

---

# Utilization of a Hidden-Markov-Model for the Prediction of Lane Change Maneuvers

---

Ein Hidden-Markov-Modell zur Vorhersage von Spurwechselmanövern

Master-Thesis von Fabian Faller

Tag der Einreichung:

1. Gutachten: Prof. Johannes Fürnkranz
2. Gutachten: Hien Dang, M.Sc.; Dr. Frederik Janssen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Computer Science Department  
Knowledge Engineering Group

Utilization of a Hidden-Markov-Model for the Prediction of Lane Change Maneuvers  
Ein Hidden-Markov-Modell zur Vorhersage von Spurwechselmanövern

Vorgelegte Master-Thesis von Fabian Faller

1. Gutachten: Prof. Johannes Fürnkranz
2. Gutachten: Hien Dang, M.Sc.; Dr. Frederik Janssen

Tag der Einreichung:

---

## Erklärung zur Master-Thesis

---

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 08.03.2016

---

(Fabian Faller)

---

## Abstract

---

The development of advanced driver assistance systems (ADAS) has been in the focus of many car manufacturers (OEMs) and technology suppliers for the last couple of years. One of the most popular assistance systems are lane keeping assists or lane departure warning systems which have already found their way into mid- to high-level serial cars.

Current systems mainly rely on external features, like distance to lane marking or time to lane crossing (TTC), and do not take the driver's intention into account. Therefore these systems are only able to generate a warning or perform an intervention very shortly before the car is crossing the lane marking, which leads to a short advance warning time for the driver and substantial intervention forces in case of active systems.

In order to maximize the advance warning time and minimize potential interventions, the utilization of machine learning techniques to predict lane change maneuvers has moved into focus of many researchers and is also investigated in this work.

It evaluates the application of a Hidden-Markov-Model for the prediction of lane change maneuver based on the inference of the driver's intention and examines the impact of combining external features and control inputs with observations of the driver's behavior.

---

## Contents

---

<b>1. Introduction</b>	<b>5</b>
1.1. Motivation . . . . .	5
1.2. Goals of this Work . . . . .	6
1.3. Structure of the Thesis . . . . .	6
<b>2. Theoretical Background</b>	<b>7</b>
2.1. Machine Learning . . . . .	7
2.1.1. Supervised Learning . . . . .	7
2.1.2. Model Parameter Estimation . . . . .	8
2.1.3. Feature Selection . . . . .	10
2.2. Hidden-Markov-Models . . . . .	12
2.2.1. Model Parameters . . . . .	12
2.2.2. Parameter Estimation . . . . .	13
2.3. Measuring a Model's Performance . . . . .	14
2.3.1. Precision, Recall & F1 . . . . .	15
2.4. Baseline HMM . . . . .	15
<b>3. Experimental Setup and Data Set</b>	<b>17</b>
3.1. The Driving Simulator . . . . .	17
3.2. The Data Set . . . . .	18
3.2.1. Coarse Feature Pre-Selection . . . . .	18
3.2.2. Coordinate Systems . . . . .	20
3.2.3. Driver Variance . . . . .	21
3.2.4. Anomalies . . . . .	22
3.3. Real World Feasibility . . . . .	23
<b>4. Related Work</b>	<b>25</b>
<b>5. Methodology</b>	<b>27</b>
5.1. Labeling the Driver's Intention . . . . .	27
5.2. Feature Pre-processing . . . . .	28
5.3. Feature Selection . . . . .	30
5.3.1. Pre-Selection . . . . .	30
5.3.2. Final Selection . . . . .	31
5.4. The Driver Intention Model . . . . .	32
5.4.1. Training the Model . . . . .	32
5.4.2. State Selection . . . . .	34
5.4.3. Applied Model Variations . . . . .	35
5.5. Inferring the Lane Change Intention . . . . .	36
5.6. Visualization of Results . . . . .	37

<b>6. Software/Implementation</b>	<b>39</b>
6.1. Probabilistic Modeling Toolkit . . . . .	39
6.2. Architecture . . . . .	39
6.2.1. Data Processing . . . . .	40
6.2.2. Model Training . . . . .	40
6.2.3. Evaluation . . . . .	40
6.2.4. Executables . . . . .	40
<b>7. Results of Feature- and State Selection</b>	<b>43</b>
7.1. Feature Pre-Selection . . . . .	43
7.2. Feature Selection . . . . .	43
7.3. State Selection . . . . .	46
<b>8. Influence of Features and Feature Categories</b>	<b>47</b>
8.1. Exterior Features . . . . .	47
8.2. Control Inputs . . . . .	48
8.3. Head- and Gaze Features . . . . .	49
8.4. Vehicle Dynamics . . . . .	50
<b>9. Evaluation of the Driver Intention Model</b>	<b>51</b>
9.1. Diver Intention Model vs. Baseline Model . . . . .	51
9.2. Cropped Records vs. Entire Records . . . . .	52
9.3. Gaussian Model vs. Gaussian/Discrete Model . . . . .	53
9.4. Driver Specific Training vs. Driver Independent Training . . . . .	53
9.5. Visualization of Prominent Results . . . . .	54
<b>10. Conclusion and Future Work</b>	<b>57</b>
10.1. Conclusion . . . . .	57
10.2. Future Work . . . . .	59
<b>Appendices</b>	<b>60</b>
<b>A. Feature Selection</b>	<b>61</b>
A.1. Pre-Selection . . . . .	61
A.2. Gaussian Model . . . . .	63
A.3. Gaussian/Discrete Model . . . . .	64
A.4. Alternative Selection process . . . . .	65
<b>B. State Selection</b>	<b>66</b>
<b>C. Model Variations</b>	<b>67</b>
C.1. Baseline Model . . . . .	67
C.2. Gaussian Model . . . . .	67
C.3. Gaussian/Discrete Model . . . . .	68

---

# 1 Introduction

---

## 1.1 Motivation

---

Increasing road safety has always been one of the most important goals since the mass distribution of cars. In the last decades, a lot of progress has been achieved and safety systems like airbags, anti-lock braking systems (ABS) or electronic stability control (ESC) have become integral parts of the automobile.

In recent years, the focus moved to advanced driver assistance systems (ADAS) which are supposed to further increase the vehicle safety by perceiving the driving context and support the driver in critical situations. Depending on the system, the provided assistance is either limited to a warning in case of a critical situation or includes active manipulation of the vehicle's controls. Prominent examples for current ADAS are lane departure warning (LDW) systems, lane keeping assistance systems (LKAS) or adaptive cruise control (ACC).

While current serial systems solely focus on the vehicle's dynamics, environment and control inputs, the recent focus of many researchers has been the incorporation of the driver's behavior into such systems. Monitoring the driver's actions and inferring the driving intention is a promising approach for the development of predictive assistance systems, especially in the area of LDW/LKAS.

By inferring the driver's lane change intention, such systems have the ability to predict certain driving maneuvers before they are actually executed and therefore are capable of generating warnings or adapting the vehicle dynamics at a much earlier point in time than current systems. As a result, the driver's ahead warning time can be significantly extended and the magnitude of possibly crucial control inputs stays relatively small.

Well performing predictive systems therefore have a high potential for increasing highway safety and reducing the amount of lane-change related accidents.

A promising approach for the development of such types of predictive systems is the utilization of machine learning techniques which can be used to recognize patterns in driving data and predict future maneuvers based on these patterns. In contrast to rule-based approaches, they are able to extract the required information directly from data and therefore require less human assumptions. Additionally to that, they might be able to recognize certain patterns that a human developer would never have found.

In this work, the widely used Hidden-Markov-Model is utilized for inferring the driver's lane change intention and predicting lane change maneuvers. The model is trained and evaluated on a substantial set of simulator data, containing a total of 23 hours of driving time, collected from a diverse selection of drivers.

---

## 1.2 Goals of this Work

---

The main goal of this thesis is the development and evaluation of a lane change prediction system, based on a Hidden-Markov-Model and the driver's lane change intention. This includes the following milestones:

- Finding a reasonable intention indicator and developing an automated labeling process.
- Determining a suitable feature selection and model configuration for the intention inference.
- Finding a meaningful evaluation method to assess the model's performance.
- Analyzing the influence of individual features and feature classes on the model's performance.

---

## 1.3 Structure of the Thesis

---

This thesis starts with a general introduction into machine learning and the theory behind the utilized models and methods in Chapter 2. After that, the experimental setup and the data set is described in Chapter 3, followed by an overview of related work in Chapter 4.

Now that the theory and application domain is introduced, the model creation process is presented in Chapter 5 followed by a description of the software, which has been developed to execute the described process, in Chapter 6.

Chapters 7,8 and 9 provide a detailed evaluation of the individual steps of the model creation, evaluate the influence of particular features and assess the performance of the final model.

In the end, Chapter 10 recapitulates the discoveries of this work and provides some impulses for possible future activities.



---

## 2 Theoretical Background

---

Since this work heavily utilizes models and techniques from the area of machine learning, a brief introduction into the used models and learning methods is given in this chapter. It starts with a brief general introduction into supervised machine learning and afterwards gives a detailed explanation of the specific model and learning methods, utilized in this work.

---

### 2.1 Machine Learning

---

The term machine learning is associated with a huge variety of models and approaches. In general, they can be classified into three main areas, known as supervised learning, unsupervised learning and reinforcement learning.

According to [19] the corresponding learning tasks are defined as follows:

In supervised learning, the available data set consists of  $N$  samples of input/output pairs  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$  where each  $y_j$  is generated by the unknown function  $y = f(x)$  and a function  $h$  (hypothesis) has to be found which approximates the true function  $f$ .

It is called supervised since the associated output value of the true function  $y_j = f(x)$  is given for each available input value  $x_j$  and can therefore be used in the training process.

An unsupervised learning task is somewhat similar, the training data, however, only consists of the training samples  $x_1, x_2, \dots, x_N$  and the correct output values  $y_1, y_2, \dots, y_N$  are unknown. The training process therefore can't resort to the output values and needs to focus on finding patterns in the input data exclusively.

A prominent example for unsupervised learning are clustering algorithms that obtain information about the data by identifying concentrations of input values.

Reinforcement Learning is an iterative learning process that usually starts with a coarse demonstration (e.g. of a desired action) and successively refines this demonstration by iteratively adapting the systems behavior based on the outcome of a reward function which is used to evaluate the changes, made in an iteration.

A prominent example for this approach can be found in the area of robotics. E.g. a certain action can be trained to a robot by iteratively refining a coarse, kinesthetically taught trajectory.

---

#### 2.1.1 Supervised Learning

---

In this work, the task of predicting upcoming lane changes is formulated as a supervised learning problem. Certain aspects from these types of models are therefore furtherly introduced in this section.

#### Regression & Classification

Supervised learning can generally be categorized into regression and classification tasks, depending on the valuation of the output value  $y$ .

In a regression task, the output typically has numerical valuation while a classification task is

---

based on a finite set of discrete categorizations/classifications.

E.g. a classical regression task would be predicting the progression of the stock market, where future values of certain company shares are supposed to be predicted based on current and past stock prices. A typical classification example would be the determination of a person's gender given a portrait photograph.

### Generative vs. Discriminative Models

Another discrimination can be made by the type of probability distribution which is defined by a model. While generative models define a joint probability distribution  $p(x, y) = p(y | x)p(x)$  over the inputs ( $x$ ) and outputs ( $y$ ), discriminative models only define the conditional probability distribution  $p(y | x)$  [19].

Due to the definition of  $p(x, y)$ , generative models have the ability to generate artificial samples which can be used for evaluating the performance of other models. The advantage of discriminative models is the lack of prior probabilities  $p(x)$  which do not need to be modeled.

In general, discriminative models are therefore better suited for small training sets as certain dependencies do not need to be modeled. To reduce the model's complexity and therefore amount of required training data generative models often define certain independence assumptions that simplify the joint probability distribution.

---

#### 2.1.2 Model Parameter Estimation

---

Fitting a particular model to a given set of training data  $\mathbf{x}$  usually requires finding a suitable assignment of the model's parameters  $\theta$  and the choice of the model's complexity. This section introduces the most common parameter estimation techniques and discusses the general aspects of a model's complexity.

---

#### Statistical Learning Methods

---

Statistical learning methods come into play when it comes to finding a parameter estimation for a particular model. They determine a suitable assignment  $\theta$  through maximization of the model's likelihood function  $f(\mathbf{x} | \theta)$  for a given  $\mathbf{x}$  with respect to  $\theta$ .

#### Maximum-a-posteriori (MAP) and Maximum-Likelihood (ML)

The parameter estimation process for models with observable parameters is usually based on the determination of the maximum-a-posteriori or maximum-likelihood estimate.

In cases where prior knowledge about the model's parameters exists in form of a probability distribution  $g(\theta)$ , the maximum-a-posteriori estimate can be determined by:

$$\hat{\theta}_{MAP} = \arg \max_{\theta} f(\mathbf{x} | \theta)g(\theta)$$

The maximum-likelihood estimate is strongly related to the MAP estimate. It is used in cases where no prior knowledge is available and is defined as:

$$\hat{\theta}_{ML} = \arg \max_{\theta} f(\mathbf{x} | \theta)$$

---

## Expectation Maximization (EM)

As denoted in [3], a powerful way of finding maximum-likelihood-solutions, in cases of incomplete data or latent variables, is the Expectation-Maximization(EM)-algorithm [6]. It is an iterative approach starting with an initial estimation of the model's parameters and alternates between two consecutive steps (E step and M step) until convergence. In the E step (expectation step), the current parameter estimate is used to evaluate the model's log likelihood function. The M step (maximization step) computes a new parameter estimate which maximizes the log likelihood function.

Let  $\{\mathbf{X}, \mathbf{Z}\}$  be the complete data set of observed data and latent variables.  $\boldsymbol{\theta}$  is considered as the set of model parameters. The log likelihood function for a model with discrete latent variables is given by:

$$\ln p(\mathbf{X} | \boldsymbol{\theta}) = \ln \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$$

In practice, the latent variables  $\mathbf{Z}$  cannot be observed and their values are only given by the posterior distribution  $p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta})$ . Therefore, the complete-data log likelihood cannot be used and the expected value under the posterior distribution of the latent variable is considered instead in the E step of the algorithm. It takes the current parameter estimate  $\boldsymbol{\theta}^{old}$  and uses the posterior distribution of the latent variable to define the Q-function:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) = \sum_{\mathbf{Z}} p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}) \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$$

The M step maximizes this function with respect to  $\boldsymbol{\theta}$  to compute a new parameter estimate which is then used in the next iteration of the algorithm:

$$\boldsymbol{\theta}^{new} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$$

Generally, the proceeding of the EM algorithm can be summarized as follows:

1. Initialize parameter estimate  $\boldsymbol{\theta}^{old}$
2. E step: Evaluate  $p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^{old})$
3. M step: Evaluate  $\boldsymbol{\theta}^{new} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old})$
4. Check if log likelihood or parameter values have converged, otherwise set  $\boldsymbol{\theta}^{old} = \boldsymbol{\theta}^{new}$  and return to 2.

When a prior  $p(\boldsymbol{\theta})$  is defined over the model parameters, the EM algorithm can also be used to find MAP solutions by evaluating  $\boldsymbol{\theta}^{new} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{old}) + \ln p(\boldsymbol{\theta})$  in the M step.

---

## Choosing the Model's Complexity

---

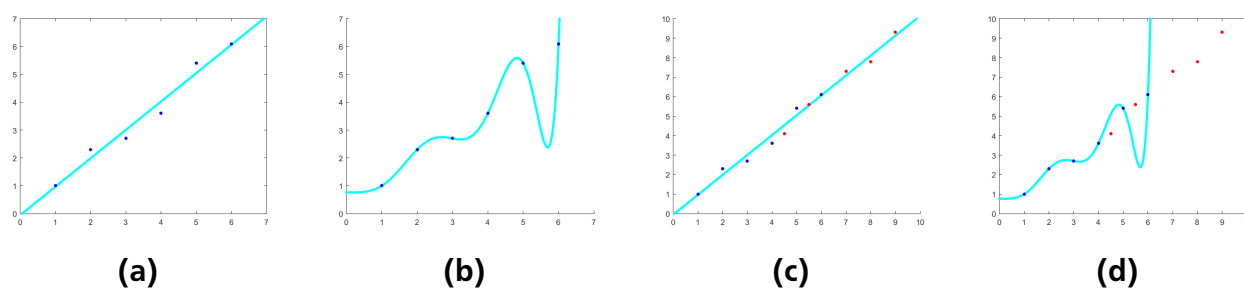
The impacting factor on the model's complexity differs between different model types (number of states, order of polynomial, number of basis functions, etc.). However, there are certain rules for choosing a model's complexity that hold true for any type of model.

An important general heuristic principle which has to be considered and is also used in many other scientific applications, is Occam's razor<sup>1</sup>. Principally, it states that if there are multiple equally likely solutions to a problem, the simplest should be preferred.

In the area of machine learning, preferring models with low complexity reduces the probability of running into overfitting problems which are described below.

### Overfitting

The meaning of overfitting could best be explained using a simple example: Consider the task of finding a polynomial that approximates the set of points shown in Figure 2.1(a),(b).



**Figure 2.1.:** Principle of overfitting

In 2.1(a) the points are approximated by a first order polynomial while in 2.1(b) a 9th order polynomial is used. As one can see, the approximation error for the given points is essentially smaller when using a high-order polynomial.

The problematic of using more complex models becomes clear, when a new set of point is observed after fitting the polynomials (Figure 2.1(c),(d)). While the first-order polynomial still provides a good approximation of these points, the 9th order polynomial does not even closely match the newly added points. It is said, that the 9th order polynomial overfits the training data and does not generalize well.

This principle holds true for almost all kinds of models used in machine learning and in many applications a tradeoff between a small training set approximation error and a good generalization needs to be found.

---

### 2.1.3 Feature Selection

---

A crucial factor for the model's prediction performance is the utilized feature set. It is therefore indispensable to perform a suitable feature selection process, before the training of the model. According to [8] there are basically three groups of feature selection processes that would be feasible to apply in this work:

---

<sup>1</sup> <http://www.math.ucr.edu/home/baez/physics/General/occam.html> (02/29/16)

**Filter Methods:** The feature selection is performed in a pre-processing step. It uses certain heuristics to assess the individual features to produce a general, model independent feature selection. However, the influence of a feature on an actual model can therefore hardly be determined by this type of approaches.

**Wrapper Methods:** The feature selection is performed by using an actual model in form of a black box to evaluate a large number of possible feature subsets based on the model's prediction performance.

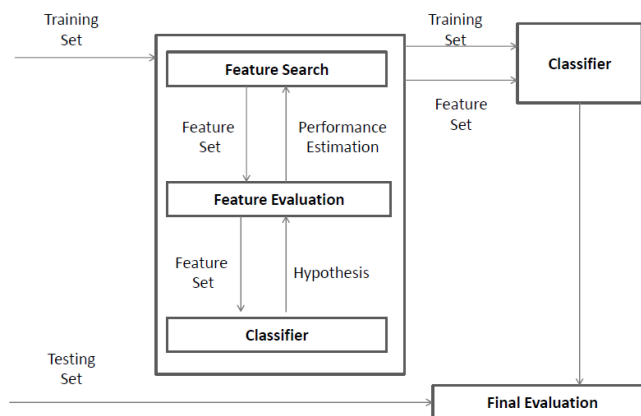
**Embedded Methods:** The feature selection is embedded into the model's learning process, resulting in a highly model specific selection.

As this work utilizes a variation of wrapper methods to determine a suitable feature selection, the remainder of this section provides a general introduction to these types of methods. More detailed information about filter- and embedded methods can be found in [8].

