2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Clustering Vehicle Maneuver Trajectories Using Mixtures of Hidden Markov Models

John Martinsson[1], Nasser Mohammadiha[2], and Alexander Schliep[3]

Abstract— **The safety of autonomous vehicles needs to be verified and validated by rigorous testing. It is expensive to test autonomous vehicles in the field, and therefore virtual testing methods are needed. Generative models of maneuvers such as cut-ins, overtakes, and lane-keeping are needed to thoroughly test the autonomous vehicle in a virtual environment. To train such models we need ground truth maneuver labels and obtaining such labels can be time-consuming and costly. In this work, we use a mixture of hidden Markov models to find clusters in maneuver trajectories, which can be used to speed up the labeling process. The maneuver trajectories are noisy, asynchronous and of uneven length, which make hidden Markov models a good fit for the data. The method is evaluated on labeled data from a test track consisting of cut-ins and overtakes with favorable results. Further, it is applied to natural data where many of the clusters found can be interpreted as driver maneuvers under reasonable assumptions. We show that mixtures of hidden Markov models can be used to find motion patterns in driver maneuver data from highways and country roads.**

I. Introduction

Large amounts of data are being collected by leading car manufacturers to improve, verify, and validate self-driving systems. The number of self-driven miles needed to show that a self-driving system is safe enough to be accepted by the public is very large; in the order of millions or billions of driven miles [1]. This makes verification and validation of self-driving cars a costly process and an effort is being made to accelerate this by testing self-driving systems in virtual environments. Two widely studied problems for such virtual testing environments are the generation of maneuvers that occur on the road [2], and the generation of naturalistic sensor output [3]–[5]. Common for both these problems is that we usually rely on ground truth scenario labels for the data to successfully train scenario-specific models.

In this paper we aim at clustering maneuver trajectories in an unsupervised manner. The clustering algorithm is applied to large amounts of field data to extract different maneuvers. Such analysis speeds up labeling and therefore also the development of maneuver generation models and has several other important applications. First, it can be used to set up and implement realistic simulations for desired scenarios such as cut-ins and overtakes. Second, it can be used to extract new scenarios to extend the autonomous driving test catalogue. Third, it can be used to learn driver intention models for these various scenarios.

The data can include sequences of observations from sensors, such as radars, cameras and LiDARs. These observations can, for example, be approximations of the relative motion between the host vehicle and other tracked vehicles on the road, the distance from the host vehicle to the lane markings, or the number of lanes on the road. The hidden Markov model (HMM) is useful here because these sequences are of varying length and exhibit an asynchronous and noisy behavior. For example, when a car cuts in front of the host vehicle the actual lane-change can be delayed for different cars, resulting in an asynchronicity in the "cut-in motion trajectories".

General methods are needed to find patterns and natural groupings in this type of data to organize and understand them. Finding groups of similar motion trajectories helps us discover typical motion patterns, detect anomalies, create models of maneuvers, and summarize the data in comprehensible ways for further analysis. This is not trivially handled by standard clustering methods such as k-means or hierarchical clustering, since standard distance measures, for example the Euclidean distance, are not well defined for sequences of unequal length and also overemphasize asynchronicity in the sequences.

In this paper we study a probabilistic model-based clustering method, which is used to cluster maneuver trajectory data. The method uses a mixture of hidden Markov models (MHMM) to find natural groupings in the trajectory data. The HMM is a flexible model which can learn asynchronous and noisy behaviors, and it is not restricted to same length sequences, which makes it a good choice for this type of data. Further, by using a mixture of HMMs we can do soft assignment of motion trajectories. This allows trajectories which lie on the border between neighboring maneuvers to be partially assigned to several clusters. We show that the method can be used to find typical motion patterns in a data set of multivariate motion trajectories where each motion trajectory consists of the change in relative position and velocity of the host vehicle and the tracked vehicle on the road, observed at discrete time steps from the time when it enters the host vehicle's field of view until it exits its field of view.

This work is supported by the BADA-SEMPA project, which is partially financed by the Swedish government agency Vinnova.

