities are depicted in Figure 6.5 and Figure 6.6, respectively. The densities in the first and the third hidden state model observations of free-driving close to the beginning and the end of drives along intersection-incoming segments. Observations of braking are frequently emitted by the second hidden state. As visible in the latter figure, transitioning from the first into the third hidden state has a probability of approx. 0.024, whereas transitioning in the second one has a probability of approx. 0.012. The former transition models observations gathered from vehicles that approach and leave intersections without braking. In a probable sequence of hidden states underlying such observations, states that frequently emit observations of braking might be dispensable. Instead, the latter transition models observations gathered from vehicles that brake during their approach. In this case, a probable sequence would most likely include hidden states that frequently emit observations of braking. In other words, two different ways of crossing intersections are modeled by this HMM. Apart from that, the former transition is more probable than the latter. As HMMs

are learned from trajectories, a reasonable explanation for this might be that specific intersection paths are followed by more vehicles than others.

As mentioned in section 5.2.1, initially, the topology for all HMMs is ergodic. In several parametrized models, characteristics of the ergodic topology are retained during the parametrization procedure. For instance, pairs of transitions with opposite directions that interconnect hidden-state pairs are noted. However, the topology of, for instance, the parametrized HMM representing the class of yielding-`priority` intersection paths of type I can be described as a left-to-right one [13]. It is parametrized in the experiment aiming at counteracting false `priority` classifications. The learned transition probabilities are visualized in Figure A.4.2. In this figure, one can see how the entirety of all transition probabilities rounded to values greater than 0.000 form a directed path through the four hidden states.

6.4.2. Trajectory-Data-Related Remarks

The experiment conducted on an augmented dataset aims at improving classification performance with regard to intersection paths belonging to the `priority` and `yield` class. This experiment is presented in section 6.2. The additionally created data increases the number of exactly these intersection paths. Yet, testing is done exclusively on the initial dataset, following the strategy which is introduced in section 4.4. With HMMs being parametrized on trajectories assigned to more different intersection paths, an improved inference performance regarding the `priority` and `yield` class is expected. Instead, results show only marginal improvements in terms of false classifications of `yield` intersection paths and a slight deterioration in terms of the classification of `priority` intersection paths. The normalized joint confusion matrices depicted on the left side of Figure 6.3 and in Figure 6.11 show the classification performance when parametrizing HMMs on the initial and on the augmented dataset, respectively. As outlined in section 6.2.2.1, the fraction of correctly classified `yield` intersection paths increases by roughly 5 percentage points when using the augmented dataset. The fraction of the correctly classified `priority` intersection paths decreases by roughly 6 percentage points. However, the test F_1 score mean achieved in this experiment is, with a value of around 0.817, only approx. 2 % lower than the one achieved by the best-performing HMMs of the grid search experiment. Considering that results can vary within small ranges which is mentioned at the beginning of section 6, it could be argued that a score difference that low can be neglected. In contrast, using an augmented dataset yields to a decrease in test F_1 score standard deviation by approx. 39 % to a value of approx. 0.035. Thus, test scores achieved by the different experiment runs cross-validating the approach in this experiment vary considerably less. This goes well with the finding presented in section 6.2.2.2. Regarding `yield` models parametrized in different cross-validation runs on different development portions of the augmented dataset, almost no deviation can be determined when visually comparing associated learned observation densities. Two instances of these `yield` models are depicted in subfigure (b) of Figure 6.12 and in Figure 6.13, respectively. In contrast, as mentioned in section 6.1.2.2.2 of the grid search experiment, best-performing `yield` models parametrized on the initially created dataset do exhibit noticeable deviations. Thus, one might argue that parametrizing `yield` models on trajectories assigned to more different intersection paths reduces deviations regarding the observation densities of produced models.

However, the volume and the variety of trajectory data still seem to limit a further analysis of the results. Especially with respect to the experiment aiming at counteracting false `priority`

classifications, this poses a problem. As explained in section 6.3.2.2, three classes instead of one are used to represent different types of `priority` intersection paths. Consequently, available `priority`-regulated development intersection paths are divided into three sets in order to parametrize three models that represent the new classes. Analogously, intersection paths belonging to the test set are also divided into three sets in order to conduct test inference. This way, these sets end up containing even less `priority`-regulated intersection paths. Thus, it can be argued that the computed test F_1 score mean is not representative of the performance of this approach. Additionally, the following has to be viewed critically. In several cases, the log-likelihoods of correctly classified yielding-`priority` intersection paths of type II suggest that correct classifications are only slightly achieved. For instance, a yielding-`priority`-regulated intersection path of type II has a difference of approx. only 0.006 between the log-likelihoods associated to the correct and a false class. This can be seen in the right subplot of Figure 6.23. It would have to be tested whether developing HMMs on even more data mitigates issues like this.

Nevertheless, also evidence supporting the suitability of above-described approach can be identified. The associated normalized joint confusion matrix is depicted in Figure 6.21. In this matrix, one can see that no `priority`-regulated intersection path is falsely classified. Instead, approx. 5 and 10 % of `yield` and `yield-to-right`-regulated intersection paths, respectively, are falsely classified as yielding-`priority`-regulated intersection paths of type II. As mentioned in section 6.3.3.1, all of these falsely classified intersection paths describe right turns at intersections. Yielding-`priority`-regulated intersection paths, however, are such that describe left turns at or going straight through intersections and are followed by vehicles that potentially have to give way to vehicles following other `priority`-regulated intersection paths. But at German intersections, vehicles following `prioritized` intersection paths that describe right turns generally do not have to give way to other vehicles. Thus, using this kind of context knowledge could postprocess classification results and possibly avoid false classifications. Certainly, if cyclists or pedestrians would be involved, this finding has to be reevaluated.

7. Conclusion and Outlook

In the following, this thesis shall be concluded. A summary of the suggested method and the most important results are given in subsection 7.1. Several directions for future work are provided in subsection 7.2.

7.1. Conclusion

The scientific interest of this thesis is the development of an approach based on Hidden Markov Models (HMM) to the inference of traffic regulations. The design of the suggested approach is as follows: HMMs are randomly initialized with an ergodic topology. Multivariate Gaussians are employed as observation probability density functions in the hidden states. Each individual traffic regulation is represented by one or more HMMs. These are parametrized using class-stereotypical trajectory data. Based on likelihood, each intersection path is assigned to a regulation class. In all experiments, approaches are cross-validated and achieved performance scores are averaged. This is done in a cross-intersection manner. Thus, trajectory data gathered from a specific intersection is always found in either the portion used for classifier development or in the portion employed for testing. It is never found in both portions simultaneously. In essence, the representation of trajectories on which HMMs are parametrized is based on series of sampled measurements of speed and acceleration. On the one hand, in a grid search experiment a variety of possible parameter and design choices further specifying this representation are evaluated. From this experiment, the combination leading to the best classification results is selected and determines how trajectories are represented throughout the remaining experiments. The result is the following: Trajectories are resampled and described using the features speed, acceleration and intersection-relative time. The latter of these three features indicates the duration between each trajectory point and the respective intersection center. In the same experiment, four-hidden-state HMMs are highlighted as providing a good trade-off between high classification performance and the risk of overfitting the data. On the other hand, several trajectory-representation-related parameters are not varied in experiments. Trajectories always begin at the point where the associated vehicle has a driving distance of 70 m to the intersection center ahead. Features are consistently normalized in all presented experiments. Of each trajectory, only the segment describing the intersection approach until the point closest to the intersection center is reached is employed as parametrization basis and inference target of HMMs. Resampled trajectory segments always have a length of 71 trajectory points.

Several performance scores achieved by above described method are presented in the following. On the initially created dataset, the HMMs, parametrized as described, achieve a mean test F_1 score of approx. 0.832. All `stop` and `traffic-light`-regulated intersection paths are correctly classified. Approx. 11 and 40 % of `priority` and `yield`-regulated intersection paths, respectively, are falsely classified. Also, a strong confusion between the classes `yield` and `yield-to-right` is noted.

In a subsequent experiment, the impact of developing models on an augmented dataset is examined. As a result, classification performance regarding the `yield` class is found to improve. False classifications of `yield`-regulated intersection paths are reduced by around 5 percentage points. However, regarding the `priority` class, a slight decrease in terms of classification performance is noted. Approx. 17 % of `priority`-regulated intersection paths are misclassified which corresponds to an increase of around 6 percentage points.

Based on an analysis of false classifications of `priority`-regulated intersection paths, the way these are represented is modified in another experiment. Instead of having one `priority` class, three classes are used to represent this traffic regulation. These shall account for the diversity of behavior characteristics that can be identified when analyzing associated trajectories. One of those three classes covers `priority`-regulated intersection paths that allow passing an intersection without having to give way to other traffic participants. Intersection paths belonging to the other two classes are followed by vehicles that potentially do have to give way to vehicles following other `prioritized` intersection paths. Furthermore, these two classes distinguish intersection paths based on the speed level vehicles initially drive at when approaching intersections. With the aid of this seven-class-approach, no `priority`-regulated intersection path is misclassified. However, approx. 5 % of `yield` and 10 % of `yield-to-right`-regulated intersection paths are falsely assigned to one of the three classes representing the `priority` regulation. Besides, deeper analyses show that several correctly classified `priority`-regulated intersection paths are close to being falsely recognized. In other words, a high risk of false classifications is assumed.