---

## Wrapper Methods

---



**Figure 2.2.:** Proceeding of a Feature Selection using the wrapper method. [20]

As Figure 2.2 illustrates, the selection process of a wrapper method iterates over three consecutive steps:

**Feature Search:** Determines the feature selections that are passed to the evaluation.

**Feature Evaluation:** Evaluates the performance of the current set of feature selections using the outcome of the classifier and a performance measure.

**Classifier:** An actual model which is trained on the current set of feature selection, extracted from the training set.

The process starts with a training set consisting of multiple features that are supposed to be reduced to a reasonable subset. As a first step, the feature search determines a set of feature selections (e.g. each feature by itself) that are passed on to the evaluation process. The feature evaluation trains a classification model on the current set of feature selections and evaluates the model's performance on each selection. The results are passed back to the feature search which

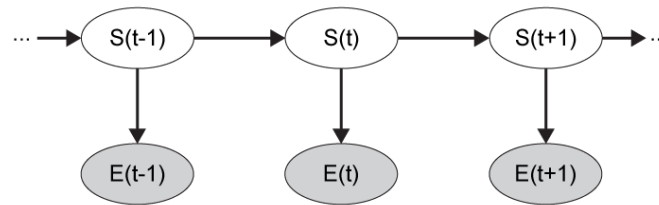
determines a new set of feature combination based on the model's performance estimation. This process repeats until the feature search does not add any new features to the current selection or every feature, contained in the training set, has been selected. The final feature selection can now be used to train the actual classification model which is afterwards evaluated on a separate test set to assess its generalization abilities.

---

## 2.2 Hidden-Markov-Models

---

Hidden Markov Models (HMMs) are widely used generative predictive models, especially suited for handling sequential data. They model a stochastic process consisting of a latent (hidden) temporal series of discrete system states ( $S(t)$ ) and corresponding observable emissions ( $E(t)$ ), generated by these states. Figure 2.3 shows the graphical model of a HMM.



**Figure 2.3.:** Graphical model of a Hidden-Markov-Model [3]

The structure of HMMs is based on two major independence assumptions regarding the system's states and emissions:

**Markov-Assumption:** Each state is only dependent on its immediate predecessor, or: given its predecessor, a state is conditionally independent of every other state. The temporal process of the states is therefore modeled as a first-order Markov process.

**Sensor-Markov-Assumption:** Each emission is exclusively dependent on its generating state, or: given the corresponding state, an emission is conditionally independent of every other emission or state.

---

### 2.2.1 Model Parameters

---

The introduced assumptions allow to define a HMM by the 3-tuple of parameters  $\theta = (\pi, A, B)$ , whereby  $\pi$ ,  $A$  and  $B$  are stationary probability distributions:

$\pi$ :  $p(s_0)$  - the initial distribution at the beginning of the Markov chain.

$A$ :  $p(s_t | s_{t-1})$  - the transition probabilities between the system's states

$B$ :  $p(e_t | s_t)$  - the emission probabilities given a state

While  $\pi$  and  $A$  are discrete distributions,  $B$  might as well be defined by a continuous distribution, depending on the nature of the observed features. A popular way to model continuous observations is the Gaussian distribution, which is also utilized in this work.

Given  $\theta$ , the joint-probability distribution for each time step of a HMM can be formulated as:

$$p(\mathbf{S}, \mathbf{E} | \theta) = p(s_0 | \pi) \prod_{i=1}^T p(s_i | s_{i-1}, A) p(e_i | s_i, B) \quad (2.1)$$

where  $\mathbf{S} = \{s_0, \dots, s_T\}$  and  $\mathbf{E} = \{e_1, \dots, e_T\}$

---

## 2.2.2 Parameter Estimation

---

Given a time series of observed emissions  $\mathbf{E} = \{e_1, \dots, e_T\}$  the parameters of a Hidden-Markov-Model can be determined by using maximum likelihood. Since the model contains latent variables, the maximum likelihood solution cannot be analytically computed and therefore a variation of the previously introduced EM-algorithm is a popular way of finding a maximum-likelihood estimate for a HMM.

As explained in sec:learningMethods, the estimation process consists of the subsequent alternation of an E- and M Step until the model's likelihood estimate converges or a maximum number of iterations is reached.

According to [3], the function  $Q(\theta, \theta^{old})$ , evaluated in the E step of the HMM based variation, is based on the marginal posterior distribution of the latent states  $\gamma(s_t) = p(s_t | \mathbf{E}, \theta^{old})$  and the joint posterior distribution of two successive states  $\xi(s_{t-1}, s_t) = p(s_{t-1}, s_t | \mathbf{E}, \theta^{old})$ . These two quantities can be efficiently evaluated using the Forward-Backward algorithm [18] which is described later in this section. The M-Step maximizes  $Q(\theta, \theta^{old})$  w.r.t.  $\theta$  to determine the new parameter estimate which is used in the next iteration of the algorithm.

As the EM-Algorithm is not guaranteed to find a globally optimal solution, it has to be initialized with reasonable values to avoid converging to a local minimum. Therefore the initial mean values of the Gaussian distributions are determined by applying the k-means clustering algorithm [3] and the prior distribution and transition matrix are initialized with normalized, randomly chosen, non-zero values.

More detailed explanations about the proceeding of the EM algorithm on Hidden-Markov-Models can be found in [3].

---

## The Forward-Backward Algorithm

---

The Forward-Backward algorithm provides the basis for the efficient evaluation of the Q function quantities  $\gamma(s_t)$  and  $\xi(s_{t-1}, s_t)$  for a given time-series of observed emissions  $E = \{e_1, \dots, e_T\}$ . It consists of two separate procedures that are described below according to the explanations in [2].

### Forward Procedure

The forward procedure determines  $\alpha_i = p(s_i | \{e_1, \dots, e_t\}, \theta)$  which denotes the probabilities of observing the partial time-series  $\{e_1, \dots, e_t\}$  of emissions and ending up in state  $s_i \in \mathbf{s}$  at time  $t$  given the the model parameters  $\theta$ . The computation is performed iteratively starting at  $t = 1$  as follows:

---

1.  $\alpha_i(1) = \pi_i b_i(\mathbf{e}_1)$

2.  $\alpha_j(t+1) = \left[ \sum_{i=1}^N \alpha_i(t) a_{ij} \right] b_j(\mathbf{e}_{t+1})$

### Backward Procedure

The backward procedure determines  $\beta_i = p(s_i | \{\mathbf{e}_t, \dots, \mathbf{e}_T\}, \boldsymbol{\theta})$  which denotes the probability of the ending partial time-series starting in state  $s_i \in \mathbf{s}$  at time  $t$  given the partial time-series  $\{\mathbf{e}_t, \dots, \mathbf{e}_T\}$  and the model parameters  $\boldsymbol{\theta}$ . It is also performed iteratively, this time starting at  $t = T$  as follows:

1.  $\beta_i(T) = 1$

2.  $\beta_j(t) = \sum_{i=1}^N a_{ij} b_j(\mathbf{e}_{t+1}) \beta_j(t+1)$

The marginal posterior distribution of the latent states  $\gamma(\mathbf{s}_t)$  and the joint posterior distribution of two successive states  $\xi(\mathbf{s}_{t-1}, \mathbf{s}_t)$ , required in the Q function, can now be evaluated using  $\alpha$  and  $\beta$  as described in [2].

---

## 2.3 Measuring a Model's Performance

---

To be able to evaluate the quality of a model or a certain parameter selection, a performance measure needs to be defined which determines the accordance of the model's outcome and the labels assigned to a data set. To produce a reasonable performance evaluation, the assessed model should be trained and evaluated on strictly separated data sets. If this principle is not applied, no statement about the generalization abilities of a model can be made: The evaluation only measures the in-sample error which may lead to the selection of complex models and an overfitting of the given data.

The separation has to be performed on a very early stage to minimize the risk of unintended data-snooping.

### Holdout

A common way of separating a given data set into a training- and test set is splitting it into two differently sized parts, where the larger part is used for the training process and the smaller part is used for testing the model's performance.

The usual splitting ratio, which is also applied in this work, is using 2/3 of the data for the training set and 1/3 of the data for the test set.

As the data, used in this work, consists of a series of time-coherent data, it is important to preserve the temporal dependencies when splitting the data. Especially because a HMM is used which is a model that incorporates these dependencies into its parameters.

The data is therefore split by using the first 2/3 of the data set for training and the remaining 1/3 for testing.



---

### 2.3.1 Precision, Recall & F1

---

The main performance measure, used in this work, is a variation of the F1 score which is defined by the harmonic mean of precision and recall.

As introduced in [10], the models inference outcome is classified as follows:

**True Predictions (tp):** The predicted maneuver matches the assigned label. Only the correct prediction of a lane change maneuver is counted as a true prediction, correctly predicted lane keeping maneuvers are left out of this measure. The reason for that is the dominance of the lane keeping maneuver. When driving on a highway, over 90% of the time lane keeping would be the correct prediction. If these were counted as true predictions, a model which does not predict any lane change maneuvers would therefore already have an exceptionally high score even though it does not detect any of the relevant maneuvers.

**False Predictions (fp):** The predicted maneuver is the opposite of the assigned label. Again, lane keeping is left out here and only lane change maneuvers are counted.

**False Positive Predictions (fpp):** The model predicts a lane change maneuver while the assigned label is lane keeping.

**Missed Predictions (mp):** The model predicts the lane keeping maneuver while the assigned label is one of the lane change maneuvers.

Based on these definitions precision, recall and F1 are defined as:

$$Pr = \frac{tp}{tp + fp + fpp}; Re = \frac{tp}{tp + fp + mp}; F1 = \frac{2 \cdot Pr \cdot Re}{Pr + Re}$$

The precision score weights the model's true predictions against all of its lane change maneuver predictions. It is a measure of how many of the model's predictions are correct. Recall, on the other hand, weights the model's true predictions against all cases where the labels are set to a lane change maneuver. It is therefore a measure of how many of the labeled maneuvers are correctly predicted by the model.

Each of these scores for itself is not a good measure of the model's performance. E.g. a model that misses many lane change maneuvers would still get a high precision score as long as its predictions are correct. On the other hand, a model with a high number of false positive predictions would get a high recall score as long as it matches the labeled lane change maneuvers.

Therefore these scores are combined to the F1 measure which forms a tradeoff between these two measures by computing their harmonic mean.

---

## 2.4 Baseline HMM

---

In order to efficiently perform the feature selection process and to evaluate the performance of the utilized model, a baseline model is introduced that offers the properties of a Hidden-Markov-Model and provides the ability of being trained in a very efficient way. The baseline model is defined as a basic Hidden-Markov-Model, consisting of three states using multivariate

---

Gaussian distributions to model the emission probabilities. Each state represents one of the labeled maneuver intentions which are supposed to be inferred given the driving data. The model therefore consists of single individual states for right lane changes, driving straight and left lane changes.

As the model's states are identical to the labeled intentions, the usually latent initial- and transition distributions become observable and the training of the model does not require running the EM-algorithm. Instead, the training process basically requires counting label frequencies and a one time fitting of the Gaussian distribution:

Considering a temporal series of observations  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ , the corresponding labels  $Y = \{y_1, \dots, y_T\}$  and the described baseline HMM.

The model's initial distribution is determined by counting the frequency of the different labels in  $Y$  and the transition probabilities are identified by counting the transitions between different labels in  $Y$  and normalizing.

Finally, the Gaussian distribution of each state needs to be fitted to the corresponding observations. For this purpose, the corresponding observations for each state are extracted from  $\mathbf{X}$  according to the labels in  $Y$  and the Gaussian distribution is fitted on that data.

The baseline HMM is therefore the simplest possible configuration of a Hidden-Markov-Model that can be used for solving the task of inferring the driver's lane change intention in this particular problem specification. Hence it provides a good baseline for the assessment of the more complex driver intention model which is presented later in this work. In addition to that it simplifies the evaluation of feature influences since it has a similar structure as the complex model while providing a highly efficient training process.

---

### 3 Experimental Setup and Data Set

---

The development of a HMM based lane change intent classifier requires a considerable amount of data containing information about the driver's behavior, vehicle dynamics, vehicle environment, et cetera.

In this work an extensive data set is used that has been captured in a simulator study in 2014. This chapter describes the experimental setup of that study and provides detailed information about the available data.

---

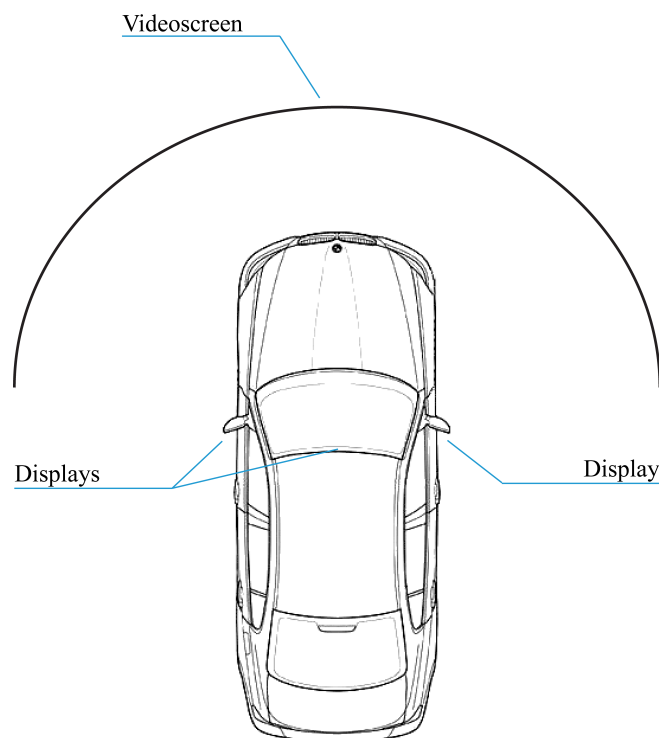
#### 3.1 The Driving Simulator

---

As Figure 3.1 illustrates, the simulator setup consists of a modified car body, statically placed in front of a large curved screen showing the simulated environment. In order to allow the driver to monitor the area behind the car, the mirrors have been replaced by similarly shaped displays, showing a simulated rear view. The car's control units (steering wheel, pedals, etc.) are connected to the simulator software and can be used in the usual manner.

To capture the driver's behavior, the interior of the car is equipped with driver facing infrared cameras, supported by infrared LEDs which are illuminating the scene without distracting the driver. Additionally to that, an accelerometer is attached to the driver's right arm.

Exterior Features like distance to lane center or time to collision are provided by the simulation environment.



**Figure 3.1.:** Birdseye view on the simulator setup.<sup>1</sup>

---

## 3.2 The Data Set

---

This section gives an introduction into the captured data set, determines a first coarse pre-selection of suitable features and discusses important properties of the data that have been discovered during the course of this work.

The overall data set consists of 34 individual records of different drivers, each containing around 40 minutes of driving data and approximately 30 lane change maneuvers in each direction. Figure 3.2 shows the simulated scenario that corresponds to a drive on a multi-lane highway.



**Figure 3.2.:** The driving scenario.

---

### 3.2.1 Coarse Feature Pre-Selection

---

As each record contains the huge number of 403 captured features, a first coarse pre-selection is performed to extract potential candidates for the inference of lane change intentions according to their denomination and description in the signal reference list which came with the data set. The resulting list of candidate features is shown in Table 3.1.

---

<sup>1</sup> Car illustration: <http://www.the-blueprints.com> (02/18/2016)

<b>Feature</b>	<b>Description</b>
<b>Head Features</b>	
Gaze Heading	Horizontal orientation of the driver's gaze direction
Gaze Intersection	Discrete mapping of driver's gaze to certain world objects (windshield, mirrors, etc.)
Head Heading	Horizontal orientation of the driver's head
<b>Exterior Features</b>	
Psi	Difference between vehicle's yaw angle and lane tangential angle
Lateral Distance	Lateral distance between center of car and center of lane
Distance Next	Longitudinal distance to the next vehicle on the current lane
<b>Control Inputs</b>	
Accelerator Pedal	Position of accelerator pedal
Brake Pedal	Position of brake pedal
Steering Wheel	Angle of the steering wheel
Steering Wheel-v	Rotational velocity of the steering wheel
Steering Moment	Applied torque to the steering wheel
<b>Vehicle Dynamics</b>	
v-yaw	The vehicle's yaw rate
a-y	The vehicle's lateral acceleration
v-y	Lateral velocity of the vehicle
<b>Arm Movement</b>	
Sensor A - AccX	Acceleration in x-direction
Sensor A - AccY	Acceleration in y-direction
Sensor A - GyroZ	Orientation w.r.t z-axis
Sensor A - LinAccX	Acceleration in x-direction
Sensor A - LinAccY	Acceleration in y-direction
Sensor B - AccX	Acceleration in x-direction
Sensor B - AccY	Acceleration in y-direction
Sensor B - GyroZ	Orientation w.r.t z-axis
Sensor B - LinAccX	Acceleration in x-direction
Sensor B - LinAccY	Acceleration in y-direction

**Table 3.1.:** Coarse pre-selection of candidate features

---

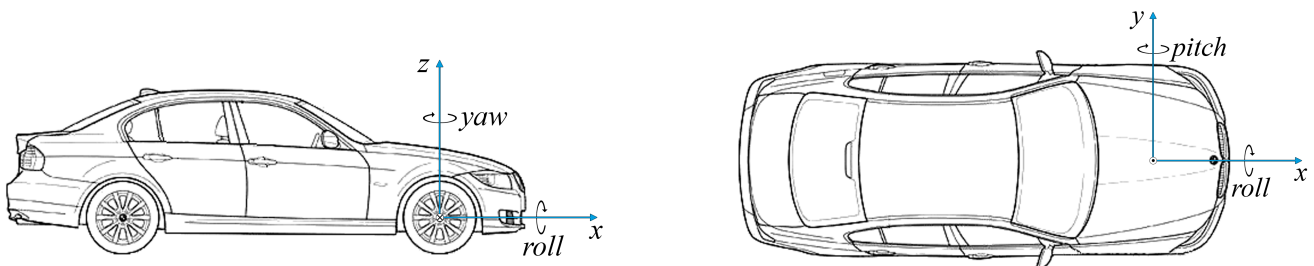
## 3.2.2 Coordinate Systems

---

The features utilized in this work are measured with respect to two different coordinate systems: The car coordinate system, responsible for external features and the head coordinate system responsible for the utilized head features.

### Car Coordinates

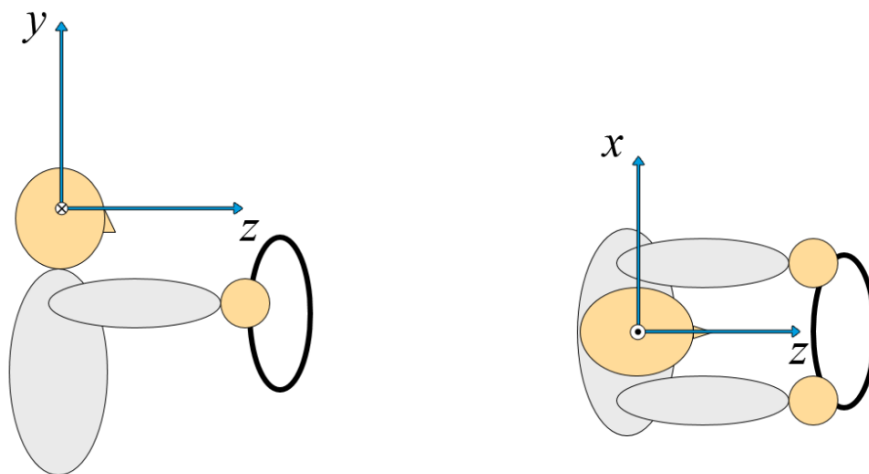
The car coordinate system, illustrated in Figure 3.3, complies with the automotive convention and is originated in the center of the front axle. It is a right-handed system oriented in the way that the x-axis is pointing in driving direction, the y-axis is pointing to the left and the z-axis is pointing upwards.