[1]John Martinsson is with the Department of Computer Science and Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden (e-mail: john.martinsson@gmail.com)

[2]Nasser Mohammadiha is with Zenuity AB, New Technology, SE-417 56 Gothenburg, Sweden, and is Adjunct Faculty of the Department of Computer Science and Engineering, Chalmers University of Technology (e-mail: nasser.mohammadiha@zenuity.com)

[3]Alexander Schliep is joint Faculty of the Department of Computer Science and Engineering, Chalmers University of Technology and University of Gothenburg, SE-412 96 Gothenburg, Sweden (e-mail: alexander@schlieplab.org)

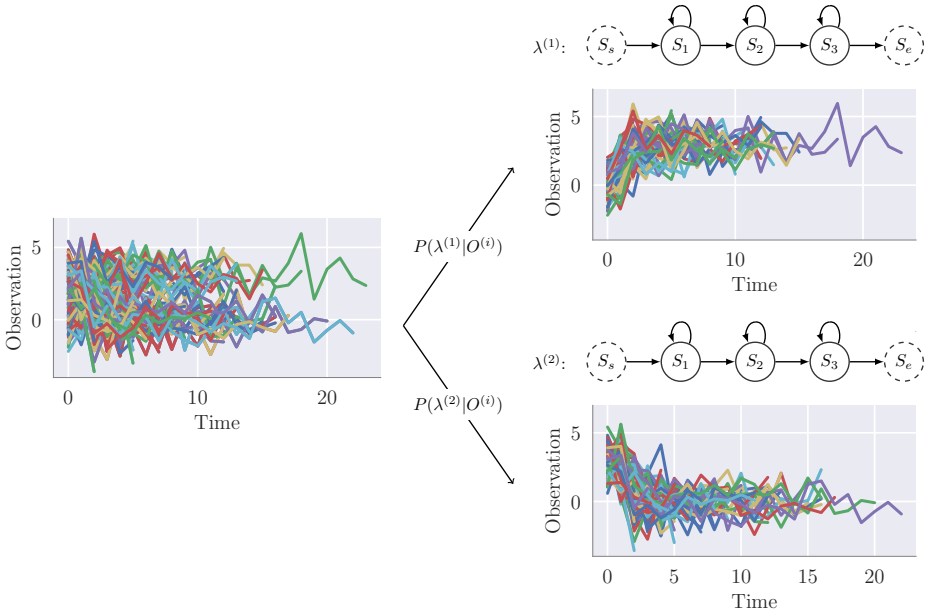

Fig. 1. We illustrate of how hard clusters can be derived from the mixture of HMMs. The mixture is first fitted to the original data (leftmost plot). The posterior probability, or responsibility, of the model given each sequence, $O^{(i)} \in \{O^{(1)}_{1:T_1}, O^{(2)}_{1:T_2}, \ldots, O^{(L)}_{1:T_L}\}$, of observations is then computed using Bayes's theorem, and a hard assignment is made for each sequence to the cluster which takes most responsibility for it. The first mixture component $\lambda^{(1)}$ learns the behaviour where the mean of the signal tends to increase over time, and the second mixture component $\lambda^{(2)}$ learns the behavior where the mean of the signal tends to decrease over time. Each mixture component is a left-to-right HMM with three hidden states, and the arrows in each HMM indicate the possible transitions. The dashed circles indicate the start state (S_s) and the end state (S_e).

II. Previous Work

To the best of our knowledge, HMMs were first used in [6] to cluster time series data using a "composite" HMM, which is equivalent to a mixture of HMMs. For the sake of clarity we keep to the mixture formulation as it is a more well-known framework given the large body of research on mixture models.

The composite HMM was evaluated on a data set generated by two different HMMs with favorable results. Note that the generated data matched the assumptions of the clustering method perfectly. A similar method was studied in [7], where a Bayesian clustering framework for temporal data was derived.

In [8] a mixture of HMMs was applied to motion trajectories derived by a face tracking system from video of a person using a "Camera Mouse" system, and in [9] it was used to analyze gene expressions in bioinformatics.