In conclusion, the results achieved by the suggested method are promising. Parametrizing HMMs using trajectory data is a suitable approach to infer traffic regulations at intersections. Based on refinements of this approach, all available intersection paths belonging to the traffic regulation classes `priority`, `stop` and `traffic light` are classified correctly. However, with respect to `yield` and `yield-to-right`-regulated intersection paths, problems persist. Showing potential solutions for solving these would go beyond the scope of this thesis. Apart from that, other weaknesses in the context of data volume and variety are identified. A repertoire of yet more intersection paths extracted from different intersections is considered necessary in order to improve and also better validate achieved results. Specifically, it is vital that the variety of data is increased. This especially refers to the variety of speed limits applying for `yield` and `stop`-regulated intersection paths. This is motivated by the fact that intersection-incoming segments of these intersection paths only appear in combination with 30 $\frac{\text{km}}{\text{h}}$ speed limits in the data.

7.2. Outlook

In this section, several possibilities to build on the initial success of this method are provided. In the context of the trajectory representation, unexplored combinations of parameters and choices remain. For instance, the intersection path distance feature is only tested with other parameters and design choices being set to fixed values and decisions, respectively. This feature specifies the distance between trajectory points and the center point of the respective intersection. Alongside other trajectory features, it could be varied in a more extensive grid search. One would then need to examine whether this yields well or even better-performing models. Additionally, a feature indicating the Euclidean distance between each trajectory point and the respective intersection center point could be introduced. It could be scaled based on the radius of the respective intersection in order to enable a more generalized comparison between differently sized intersections. The usefulness of such a feature is illustrated as follows: There are `priority`-regulated intersection paths that are followed by vehicles that potentially have to give way to other traffic participants. Often, such intersection paths describe left turns at intersections. On these, slowing down or stopping in order to give way generally occurs close to the intersection center point, hence often inside of intersections. In contrast, vehicles following `yield`, `stop` or `yield-to-right`-regulated intersection paths typically slow down or stop in order to give way before entering the intersection, hence generally outside of it. Thus, in the context of intersection paths being regulated according to these traffic regulations, often comparatively high distance values would be assigned to trajectory points at which stopping or slowing down occurs. The corresponding points occurring along `prioritized` intersection paths, however, would often be assigned comparatively low distance values. With described characteristics, this feature could be suitable to provide an alternative solution to the problem of falsely classified `priority`-regulated intersection paths.

In the context of HMM-related design choices, the way observation signals are modeled could be modified. Certain well-performing models turn out to develop observation densities in different hidden states that are redundant. The development process could be repeated with lower hidden-state counts exclusively for such HMMs. One would then need to examine whether original performances can be maintained. Alternatively, the use of mixture models as observation probability density functions could be evaluated. More precisely, it could be examined whether multiple hidden states with partially redundant observation densities can effectively be substituted by a lower number of hidden states with mixture models. This is motivated by the ability of mixture models to model complex and versatile phenomena [50]. What should also be mentioned is that, depending on the actual design of an approach employing these, mixture models are generally defined by a higher number of parameters than multivariate Gaussians employed to model the same data. Thus, this could result in greatly increased development durations. In both described solution approaches, HMM sets can, in contrast to those applied in this thesis, comprise HMMs with different hidden-state counts. For this reason, one needs to examine whether this inference design still yields classification performances as high as before. If this is not the case, normalization of log-likelihoods could be a suitable modification. How well an HMM describes a set of trajectories is quantified by calculating the log-likelihood of these given the HMM. The normalization could be realized as a scaling based on value ranges of log-likelihoods observed for each individual model. Log-likelihood values achieved during inference that are close to the observed log-likelihood maximum are mapped to high values, whereas values closer to the observed minimum are mapped to lower values.

Several experiments are conducted in order to find solution approaches to the confusion between `yield` and `yield-to-right`-regulated intersection paths. As no solution solely using HMMs is found, the thesis focuses on other experiments for reasons of consistency. However, a possible approach including a mixture of methods could be designed as follows: As specified by the inference design suggested in this thesis, each intersection path is assigned a vector of log-likelihoods. Each of these log-likelihoods is calculated based on one model of the set of HMMs used for inference. Thus, the log-likelihoods calculated based on the `yield` and `yield-to-right` model could, for instance, augment feature vectors designed in related works [4–7]. In these works, feature vectors serve as input for discriminative classifiers, as, for instance, Neural Networks, Support Vector Classifiers or Random Trees. One would then need to evaluate whether combinations of existing features and log-likelihoods are suitable to improve classification performances.

References

[1] J. Ziegler *et al.,* "Making Bertha Drive — An Autonomous Journey on a Historic Route," *IEEE Intell. Transport. Syst. Mag.*, no. 6, pp. 8–20, 2014.

[2] F. Kunz *et al.,* "Autonomous Driving at Ulm University: A Modular, Robust, and Sensor-Independent Fusion Approach," in *2015 IEEE Intelligent Vehicles Symposium*, Seoul, Korea, 2015, pp. 666–673.

[3] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, 2008.

[4] C. Ruhhammer, *Inferenz von Kreuzungsinformationen aus Flottendaten: Dissertation*. Karlsruhe: KIT Scientific Publishing, 2017. Accessed: Feb. 12 2020. [Online]. Available: https://www.ksp.kit.edu/9783731507215

[5] C. A. Pribe and S. O. Rogers, "Learning to Associate Observed Driver Behavior with Traffic Controls," *Transportation Research Record Journal of the Transportation Research Board*, vol. 1679, no. 1, pp. 95–100, 1999.

[6] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "SmartRoad: Smartphone-Based Crowd Sensing for Traffic Regulator Detection and Identification," *ACM Trans. Sen. Netw.*, vol. 11, no. 4, pp. 1–27, 2015.

[7] F. Saremi and T. Abdelzaher, "Combining Map-Based Inference and Crowd-Sensing for Detecting Traffic Regulators," in *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems: Proceedings*, Dallas, TX, USA, 2015, pp. 145–153.

[8] S. Zourlidou, C. Fischer, and M. Sester, "Classification of street junctions according to traffic regulators," in *Accepted Short Papers and Posters from the 22nd AGILE Conference on Geo-information Science*, Limassol, Cyprus, 2019.

[9] R. Carisi, E. Giordano, G. Pau, and M. Gerla, "Enhancing in Vehicle Digital Maps via GPS Crowdsourcing," in *2011 Eighth International Conference on Wireless On-Demand Network Systems and Services*, Bardonecchia, Italy, 2011, pp. 27–34.

[10] S. Dasgupta, D. Hsu, and C. Monteleoni, "A general agnostic active learning algorithm," Department of Computer Science and Engineering, University of California, San Diego CS2007-0898, 2007.

[11] Federal Highway Administration, *Intersection Safety: Unsignalized Intersections.* [Online]. Available: https://safety.fhwa.dot.gov/intersection/conventional/unsignalized/ (accessed: Feb. 12 2020).

[12] T. M. Mitchell, *Machine Learning*. New York, NY: McGraw-Hill, 1997.

[13] L. R. Rabiner, “A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[14] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,* 3rd ed., 2018.

[15] L. Damiano, B. Peterson, and M. Weylandt, “A Tutorial On Hidden Markov Models Using Stan,” 2018.

[16] M. Awad and R. Khanna, Eds., *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Berkley, CA: Apress Open, 2015.

[17] C. Manning, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2009.

[18] V. Labatut and H. Cherifi, “Evaluation of Performance Measures for Classifiers Comparison,” *Ubiquitous Computing and Communication Journal*, no. 6, pp. 21–24, 2011.

[19] H. Narasimhan, W. Pan, P. Kar, P. Protopapas, and H. G. Ramaswamy, “Optimizing the Multiclass F-Measure via Biconcave Programming,” in *16th IEEE International Conference on Data Mining*, Barcelona, Spain, 2016, pp. 1101–1106.

[20] P. A. Lopez *et al.,* “Microscopic Traffic Simulation using SUMO,” in *2018 IEEE Intelligent Transportation Systems Conference: November 4-7, Maui, Hawaii*, Maui, HI, USA, 2018, pp. 2575–2582.

[21] German Aerospace Center (DLR), *SUMO: FAQ.* [Online]. Available: https://sumo.dlr.de/docs/FAQ.html (accessed: Feb. 13 2020).

[22] OpenStreetMap contributors, *Planet dump retrieved from https://planet.osm.org*, 2017. Accessed: Feb. 13 2020. [Online]. Available: https://www.openstreetmap.org/

[23] German Aerospace Center (DLR), *SUMO: Documentation.* [Online]. Available: https://sumo.dlr.de/docs/SUMO_User_Documentation.html (accessed: Feb. 13 2020).

[24] Google LLC, *Google Street View.* Accessed: Feb. 13 2020. [Online]. Available: https://maps.google.com/

[25] M. Pourabdollah, E. Bjärkvik, F. Furer, B. Lindenberg, and K. Burgdorf, “Calibration and Evaluation of Car Following Models Using Real-World Driving Data,” in *IEEE ITSC 2017: 20th International Conference on Intelligent Transportation Systems*, Yokohama, Kanagawa, Japan, 2017, pp. 1–6.