**Figure 3.3.:** Illustration of the car coordinate system. <sup>2</sup>

### Head Coordinates

The head coordinate system, illustrated in Figure 3.4 is defined to be originated in the middle of the driver's head with the x-axis pointing to the left, the y-axis pointing upwards and the z-axis pointing in driving direction. As the driver's head does not remain on a static position, the actual origin is estimated by the camera system which is monitoring the driver.



**Figure 3.4.:** Illustration of the head coordinate system

---

<sup>2</sup> Car illustration: <http://www.the-blueprints.com> (02/18/2016)

---

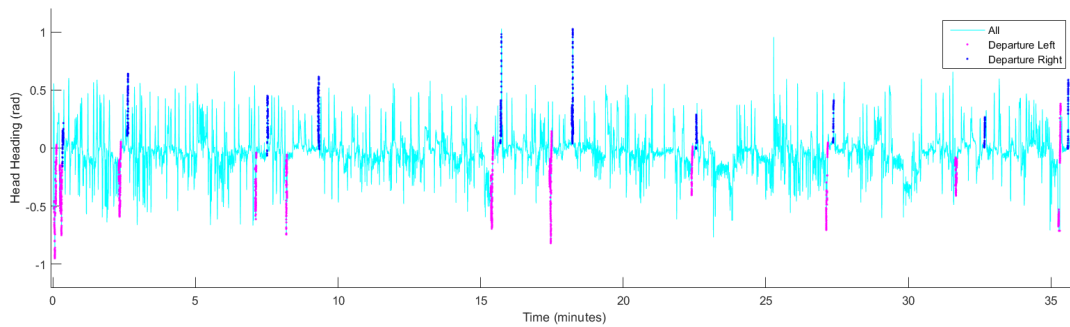
### 3.2.3 Driver Variance

---

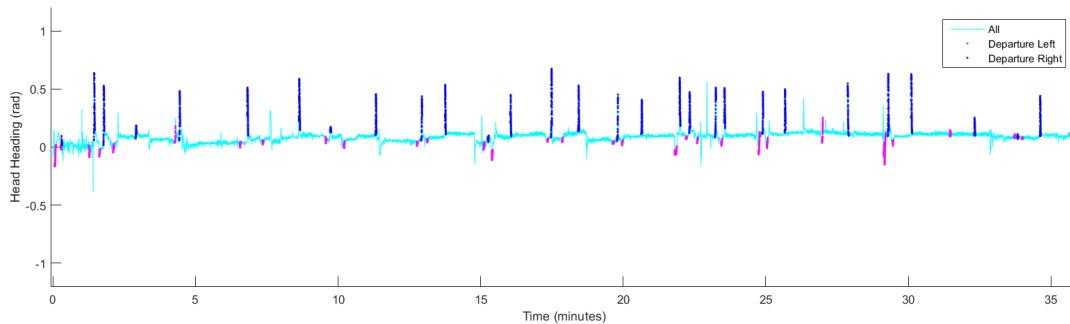
The evaluation of different driving records shows that there is a huge variance in the behavior of different drivers.

A prominent example for this behavioral difference between drivers is the nature of head movement which is illustrated in Figure 3.5.

As one can easily see, the amount and magnitude of head movement significantly differs between the two drivers. While driver 33's head is constantly moving throughout the course of the record, driver 34 mainly shows head activity in close proximity to lane change maneuvers.



Driver 33



Driver 34

**Figure 3.5.:** Illustration of variance in head movement between driver 33 and 34.

---

### 3.2.4 Anomalies

---

Since the data has been recorded in a driving simulator, the overall quality is on a very high level. However, inspecting the data reveals several anomalies that may impact the performance of the applied machine learning approaches.

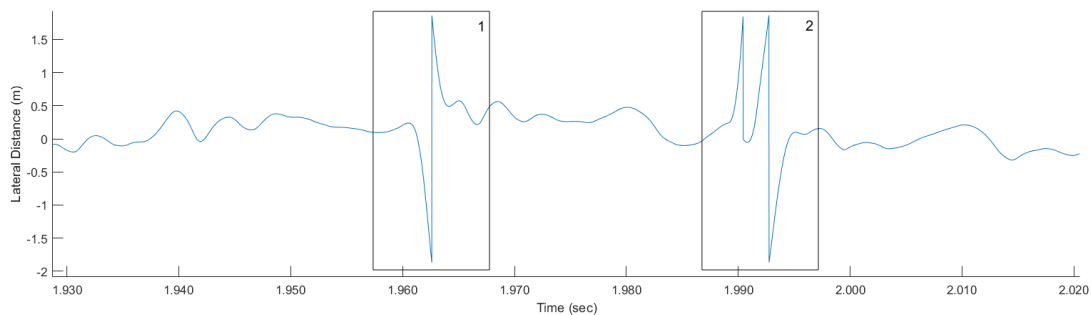
The remainder of this section furtherly examines the discovered anomalies and provides an assessment of their potential influence on the learning and evaluation process which is presented later in this work.

#### Lateral Distance at Highway Exits

The first anomaly regards the Lateral Distance feature which is provided by the simulator software and measures the deviation between the vehicle's center and the center of the current lane in lateral direction. The correlation between the value of this feature and lane change maneuvers is very high, since it does show a significant gain when such a maneuver is executed and only shows little activity when driving straight. In addition to that, it shows a significant "jump" when the vehicle crosses the lane marking which will play an important role in the labeling process which is presented later in this work.

The observed anomaly arises when the vehicle passes a highway exit. As Figure 3.6 illustrates, the feature shows a behavior which is similar to a lane change maneuver, even though the vehicle is driving straight.

The reason for that anomaly is the way, a lane is defined in the simulator software. It seems like the outer lane marking follows the highway exit, which leads to a deviation between the vehicle's center and the lane center in the opposite direction.



**Figure 3.6.:** Illustration of the lateral distance anomaly. Box 1 shows the usual behavior of lateral distance at lane change maneuvers. Box 2 shows the described anomaly, followed by an actual lane change maneuver.

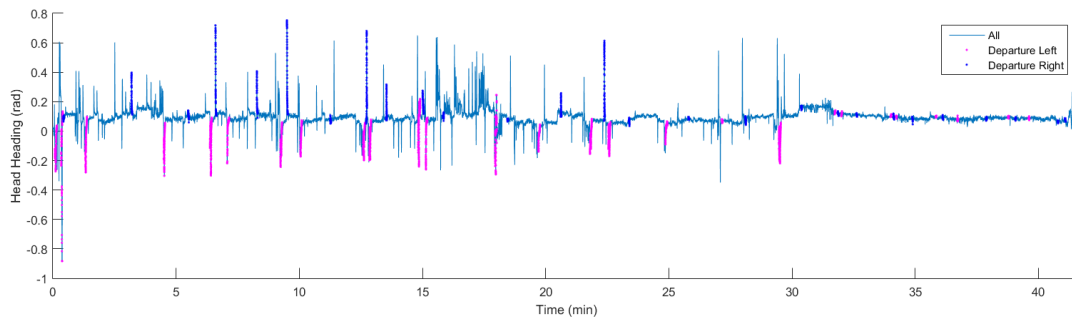
When using this feature in the machine learning process, this may cause false positive predictions at highway exits and also could reduce the overall prediction precision. However, since the number of highway exits on the simulated route is relatively small the negative influence of this anomaly should also be very limited.



---

## The Driver's Activity

Another discovered anomaly concerns the driver's activity over the course of the drive. Every driver shows a significant reduction of head activity after approximately 30 minutes of driving. As Figure 3.7 shows, there is almost no head movement at lane changes in the last third of the record while the first 30 minutes show a distinct correlation between lane changes and head movement.



**Figure 3.7.:** Illustration of the head heading anomaly. The head activity drastically decreases after 30 minutes of driving. Even at lane change maneuvers, there is no significant head movement in the last third of the record.

As it turned out, after 30 minutes of driving, the participants of the study were told to stop checking the car's environment before lane changes.

Since this does not correspond to a natural human behavior, the last 15 minutes of each record are discarded and not used for any of the processes described in 5.

To examine the influence of such drastic behavioral changes on the model's performance, the final model is however evaluated on cropped records as well as on entire records.

---

## 3.3 Real World Feasibility

---

Even though the data has been captured in a driving simulator, the model is designed with a real world application in mind. All exterior feature candidates are yet available in many mid- to high-level serial cars, as they are already used by current ADAS applications.

Table 3.2 shows all short-listed exterior features and provides information about their required sensor technology as well as current ADAS applications based on these features.

Feature	Description	ADAS	Sensor Technology
Lateral Distance	Lateral distance between center of car and center of lane	ACC, LKAS, LDW	Camera
Psi	Angle between car and lane	ACC, LKAS, LDW	Camera
Distance Ahead	Longitudinal distance to ahead car on same lane	ACC, Traffic Jam Assist, Emergency Braking Assist	RADAR <sup>1</sup> , LIDAR <sup>2</sup> , Camera

**Table 3.2.:** Utilization of exterior features in real world applications.

---

<sup>1</sup> Radio Detection And Ranging

<sup>2</sup> Light Detection And Ranging

---

## 4 Related Work

---

The application of machine learning methods in the automotive environment is in focus of many recent publications which show a wide variety in the utilized machine learning methods and applications.

As this work focuses on inferring the driver's lane change intention, the related work, presented in this chapter, is limited to publications with a similar area of application.

The approach, presented in [12] utilizes a probabilistic variation of a multiclass Support Vector Machine (SVM) [4] in combination with Bayesian Filtering to predict lane changes, solely based on lane information, vehicle speed and steering angle. They distinguish between three maneuver classes (right lane change, left lane change and lane keeping) which are inferred from real-world driving data.

The presented approach acquires probabilistic output from the SVM by using a generalized Bradley-Terry model [13] and feeds it into a Bayesian Filter that recursively determines the maneuver probabilities by combining the SVM output with the maneuver probabilities of the previous time step.

The presented results show that the average prediction time is only 1 sec which might not be sufficient for an efficient prevention of accidents.

Due to the findings of this work, I suspect that this time could very likely be improved by adding head motion to the utilized feature set.

In [15] a system is presented that infers the driver's intention based on lane positional information, vehicle parameters and the driver's head motion using Sparse Bayesian Learning (SBL) [21]. SBL is a discriminative approach that has the property of automatically pruning irrelevant features and creating a sparse representation by applying independent Gaussian prior distribution to the weighting parameters of the underlying set of basis functions. The variance of the Gaussian distribution is estimated from data by using Evidence Maximization and is forced to be zero if the corresponding feature is deemed irrelevant which leads to the automatic pruning of unnecessary features. As an additional step, a quantile filter is applied to smooth the inference results.

The evaluation of this approach shows, that the utilization of head features significantly improves the model's performance, which is in accordance to the findings of this work.

In [14] a driver specific, sophisticated variation of a Hidden-Markov-Model (HMM) is used to detect unintentional lane departures at an early stage. It differentiates three control strategies (lane change right, lane keeping, lane change left) and trains a separate HMM for each control strategy in a first step, whereby the number of states for each HMM is determined using the Bayesian information criterion (BIC). Afterwards these HMMs are combined to a global driver model to determine the most likely control strategy at the current point in time, given the history of driving situations. Subsequently, the most likely strategy is used to predict the future driving inputs using Gaussian Mixture Regression (GMR) [5] and a nonlinear bicycle vehicle model.

This approach is again solely based on lane information and vehicle dynamics and therefore

---

might be enhanced by including head features into the process.

As the combination of three control strategies to a global driver model seem to be a promising enhancement of HMMs, the model presented in this work adopts several aspects of this approach.

A variation of HMMs, called AIO-HMM (Autoregressive Input-Output HMM), is used in [10] for the anticipation of lane change maneuvers on highways and turns at lane intersections. The standard HMM is extended by an additional input-Layer, modeling external features and conditioning the hidden states as well as the output-layer which models internal features. In addition to that, a temporal dependency between successive outputs is introduced.

In this approach a separate AIO-HMM is trained for each maneuver and the current maneuver is determined using the maneuver that best explains the past  $T$  seconds of driving context and a threshold that is supposed limit the prediction fluctuation.

In [11] the combination of Recurrent Neural Networks (RNNs) [17] and Long Short-Term Memory (LSTM) [9] is proposed for the anticipation of lane change maneuvers on highways, as well as turns at lane intersections. LSTM cells allow maintaining their state over time are used in the RNN to remember long-term context dependencies.

Another widely related application is presented in [1]. It describes a context aware driver behavior detection system that is supposed to assess the driver's behavior and, if necessary, warn other vehicles using a vehicular ad-hoc network (VANET).

A probabilistic model, based on a dynamic Bayesian network (DBN) is used to infer different types of driving behavior (normal, drunk, reckless and fatigue) based on internal and external features including gaze direction, the vehicle's velocities and accelerations, lane information, information about other traffic participants, etc..

Including a lane change intention classifier into such a connected environment would have a high potential of increasing the benefit of both systems: The lane change classifier would be able to incorporate more detailed information of other traffic participants into the classification process and the information about intended lane changes could be transmitted to surrounding vehicle's, so that they can react before potential dangerous situations occur.

---

## 5 Methodology

---

This chapter introduces the process, carried out to train a Hidden-Markov-Model that allows the inference of lane change intention on the available data set.

It starts with a detailed introduction into the developed labeling algorithm and afterwards describes the individual steps which are required to find a suitable feature selection and parameter estimation for the final driver intention model.

---

### 5.1 Labeling the Driver's Intention

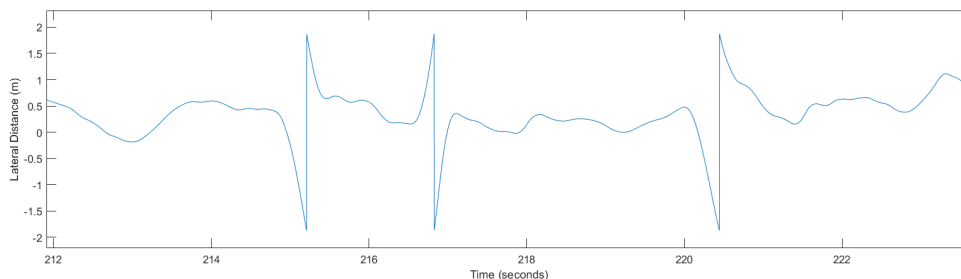
---

One of the most crucial parts of this work is the labeling of the driver's intention. It builds the basis for almost every other part of the proposed approach as it is used for the feature selection, the training of the Hidden-Markov-Model as well as its performance evaluation.

The critical point is, that it is nearly impossible to find a "Ground Truth" of the driver's actual intention. Even for a human observer, it is hard to tell, when intention starts and how it is shown. It gets even harder if the same criteria are supposed to work for different drivers due to the huge variance in their behavior, described in 3.2.3. In addition to that, even a single driver shows varying behavior in different situations.

In order to avoid false positive labels and to render possible an automated labeling method, an algorithm is developed that focuses on actually performed lane change maneuvers and defines a certain time window before the crossing of the lane marking. The data points inside that window are labeled as intention for the corresponding maneuver (right/left lane change) and all remaining data points in the data set are labeled as lane keeping.

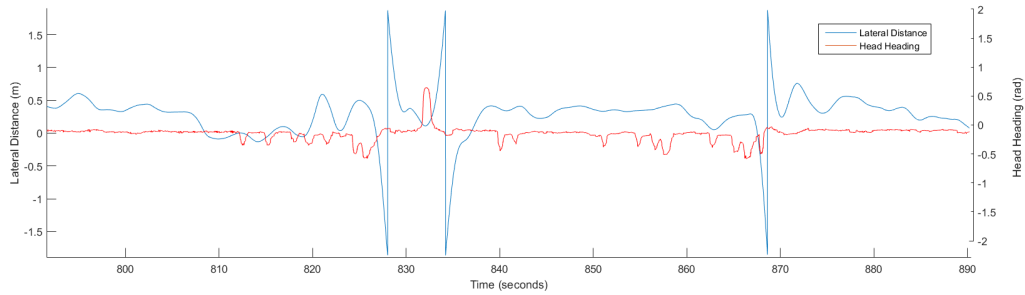
The end point of each window is determined using the distance between vehicle and lane center that, as Figure 5.1 illustrates, shows a significant "jump" when the vehicle crosses the lane marking, caused by a change of the reference system to the center of the new lane.



**Figure 5.1.:** Lateral Distance at lane crossings.

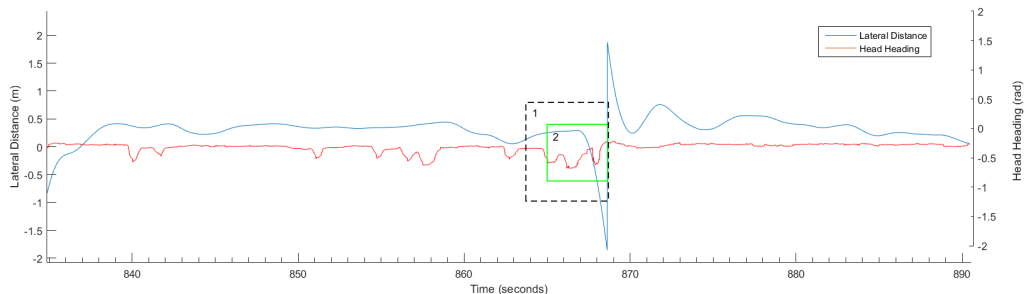
To set the beginning of the time window, a feature needs to be determined, which indicates the drivers intention of performing a lane change. A simple solution, that is used in many publications, is defining a certain fixed time interval and use it throughout all records. However, this does not map the actual intention very well, since its duration is highly situation and driver dependent. Hence, in this work a more sophisticated labeling approach is used which dynamically incorporates the drivers behavior to find a suitable window size.

The visual inspection of the recorded driving data, described in 5.3.1, revealed a high correlation between lane change maneuvers and the driver's head movement (see Figure 5.2). This arises from the fact that a driver usually assures that a lane change maneuver can be safely carried out before it is actually executed. Therefore the increase of the driver's head motion in close temporal proximity to a lane change can be considered as a good approximation for the beginning of the lane change intention.



**Figure 5.2.:** Lateral Distance and Head Heading at lane crossings.

To determine the start of the increased activity, a sector of 5 seconds before crossing the lane marking is searched for peaks in the driver's head heading. As Figure reffig:labelingAlgo illustrates, the peak which is the furthest away from the lane crossing is set as beginning and all points in between are marked as intention for a lane change in the corresponding direction. In cases of no, or only little head activity, a minimum sector of 2 seconds before the crossing of the lane marking is labeled as intention.



**Figure 5.3.:** Illustration of the labeling method. Box 1 defines the 5 second search window which is used to find the start of the labels by finding peaks in the driver's head movement. Box 2 marks the area which is actually labeled as a lane change intention, starting at the first peak of head movement inside the search window.