A similar system to the one presented in this paper, using a mixture of linear dynamic models, was used for unsupervised motion learning of relative movement between pedestrians and the host vehicle in urban environments [10], and the authors stress that unsupervised motion learning will have a great impact on driver assistance systems.

In this work we will further study a probabilistic model-based clustering approach for motion trajectories. In particular, we use mixtures of HMMs to perform unsupervised motion learning of the relative movement between the host vehicle and other tracked cars on the road.

III. Clustering Method

In this section we give a brief introduction to HMMs and mixtures of HMMs. See [11] for a more detailed introduction to HMMs and [8] for a more detailed overview of the parameter estimation formulas used to learn the model parameters.

A. Hidden Markov Model

A HMM can be used to model stochastic processes with an implicit temporal dependency among the observed values. In this paper, we use a first-order HMM with multivariate Gaussian observation densities, and assume that the states of the data-generating system fulfills the Markov property, that is, $\mathbb{P}(Q_t|Q_{t-1}, \ldots, Q_1) = \mathbb{P}(Q_t|Q_{t-1})$, see [11] for details.

An HMM is defined by a set of N states $S = \{1, 2, \ldots, N\}$, a transition probability matrix $A = \{a_{ij}\}$, a set of multivariate Gaussian emission distributions $B = \{b_j(\cdot)\}$ where $b_j(\cdot) = \mathcal{N}(\cdot; \mu_j, \Sigma_j)$, and the initial transition vector $\pi = \{\pi_j\}$, where $1 \leq i, j \leq N$.

Here, a_{ij} is the probability of making a transition from state i to state j, $b_j(\cdot)$ is the Gaussian density of state j defined by the mean vector μ_j and the covariance matrix Σ_j, and π_j is the probability of starting in state j; $1 \leq i, j \leq N$.

Let $\lambda = (\pi, A, \mu, \Sigma)$ be the complete parameter set of such a HMM. The likelihood of an observed sequence O can then be computed by

$$\mathbb{P}(O|\lambda) = \sum_{\text{all } Q} \mathbb{P}(O|Q, \lambda)\mathbb{P}(Q|\lambda), \tag{1}$$

using the forward-backward procedure [11], where $Q = \{q_1, \ldots, q_T\}$ is a fixed latent state variable sequence for the observed sequence $O = \{o_1, \ldots, o_T\}$.

To reflect asynchronous behavior in the same driving scenario, we use a left-to-right topology for each HMM in the mixture. Figure 1 illustrates how a mixture of HMMs can be fitted to synthetic data to learn two noisy asynchronous behaviors. The left-to-right topology restricts the transitions of the HMM, and it can thus be thought of as a piecewise constant stochastic function. Figure 1 shows that the first mixture component $\lambda^{(1)}$ learns a behavior where the mean of the signal tends to increase over time. That is, the Gaussian observation distributions in state one, two and three of the HMM each learn a mean that increases with the state number, and a roughly unit variance explaining the noise. The left-to-right transitions, illustrated with arrows in the figure, learn the asynchronous behavior, as is evident from the different time points at which change occurs. Similarly, the second mixture component $\lambda^{(2)}$ learns a behavior where the mean of the signal tends to decrease over time. The parameters $\lambda^{(1)}$ and $\lambda^{(2)}$ for the two HMMs can be learned from the data by estimating a solution to a maximum likelihood (ML) problem.

Given a finite set of finite multivariate sequences $\mathcal{O} = \{O^{(1)}_{1:T_1}, O^{(2)}_{1:T_2}, \ldots, O^{(L)}_{1:T_L}\}$, where $O_{1:T} = \{O_1, O_2, \ldots, O_T\}$ and T_i is the length of sequence i, we want to solve the ML problem

$$\lambda^* = \arg\max_{\lambda} \prod_{l=1}^{L} \mathbb{P}(O^{(l)}_{1:T_l}|\lambda). \tag{2}$$

The latent state variables make solving for global maximums impossible, therefore we use the Expectation-Maximization (EM) algorithm, known as Baum-Welch in the HMM literature [11], to find a local maximum. (See [8] for the complete parameter estimation formulas.)