[26] L. Bieker-Walz, M. Behrisch, M. Junghans, and K. Gimm, “Evaluation of car-following-models at controlled intersections,” in *ESM 2017 European Simulation and Modelling Conference*, Lisbon, Portual, 2017, pp. 247–251.

[27] German Aerospace Center (DLR), *SUMO: Car-Following-Models.* [Online]. Available: https://sumo.dlr.de/docs/Car-Following-Models.html (accessed: Feb. 13 2020).

[28] F. Viti, S. P. Hoogendoorn, H. J. van Zuylen, I. R. Wilmink, and B. van Arem, “Microscopic Data for Analyzing Driving Behavior at Traffic Signals,” in *International Series in Operations Research & Management Science, Traffic Data Collection and its Standardization*, J. Barceló and M. Kuwahara, Eds., New York, NY: Springer-Verlag New York, 2010, pp. 171–191.

[29] M. Liebner, F. Klanner, M. Baumann, C. Ruhhammer, and C. Stiller, "Velocity-Based Driver Intent Inference at Urban Intersections in the Presence of Preceding Vehicles," *IEEE Intell. Transport. Syst. Mag.*, vol. 5, no. 2, pp. 10–21, 2013.

[30] E. Bjärkvik, F. Fürer, M. Pourabdollah, and B. Lindenberg, "Simulation and Characterisation of Traffic on Drive Me Route around Gothenburg using SUMO," in *SUMO User Conference 2017: SUMO 2017 - Towards Simulation for Autonomous Mobility*, Berlin, Germany, 2017.

[31] German Aerospace Center (DLR), *SUMO: Vehicle Type Parameter Defaults.* [Online]. Available: https://sumo.dlr.de/docs/Vehicle_Type_Parameter_Defaults.html (accessed: Feb. 13 2020).

[32] M. Treiber and V. Kanagaraj, "Comparing Numerical Integration Schemes for Time-Continuous Car-Following Models," *Physica A: Statistical Mechanics and its Applications*, vol. 419, pp. 183–195, 2015.

[33] S. Shekhar and H. Xiong, *Encyclopedia of GIS*. Boston, MA: Springer-Verlag Berlin Heidelberg, 2008. Accessed: May 17 2020. [Online]. Available: http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10228810

[34] G. Klunder, *AMITRAN: Assessment Methodologies for ICT in multimodal transport from User Behaviour to CO2 reduction*. ITS World Congress Orlando 19.10.2011. [Online]. Available: http://www.ecomove-project.eu/assets/Documents/Presentations/ITS-Orlando-2011/ITS-World-Orlando2011-SS64-AMITRAN.pdf (accessed: Feb. 13 2020).

[35] J. Wang, K. Dixon, H. Li, and J. Ogle, "Normal Deceleration Behavior of Passenger Vehicles at Stop Sign-Controlled Intersections Evaluated with In-Vehicle Global Positioning System Data," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1937, pp. 120–127, 2005.

[36] G. S. Aoude, V. R. Desaraju, L. H. Stephens, and J. P. How, "Driver Behavior Classification at Intersections and Validation on Large Naturalistic Data Set," *IEEE Trans. Intell. Transport. Syst.*, vol. 13, no. 2, pp. 724–736, 2012.

[37] S. B. Amsalu, A. Homaifar, F. Afghah, S. Ramyar, and A. Kurt, "Driver Behavior Modeling near Intersections Using Support Vector Machines based on Statistical Feature Extraction," in *2015 IEEE Intelligent Vehicles Symposium*, Seoul, Korea, 2015, pp. 1270–1275.

[38] B. Tang, S. Khokhar, and R. Gupta, "Turn Prediction at Generalized Intersections," in *2015 IEEE Intelligent Vehicles Symposium*, Seoul, Korea, 2015, pp. 1399–1404.

[39] A. David and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding," Stanford InfoLab 2006-13, 2006. Accessed: Mar. 12 2020. [Online]. Available: http://ilpubs.stanford.edu/778/

[40] C. M. Bishop, *Pattern Recognition and Machine Learning,* 8th ed. New York, NY: Springer, 2009.

[41] G. van Rossum and F. L. Drake Jr, *Python 3 Reference Manual*: Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[42] J. Schreiber, "pomegranate: Fast and Flexible Probabilistic Modeling in Python," *Journal of Machine Learning Research*, no. 18, pp. 1–6, 2018.

[43] F. Pedregosa *et al.,* "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[44] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 22–30, 2011.

[45] W. McKinney, "Data Structures for Statistical Computing in Python," in *Proceedings of the 9th Python in Science Conference*, Austin, Texas, 2010, pp. 56–61.

[46] K. Greff, A. Klein, M. Chovanec, F. Hutter, and J. Schmidhuber, "The Sacred Infrastructure for Computational Research," in *Proceedings of the 16th Python in Science Conference*, Austin, Texas, Jul. 2017-Jul. 2017, pp. 49–56.

[47] J. Dai and J. Cheng, "HMMEditor: a visual editing tool for profile hidden Markov model," *BMC genomics*, 9 Suppl 1, 8, 2008.

[48] D. Hajek and J. Nouza, "Unhiding Hidden Markov Models by Their Visualization (Application in Speech Processing)," in *Eurographics, Virtual Environments and Scientific Visualization '96: Proceedings of the Eurographics Workshops in Monte Carlo, Monaco, February 19-20, 1996, and in Prague, Czech Republic, April 23-25, 1996*, M. Göbel, J. David, P. Slavik, and J. J. Wijk, Eds., Vienna: Springer, 1996, pp. 277–285.

[49] L. J. Kazmier, *Schaum's Outline of Theory and Problems of Business Statistics,* 4th ed. New York: McGraw-Hill, 2004.

[50] C. Sammut and G. I. Webb, Eds., *Encyclopedia of Machine Learning*. Boston, MA: Springer, 2011.

List of Figures

List of Tables

A. Appendix

A.1. Distance-based Trajectory Representation

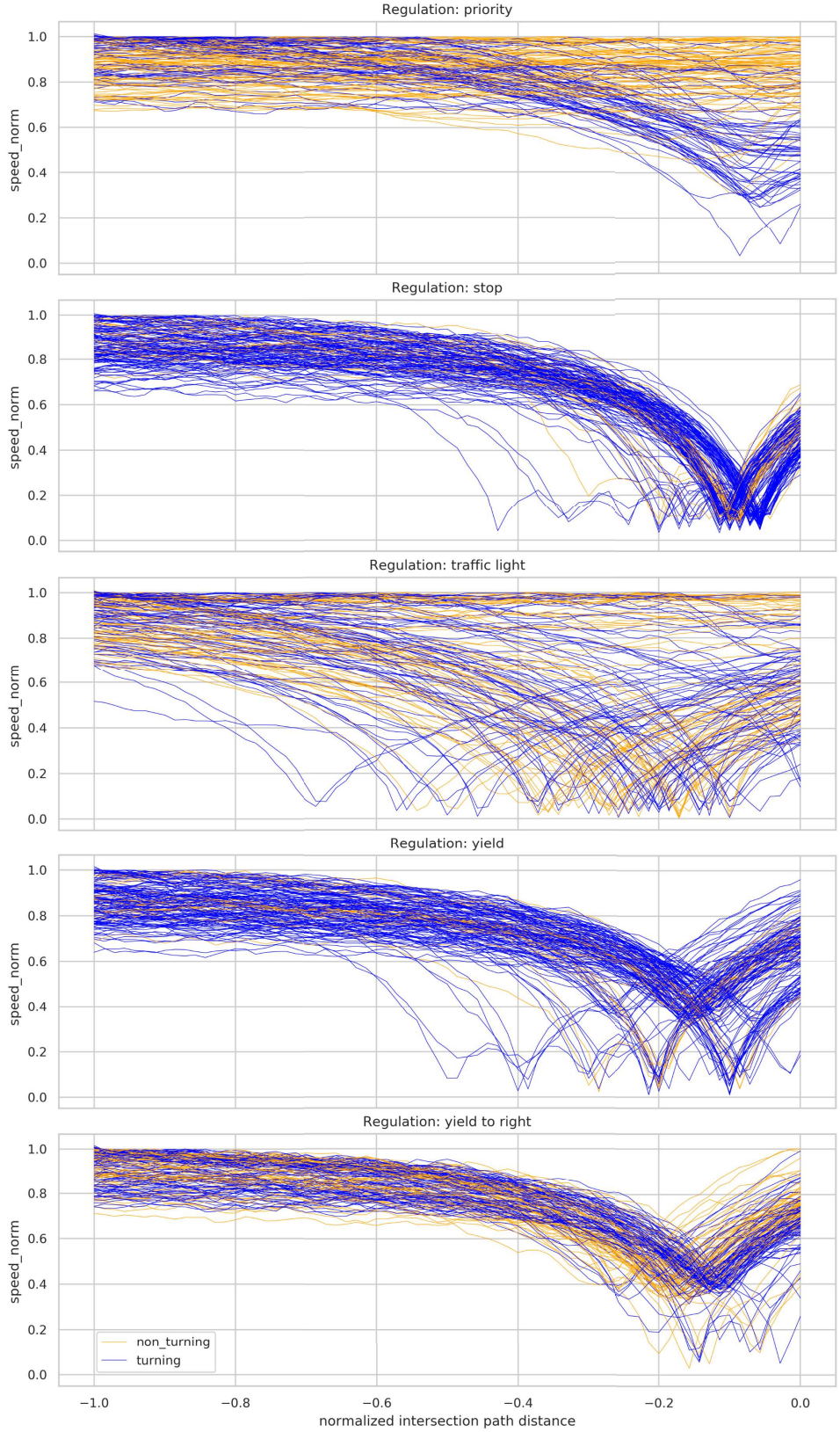


Figure A.1.1: Normalized speed of incoming trajectory segments following differently regulated intersection paths, plotted against the normalized intersection path distance feature. Depicted trajectory segments are resampled to a distance-based representation and have a length of 71 points. Dark yellow lines indicate vehicles going straight, blue lines indicate such executing turning maneuvers. Each subplot shows 150 trajectory segments.