## 5.2 Feature Pre-processing

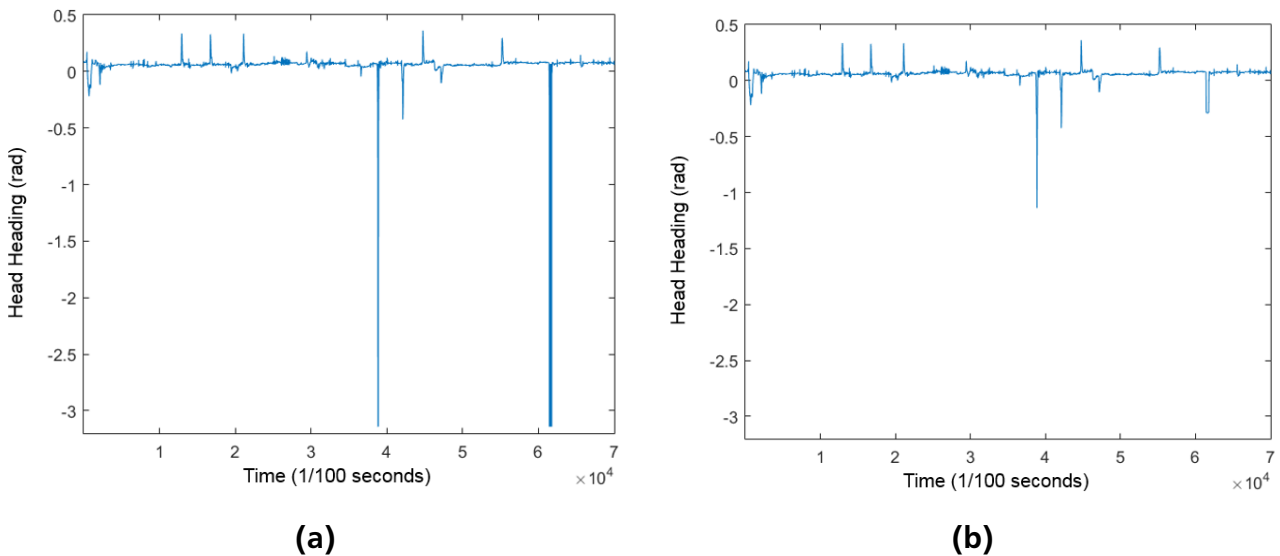
One of the advantages of using a driving simulator is the quality of the provided features. Hence, most of them do not require a pre-processing step and can be used in their original form. However, this does not hold true for every feature and therefore the remainder of this section presents those features that need to undergo a pre-processing step and describes the applied modifications.

## Head- and Gaze Features

As described in 3.1, the features providing information about the driver's head movement and gaze direction captured by a camera system. The quality of these features is therefore not on the same high level as those which are provided by the simulator software.

The largest abbreviations between those features and reality are caused by the occlusion of the driver's face, mostly induced by his hand. To assess the validity of the provided values, the utilized computer vision toolchain therefore provides a separate signal, rating the quality of its outcome.

Whenever this quality signal falls below 50 % the value of the corresponding feature is replaced by its last valid predecessor. Figure 5.5 shows an excerpt of head heading values before and after replacing low quality values.



**Figure 5.4.:** Excerpt of head heading values. (a) shows the original values, (b) shows the values after replacing low quality values.

## Distance to the Next Vehicle Ahead (Distance Next)

Even though this feature is provided by the simulation environment, it needs to undergo a pre-processing step to make it usable for the proposed model.

As Figure 5.5(a) shows, the distance to the next vehicle ahead (Distance Next) is artificially set to 0, if no vehicle is closer than 900 m. This causes several problems if Gaussian distributions are used to model the observations.

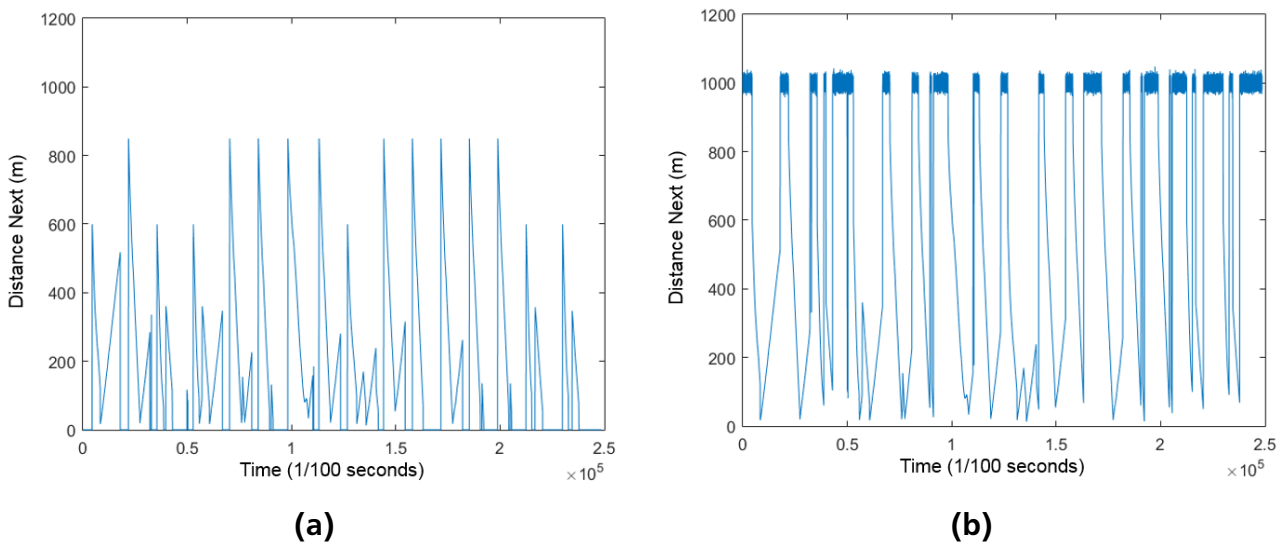
First of all, it is disadvantageous that being in close proximity to a car ahead and the complete absence of a car are mapped to numbers of similar magnitude. Considering a Gaussian HMM with two states, whereby state A models the proximity to a car and a state B models the absence of a car. The mean of the fitted Gaussian distributions would be somewhere around 0 for both states. If a new data point is observed which also has a value close to 0, the Gaussian probability density function of both states also evaluates to value of similar magnitude which leads to a high probability of misclassifications.

Another problem arising with the artificial 0 is the fact, that many data points have the exact same value. If a HMM with multiple states is used, there is the possibility that one of these states is supposed to map data points with 0 value only. This leads to the problem, that the Gaussian

---

distribution collapses to a single point which results to a variance of zero and a corresponding density function that goes to infinity at that point.

To overcome the explained problems, the zero valued data points are set to a better suited value by sampling a Gaussian distribution centered right above the highest value of valid observations (1000 m). This establishes reasonable distance between the values of close proximity and no car ahead as well as eliminates the issue of a collapsing Gaussian distribution while preserving the feature's properties. The result of this modification is shown in Figure 5.5.



**Figure 5.5.:** Distance to vehicle ahead. (a) shows the original values, (b) shows the values after applying the described modifications.

---

### 5.3 Feature Selection

---

Another crucial factor for the system's prediction performance is the selection of the utilized features. Ideally, a feature should have a distinctive correlation to the driver's lane-change intention while containing a low amount of peaks when driving straight. Additionally to that, the feasibility of capturing the features in a real-world scenario, and the amount of noise are also important criteria. In order to extract an appropriate feature subset, a substantial signal analysis has been carried out, including manual pre-selection and an automated evaluation of the candidate features. To prevent any kind of unintended data-snooping, the selection process is limited to the records 1 to 25 while the remaining records are set aside for the evaluation of the final driver intention model.

---

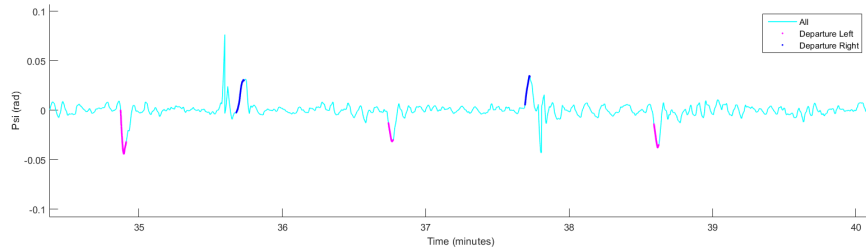
#### 5.3.1 Pre-Selection

---

The pre-selection process starts with the coarse selection presented in 3.2.1. These candidate features are furtherly evaluated by a manual visual inspection of the augmented temporal progression of the feature's values which is illustrated in Figure 5.6. To allow a reliable assessment of the feature's correlation to lane change intentions, the previously introduces labeling algorithm is used to highlight the feature values in the labeled areas. In addition to that, the



visualization provides a short video clip of the corresponding driving context when clicking on a certain point in the plot.



**Figure 5.6.:** Illustration of the visualization plot showing the angle between vehicle and lane over time. The values which are labeled as a lane change intention are highlighted in blue and magenta.

The visual inspection has been carried out on three randomly chosen records which have been evaluated using the following criteria:

1. **Overall signal quality:** Is it stable, are there many outliers, how does the corresponding quality signal behave, etc.
2. **Correlation to the labeled lane change intention:** Is there a distinct behavior in the labeled areas, how many peaks occur in unlabeled areas, etc.

The assessment of the inspected features can be found in Appendix A in Table A.1.

---

### 5.3.2 Final Selection

---

For the final feature selection, a variation of the wrapper method, introduced in 2.1.3, is used starting with 11 candidate features that returned from the pre-selection.

The following configuration of the wrapper method is used for the selection process:

**Feature Search:** Generates every possible subset of the pre-selected signal, resulting in a total of 2047 evaluated combinations. It starts with each feature individually, then uses each pairwise combination, each triple, et cetera.

**Feature Evaluation:** Evaluates the F1 measures of every feature combination, using records 1-13 for the training of the classifier and performing the inference and performance evaluation on each of the records 15-25 individually.

**Classifier:** The baseline HMM.

#### Alternative Feature Search Method

As the number of feature subsets, which are generated by the described feature search method, grows exponentially with the amount of candidate features, it can only be used with the fairly small pre-selection which requires much manual human assessment. Besides that, the pre-selection is only performed on three randomly chosen records which might not be entirely representative for the remainder of the records.

Hence, to validate the described feature selection, an alternative selection process is carried

---

out, which involves less manual assessment and includes the entire coarse feature pre-selection containing 24 candidate features.

In order to evaluate these features, the feature search method is replaced by an iterative, greedy process that generates only a fraction of combinations of the original method. At each iteration, the combination of the previously selected features and each individual remaining candidate feature is evaluated and the feature which leads to the highest evaluation score is added to the set of selected features. The process terminates when either selecting an additional feature does not increase the evaluation score by at least 1 % or all candidate features have been selected.

After performing both of the processes, final selection is determined by analyzing the frequency of feature occurrences in the best performing feature selections of each evaluated record. The results of that process are presented in 5.3.

---

## 5.4 The Driver Intention Model

---

As stated in 1, this work utilizes a Hidden-Markov-Model for the prediction of lane change maneuvers based on the driver's intention. The model's design is adapted from [14] and hence consists of three separately trained maneuver HMMs which are afterwards combined to the driver intention model. As their name indicates, the maneuver models represent the labeled maneuver intentions and therefore correspond to right lane change intention, lane keeping and left lane change intention. The global driver intention model combines these maneuver models into a single HMM which is finally used to infer the driver's lane change intention.

---

### 5.4.1 Training the Model

---

The process of training the driver intention model follows multiple consecutive steps, illustrated in Figure 5.7, which are described in the remainder of this section.

#### Feature Extraction and Labeling

As a first step, the features that are supposed to be used in the training and execution of the model need to be extracted from the records used for the training process. In addition to that, the labeling algorithm, described in 5.1 is executed on the extracted data set to create the required ground truth for the training process.

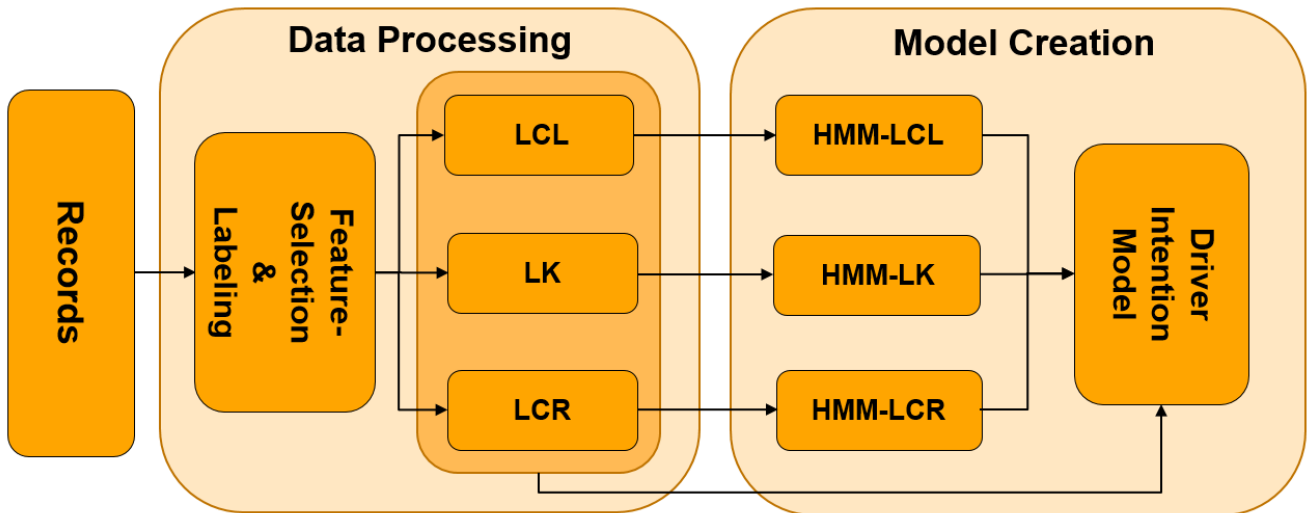
Based on these labels, the data set is divided into a separate maneuver training sets which are used for the subsequent creation of maneuver models.

#### Training separate Maneuver Models

Given the separated training set, individual maneuver models can be trained on the corresponding data using the EM-algorithm.

This separate training process allows choosing an arbitrary number of states for each of the maneuver HMMs. That allows a fine-grained modeling of the observed features and their temporal dependencies which enhances the accuracy of the mapping between the model's emissions and the observed features, especially when using Gaussian distributions.

E.g. considering a model uses multivariate Gaussian probability distributions to model the emission probabilities of two features and a given data set of observations, shown in Figure 5.8 (a).



**Figure 5.7.:** Overall process of training the driver intention model. Starting with a set of records, the desired features are extracted and the data set is split into three subsets according to the assigned labels(LCL = Lane Change Left, LK = Lane Keeping, LCR = Lane Change Right). After that, an individual maneuver HMM is trained on each subset and finally combined to the driver intention model. It is trained on the combined data set in a way that reuses certain key aspects of the maneuver HMMs.

Figure 5.8 (b) and (c) show the result of training a model with one and two states on these observations. As one can see, using two states increases the model’s ability of capturing the structure of the observations and therefore results in a more precise model. However, it has to be considered that increasing the number of states also increases the model’s complexity and may lead to an overfitting of the observed data.

The separate maneuver HMMs could already be used to predict the driver’s intention by choosing the most probable maneuver for a given sequence of observations. However, this does not fully utilize the model’s capability of modeling sequential data.

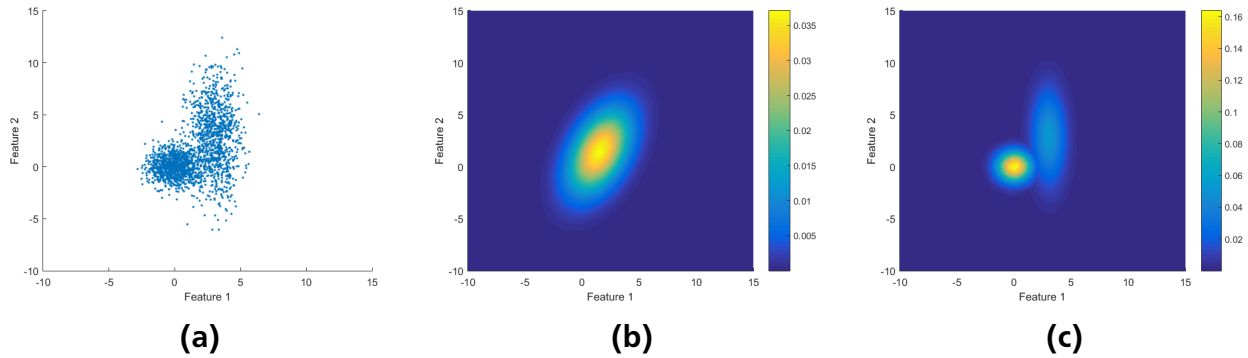
Therefore these HMMs are finally combined to the driver intention model which incorporates the probabilities of transitioning between the individual maneuvers into a global HMM.

### Training the Driver Intention Model

As a final step, the separate maneuver models are combined to the driver intention model which again requires the execution of the EM-algorithm. Fortunately the final training process can be initialized using the results of the maneuver training. Certain aspects, like the emission probabilities can even be completely inherited.

Training the final model therefore requires the following steps:

1. Determine the prior probabilities of the maneuver models by counting the number of data points in each part of the training set and normalizing
2. Determine the transition probabilities between these models by counting the transitions in the labeled training set and normalizing
3. Initialize the global transition matrix using the results of 1., 2. and the maneuver transition matrices. Inherit the emission probabilities from the maneuver models.



**Figure 5.8.:** Illustration of the Gaussian emission probabilities of two different models, fitted on the observations shown in (a). (b) and (c) show the corresponding Gaussian probability density functions of a model using a single state (b) and a model using two states (c). It can be seen that a model, using two states (c) provides a more precise mapping of the observations than a model using a single state (b).

4. Train the initialized global model on the entire training set using the EM-algorithm while leaving the emission probabilities untouched

---

## 5.4.2 State Selection

---

As mentioned before, training individual maneuver HMMs provides the possibility of arbitrarily choosing the number of states for each model which leads to a finer grained mapping of the features properties and temporal dependencies.

However, the number of states needs to be chosen carefully to find a good tradeoff between the model's complexity and generalization ability. A common way of choosing the number of states is evaluating the Bayesian Information Criterion (BIC) [3] on a large number of combinations and choosing the combination that results in the lowest score, as performed in [14]. However, this approach is solely based on the model's likelihood on a given training set and only allows a vague estimation of the model's generalization abilities.

Hence, in this work the state selection process includes the evaluation of the model's performance on a separate test set which ensures a model with good generalization abilities and therefore implicitly limits the model's complexity.

To find a well performing, driver independent state combination, a process is performed which is somewhat similar to the wrapper method, described in 2.1.3.

As Listing 1 illustrates, the process iterates over a set of state configurations, trains and evaluates a model for each of them and selects the configuration that results in the best average performance. Again the model is trained on records 1-13 and evaluated on each record 15-25 individually.

Since the training process of the driver intention model is significantly slower than training the baseline model, the evaluated variations need to be limited to a reasonable number.

Experiments that have been performed in advance of the selection process showed that the model does not require a huge number of states to achieve good prediction results. This is especially true for the lane change maneuvers, since the amount of observed data for these maneuvers is considerably smaller than the data set for lane keeping.

---

**Data:** Records 1-25

**Result:** A driver independent state selection

initialization;

**foreach** *Combination of States* **do**

    train driver intention model on records 1-13;

**foreach** *Record 15-25* **do**

        evaluate the model's performance using F1-measure;

**end**

**end**

choose combination resulting in best average F1-score;

**Algorithm 1:** State selection process.

The limits for the generation of state combinations are therefore to that observations an set to the values, shown in Table 5.1, resulting in a total of 360 different state combinations that are evaluated in the selection process.