B. Finite Mixture of Hidden Markov Models

In this section we present the model-based clustering method used in this paper, where the mixture components are HMMs. We assume that the true stochastic process, from which we have observed the sequences $O = \{O^{(1)}_{1:T_1}, \ldots, O^{(L)}_{1:T_L}\}$, is a mixture distribution of K hidden Markov models.

Let $\mathbb{P}(O|\lambda^{(k)})$ be the likelihood of sequence O given the kth HMM parameterized by $\lambda^{(k)}$, and let $w_l = (w_{l1}, w_{l2}, \ldots, w_{lK})$ be the cluster responsibility vector for the lth sequence, where K is the number of mixture components, or clusters, used interchangeably throughout the rest of the paper.

The cluster responsibility vectors constitute a soft clustering of the sequences. There is nothing, to the best of our knowledge, suggesting that there are only a finite set of driver maneuvers that occur on the road. Instead it is reasonable to believe that there are commonly occurring maneuvers leading to dense regions in a continuous space, with ambiguous maneuvers occurring between these dense regions. The clustering should reflect this. We thus defer the hard assignments until it is needed, and preserve the information about, and quantify, the uncertainty in the clustering.

The mixture of HMMs is parameterized by $\theta = \{\lambda^{(k)}, p^{(k)}\}_{k=1}^{K}$, where $\lambda^{(k)}$ is the kth HMM of the mixture, and $p^{(k)}$ is the mixing coefficient of mixture component k. The likelihood of an observed sequence $O^{(l)}_{1:T_l}$ given the MHMM parameterized by θ can then be computed using

$$\mathbb{P}(O^{(l)}_{1:T_l}|\theta) = \sum_{k=1}^{K} \mathbb{P}(O^{(l)}_{1:T_l}|\lambda^{(k)})p^{(k)}, \tag{3}$$

if we assume that $p^{(k)}$ is the prior probability of the kth mixture component. We then want to solve the ML problem

$$\theta^* = \arg\max_{\theta} \prod_{l=1}^{L} \sum_{k=1}^{K} \mathbb{P}(O^{(l)}_{1:T_l}|\lambda^{(k)})p^{(k)}, \tag{4}$$

but it is not known how to solve this optimally. Therefore, we use the EM algorithm. The iteration continues until a local optimum is reached. Note that the optimization surface may be non-convex with many local optima. It is therefore necessary to let the model converge many times with different random initializations. (See [8] for the complete parameter estimation formulas.)

C. Random Initialization

Initially each HMM has identical multivariate Gaussian emission distributions centered around zero with unit variance for each state. Because of the left-to-right topology of the HMM the transition probabilities are completely specified by the self transition probabilities in each state. The self transitions for each HMM are chosen such that the expected durations of the HMMs are uniformly distributed in the interval $[T_{\min}, T_{\max}]$, where $T_{\min}$ and $T_{\max}$ are the length of the shortest and longest sequences in the data respectively. The HMMs are then each fitted to the data set with weights $w \in [0, 1]$ drawn uniformly at random for each sequence. The HMMs, or mixture components, will then explain random sub-populations of the data which has been shown to be more effective than randomizing the mixture parameters [9]. The mixture of these HMMs is then fitted to the data.

IV. Model Selection

The clustering method presented requires that the number of clusters K, and the topology of the HMMs, is known beforehand. In most real-world applications this is not the case. Therefore, a method to choose among the many possible mixture configurations is needed. In statistics this problem is known as *model selection*, and a number of different criteria have been proposed to estimate solutions to this problem [12].

A commonly used criterion is the Bayesian information criterion (BIC) [13]. BIC is based on Bayes factors, and balances the likelihood of the data and the number of free parameters.