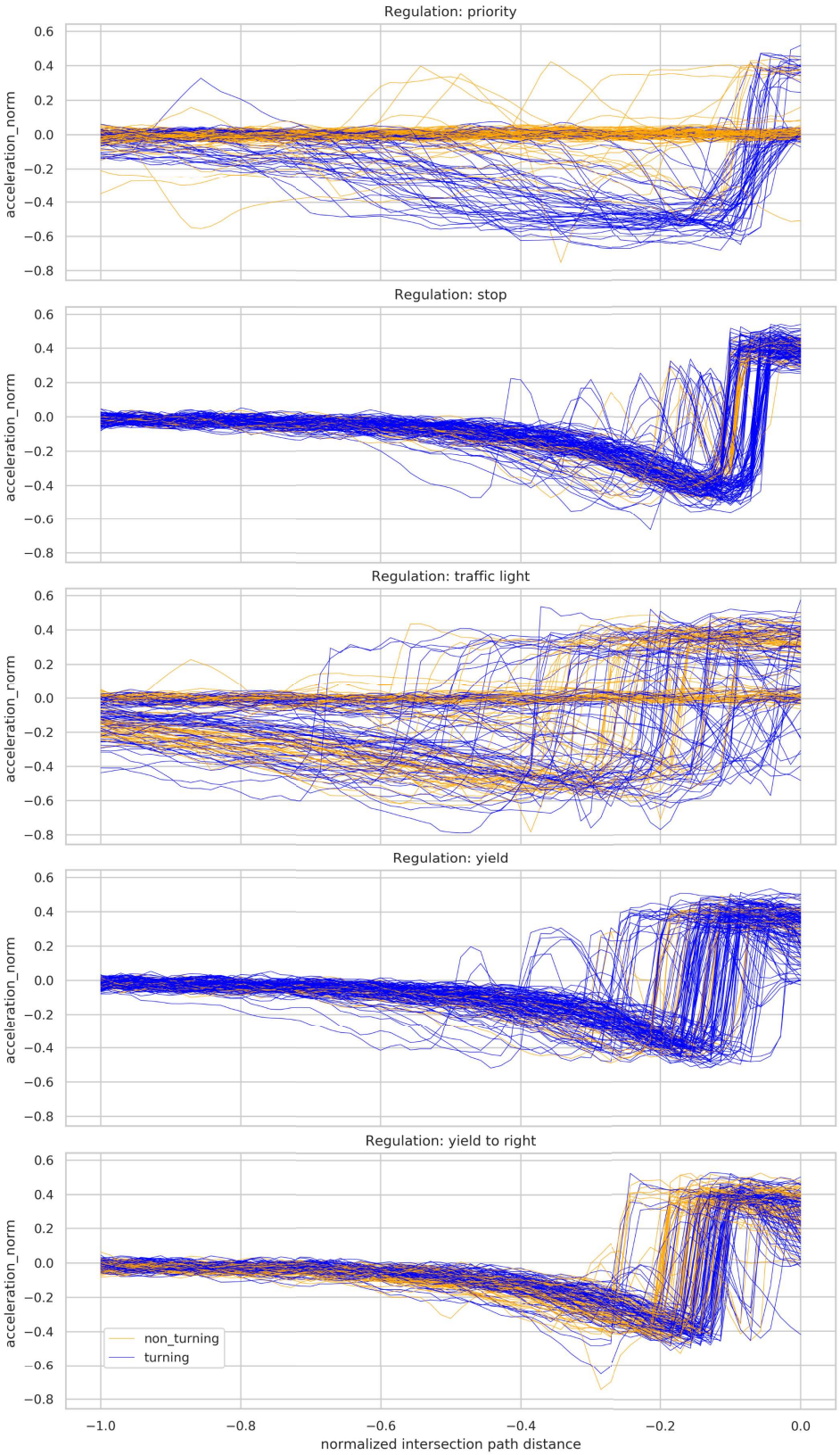


Figure A.1.2: Normalized acceleration of 150 incoming trajectory segments following differently regulated intersection paths, plotted against the normalized intersection path distance feature. Depicted trajectory segments are resampled to a distance-based representation and have a length of 71 points. Dark yellow lines indicate vehicles going straight, blue lines indicate such executing turning maneuvers. Each subplot shows 150 trajectory segments.

A.2. Additional Results of Grid Search Experiment of Hidden-Markov-Model-Based Inference Approach

A.2.1. Score Results

Number of hidden states	Distance-based				Time-based			
	Training score		Test score		Training score		Test score	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
Normalized speed, normalized acceleration, normalized intersection-relative time								
2	0.839	0.031	0.740	0.104	0.805	0.018	0.738	0.066
3	0.858	0.033	0.752	0.094	0.854	0.017	0.805	0.055
4	0.869	0.034	0.795	0.070	0.878	0.037	0.832	0.057
5	0.860	0.050	0.759	0.044	0.889	0.025	0.777	0.080
6	0.860	0.036	0.743	0.102	0.875	0.062	0.813	0.105
7	0.864	0.038	0.786	0.069	0.896	0.018	0.795	0.077
8	0.862	0.027	0.750	0.045	0.905	0.018	0.841	0.071
9	0.858	0.057	0.763	0.027	0.907	0.009	0.829	0.082
10	0.850	0.031	0.761	0.132	0.898	0.025	0.828	0.066
Normalized speed, normalized acceleration								
2	0.620	0.188	0.526	0.171	0.692	0.091	0.709	0.153
3	0.794	0.059	0.703	0.140	0.786	0.037	0.755	0.079
4	0.798	0.043	0.699	0.167	0.867	0.027	0.805	0.060
5	0.805	0.052	0.702	0.144	0.851	0.064	0.771	0.025
6	0.803	0.063	0.739	0.101	0.810	0.098	0.741	0.079
7	0.817	0.052	0.754	0.109	0.865	0.037	0.784	0.049
8	0.811	0.029	0.759	0.115	0.818	0.073	0.801	0.122
9	0.752	0.026	0.652	0.149	0.858	0.011	0.822	0.123
10	0.781	0.048	0.706	0.121	0.859	0.026	0.845	0.052
Normalized speed, normalized intersection-relative time								
2	0.741	0.027	0.650	0.081	0.772	0.019	0.730	0.045
3	0.753	0.020	0.679	0.062	0.736	0.055	0.711	0.048
4	0.714	0.059	0.687	0.045	0.763	0.024	0.703	0.040
5	0.764	0.051	0.697	0.077	0.760	0.011	0.688	0.085
6	0.744	0.076	0.705	0.077	0.791	0.042	0.690	0.066
7	0.747	0.043	0.674	0.066	0.789	0.034	0.729	0.078
8	0.752	0.026	0.686	0.061	0.803	0.049	0.720	0.041
9	0.741	0.024	0.674	0.064	0.800	0.032	0.736	0.055
10	0.773	0.020	0.699	0.047	0.812	0.031	0.730	0.109

Number of hidden states	Distance-based				Time-based			
	Training score		Test score		Training score		Test score	
	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation	Mean	Standard Deviation
Normalized speed								
2	0.694	0.046	0.583	0.143	0.739	0.020	0.744	0.047
3	0.717	0.045	0.647	0.128	0.645	0.061	0.603	0.064
4	0.714	0.034	0.675	0.124	0.692	0.014	0.631	0.040
5	0.724	0.046	0.656	0.086	0.736	0.036	0.671	0.025
6	0.722	0.033	0.658	0.118	0.746	0.038	0.680	0.113
7	0.720	0.038	0.658	0.120	0.768	0.043	0.691	0.105
8	0.723	0.033	0.665	0.103	0.697	0.120	0.638	0.161
9	0.714	0.033	0.647	0.120	0.722	0.114	0.603	0.122
10	0.718	0.042	0.664	0.114	0.535	0.074	0.451	0.112

Table A.2.1: Mean and standard deviation of F_1 scores achieved in the context of the grid search experiment. In this experiment, several parameter combinations specifying trajectory representation and Hidden Markov Model training characteristics are examined. Cross-validation is performed on each examined parameter combination. Therefore, mean and standard deviation are calculated from individual F_1 scores achieved by a set of experiment runs cross-validating the same parameter combination. In all instances, resampled trajectory data is used. Inference is conducted on both training and test intersection paths using parametrized Hidden Markov Models. Thus, scores are organized in the columns named training score and test score, respectively. Each row of the table represents scores achieved in the context of two different types reference axes being used. The left half of each row refers to experiment runs in which trajectories are resampled to a distance-based representation. The right half refers to experiment runs in which trajectories are resampled to a time-based representation. The subheadings indicate the features used to describe trajectories in experiment runs. The leftmost column provides the number of states set for Hidden-Markov-Model parametrization.