	Lane Change Right	Lane Keeping	Lane Change Left
States	1 - 6	1 - 10	1 - 6

**Table 5.1.:** Search space of the state selection process.

---

### 5.4.3 Applied Model Variations

---

The family of Hidden-Markov-Models comprise a large variety of distinctive configurations. E.g. as described in 2.2, the emissions of a HMM can be modeled by different types of probability distributions. Besides that, there exist several HMM extension which add extra model-parameters and dependencies to increase the model's capability of mapping certain structures of the world. During the progress of this work, two variations of HMMs have been applied whose properties are described in this section.

---

#### Gaussian Observations

---

A commonly used variant of HMMs utilizes Gaussian probability distributions to model emissions with continuous valuation. In this work a single multivariate Gaussian distribution is used to model the emission probabilities of a hidden state, whereby each of the selected features is incorporated as a separate dimension of the distribution. In the inference process, the d-dimensional probability density function of the state's Gaussian distribution is used to determine the probability of a state given an emission:

$$y = f(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|\boldsymbol{\Sigma}|}(2\pi)^d} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

where  $\mathbf{x}$  is a d-dimensional input data,  $\boldsymbol{\mu}$  is a d-dimensional vector denoting the mean of the Gaussian distribution and  $\boldsymbol{\Sigma}$  is a d-by-d symmetric positive definite matrix denoting the covariance of the Gaussian distribution.

Even though Gaussians are a very popular and convenient way to model continuously valued features, there are certain issues one has to consider when using this approach: Depending on the nature of a feature, a Gaussian distribution might not be a good mapping of the feature's behavior or even infeasible to apply. As shown in Figure 5.8(b), in some situations, increasing the number of states is a possible way to overcome the issue of a bad mapping, but is also increasing the model's complexity.

---

### Combination of Gaussian and Discrete Observations

---

The data records contain several features with discrete valuations which cannot be sufficiently modeled by a Gaussian distribution. Especially one of them seems to be a promising candidate for the inference of the driver's intention. It determines the intersection of the driver's gaze and real world objects and maps it to a predefined set of areas of interest. These areas are defined as shown in Table 5.2.

Value	Area of Intersection
0	Windshield
1	Left Outer Mirror
2	Internal Rear Mirror
3	Right Outer Mirror
4	Speed Indicator
5	Display (placed at the center of the dashboard)
6	Middle Console
65535	Unknown

**Table 5.2.:** Discrete Gaze Intersection areas

In the remainder of this work, this feature is referred to as Gaze Intersection. In order to evaluate its contribution to the prediction quality, the model needs to be enabled to simultaneously handle continuous and discrete features. This is achieved by adding an additional parameter to the HMM, denoting the emission probabilities for the discrete observations. Therefore the Gaussian/Discrete HMM is defined by the parameter 4-tuple  $\theta = (\pi, A, B_{gauss}, B_{discrete})$  and a weighted combination of both distributions is used in the training and inference processes.

---

## 5.5 Inferring the Lane Change Intention

---

After training the driver intention model, it can now be used for the prediction of lane change maneuvers, working on a stream of driving data consisting of the same set of features which have been used in the training process.

The inference is performed using the forward procedure of the Forward-Backward algorithm, described in 2.2.2, which iteratively determines the probabilities of being in each of the model's hidden states for every newly observed feature vector.

Since each of the maneuvers is modeled by multiple hidden states, the probabilities of each maneuver's states are accumulated to form the final maneuver probabilities and the maneuver

---

with the highest probability is chosen as the current maneuver.

Since the forward procedure is only dependent on the outcome of its prior iteration, the performance of the inference process does not depend on the duration of execution.

It might have been noticed, that the process which is described here, is purely based on inferring the driver's intention using current observations. The predictive nature of this process is achieved through the applied labeling of the maneuvers which starts several seconds before the lane marking is crossed.

One could also use the described process to compute actual predictions of future state probabilities by running the forward procedure without adding new observations and solely using the transition probabilities to determine the succeeding state probabilities. However the quality of these predictions is known to decrease rapidly and the algorithm converges to a stationary distribution shortly [19].

---

## 5.6 Visualization of Results

---

In order to allow an assessment of the model's performance which is more intuitive than the numerical F1 score, two variants of graphical outputs are introduced that allow a detailed visual inspection of the inference results.

### Variant 1: Plot

The first visualization variant is a plot that creates an overlay of predicted maneuvers and corresponding labels for each data point in the data set which is used for the evaluation of a model.

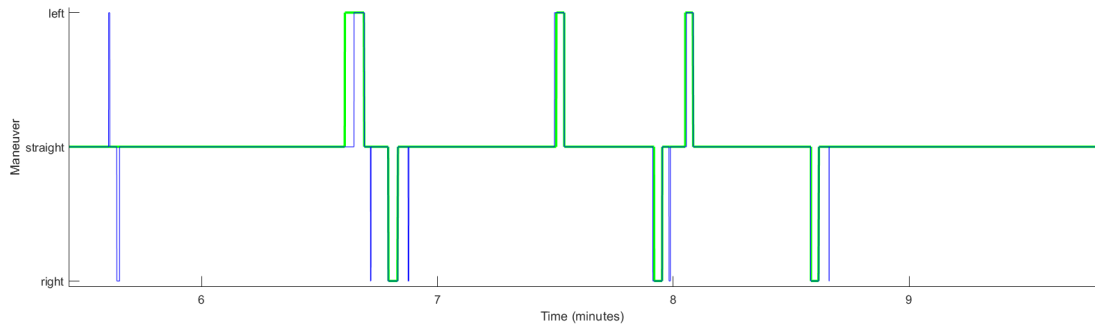
This rather simple visualization already enables an extensive evaluation of the prediction results. In contrast to the general F1 score it allows the inspection of particular areas as well as an overall assessment of the model's performance.

This visualization also allows an experienced observer to estimate precision and recall and therefore the F1 score.

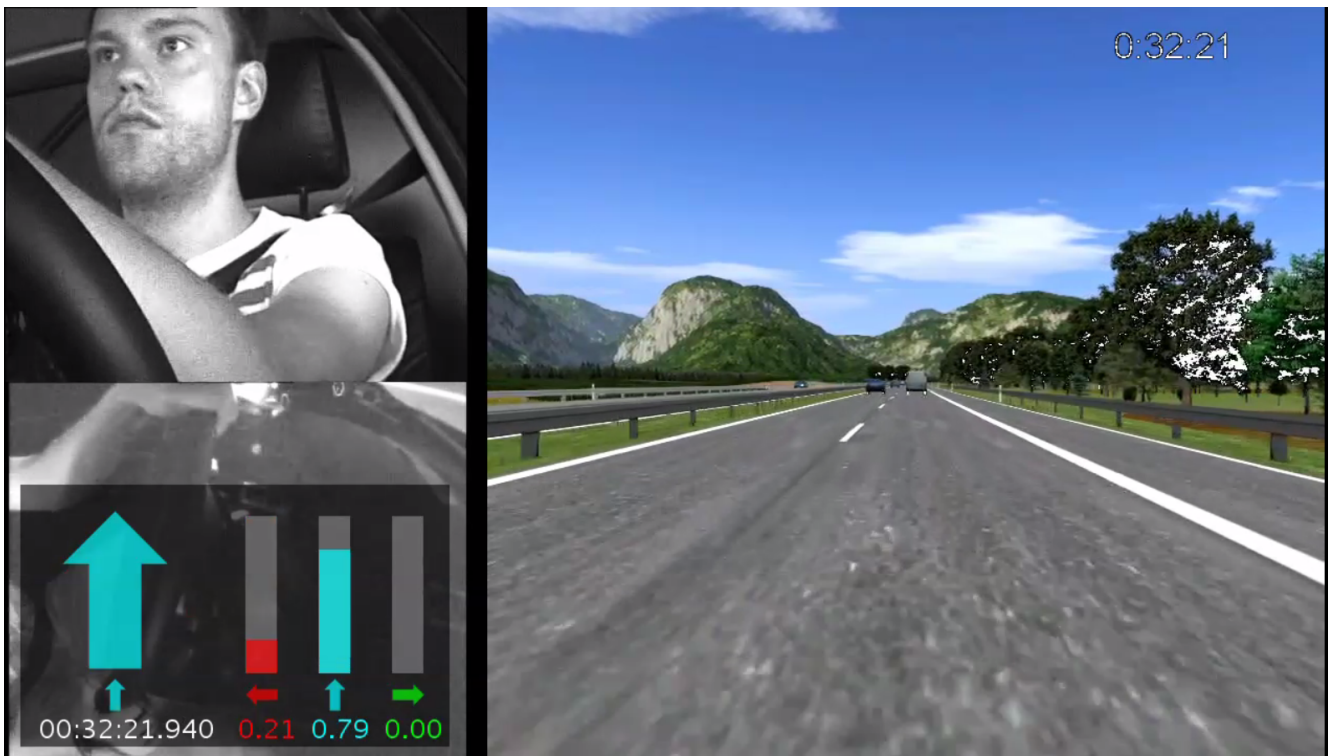
Figure 5.9 shows an excerpt of the described visualization variant. The best possible result would be an exact match of the predictions (blue) and the corresponding labels (green). Deviations between these lines represent discrepancies between prediction and label and can be used to estimate precision and recall of the visualized prediction.

### Variant 2: Video Overlay

An even more demonstrative visualization can be generated by adding information about the maneuver predictions and associated labels to the corresponding video of a record as shown in Figure 5.10. That allows a real time assessment of the model's classifications and the evaluation of particular feature influences onto the model's performance.



**Figure 5.9.:** Excerpt of the model’s predictions (blue) plotted against the assigned labels (green) over time. The x-axis shows the elapsed time in hundredth of a second, the y-axis shows the corresponding maneuver (1 = right lane change, 2 = lane keeping, 3 = left lane change)



**Figure 5.10.:** Frame of a driving video with overlay in the bottom left corner. The lefthand side of the box shows the current maneuver prediction and current label in form of a large and a small arrow. The righthand side contains the probabilities of each maneuver in form of a bar chart and corresponding numerical values.



---

## 6 Software/Implementation

---

The realization of the described process is implemented in MATLAB<sup>1</sup> as it provides an extensive set of tools and functionality for the processing of large data sets and the fast development of demonstrator software. Besides that, the recorded data is provided in a proprietary MATLAB format which can be comfortably used in the MATLAB-Environment.

---

### 6.1 Probabilistic Modeling Toolkit

---

As basis for the implementation of the Hidden-Markov-Model, the open source Probabilistic Modeling Toolkit for Matlab (PMTK3)<sup>2</sup> is used. It has been originated by the former UBC<sup>3</sup> professor Kevin P. Murphy as a companion to his Machine Learning textbook [16] and has been under active development ever since.

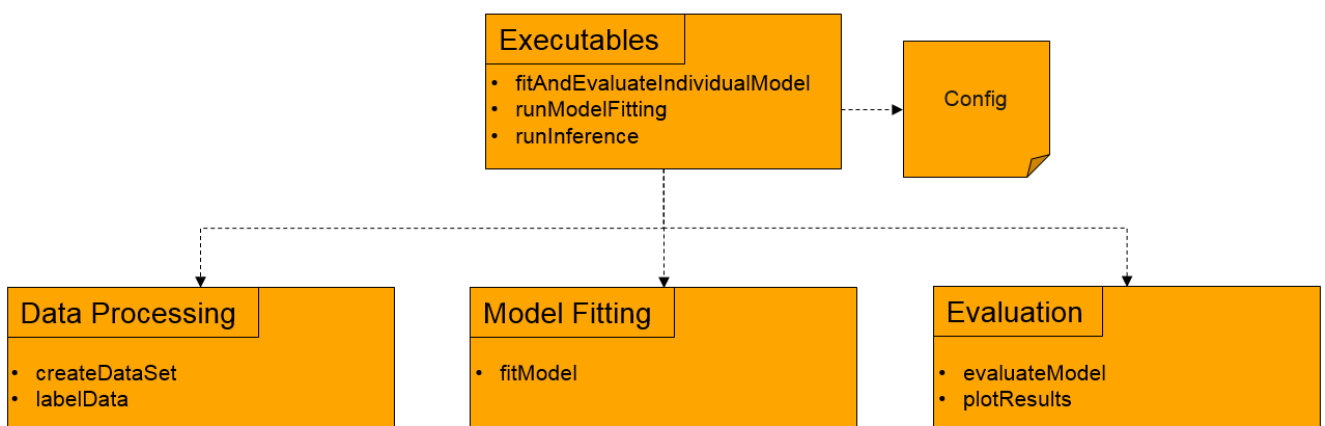
In this work, the implementation of HMMs included in the toolbox is used as a basis for the implementation of the presented models. That significantly reduces the implementation and testing effort and allows a more comprehensive model evaluation within the time available.

---

### 6.2 Architecture

---

Since the evaluation of feature influence, number of states, etc. requires the efficient execution of a huge number of model configurations, a simple software architecture, illustrated in Figure 6.1, is developed that encapsulates the fundamental steps which are required to train and evaluate a model and allows a flexible execution of different model configurations. The remainder of this section describes the individual parts of the architecture and how the software can be used to train and evaluate a model.



**Figure 6.1.:** Schematic illustration of the software architecture.

---

<sup>1</sup> <http://www.mathworks.com/products/matlab/>(19.02.2016)

<sup>2</sup> <https://github.com/probml> (19.02.2016)

<sup>3</sup> University of British Columbia (Vancouver, Canada)

---

As Figure 6.1 shows, the architecture mirrors the structure of the overall concept and can also be divided into a data processing and model creation part. In addition to that, it provides the functionality of running and evaluating a trained model.

## Model Configurations

In order to keep track of different model configurations and utilized feature sets, a model configuration can be defined using the createConfig script, that creates and stores a new configuration object defining the model's parameter configuration, the feature selection, etc.. The attributes, contained in such a configuration object are listed in Table 6.1.

---

### 6.2.1 Data Processing

---

The data processing package encapsulates the preparation of data sets for the training and evaluation of a model. It provides the functionality of extracting a desired feature selection from a given set of records and performing the maneuver labeling. This allows the comfortable execution of the model creation process on multiple records using different variations of feature selections.

---

### 6.2.2 Model Training

---

The model package contains the functionality of training and evaluating different types of models and model configurations on pre-processed data sets. It provides a single entry-point for training a model based on the model type which is defined in the config file.

---

### 6.2.3 Evaluation

---

The implemented evaluation methods are combined in the evaluation package. It provides the possibility to evaluate a trained model on a test data set and to visualize the results.

---

### 6.2.4 Executables

---

This package provides the entry points for training and evaluating arbitrary model configurations on the simulator data set. As Figure 6.1 illustrates, the contained scripts are based on the functionality provided by the other packages. It provides:

#### **Training and Evaluation of Driver Specific Models (fitAndEvaluateIndividualModel)**

Trains and evaluates a driver specific model on each of the defined records using the specified configuration. Each record is split into a training and a test set using the holdout method described in 2.3.

#### **Training of a Model (runModelFitting)**

Trains a single model on the defined records using the specified configuration.

---

### **Execution/Evaluation of a Driver Independent Model (runInference)**

Executes a previously trained model on each of the defined records individually and evaluates the prediction performance.

Attribute	Description	Possible Values
id	Identifier of the configuration	Any integer value
modelType	The type of the model that is supposed to be executed.	gauss, baelineGauss, gauss-Disc, baselineGaussDisc
statesRight	Number of states utilized to model right lane changes (ignored when using baseline models)	Any positive integer value. Increasing this number also increases the model's complexity and the runtime of training/evaluating the model.
statesStraight	Number of states utilized to model driving straight (ignored when using baseline models)	Any positive integer value. Increasing this number also increases the model's complexity and the runtime of training/evaluating the model.
statesLeft	Number of states utilized to model left lane changes (ignored when using baseline models)	Any positive integer value. Increasing this number also increases the model's complexity and the runtime of training/evaluating the model.
desiredGaussSignals	The feature selection that is supposed to be modeled by a Gaussian distribution.	Any subset of continuously valuated features.
desiredDiscSignals	The feature selection that is supposed to be modeled by a discrete distribution. (ignored when using gauss or baselineGauss as modelType)	Any subset of discrete features.
predictionThreshold	The probability a maneuver needs to overcome to get selected in the inference process. (0 by default)	Any real number between 0 and 1
labeling	The method used to label the maneuvers. The labeling can either be done dynamically as described in METHODOLOGY or static by simply labeling a certain time period before the car crosses a lane marking. (dynamic by default)	dynamic, static
labelingDuration	The labeling duration when using static labeling. (Ignored when using dynamic, 2.5 sec by default)	Any positive real number.

**Table 6.1.:** Attributes of a configuration.

---

## 7 Results of Feature- and State Selection

---

This chapter presents the findings of the individual steps of the feature- and state selection processes which are described in 5 and determines the final feature and state selection that is used for the evaluation of the driver intention model, presented in 9.

---

### 7.1 Feature Pre-Selection

---

As described in 5.3.1, the list of coarse feature pre-selections, presented in 3.2.1, in this step is furtherly refined by the visual inspection of three randomly chosen records. Several features from that list are assessed as not relevant for the prediction of lane change maneuvers and are therefore discarded, resulting in the shortlist of candidates shown in Table 7.1.

Category	Features
Head Features	Head Heading, Gaze Heading
Exterior Features	Psi, Lateral Distance, Distance to Vehicle Ahead
Control Inputs	Steering Moment, Steering Wheel, Steering Wheel-v
Vehicle Dynamics	y-yaw, a-y, v-y

**Table 7.1.:** Short-listed Features

The detailed assessment of the inspected features can be found in Appendix A Table A.1.

---

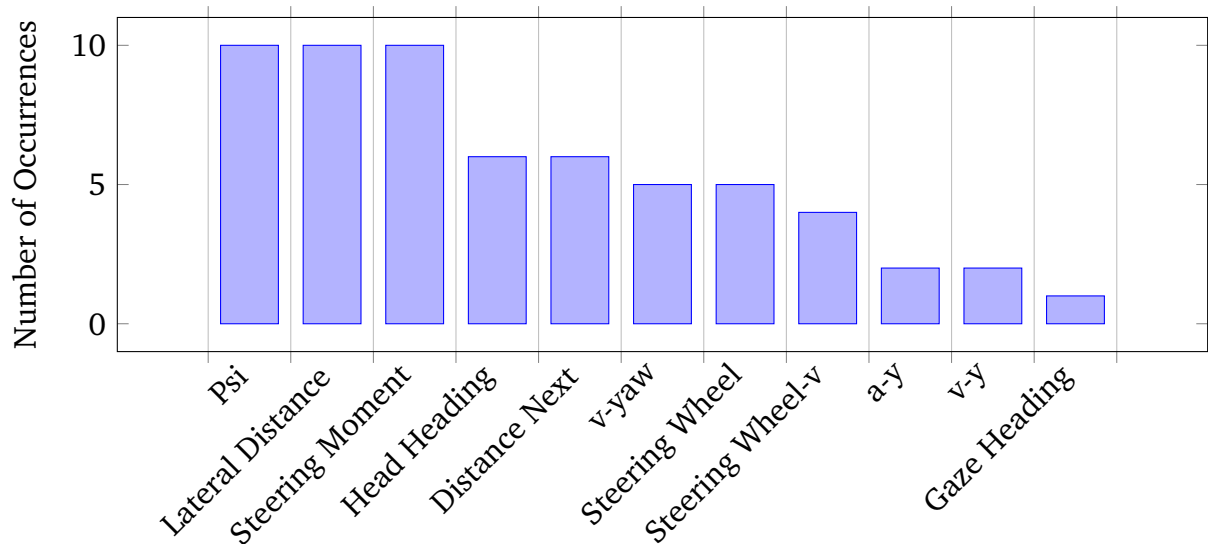
### 7.2 Feature Selection

---

The short-listed features resulting from the prior pre-processing step are furtherly assessed using the wrapper method as described in 5.3.2.