A. Integrated Classification Likelihood

We use the integrated classification likelihood criterion (ICL) [14], an extension of BIC that penalizes clusterings with a high uncertainty, or entropy. It is defined as

$$ICL_{K,M} = -2\mathcal{L}(\hat{\theta}_{K,M}|\mathcal{O}) + d_{K,M}\log N + 2EN_{K,M}, \quad (5)$$

where

$$EN_{K,M} = \sum_{k=1}^{K}\sum_{i=1}^{L} -\hat{w}_{ij}\log\hat{w}_{ij}, \quad (6)$$

is the entropy in the clustering, and

$$\hat{w}_{ij} = \mathbb{P}(\lambda^{(i)}|O^{(j)}_{1:T_j}; \hat{\theta}_{K,M}). \quad (7)$$

is the responsibility of the ith mixture component for the jth sequence. In (5) K denotes the number of mixture components, M denotes the number of hidden states in each mixture component, $\mathcal{L}(\cdot|\mathcal{O})$ is the likelihood function of the model parameters given the data, $\hat{\theta}_{K,M}$ is the ML estimate of the model parameters, and $d_{K,M}$ is the number of free parameters in the model. We then choose

$$K^*, M^* = \arg\min_{K,M} ICL_{K,M}, \quad (8)$$

where $1 \leq K \leq K_{\max}$ and $1 \leq M \leq M_{\max}$, which will be referred as a search over the model configurations.

B. Consensus Index

We also use the consensus index (CI) [15], which is based on the hypothesis that a good model configuration will find similar clusterings with different random initializations. Let $\mathcal{U}_{K,M} = \{U_1, U_2, \ldots U_B\}$ be a set of B clusterings, each found with K mixture components and M hidden states. The CI is then defined as

$$CI(\mathcal{U}_{K,M}) = \sum_{i<j} AM(U_i, U_j), \quad (9)$$

where AM is an agreement measure for the clusterings. We use the extended corrected rand index [16] as the agreement measure for the clusterings, which is an extension of the corrected rand index to soft clusterings. We then choose

$$K^*, M^* = \arg\max_{K,M} CI(\mathcal{U}_{K,M}), \quad (10)$$

where $2 \leq K \leq K_{\max}$ and $2 \leq M \leq M_{\max}$ as the optimal number of mixture components and hidden states.

V. Data Set

All data have been collected by the Volvo Car Corporation. The data consists of detected and tracked vehicles, obtained using a radar and a camera sensor. In addition, collected and processed data from a LiDAR sensor is also available in the dataset. The relative position between the host vehicle and other tracked vehicles, and their relative velocity are the features considered in this work.

Let the point $(x_t^{(l)}, y_t^{(l)})$ be a measurement of the relative lateral and longitudinal distance, at time t, between the host vehicle and the lth tracked car on the road, and let $(\dot{x}_t^{(l)}, \dot{y}_t^{(l)})$ be the relative velocity, where $1 \leq t \leq T_l$ and $1 \leq l \leq L$. A complete data set is defined by $\mathcal{O} = \{O^{(l)}_{1:T_l}\}_{l=1}^{L}$ where $O_t^{(l)} = (x_t^{(l)}, y_t^{(l)}, \dot{x}_t^{(l)}, \dot{y}_t^{(l)})$, L is the number of tracked cars and T_l is the total length of the lth tracking.

We have studied two different data sets of this type. One collected in a controlled environment, which means that we have access to ground truth labels, and one collected mainly on highways and country roads.

A. Data from Test Track

This data set has been collected at the Astra Zero test track near Gothenburg, Sweden. We refer to the data as data set 1. There are two classes of maneuvers in the data called overtakes and cut-ins. These are real car maneuvers but collected in a controlled environment. An overtake is when the host vehicle enters the tracked vehicle's lane and the tracked vehicle changes lane to give way for the host vehicle, and a cut-in is when the tracked vehicle enters the host vehicles lane close to the host vehicle. Each class has four distinct sub-classes, and each sub-class was measured 8–10 times. Resulting in a total of 77 motion trajectories for the whole data set.