Number of hidden states	Training score		Test score	
	Mean	Standard Deviation	Mean	Standard Deviation
Normalized speed, normalized acceleration, normalized intersection-relative time				
2	0.466	0.136	0.421	0.090
3	0.219	0.172	0.240	0.152
4	0.307	0.345	0.309	0.315
5	0.371	0.330	0.418	0.353
6	0.531	0.280	0.500	0.254
7	0.492	0.349	0.456	0.249
8	0.384	0.229	0.341	0.193
9	0.288	0.339	0.222	0.195
10	0.185	0.184	0.120	0.118
Normalized speed, normalized acceleration				
2	0.745	0.114	0.593	0.143
3	0.786	0.047	0.660	0.073
4	0.656	0.134	0.584	0.114
5	0.752	0.082	0.708	0.084
6	0.691	0.064	0.630	0.179
7	0.670	0.081	0.567	0.084
8	0.679	0.094	0.527	0.143
9	0.643	0.198	0.506	0.177
10	0.461	0.073	0.366	0.044

Number of hidden states	Training score		Test score	
	Mean	Standard Deviation	Mean	Standard Deviation
Normalized speed, normalized intersection-relative time				
2	0.711	0.057	0.611	0.197
3	0.706	0.180	0.617	0.196
4	0.680	0.045	0.590	0.108
5	0.603	0.221	0.445	0.152
6	0.571	0.256	0.544	0.223
7	0.554	0.173	0.447	0.117
8	0.586	0.084	0.518	0.063
9	0.594	0.025	0.474	0.118
10	0.592	0.040	0.446	0.155
Normalized speed				
2	0.737	0.068	0.626	0.104
3	0.702	0.039	0.591	0.148
4	0.816	0.051	0.731	0.092
5	0.538	0.034	0.456	0.108
6	0.529	0.053	0.474	0.148
7	0.502	0.063	0.408	0.056
8	0.471	0.060	0.430	0.050
9	0.473	0.034	0.414	0.166
10	0.422	0.014	0.403	0.055

Table A.2.2: Mean and standard deviation of F_1 scores achieved in the context of the grid search experiment. In this experiment, several parameter combinations specifying trajectory representation and Hidden Markov Model training characteristics are examined. Cross-validation is performed on each examined parameter combination. Therefore, mean and standard deviation are calculated from individual F_1 scores achieved by a set of experiment runs cross-validating the same parameter combination. In all instances, trajectory data is used without being resampled. This way, all trajectories are represented with reference to a time-based axis. Inference is conducted on both training and test intersection paths using parametrized Hidden Markov Models. Thus, scores are organized in the columns named training score and test score, respectively. Each row of the table represents scores achieved from examining one parameter combination. The subheadings indicate the features used to describe trajectories in experiment runs. The leftmost column provides the number of states set for Hidden-Markov-Model parametrization.

A.2.2. Parametrized Hidden Markov Models

State number	Mean vector	Covariance Matrix		
1	0.7470	0.0058	-0.0001	-0.0007
	-0.0024	-0.0001	0.0005	0.0000
	-0.7956	-0.0007	0.0000	0.0162
2	0.6907	0.0103	0.0068	-0.0116
	-0.2605	0.0068	0.0360	-0.0149
	-0.5759	-0.0116	-0.0149	0.0345
3	0.7470	0.0057	0.0000	0.0001
	0.0015	0.0000	0.0004	0.0000
	-0.2858	0.0001	0.0000	0.0274
4	0.4432	0.0219	-0.0055	0.0009
	-0.1617	-0.0055	0.0976	0.0209
	-0.1551	0.0009	0.0209	0.0105

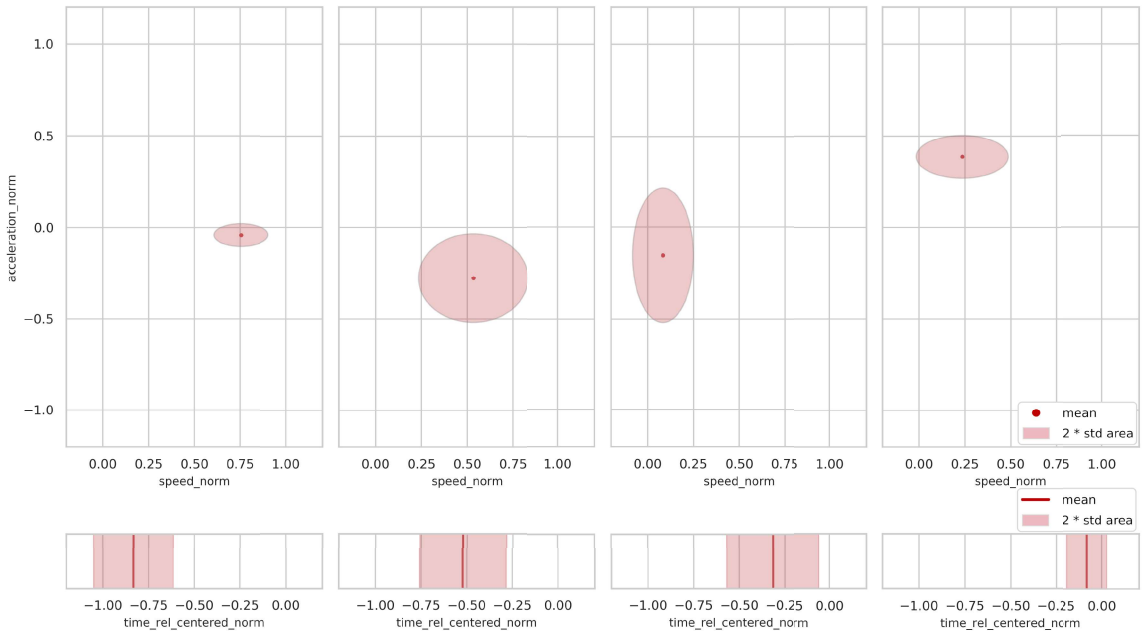
Table A.2.3: Parameter values of the observation densities depicted in Figure 6.5. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

State number	Mean vector	Covariance Matrix		
1	0.7624	0.0056	-0.0002	-0.0035
	-0.0371	-0.0002	0.0009	-0.0020
	-0.7951	-0.0035	-0.0020	0.0164
2	0.2583	0.0577	-0.0187	-0.0278
	-0.1966	-0.0187	0.0314	0.0085
	-0.4211	-0.0278	0.0085	0.0290
3	0.5954	0.0143	0.0093	-0.0127
	-0.2225	0.0093	0.0119	-0.0097
	-0.3875	-0.0127	-0.0097	0.0137
4	0.4498	0.0223	0.0010	0.0048
	0.3152	0.0010	0.0168	0.0015
	-0.0928	0.0048	0.0015	0.0037

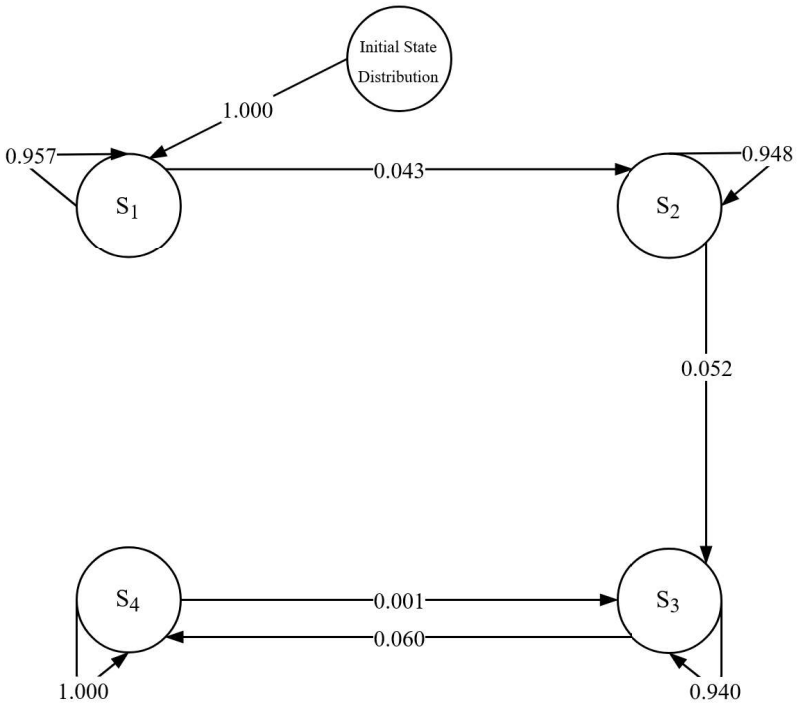
Table A.2.4: Parameter values of the observation densities depicted in subfigure (a) of Figure 6.7. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

State number	Mean vector	Covariance Matrix		
1	0.7741	0.0053	-0.0002	-0.0036
	-0.0382	-0.0002	0.0009	-0.0021
	-0.7944	-0.0036	-0.0021	0.0161
2	0.6003	0.0145	0.0103	-0.0125
	-0.2327	0.0103	0.0124	-0.0102
	-0.3999	-0.0125	-0.0102	0.0129
3	0.2876	0.0355	-0.0125	-0.0098
	-0.0813	-0.0125	0.0569	0.0241
	-0.3168	-0.0098	0.0241	0.0325
4	0.5127	0.0104	-0.0004	0.0043
	0.3545	-0.0004	0.0058	-0.0008
	-0.0862	0.0043	-0.0008	0.0032

Table A.2.5: Parameter values of the observation densities depicted in subfigure (a) of Figure 6.8. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.