Looking at the detailed results of this process, shown in AppendixA Table A.2, reveals that the best performing feature selections for the individually evaluated records as well as the achieved scores show a significant variance between the evaluated records.

However, as Figure 7.1 illustrates, some features can be identified which appear in a majority of the individual best performing selections and can therefore be considered as well suited features for the final model.

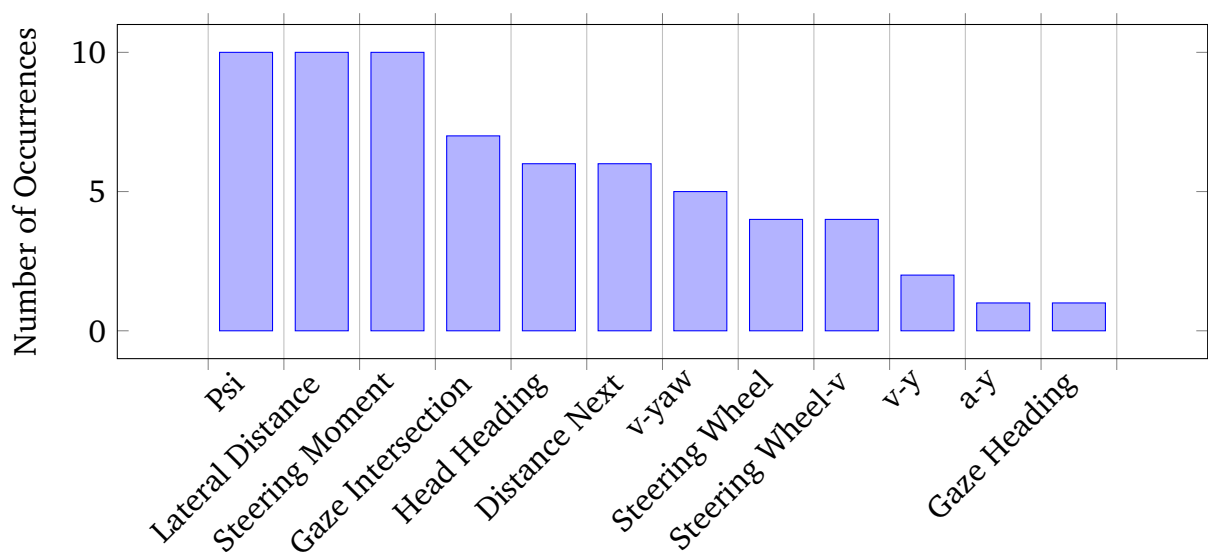


**Figure 7.1.:** Frequency of feature selections using a Gaussian model.

### Discrete Emission

As described in 5.4.3, the discrete Gaze Intersection feature seems to be a promising candidate for the prediction of lane change maneuvers. To evaluate the benefit of that feature and the described model variation, the wrapper method is performed again using a Gaussian/Discrete baseline model. The list of candidate features is extended by the Gaze Intersection feature, leading to a total number of 4094 evaluated combinations.

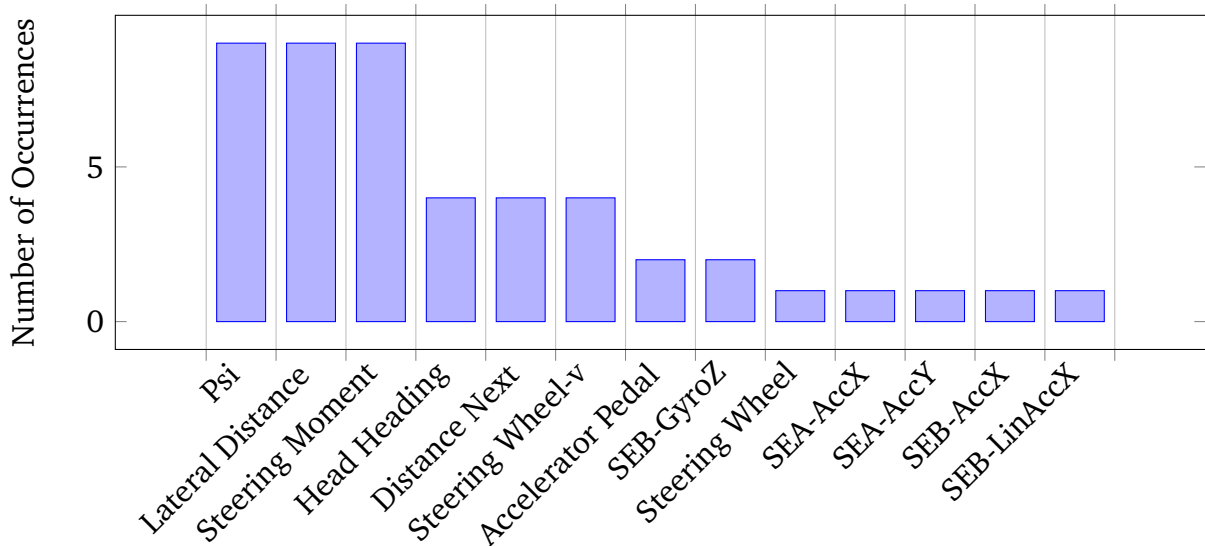
As Figure 7.2 shows, Gaze Intersection appears in half of the best performing selections, which makes it a promising candidate for the final selection. However, looking at the detailed results in Appendix A Table A.3 reveals that it does only marginally increase the score in comparison to the purely Gaussian model. A more detailed analysis of this feature's influence can be found in 8.3.



**Figure 7.2.:** Frequency of feature selections using a Gaussian/Discrete model.

## Alternative Selection Process

In addition to the combination of visual inspection and wrapper method, the alternative selection process, described in 5.3.2, is performed to validate the selected features. This iterative process is executed on the same records as before to generate comparable results. Figure 7.3 shows the frequency of the feature occurrence in the driver specific selections. A complete list of the selected features and their corresponding F1-Score can be found in Appendix A.4 Table A.4



**Figure 7.3.:** Frequency of feature selections using the alternative selection process and a Gaussian model.

## Final Selection

The determination of a final driver independent feature selection depends on the way the features are chosen, given their number of occurrences in the best performing selections. As the results in Appendix A show, choosing an optimal driver independent feature set is hardly possible, due to the variance between the different records. Therefore a tradeoff needs to be found which works reasonably well for multiple drivers.

In this work, the features are chosen according to their occurrences in the driver specific selections shown in Figure 7.1 and 7.3.

The three different feature sets, listed in Table 7.2 are evaluated against each other

Selection	Features
Shortlist	Head Heading, Gaze Heading, Lateral Distance, Psi, Distance Next, a-y, v-y, v-yaw, Steering Wheel, Steering Wheel-v, Steering Moment
Liberal Selection	Lateral Distance, Psi, Steering Moment, Head Heading, Distance Next, v-yaw, Steering Wheel
Strict Selection	Lateral Distance, Psi, Steering Moment

**Table 7.2.:** Evaluated feature selections.

<b>Selection</b>	<b>ØF1-Score</b>
Driver-specific	0.6664
Liberal Selection	0.6289
Liberal Selection + Gaze Intersection	0.6282
Strict Selection + Gaze Intersection	0.5701
Strict Selection	0.5540
Shortlist	0.4994

**Table 7.3.:** Performance of evaluated feature selections.

As Table 7.3 shows, the average results of a liberal feature selection come closest to the results of driver specific selections. Even though the Gaze Intersection is part of many driver specific selections, it does not increase the result when added to the liberal selection. Therefore, in the following processes and for the training of the final driver intention model, the liberal selection is used.

### 7.3 State Selection

The remaining model parameter which needs to be chosen, is the number of states used for the individual maneuvers. As described in 5.4.2, this includes an iterative process that evaluates an extensive number of state configurations. The feature selection is set to the liberal selection, presented in 7.2 as it showed the best driver independent performance. Inspecting the detailed results in Appendix B, Table B.1 reveals that the state selection, leading to the best evaluation, again varies between different drivers. As presumed in 5.4.2, almost every best performing selection only chooses a small number of states for lane change maneuvers and a significantly higher number of states for lane keeping.

This is due to the fact that the training data contains a substantially larger amount of data for lane keeping than for lane change maneuvers. The variance in the observation when driving straight is therefore a lot higher than those in the maneuver data.

The final state selection, that results in the best average performance is shown in Table 7.4. Looking at Table 7.5 reveals that the final driver independent selection performs again worse than the best performing driver individual selections. However, it is clearly better than the baseline model.

	<b>Lane Change Right</b>	<b>Lane Keeping</b>	<b>Lane Change Left</b>
<b>States</b>	<b>1</b>	<b>7</b>	<b>1</b>

**Table 7.4.:** Final state selection.

<b>Selection</b>	<b>ØF1-Score</b>
Driver Specific	0.7428
Driver Independent	0.7090
Baseline	0.6289

**Table 7.5.:** Evaluation of different state selections.



---

## 8 Influence of Features and Feature Categories

---

In order to get more insight into the selection process and to assess the importance of particular features and feature categories, the shortlisted features are individually evaluated.

These features can be grouped in four different categories which seem to have differently strong influence on the model's performance as they occur in the best performing selections for records 15-25 with varying frequency (see Figure 8.1).

Exterior features and control inputs seem to be of special importance, as they are contained in every selection that results in the best performance on the evaluated records. Head features and vehicle dynamics are also frequently selected, however they occur a bit less frequent. In the remainder of this chapter, the feature utilization is furtherly analyzed and the influence of the different categories as well as individual features is evaluated.

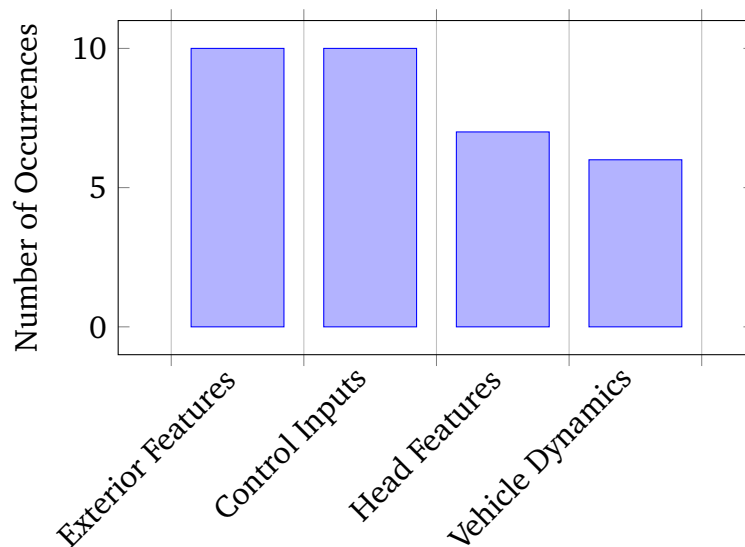


Figure 8.1.: Frequency of features from different categories.

---

### 8.1 Exterior Features

---

Looking at the results in Figure 8.1 indicates, that exterior features seem to have a huge influence in the model's performance. In particular, Lateral Distance and Psi are of special interest, as they are part of every single best performing feature selection.

This is comprehensible, considering that the labeling algorithm marks a certain time slot before actual lane changes as intention. In these marked areas, Lateral Distance and Psi show a distinct behavior which is unique for the labeled maneuver: the angle between vehicle and lane increases when the driver starts to pull over and lateral distance raises when the vehicle approaches the new lane. That makes these features to reliable and strong indicators for lane change maneuvers.

Distance to vehicle ahead is also an indicator for a lane change maneuver, but it is not as strong as both of the other exterior features. The reason for that is, that the feature's behavior, in the

areas marked as intention, is not completely unique and can also be observed in areas marked as lane keeping.

As one can see in Table 8.1, removing all exterior features results in a extremely poor model performance. Removing Lateral Distance and Psi individually has a similar, but not as drastic effect. Distance Ahead seems to have the smallest influence on the performance since its removal only causes a small drop in the evaluation score. These observations reflect prior observations, and confirm the importance of Lateral Distance and Psi.

Feature Selection	ØF1-Score
Shortlist	0.4994
Shortlist w/o Exterior Features + Lateral Distance + Psi	0.4603
Shortlist w/o Exterior Features + Lateral Distance + Distance Next	0.3886
Shortlist w/o Exterior Features + Psi + Distance Next	0.3801
Shortlist w/o Exterior Features + Lateral Distance	0.3430
Shortlist w/o Exterior Features + Psi	0.3340
Shortlist w/o Exterior Features + Distance Next	0.2558
Shortlist w/o Exterior Features	0.2088

**Table 8.1.:** Influence of Exterior Features.

## 8.2 Control Inputs

As shown in Figure 8.1, features from the control input category are also part of every driver specific feature selection. In particular, Steering Moment appears in each of the best performing selections.

The further analysis of this category, shown in Table 8.2, indicates that the influence of these features is in fact relatively small. Adding and removing single features only results in marginal changes of the evaluation score. Steering Moment seems to be the feature with the most positive influence but can also lead to a small decrease of the performance score when used in combination with Steering Wheel.

However, due to their small magnitude, it might be that not all of the observed impacts are statistically relevant.

Feature Selection	ØF1-Score
Shortlist w/o Control Inputs + Steering Wheel-v + Steering Moment	0.5485
Shortlist w/o Control Inputs + Steering Moment	0.5328
Shortlist w/o Control Inputs + Steering Wheel-v	0.5280
Shortlist w/o Control Inputs	0.5084
Shortlist	0.4994
Shortlist w/o Control Inputs + Steering Wheel + Steering Moment	0.4903
Shortlist w/o Control Inputs + Steering Wheel + Steering Wheel-v	0.4836
Shortlist w/o Control Inputs + Steering Wheel	0.4740

**Table 8.2.:** Influence of Control Inputs.

### 8.3 Head- and Gaze Features

Features from this category are also present in most of the best performing feature selections. The dominant feature seems to be Head Heading as it appears much more often than Gaze Heading. The results in Table 8.3 also show, that Head Heading is in fact the feature that has the positive influence while Gaze Heading seems to decrease the model's performance.

As Table 8.4 shows, adding the discrete Gaze Intersection feature does not have a negative influence but also does not lead to a better result than using Head Heading alone.

This discovery is in accordance with the examination published in [7] which states that the driver's head movement is the sufficient feature for the determination of the drivers head activity in the domain of lane change prediction.

Another explanation for Gaze Heading's negative influence might be its poor quality in comparison to Head Heading. As Table 8.5 shows, the quality of Gaze Heading is significantly lower than those of Head Heading, especially in areas marked as a right lane change.

Feature Selection	ØF1-Score
Shortlist w/o Head Features + Head Heading	0.5657
Shortlist	0.4994
Shortlist w/o Head Features	0.4966
Shortlist w/o Head Features + Gaze Heading	0.4740

**Table 8.3.:** Influence of Head Features.

Feature Selection	ØF1-Score
Shortlist w/o Head Features + Head Heading + Gaze Intersection	0.5665
Shortlist w/o Head Features + Gaze Intersection	0.5004
Shortlist	0.4994
Shortlist + Gaze Intersection	0.4988
Shortlist w/o Head Features + Gaze Heading + Gaze Intersection	0.4756

**Table 8.4.:** Influence of Head Features + Gaze Intersection.

Feature	Ø Loss (Overall)	Ø Loss (Labeled Right)	Ø Loss (Labeled Left)
Head Heading	2.74 %	3.22 %	3.17 %
Gaze Heading	14.66 %	17.29 %	14.85 %

**Table 8.5.:** Amount of invalid data in Head Heading and Gaze Heading, determined on records 1-25.

---

## 8.4 Vehicle Dynamics

---

As Figure 8.1 shows, features representing the vehicle's dynamics occur least frequently in the driver specific selections. The further analysis of the individual features of that category, shown in Figure 8.6, reveal that the feature's influences are again quite small. A distinct evaluation of the feature's influences can therefore hardly be performed as the variations might again be not statistically relevant.

Feature Selection	ØF1-Score
Shortlist w/o Vehicle Dynamics + v-yaw	0.5512
Shortlist w/o Vehicle Dynamics + v-y + v-yaw	0.5364
Shortlist w/o Vehicle Dynamics	0.5313
Shortlist w/o Vehicle Dynamics + v-y	0.5277
Shortlist w/o Vehicle Dynamics + a-y + v-yaw	0.5231
Shortlist w/o Vehicle Dynamics + a-y	0.5153
Shortlist	0.4994
Shortlist w/o Vehicle Dynamics + a-y + v-y	0.4743

**Table 8.6.:** Influence of Vehicle Dynamics.

---

## 9 Evaluation of the Driver Intention Model

---

After evaluating the influence of individual features and finding a driver independent state selection, the final driver intention model is evaluated on the remaining records 26-37 (32 is missing). These records have not been touched so far and therefore allow an unaffected assessment of the model's performance and its generalization abilities.

---

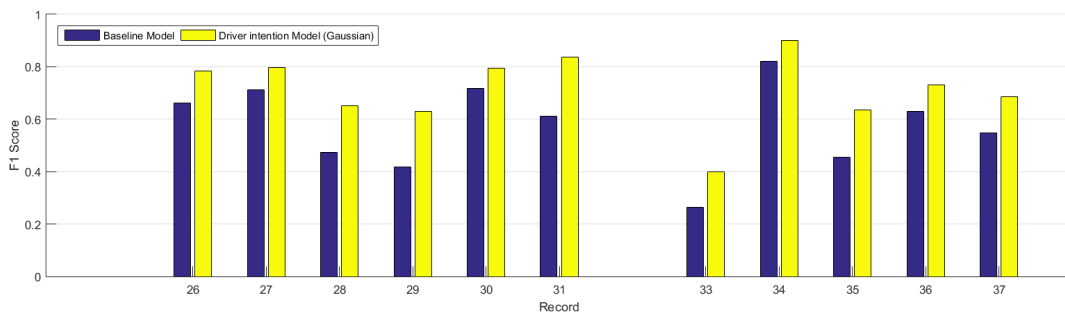
### 9.1 Diver Intention Model vs. Baseline Model

---

In this section the performance of the driver intention model is evaluated against the baseline model to examine if the effort of training three separate models and combining them to a driver model is justified by the fact that it results in a distinctly better performance.

Both models are trained on records 1-25 using the liberal feature selection and are tested on each individual record 26-37. The number of states of the driver intention model are furthermore set to the final state selection.

This process allows the assessment of the model's ability to infer the intention of an unknown driver after being trained on a data set containing the driving data of multiple drivers.



**Figure 9.1.:** Performance comparison of the driver intention model and the baseline model.

As Figure 9.1 illustrates, the driver model clearly outperforms the baseline HMM in every instance and increases the F1 score by around 15 % on average.

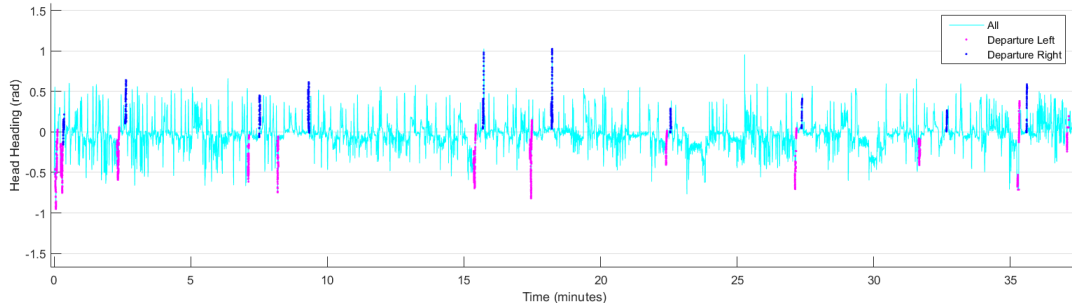
It can also be observed, that there is a huge variance in the model's performance between different records which can again be explained by the variance between the behavior of different drivers, described in 3.2.3. Especially record 33 causes a very poor performance and is therefore furtherly analyzed.

Detailed scores of these models can be found in Appendix C Table C.1 and C.2.

#### Record 33

Inspecting the course of the driver's head movement, shown in Figure 9.2, reveals that the reason for the poor performance lies in the driver's exceptionally high frequency of head activity, compared to other drivers. While most of them show a rather moderate frequency of head movements, this driver constantly shows a high head activity throughout the entire record.

Here lies one of the essential disadvantages of driver independent models: If the behavior of the driver, whose intention is supposed to be inferred, fundamentally differs from the behavior of the drivers used for the training of the model, it results in a poor performance. This leads to the concept of driver specific models which is discussed in 9.4.



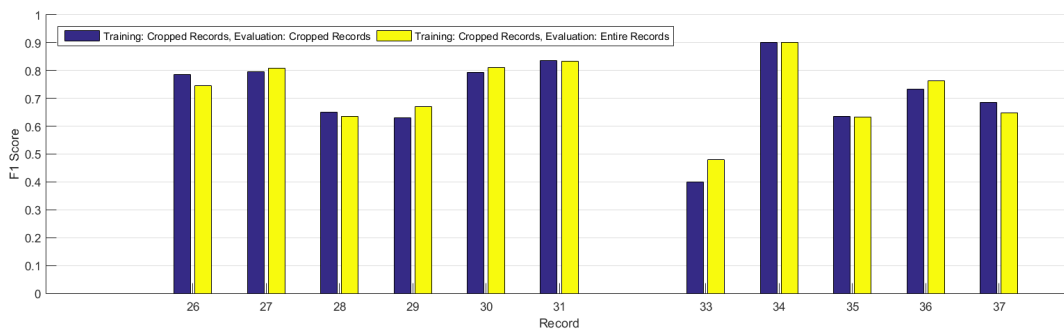
**Figure 9.2.:** Extraordinary head movement of driver 33.

## 9.2 Cropped Records vs. Entire Records

As described in 3.2.4, the feature- and state selection processes are performed on cropped versions of the records where the last 15 minutes of each records are discarded due to the fact that the drivers were ordered to stop checking the car’s environment in that part of the records.

It is nevertheless interesting to see how that drastic change in behavior influences the performance of a driver independent model.

To evaluate that impact, the model which has been trained on the cropped version of the records is now evaluated on the entire records 26-37.



**Figure 9.3.:** Performance comparison of the driver intention model on cropped and uncropped versions of the records.

The results, illustrated in Figure 9.3, show that the performance impact of the reduced activity is actually relatively small and the model does even achieve slightly better results on some of the uncropped records. Since the uncropped version contains the same data as the cropped version plus some extra data in the end, an improvement of the performance means that the model works exceptionally well on that last 15 minutes of data.

That might first be surprising but can be explained by the way the intention is labeled by the labeling algorithm.

Since the label duration is based on the driver’s head activity, the labeled lane change maneuvers

in the last 15 minutes of the records are quite short (mainly 2 seconds) and therefore contain distinct values for Psi and Lateral Distance as the car already starts to deviate from the center of the current lane.

The model is therefore still able to predict lane changes even though there is no head activity. The prediction time, however, is significantly shorter as in cases where the driver actively checks the car's environment.

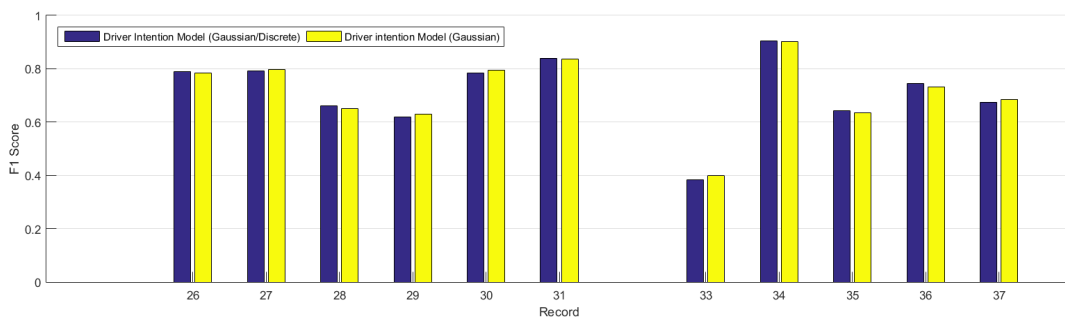
---

### 9.3 Gaussian Model vs. Gaussian/Discrete Model

---

Even though the feature selection process comes to the conclusion that adding the discrete Gaze Intersection feature does not noticeably improve the model's performance it is still evaluated on the independent test set to support the results of the selection process and to see if they hold true for the yet unseen data.

As expected, the results in Figure 9.4 show that adding Gaze Intersection to the list of utilized features does neither considerably increase nor decrease the performance of the Gaussian Model throughout all records.



**Figure 9.4.:** Performance comparison of Gaussian and Gaussian/Discrete model.

Detailed scores of these models can be found in Appendix C Table C.2 and C.4.

---

### 9.4 Driver Specific Training vs. Driver Independent Training

---

All evaluation so far are limited to driver independent models that are trained on a broad variety of drivers and are supposed to work for every unknown driver.

This concept is based on the premise, that different drivers, to some extent, show a similar behavior when driving on a highway and performing lane change maneuvers. The results, illustrated in Figure 9.1, reveal that this does not hold true for every driver and therefore causes the concept to work differently well for different drivers.

To overcome that issue, one could use individually trained models for each driver which would be able to adapt to specific characteristics of the driver's behavior. In contrast to a driver independent model, such an individual model also requires an individual training process before it can be used for the inference of the driver's intention. In a real world application it would therefore need a certain training time before it could actually be used on a new driver while a driver independent model could be used instantly, as the training process has been performed beforehand.

An individual training process however has the advantage that it does not need to cover the

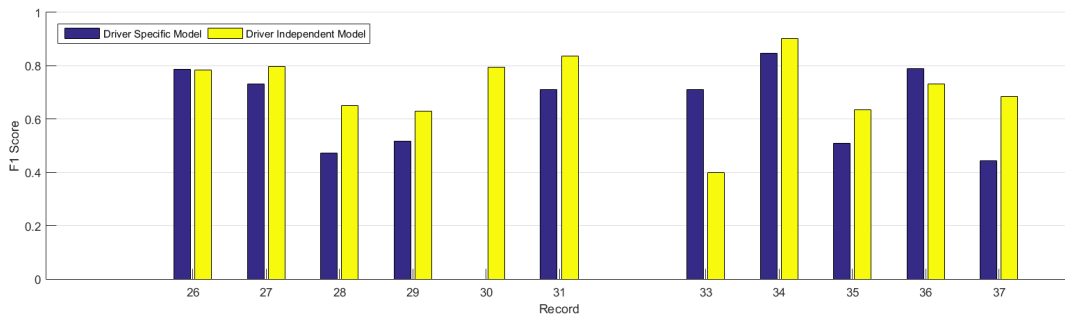
---

variety of different drivers and can be fitted on the specific behavior of a single driver. In order to evaluate that concept, a driver specific model is trained and evaluated on each of the records 26-37.

Unfortunately the records do not contain enough lane change maneuvers to run an entirely individual process including feature- and state selection, especially after discarding the last 15 minutes of driving time. Executing the entire selection machinery would require splitting each record into a training, validation and test part resulting in very small test- and validation sets containing only 3-5 lane change maneuvers in each direction.

Therefore, the individually trained models are not entirely driver specific as they use the results of the feature- and state selection, presented in 7.

The actual training process is however performed individually on each record using the holdout approach described in 2.3.



**Figure 9.5.:** Performance comparison of driver specific models and driver independent model.

As Figure 9.5 shows, the driver specific models, in most cases, do not perform as well as the driver independent model. It has to be considered, that these models are trained on a very small training set, containing only around 20 minutes of driving time and a couple of lane change maneuvers. The quality of the specific models might therefore not be as high as it could be if it was trained on a sufficient amount of data. It was even not possible to fit a model with the same configuration as the driver independent model on record 30 due to insufficient data. A however promising result is achieved on the critical record 33. Compared to the original model, the driver specific model works exceptionally well and seems to be able to adapt to the unusual behavior of that driver.

This indicates, that driver specific models have the ability to work even on exceptional drivers.

Detailed scores of these models can be found in Appendix C Table C.2 and C.3.

---

## 9.5 Visualization of Prominent Results

---

In order to allow the reader to obtain a better understanding of the evaluation results and to provide a more comprehensible representation, some meaningful results are visualized here.

### Best Performance

The best result is achieved using the driver intention model is shown in Figure 9.6. It visualizes the classifications of the Gaussian/Discrete model on record 34 which almost perfectly match the labeled maneuvers. Inspecting the corresponding head activity of that driver, illustrated in Figure 9.7 shows that the driver moves his head in a very controlled way with only



little activity when driving straight and distinct values when a lane change is executed. This is also true for other features which also show very little noise throughout the course of the drive.

The model's classifications are therefore made on the basis of a high quality test set which leads to an exceptionally high performance.

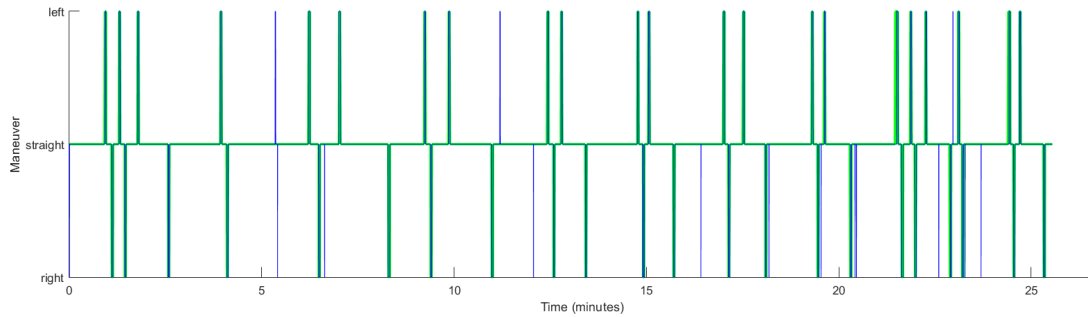


Figure 9.6.: Best overall prediction result. Achieved by Gaussian/Discrete model on record 34.

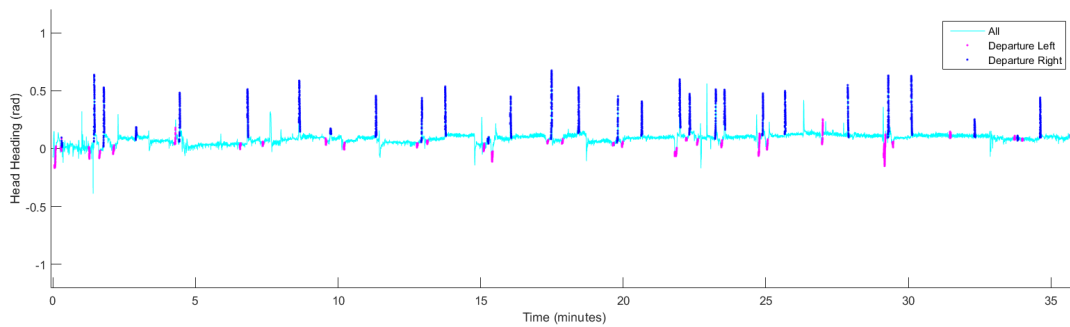


Figure 9.7.: Illustration of head movement of driver 34.

### Worst Performance

Inspecting the record that causes the worst classification results shows quite the opposite feature behavior to the record that caused the best result. As illustrated in Figure 9.9, the features show a high variance throughout the entire drive, which makes it hard for the model to detect actual lane change maneuvers.

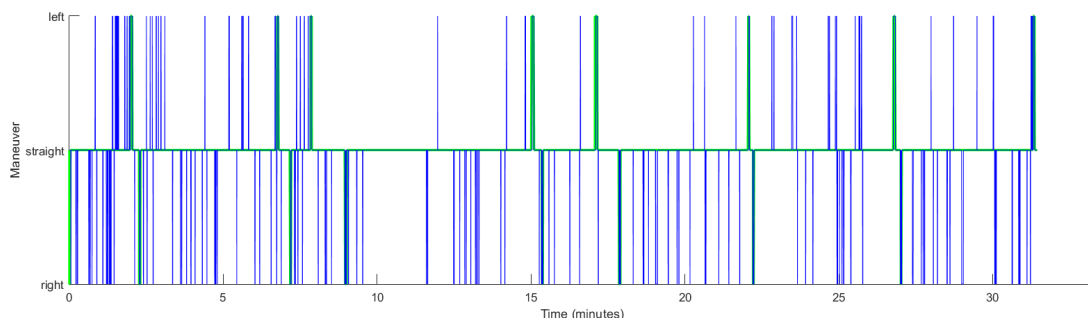
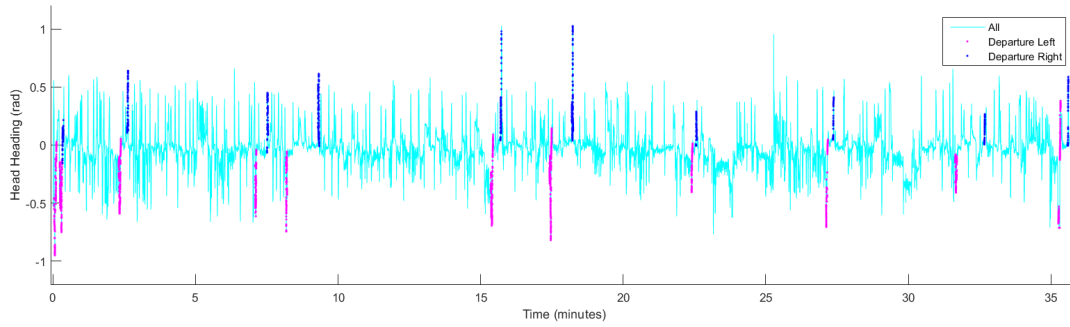


Figure 9.8.: Worst overall prediction result. Achieved by Gaussian/Discrete model on record 33.



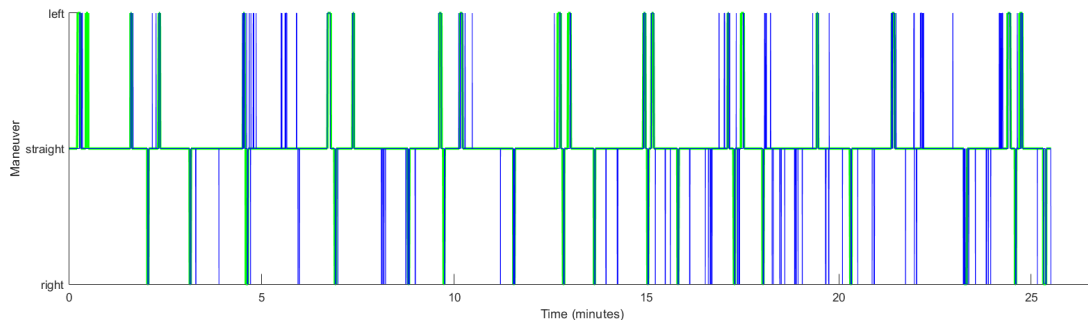
**Figure 9.9.:** Illustration of head movement of driver 33.

### Average Performance

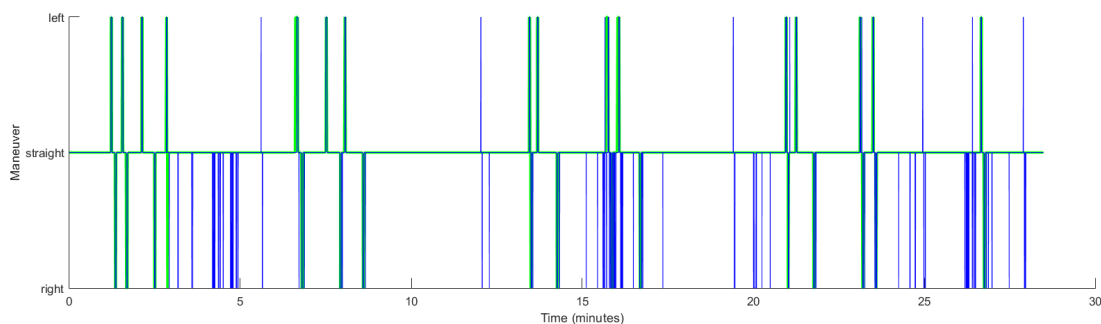
Figures 9.10 and 9.11 show the visualization of two average results, both achieving a F1 score of around 65 %.

The key difference between them is that even though the F1 score is nearly identical, precision and recall are off by almost 10 %.

Looking at the visualizations reveals that these two records show different patterns of false positive predictions and missed predictions even though they return the same F1 score. This shows the importance of using such a comprehensible visualization in addition to a numerical performance measure as it allows a far more detailed assessment of the model's performance.



**Figure 9.10.:** Average prediction result. Achieved by Gaussian model on record 28.



**Figure 9.11.:** Average prediction result. Achieved by Gaussian model on Record 29.

---

## 10 Conclusion and Future Work

---

This work presents an extensive analysis of different aspects of training and evaluating a Hidden-Markov-Model on simulated driving data. In this chapter an overall conclusion regarding the utilization of Hidden-Markov-Models for the prediction of lane change maneuvers is given and the key discoveries which have been made during the progress of this work are recapitulated.

---

### 10.1 Conclusion

---

Looking at the results of the previous chapter shows that Hidden-Markov-Models are well suited for the prediction of lane change maneuvers based on the inference of the driver's intention. Their unique temporal structure makes them particularly eligible for handling arbitrary time sequences of driving data and defines a fast and easy to implement inference algorithm that does allow the online execution on limited hardware.

The remainder of this section describes the key discoveries of this work, discusses their impact on the evaluation results and ponders on possible alternate approaches.

#### Labeling the Intention

One of the most crucial factors that influences almost every subsequent part of the learning and evaluation process is the labeling of the driver's intention. As described in 5.1, this work utilizes a automatic labeling method based on the crossing of lane markings and the driver's head movement. One of the drawbacks of this approach is, that it is not able to label lane change intentions when no actual maneuver is executed afterwards. This might be the case when a driver intends to perform a lane change but does not actually execute it because of the current traffic situation. That leads to a false negative labeling of such events which might decrease the model's performance.

A possible alternative method of labeling the intention could be using clusters of high head activity as the indicator for intended lane change maneuvers without taking actual lane changes into account. This approach should be able to overcome the described problem, but might instead cause false positive labels.

#### Driver Variance

Another important discovery, which manifests itself in almost every part of this work, is the huge variance between the behavior of different drivers. Almost every result, presented in 7, returns distinctly different outcomes for different drivers. It is therefore nearly impossible to find a driver independent model configuration that is guaranteed to perform reasonably well for every possible driver.

That leads to the concept of driver specific models which are individually trained for a certain driver. As the Figure 9.5 shows, a driver specific model seems to have the ability to also perform well on exceptional drivers like driver 33. Unfortunately the available amount of data per driver does not allow a more detailed examination of driver specific models in this work.