(a) Visualization of learned observation densities.

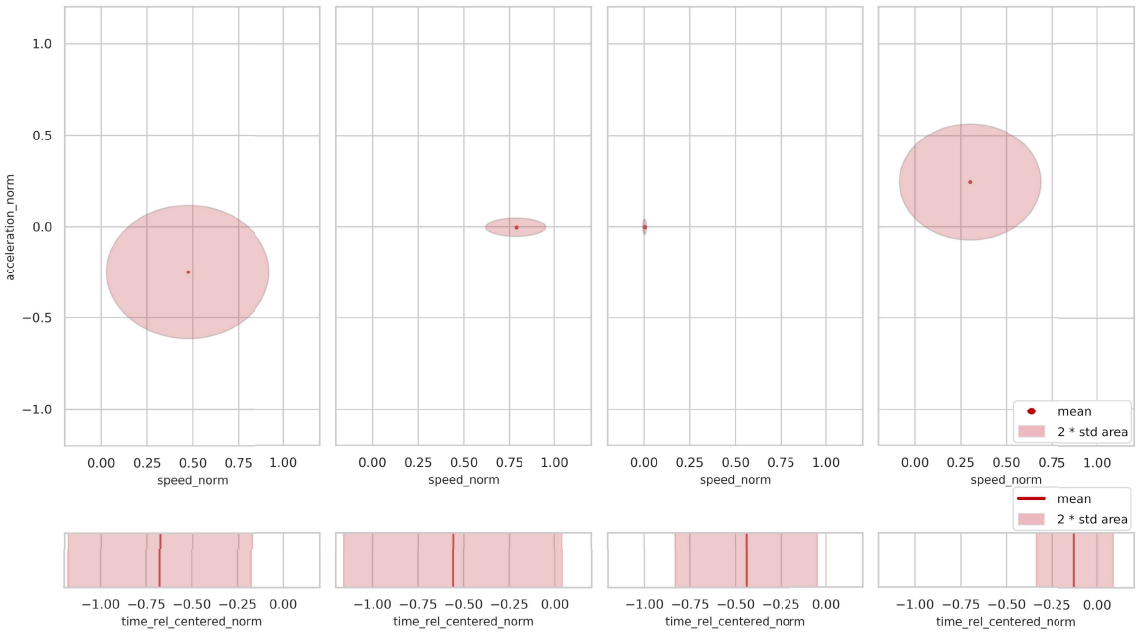


(b) Visualization of learned initial state and state transition probabilities.

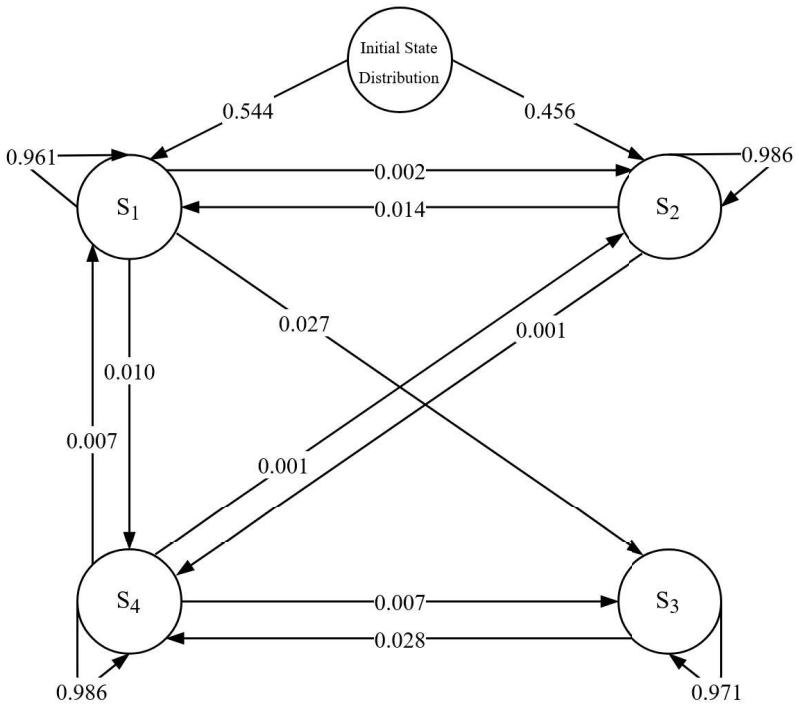
Figure A.2.1:. Visualization of learned observation densities (a) and initial state and state transition probabilities (b) of a four-hidden-state Hidden Markov Model representing the `stop` class. Each column of plots in (a) visualizes a Gaussian in one hidden state. The column order reflects the hidden-state numbering of the model. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. Values in (b) are rounded, hence all of a state's outgoing transition probabilities do not necessarily sum up to 1. The values are printed on arrows illustrating the direction of transitions. The circles labeled S_i with $1 \leq i \leq 4$ mark the model's hidden states.

State number	Mean vector	Covariance Matrix		
1	0.7542	0.0055	-0.0001	-0.0035
	-0.0395	-0.0001	0.0010	-0.0019
	-0.8323	-0.0035	-0.0019	0.0118
2	0.5328	0.0223	0.0146	-0.0138
	-0.2766	0.0146	0.0148	-0.0094
	-0.5214	-0.0138	-0.0094	0.0141
3	0.0836	0.0069	-0.0084	-0.0038
	-0.1536	-0.0084	0.0336	0.0043
	-0.3143	-0.0038	0.0043	0.0163
4	0.2354	0.0157	0.0017	0.0059
	0.3866	0.0017	0.0035	0.0006
	-0.0811	0.0059	0.0006	0.0029

Table A.2.6: Parameter values of the observation densities depicted in subfigure (a) of Figure A.2.1. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.



(a) Visualization of learned observation densities.



(b) Visualization of learned initial state and state transition probabilities.

Figure A.2.2: Visualization of learned observation densities (a) and initial state and state transition probabilities (b) of a four-hidden-state Hidden Markov Model representing the `traffic light` class. Each column of plots in (a) visualizes a Gaussian in one hidden state. The column order reflects the hidden-state numbering of the model. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. Values in (b) are rounded, hence all of a state's outgoing transition probabilities do not necessarily sum up to 1. The values are printed on arrows illustrating the direction of transitions. The circles labeled S_i with $1 \leq i \leq 4$ mark the model's hidden states.

State number	Mean vector	Covariance Matrix		
1	0.4748	0.0500	0.0117	-0.0065
	-0.2488	0.0117	0.0336	-0.0048
	-0.6771	-0.0065	-0.0048	0.0629
2	0.7860	0.007	0.0001	0.0018
	-0.0023	0.0001	0.0006	0.0014
	-0.5574	0.0018	0.0014	0.0895
3	0.0036	0.0000	0.0000	0.0000
	-0.0003	0.0000	0.0004	0.0003
	-0.4391	0.0000	0.0003	0.0384
4	0.3016	0.0374	-0.0054	0.0057
	0.2457	-0.0054	0.0254	0.0056
	-0.1243	0.0057	0.0056	0.0112

Table A.2.7: Parameter values of the observation densities depicted in subfigure (a) of Figure A.2.2. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

State number	Mean vector	Covariance Matrix		
1	0.7684	0.0057	-0.0004	-0.0016
	-0.0288	-0.0004	0.0007	-0.0006
	-0.8725	-0.0016	-0.0006	0.0067
2	0.7062	0.0061	0.0009	-0.0059
	-0.0952	0.0009	0.0027	-0.0037
	-0.5663	-0.0059	-0.0037	0.0123
3	0.4933	0.0189	0.0079	-0.0071
	-0.3109	0.0079	0.0084	-0.0054
	-0.4171	-0.0071	-0.0054	0.0312
4	0.3177	0.0556	0.0318	0.0295
	0.1950	0.0318	0.0454	0.0244
	-0.1974	0.0295	0.0244	0.0303

Table A.2.8: Parameter values of the observation densities depicted in Figure 6.9. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