That would require a substantially larger data set to be able perform an individual feature- and state selection process as well as the final evaluation. Optimally that data would consist

---

of several records, captured on different days and times to capture the natural variance in the driver's behavior.

### **Artificial Data**

As described in 3, the data basis of this work is provided by a driving simulator. All processes, presented in this work, are therefore executed on data with superior quality, containing little noise and very few false measurement (with the exception of Gaze Features). That allows the training and evaluation of the models on almost unedited and unfiltered data using training and inference algorithms evaluating individual samples of the utilized features.

When it comes to a real world application, the presented processes would, most likely, require several additional pre-processing/filtering steps to handle the much poorer data quality of real world measurements. It might also be reasonable to use window based features to increase the model's robustness against outliers.

### **Performance Measure**

The measure used in this work evaluates the performance of a model in a sample based way, comparing the outcome of a model and the corresponding label for each individual sample point. This is a very strict evaluation that does not consider any conceptual errors, which could be removed in a post-processing step.

E.g. a model that does not cause any false positive predictions and identifies every single intention, but with a short delay of 0.5 seconds, would lead to a rather low score even though the model does in fact work quite well.

This is not an essential issue since all evaluations, presented in this work, consistently used this performance measure and therefore have been assessed equally strict. However, a well performing model should also result in a reasonably high evaluation score.

An interesting alternative approach is presented in [15], where two different data sets are used for evaluation. In addition to the regular data set, which contains the lane change maneuvers, a data set is recorded that solely contains lane keeping data. This lane keeping data is used to evaluate false positives, as it certainly does not contain any intention. The regular data set is used to evaluate missed predictions and false predictions but in contrast to the method, utilized in this work, only evaluates a single data point on a certain point in time (e.g. 2 sec) before a lane change maneuver.

Using this approach would eliminate the disregard of conceptual errors and would also resolve the previously described issue of false negative labeling.

Unfortunately the available data set does not contain pure lane keeping records and therefore this method could not be utilized in this work.

---

## 10.2 Future Work

---

In addition to the adaptations, described in the previous section, there are several examinations that would be interesting to carry out.

Until now, all analyses and evaluations have been performed offline on already captured data. Even though the video overlay allows a quite well inspection of the model's behavior, it would be desirable to experience the model in a real-time execution.

The first steps, required to run the model in the driving simulator, have already been executed, however an actual test run could not be performed during the period of this work.

Even more interesting would be the integration into a real car, however, as mentioned before, this would very likely require putting a lot more effort in feature pre-processing steps and the usage of window based features to enable the model to run robustly on incomplete and noisy real world data.

Another valuable addition would be the investigation of online learning approaches, that are able to adapt to a particular driver over time.

A possible scenario would be training a general model in the way which is presented in this work and afterwards iteratively adapt to an individual driver. That would combine the best of both worlds, as the model could be used instantly for most drivers and is able to adapt to exceptional drivers over time.

Training a completely individual model would also be a worthwhile examination. This would include a driver specific feature- and state selection as well as the actual training process. As stated earlier, that would require substantially more data for an individual driver, preferably recorded in multiple test drives.

Comparing the performance of the utilized HMM variation to other approaches like AIO-HMMs [10] or RNNs [11] would also be an interesting thing to do. However, this requires a considerable amount of effort since certain testing conditions need to be established that allow an unbiased comparison of different models.

---

# Appendices

## A Feature Selection

### A.1 Pre-Selection

**Table A.1.:** Manual assessment of coarsely pre-selected features (++ = very good, + = good, o = OK, - = bad, – = very bad)

Feature	Record 02	Record 10	Record 25	Assessment
Gaze Heading	+	o	o	Increase of activity in proximity to lane change maneuvers noticeable, however also lots of activity throughout the record
Gaze Intersection	+	o	o	Correlation between gaze sectors and maneuver intention recognizable, however there is a high fluctuation throughout the record
Head Heading	++	++	++	Very stable feature with a high correlation to lane change maneuvers. Only little activity in the remainder of the records.
Psi	++	++	++	Very little noise and a distinct correlation to lane change maneuvers.
Lateral Distance	++	++	++	Very little noise and a distinct correlation to lane change maneuvers.
Distance Next	+	o	+	Correlation to left lane changes observable, however no apparent correlation to right lane changes
Accelerator Pedal	o	o	o	No distinctly different behavior in proximity to lane changes
Brake Pedal	-	o	o	No distinctly different behavior in proximity to lane changes
Steering Wheel	+	o	o	Little correlation to lane change maneuvers

*Continued on next page*

Table A.1 – Continued from previous page

Feature	Record 02	Record 10	Record 25	Assessment
Steering Wheel-v	o	o	+	Little correlation to lane change maneuvers
Steering Moment	+	o	+	Slightly increased values in proximity to lane change maneuvers noticeable.
v-yaw	+	+	+	Correlation to lane change maneuvers observable.
ay	+	+	o	Correlation to lane change maneuvers observable.
v-y	o	o	+	Correlation to lane change maneuvers observable.
Sensor A - AccX	-	o	-	Highly active signal, no noticeable correlation
Sensor A - AccY	o	-	-	Highly active signal, no noticeable correlation
Sensor A - GyroZ	-	o	-	Highly active signal, no noticeable correlation
Sensor A - LinAccX	-	-	-	Highly active signal, no noticeable correlation
Sensor A - LinAccY	-	-	-	Highly active signal, no noticeable correlation
Sensor B - AccX	o	o	o	Highly active signal, no noticeable correlation
Sensor B - AccY	-	-	-	Highly active signal, no noticeable correlation
Sensor B - GyroZ	-	-	-	Highly active signal, no noticeable correlation
Sensor B - LinAccX	-	-	-	Highly active signal, no noticeable correlation
Sensor B - LinAccY	-	o	-	Highly active signal, no noticeable correlation



---

## A.2 Gaussian Model

---

**Table A.2.:** Driver-Individual Feature Selection using the extensive search method and Gaussian Model

Record	Feature Selection	F1-Score
15	Steering Wheel, Steering Moment, a-y, Lateral Distance, psi, Distance Next	0.5313
16	Head Heading, Steering Wheel, Steering Moment, v-yaw, Lateral Distance, psi	0.7354
17	Gaze Heading, Steering Wheel-v, Steering Moment, v-yaw, v-y, a-y, Lateral Distance , psi, Distance Next	0.6908
18	Steering Moment, Lateral Distance, psi	0.5078
19	Head Heading, Steering Wheel, Steering Wheel-v, Steering Moment, v-yaw, Lateral Distance, psi, Distance Next	0.5897
20	Head Heading, Steering Wheel, Steering Moment, v-yaw, v-y, Lateral Distance, psi, Distance Next	0.6677
21	Steering Wheel-v, Steering Moment, Lateral Distance, psi, Distance Next	0.7512
22	Head Heading, Steering Wheel, Steering Wheel-v, Steering Moment, v-yaw, Lateral Distance, psi, Distance Next	0.7089
23	—	—
24	Head Heading, Steering Moment, Lateral Distance, psi	0.6914
25	Head Heading, Steering Moment, Lateral Distance, psi	0.7895

---

### A.3 Gaussian/Discrete Model

---

**Table A.3.:** Feature Selection using the extensive search method and Gaussian/Discrete Model

Record	Feature Selection	F1-Score
15	Steering Wheel, Steering Moment, a-y, Lateral Distance, psi, Distance Next, Gaze Intersection	0.5424
16	Head Heading, Steering Wheel, Steering Moment, v-yaw, Lateral Distance, psi , Gaze Intersection	0.7384
17	Gaze Heading, Steering Wheel-v, Steering Moment, v-yaw, v-y, a-y, Lateral Distance , psi, Distance Next	0.6907
18	Steering Moment, Lateral Distance, psi, Gaze Intersection	0.5192
19	Head Heading, Steering Wheel, Steering Wheel-v, Steering Moment, v-yaw, Lateral Distance, psi, Distance Next, Gaze Intersection	0.5916
20	Head Heading, Steering Wheel, Steering Moment, v-yaw, v-y, Lateral Distance, psi, Distance Next	0.6677
21	Steering Wheel-v, Steering Moment, Lateral Distance, psi, Distance Next, Gaze Intersection	0.7597
22	Head Heading, Steering Wheel, Steering Wheel-v, Steering Moment, v-yaw, Lateral Distance, psi, Distance Next	0.7088
23	—	—
24	Head Heading, Steering Moment, Lateral Distance, psi, Gaze Intersection	0.6915
25	Head Heading, Steering Moment, Lateral Distance, psi	0.7895

---

## A.4 Alternative Selection process

---

**Table A.4.:** Feature Selection using the alternative sequential search method.

Record	Feature Selection	F1-Score
15	psi, Steering Moment, Lateral Distance, SEB-LinAccX, SEB-GyroZ	0.5578
16	psi Head Heading, Lateral Distance, Steering Moment, Distance Next	0.6964
17	psi, Lateral Distance, Steering Wheel-v, Steering Moment, Accelerator Pedal, Distance Next	0.6964
18	psi, Steering Moment, Lateral Distance, SEB-GyroZ	0.5189
19	psi, Steering Wheel-v, Head Heading, Steering Moment, Lateral Distance, Steering Wheel, SEB-AccX	0.5890
20	psi, Lateral Distance, Steering Moment, SEA-AccX	0.6468
21	psi, Lateral Distance, Steering Moment, Accelerator Pedal, Distance Next, Steering Wheel-v	0.7661
22	psi, Lateral Distance, Steering Wheel-v, Steering Moment, Distance Next	0.6805
23	—	—
24	psi, Head Heading, Lateral Distance, Steering Moment, SEA-AccY	0.7125
25	Head Heading	0.7503

---

## B State Selection

---

**Table B.1.:** Results of the State Selection Process

Record	State Selection (right, lane keeping, left)	F1-Score	Baseline F1-Score
15	1,8,1	0.6344	0.4994
16	1,10,6	0.8309	0.6995
17	1,7,1	0.7658	0.6595
18	1,9,4	0.6031	0.4409
19	1,5,1	0.6499	0.5889
20	1,10,4	0.7772	0.6489
21	1,10,3	0.7968	0.6431
22	1,8,1	0.7669	0.7028
23	—	—	—
24	3,10,2	0.7346	0.6403
25	1,7,1	0.8680	0.7665

---

## C Model Variations

---

### C.1 Baseline Model

---

# Record	Precision	Recall	F1-Score
26	0.5302	0.8768	0.6608
27	0.6509	0.7878	0.7129
28	0.3432	0.7642	0.4736
29	0.2762	0.8623	0.4183
30	0.6690	0.7760	0.7185
31	0.4514	0.9442	0.6108
33	0.1550	0.8854	0.2637
34	0.7520	0.9034	0.8208
35	0.3203	0.7773	0.4537
36	0.5295	0.7768	0.6297
37	0.4268	0.7666	0.5483

Table C.1.: Performance of Baseline-HMM on Different Records

---

### C.2 Gaussian Model

---

#### Driver Independent

Record	Precision	Recall	F1-Score
26	0.7414	0.8329	0.7845
27	0.8758	0.7307	0.7967
28	0.6401	0.6625	0.6511
29	0.5469	0.7441	0.6304
30	0.8363	0.7559	0.7941
31	0.7599	0.9281	0.8357
33	0.2688	0.7753	0.3992
34	0.9248	0.8793	0.9014
35	0.6326	0.6365	0.6345
36	0.7702	0.6972	0.7318
37	0.7375	0.6410	0.6859

Table C.2.: Performance of Gaussian-HMM on Different Records

---

## Driver Specific

Record	Precision	Recall	F1-Score
26	0.7827	0.7886	0.7857
27	0.6573	0.8241	0.7313
28	0.4517	0.4954	0.4725
29	0.4131	0.6940	0.5179
30	0.0	0.0	0.0
31	0.5806	0.9122	0.7096
33	0.6388	0.8042	0.7120
34	0.9247	0.7833	0.8482
35	0.4453	0.5957	0.5096
36	0.8359	0.7470	0.7889
37	0.4634	0.4253	0.4436

**Table C.3.:** Performance of Gaussian-HMM trained and evaluated on each record individually.

---

## C.3 Gaussian/Discrete Model

---

Record	Precision	Recall	F1-Score
26	0.7468	0.8384	0.7900
27	0.8634	0.7326	0.7927
28	0.6598	0.6609	0.6604
29	0.5332	0.7351	0.6181
30	0.8325	0.7427	0.7850
31	0.7636	0.9300	0.8386
33	0.2568	0.7674	0.3848
34	0.9120	0.8976	0.9047
35	0.6342	0.6497	0.6418
36	0.7798	0.7105	0.7435
37	0.7202	0.6354	0.6752

**Table C.4.:** Performance of Gaussian/Discrete-HMM on Different Records

---

## List of Figures

---

2.1. Principle of overfitting . . . . .	10
2.2. Proceeding of a Feature Selection using the wrapper method. [20] . . . . .	11
2.3. Graphical model of a Hidden-Markov-Model [3] . . . . .	12
3.1. Birdseye view on the simulator setup. . . . .	17
3.2. The driving scenario. . . . .	18
3.3. Illustration of the car coordinate system. . . . .	20
3.4. Illustration of the head coordinate system . . . . .	20
3.5. Illustration of variance in head movement between driver 33 and 34. . . . .	21
3.6. Illustration of the lateral distance anomaly. . . . .	22
3.7. Illustration of the head heading anomaly. . . . .	23
5.1. Lateral Distance at lane crossings. . . . .	27
5.2. Lateral Distance and Head Heading at lane crossings. . . . .	28
5.3. Illustration of the labeling method. . . . .	28
5.4. Excerpt of head heading values. . . . .	29
5.5. Distance to vehicle ahead. . . . .	30
5.6. Illustration of the visualization plot. . . . .	31
5.7. Overall process of training the driver intention model. . . . .	33
5.8. Illustration of the Gaussian emission probabilities of two different models. . . . .	34
5.9. Visualization: Plot . . . . .	38
5.10. Visualization: Video overlay. . . . .	38
6.1. Schematic illustration of the software architecture. . . . .	39
7.1. Frequency of feature selections using a Gaussian model. . . . .	44
7.2. Frequency of feature selections using a Gaussian/Discrete model. . . . .	44
7.3. Frequency of feature selections using the alternative selection process and a Gaussian model. . . . .	45
8.1. Frequency of features from different categories. . . . .	47
9.1. Performance comparison of the driver intention model and the baseline model. . .	51
9.2. Extraordinary head movement of driver 33. . . . .	52
9.3. Performance comparison of the driver intention model on cropped and uncropped versions of the records. . . . .	52
9.4. Performance comparison of Gaussian and Gaussian/Discrete model. . . . .	53
9.5. Performance comparison of driver specific models and driver independent model. .	54
9.6. Best overall prediction result. Achieved by Gaussian/Discrete model on record .	55
9.7. Illustration of head movement of driver 34. . . . .	55
9.8. Worst overall prediction result. Achieved by Gaussian/Discrete model on record 33. . . . .	55
9.9. Illustration of head movement of driver 33. . . . .	56

---

9.10.Average prediction result. Achieved by Gaussian model on record 28. . . . .	56
9.11.Average prediction result. Achieved by Gaussian model on Record 29. . . . .	56



---

## List of Tables

---

3.1. Coarse pre-selection of candidate features . . . . .	19
3.2. Utilization of exterior features in real world applications. . . . .	24
5.1. Search space of the state selection process. . . . .	35
5.2. Discrete Gaze Intersection areas . . . . .	36
6.1. Attributes of a configuration. . . . .	42
7.1. Short-listed Features . . . . .	43
7.2. Evaluated feature selections. . . . .	45
7.3. Performance of evaluated feature selections. . . . .	46
7.4. Final state selection. . . . .	46
7.5. Evaluation of different state selections. . . . .	46
8.1. Influence of Exterior Features. . . . .	48
8.2. Influence of Control Inputs. . . . .	48
8.3. Influence of Head Features. . . . .	49
8.4. Influence of Head Features + Gaze Intersection. . . . .	49
8.5. Amount of invalid data in Head Heading and Gaze Heading, determined on records 1-25. . . . .	49
8.6. Influence of Vehicle Dynamics. . . . .	50
A.1. Manual assessment of coarsly pre-selected features . . . . .	61
A.2. Driver-Individual Feature Selection using the extensive search method and Gaussian Model . . . . .	63
A.3. Feature Selection using the extensive search method and Gaussian/Discrete Model . . . . .	64
A.4. Feature Selection using the alternative sequential search method. . . . .	65
B.1. Results of the State Selection Process . . . . .	66
C.1. Performance of Baseline-HMM on Different Records . . . . .	67
C.2. Performance of Gaussian-HMM on Different Records . . . . .	67
C.3. Performance of Gaussian-HMM trained and evaluated on each record individually. . . . .	68
C.4. Performance of Gaussian/Discrete-HMM on Different Records . . . . .	68

---

## Bibliography

---

- [1] S. Al-Sultan, A.H. Al-Bayatti, and H. Zedan. Context-aware driver behavior detection system in intelligent transportation systems. *Vehicular Technology, IEEE Transactions on*, 62(9):4264–4275, Nov 2013.
- [2] Jeff Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, Berkeley, CA: International Computer Science Institute, 1998.
- [3] C. Bishop. *Pattern Recognition and Machine Learning*. Springer Science+Business Media.
- [4] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.
- [5] S. Calinon, F. D’halluin, E.L. Sauser, D.G. Caldwell, and A.G. Billard. Learning and reproduction of gestures by imitation. *Robotics Automation Magazine, IEEE*, 17(2):44–54, June 2010.
- [6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.
- [7] A. Doshi and M. Trivedi. A comparative exploration of eye gaze and head motion cues for lane change intent prediction. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 49–54, June 2008.
- [8] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003.
- [9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation* 9(8), pages 1735–1780, 1997.
- [10] Ashesh Jain, Hema Swetha Koppula, Bharad Raghavan, and Ashutosh Saxena. Know before you do: Anticipating maneuvers via learning temporal driving models. *CoRR*, abs/1504.02789, 2015.
- [11] Ashesh Jain, Avi Singh, Hema Swetha Koppula, Shane Soh, and Ashutosh Saxena. Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. *CoRR*, abs/1509.05016, 2015.
- [12] P. Kumar, M. Perrollaz, S. Lefevre, and C. Laugier. Learning-based approach for online lane change intention prediction. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 797–802, June 2013.
- [13] Tzu kuo Huang, Ruby C. Weng, Chih jen Lin, and Greg Ridgeway. Generalized bradley-terry models and multi-class probability estimates. *Journal of Machine Learning Research*, 7:2006.

- 
- [14] S. Lefevre, Y. Gao, D. Vasquez, E. Tseng, R. Bajcsy, and F. Borrelli. Lane keeping assistance with learning-based driver model and model predictive control. In *12th International Symposium on Advanced Vehicle Control, 2014*, June 2014.
- [15] J.C. McCall, D.P. Wipf, M.M. Trivedi, and B.D. Rao. Lane change intent analysis using robust operators and sparse bayesian learning. *Intelligent Transportation Systems, IEEE Transactions on*, 8(3):431–440, Sept 2007.
- [16] Kevin P. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.
- [17] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. 2013.
- [18] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.
- [19] S. Russel and P. Norvig. *Künstliche Intelligenz: Ein moderner Ansatz*, volume 3. Pearson Studium.
- [20] Jiliang Tang, Salem Alelyani, and Liu Huan. Feature selection for classification: A review. Technical report, Arizona State University.
- [21] Michael E. Tipping and Alex Smola. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.