State number	Mean vector	Covariance Matrix		
1	0.7468	0.0061	0.0000	0.0000
	-0.0032	0.0000	0.0006	0.0000
	-0.9271	0.0000	0.0000	0.0022
2	0.7411	0.0056	-0.0001	-0.0001
	-0.0025	-0.0001	0.0005	0.0000
	-0.7585	-0.0001	0.0000	0.0029
3	0.7135	0.0091	0.0013	-0.0104
	-0.1510	0.0013	0.0175	-0.0006
	-0.6404	-0.0104	-0.0006	0.0349
4	0.7361	0.0060	0.0000	0.0001
	-0.0001	0.0000	0.0004	-0.0001
	-0.5659	0.0001	-0.0001	0.0039
5	0.5584	0.0173	0.0002	-0.0112
	-0.4764	0.0002	0.0039	-0.0014
	-0.3783	-0.0112	-0.0014	0.0174
6	0.7402	0.0059	0.0000	-0.0001
	0.0015	0.0000	0.0004	-0.0001
	-0.3459	-0.0001	-0.0001	0.0046
7	0.4324	0.0267	-0.0072	0.0024
	-0.0204	-0.0072	0.0784	0.0098
	-0.1140	0.0024	0.0098	0.0070
8	0.7405	0.0056	0.0000	0.0000
	0.0018	0.0000	0.0004	0.0000
	-0.1131	0.0000	0.0000	0.0050

Table A.2.9: Parameter values of the observation densities depicted in Figure 6.10. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

A.3. Additional Results of Experiment using an Increased Trajectory Data Volume

State number	Mean vector	Covariance Matrix		
1	0.7467	0.0059	0.0000	-0.0007
	-0.0025	0.0000	0.0005	0.0000
	-0.7912	-0.0007	0.0000	0.0168
2	0.6918	0.0106	0.0071	-0.0118
	-0.2440	0.0071	0.0355	-0.0144
	-0.5856	-0.0118	-0.0144	0.0348
3	0.7475	0.0057	0.0000	-0.0001
	0.0015	0.0000	0.0004	-0.0001
	-0.2800	-0.0001	-0.0001	0.0267
4	0.4326	0.0255	-0.0064	0.0010
	-0.1583	-0.0064	0.1016	0.0220
	-0.1589	0.0010	0.0220	0.0110

Table A.3.1: Parameter values of the observation densities depicted in subfigure (a) of Figure 6.12. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

State number	Mean vector	Covariance Matrix		
1	0.7603	0.0054	-0.0002	-0.0033
	-0.0374	-0.0002	0.0009	-0.0018
	-0.8045	-0.0033	-0.0018	0.0151
2	0.2513	0.0493	-0.0177	-0.0195
	-0.1718	-0.0177	0.0335	0.0116
	-0.4010	-0.0195	0.0116	0.0320
3	0.6002	0.0137	0.0099	-0.0125
	-0.2202	0.0099	0.0125	-0.0105
	-0.3936	-0.0125	-0.0105	0.0142
4	0.4478	0.0230	-0.0004	0.0049
	0.3417	-0.0004	0.0092	-0.0006
	-0.0889	0.0049	-0.0006	0.0035

Table A.3.2: Parameter values of the observation densities depicted in subfigure (b) of Figure 6.12. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

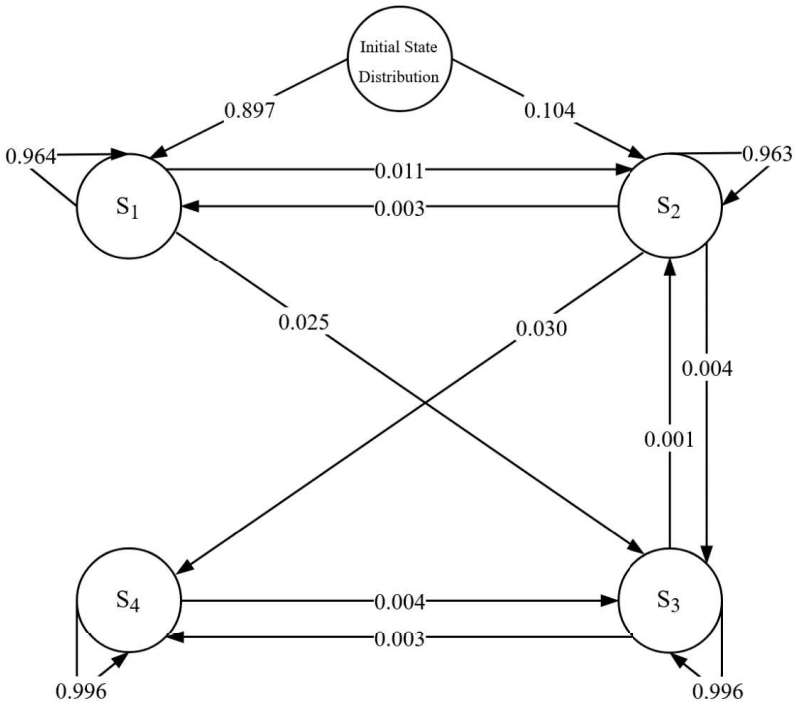


Figure A.3.1: Visualization of the learned initial state and state transition probabilities of a four-state Hidden Markov Model representing the `priority` class. Values are rounded, hence all of a state's outgoing transition probabilities do not necessarily sum up to 1. The probabilities are printed on arrows illustrating the direction of transitions. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. The circles labeled S_i, with $1 \leq i \leq 4$, mark the model's hidden states.

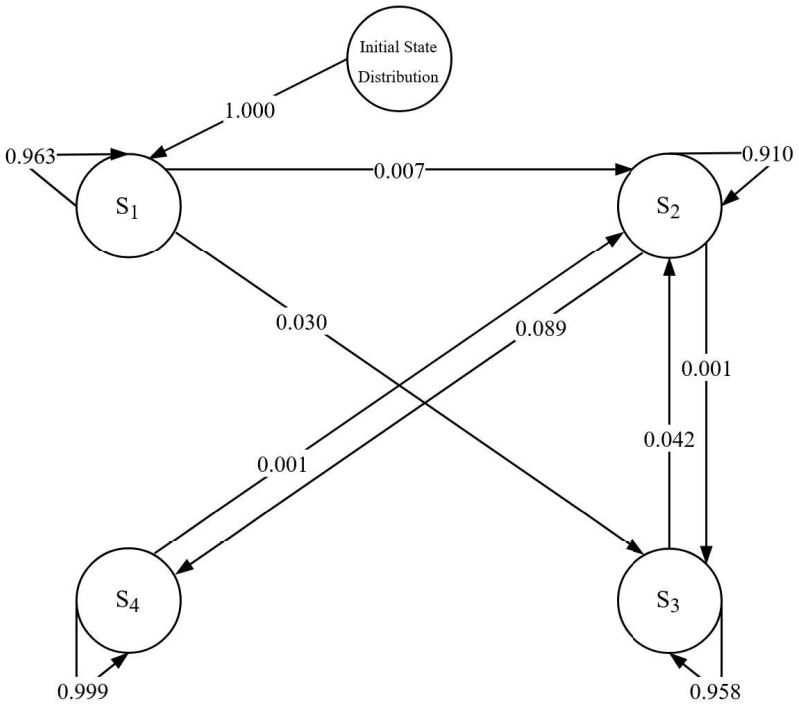


Figure A.3.2: Visualization of the learned initial state and state transition probabilities of a four-state Hidden Markov Model representing the `yield` class. Values are rounded, hence all of a state's outgoing transition probabilities do not necessarily sum up to 1. The probabilities are printed on arrows illustrating the direction of transitions. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. The circles labeled S_i, with $1 \leq i \leq 4$, mark the model's hidden states.

State number	Mean vector	Covariance Matrix		
1	0.7601	0.0054	-0.0002	-0.0032
	-0.0379	-0.0002	0.0009	-0.0018
	-0.8077	-0.0032	-0.0018	0.0149
2	0.2291	0.0536	-0.0210	-0.0239
	-0.1734	-0.0210	0.0328	0.0093
	-0.4481	-0.0239	0.0093	0.0328
3	0.5943	0.0148	0.0095	-0.0132
	-0.2201	0.0095	0.0124	-0.0100
	-0.4025	-0.0132	-0.0100	0.0149
4	0.4401	0.0255	0.0016	0.0060
	0.3261	0.0016	0.0137	0.0008
	-0.0975	0.0060	0.0008	0.0043

Table A.3.3: Parameter values of the observation densities depicted in Figure 6.13. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

A.4. Additional Results of Experiment Counteracting False Classifications of Priority Intersection Paths

State number	Mean vector	Covariance Matrix		
1	0.7465	0.0059	0.0001	0.0000
	-0.0019	0.0001	0.0006	0.0000
	-0.8649	0.0000	0.0000	0.0070
2	0.7410	0.0057	0.0000	0.0003
	0.0002	0.0000	0.0005	0.0001
	-0.5568	0.0003	0.0001	0.0100
3	0.5963	0.0202	-0.0033	-0.0218
	-0.2654	-0.0033	0.0574	0.0076
	-0.3668	-0.0218	0.0076	0.0570
4	0.7426	0.0058	0.0000	-0.0003
	0.0014	0.0000	0.0004	0.0000
	-0.1887	-0.0003	0.0000	0.0130

Table A.4.1: Parameter values of the observation densities depicted in Figure 6.24. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

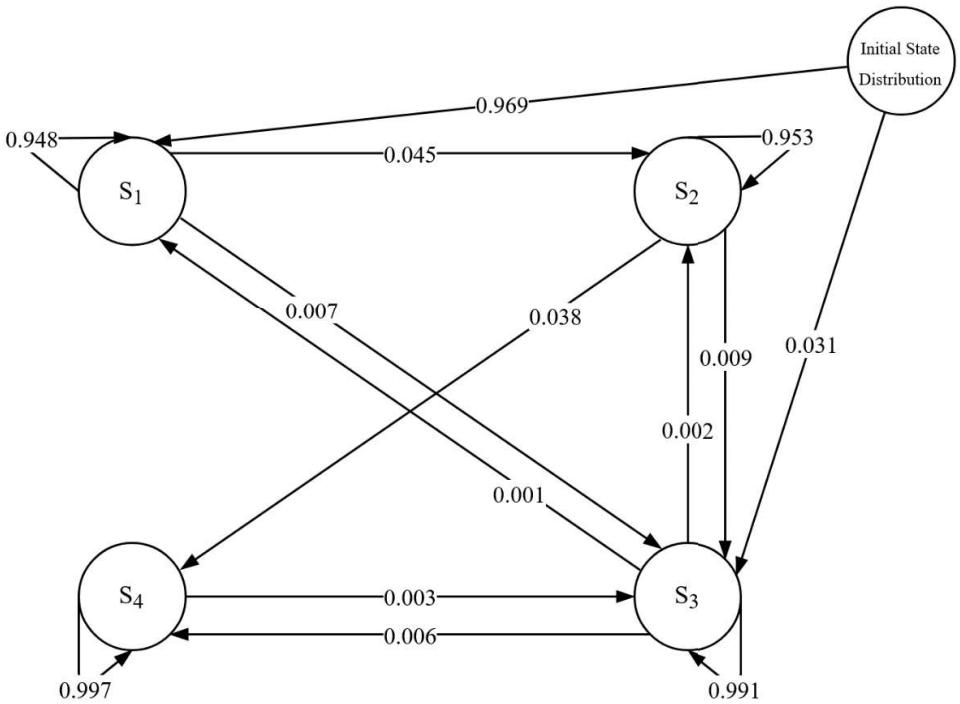


Figure A.4.1: Visualization of the learned initial state and state transition probabilities of a four-state Hidden Markov Model representing the class which covers `prioritized` intersection paths that are followed by vehicles not having to give way to others. Values are rounded, hence all of a state's outgoing transition probabilities do not necessarily sum up to 1. The probabilities are printed on arrows illustrating the direction of transitions. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. The circles labeled S_i, with $1 \leq i \leq 4$, mark the model's hidden states.

State number	Mean vector	Covariance Matrix		
1	0.8070	0.0042	-0.0012	-0.0026
	-0.1336	-0.0012	0.0025	-0.0018
	-0.8724	-0.0026	-0.0018	0.0066
2	0.6713	0.0063	0.0028	-0.0073
	-0.2996	0.0028	0.0068	-0.0054
	-0.5858	-0.0073	-0.0054	0.0110
3	0.4046	0.0104	0.0013	-0.0092
	-0.4874	0.0013	0.0034	-0.0023
	-0.3263	-0.0092	-0.0023	0.0147
4	0.2397	0.0131	0.0174	0.0065
	0.1926	0.0174	0.0764	0.0215
	-0.1128	0.0065	0.0215	0.0105

Table A.4.2: Parameter values of the observation densities depicted in subfigure (a) of Figure 6.25. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

State number	Mean vector	Covariance Matrix		
1	0.7808	0.0054	-0.0004	-0.0018
	-0.0220	-0.0004	0.0005	-0.0006
	-0.8522	-0.0018	-0.0006	0.0084
2	0.7279	0.0054	0.0003	-0.0052
	-0.0795	0.0003	0.0019	-0.0026
	-0.5414	-0.0052	-0.0026	0.0112
3	0.5266	0.0133	0.0073	-0.0078
	-0.2997	0.0073	0.0083	-0.0052
	-0.2877	-0.0078	-0.0052	0.0121
4	0.3395	0.0284	0.0251	0.0098
	0.2252	0.0251	0.0604	0.0164
	-0.0934	0.0098	0.0164	0.0074

Table A.4.3: Parameter values of the observation densities depicted in subfigure (b) of Figure 6.25. Each row of the table provides the mean vector and the covariance matrix of the observation density in a certain state of the corresponding Hidden Markov Model.

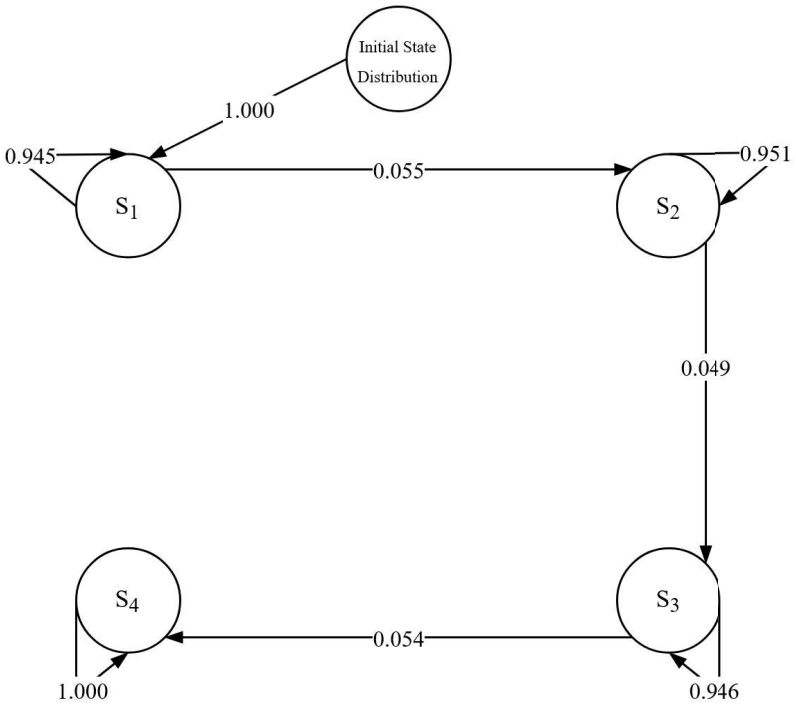


Figure A.4.2: Visualization of the learned initial state and state transition probabilities of a four-state Hidden Markov Model representing a class covering intersection paths followed by vehicles that potentially have to give way to vehicles following other `prioritized` ones. These vehicles approach intersections at comparatively high speeds. Values are rounded, hence all of a state's outgoing transition probabilities do not necessarily sum up to 1. The probabilities are printed on arrows illustrating the direction of transitions. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. The circles labeled S_i, with $1 \leq i \leq 4$, mark the model's hidden states.

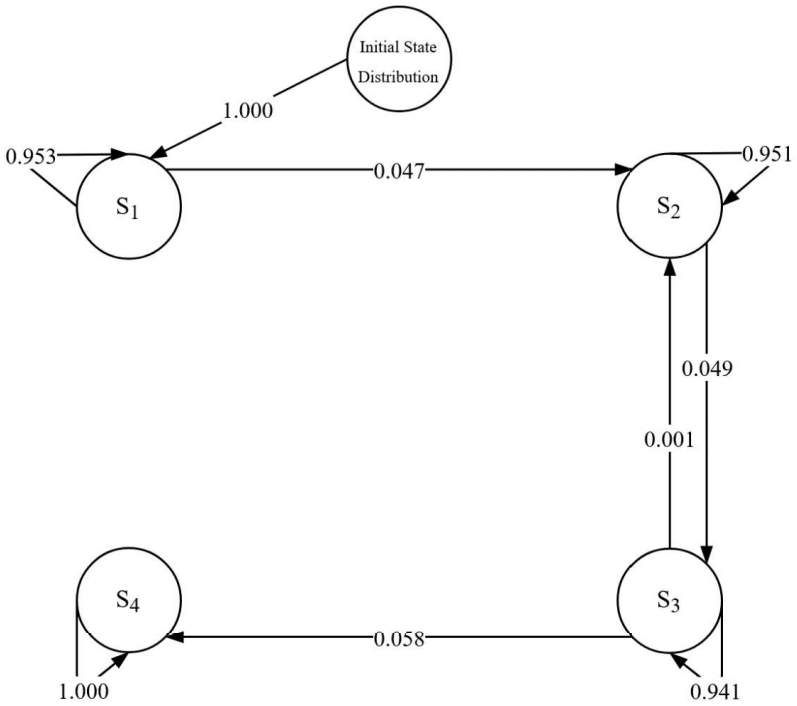


Figure A.4.3: Visualization of the learned initial state and state transition probabilities of a four-state Hidden Markov Model representing a class covering intersection paths followed by vehicles that potentially have to give way to vehicles following other `prioritized` ones. These vehicles approach intersections at comparatively low speeds. Values are rounded, hence all of a state's outgoing transition probabilities do not necessarily sum up to 1. The probabilities are printed on arrows illustrating the direction of transitions. The model is parametrized during the cross-validation of a parameter combination determining the use of trajectories resampled to a time-based representation and described by the feature tuple $(\tilde{v}, \tilde{a}, \tilde{t}^{Int})$. The circles labeled S_i, with $1 \leq i \leq 4$, mark the model's hidden states.

B. DVD with Data Creation Basis and Program Code