TABLE I

Descriptive statistics of data set 1. The relative position is in meters and the relative velocity in meters per second.

Descriptive Statistics				
	$x_t^{(l)}$	$y_t^{(l)}$	$\dot{x}_t^{(l)}$	$\dot{y}_t^{(l)}$
mean	0.93	24.22	0.06	0.07
std	1.06	13.86	0.34	0.09
min	-1.10	9.42	-2.50	-0.32
max	3.77	74.34	1.62	0.69
count	4,677			
mean length	60.74			
std length	11.28			

There are therefore two different ground truth label sets for data set 1. In label set 1 each of the eight sub-classes is considered a unique label. And in label set 2 each of the two main classes, that is, overtake and cut-in, is considered a unique label. (See Table I for summary statistics of data set 1.)

B. Data from Highways and Country Roads

This data was collected by the Volvo Car Corporation on a several-day expedition in Europe, passing through different countries, and will be referred to as data set 2. It was processed by a motion detection algorithm and an image recognition algorithm to classify the tracked objects, and only motion trajectories of other cars are kept. The tracking of the relative motion of the host vehicle and the tracked vehicle starts when the tracked vehicle enters the sensor's field of view, and ends when it exits its field of view. Note that, for example, occlusion of a tracked car by another object may result in several trackings of the same car; meaning that some trajectories are partial trackings of the same car.

We consider only motion trajectories between 4 and 8 seconds. The sampling rate of the sensor is 10 hertz, so

TABLE II

SUMMARY STATISTICS OF DATA SET 2. THE RELATIVE POSITION IS IN METERS AND THE RELATIVE VELOCITY IN METERS PER SECOND.

Descriptive Statistics				
	$x_t^{(l)}$	$y_t^{(l)}$	$\dot{x}_t^{(l)}$	$\dot{y}_t^{(l)}$
mean	-0.47	28.38	0.14	1.34
std	3.05	16.80	0.60	4.80
min	-4.49	-11.16	-5.18	-18.06
max	4.50	109.34	7.17	15.33
count		81,188		
mean length		59.83		
std length		11.83		

the length of the sequences range between 40 and 80 observations. We assume that the lane width on European highways and country roads is 3 meters and consider only tracked cars which are within one and a half lane width of the host vehicle. The resulting data set consists of 1,357 trajectories with a total of 81,188 discrete-time observations. (See Table II for summary statistics on data set 2.)

C. Data Standardization

All features are standardized into a $[0, 1]$ range before training. This is done by

$$z_t^{(l)} = \frac{x_t^{(l)} - \min(x)}{\max(x) - \min(x)}, \tag{11}$$

where $z_t^{(l)} \in [0, 1]$ is the standardized feature, $1 \leq l \leq L$, $1 \leq t \leq T_l$, and $\min(x)$ and $\max(x)$ are the minimum and maximum value, respectively, of the feature x, in the whole data set. The standardization is invertible.

VI. RESULTS AND DISCUSSION

In this section we present the results of applying the clustering method to the two data sets presented in the previous section. The method has been implemented using the Python package pomegranate [17], which is a Python/Cython implementation of probabilistic models. The implementation allows stacked models, which is key for the implementation of the mixture model, since we have stacked a general mixture model on top of hidden Markov models, which in turn are stacked onto multivariate Gaussian distributions.

TABLE III

A SUMMARY OF THE TRAINING PARAMETERS USED WHEN FITTING THE MIXTURE OF HMMS TO THE DATA SET.

Simulation	B	τ	$n_{\max}$
D1F1	20	10	1000
D1F2	20	10	1000
D2F3	3	200	1000

Table III show the training parameters used for each simulation. The simulations are named D1F1 to mean data set 1 using feature set 1, D1F2 to mean data set 1 using feature set 2, and so on. In the table, B is the number of cluster solutions found for each model configuration, that is, the number of random initializations, τ is the stopping threshold for the EM algorithm when fitting the mixture model, and $n_{\max}$ is the maximum number of iterations for the EM algorithm. We have used a linear decay of 0.8 in all simulations.

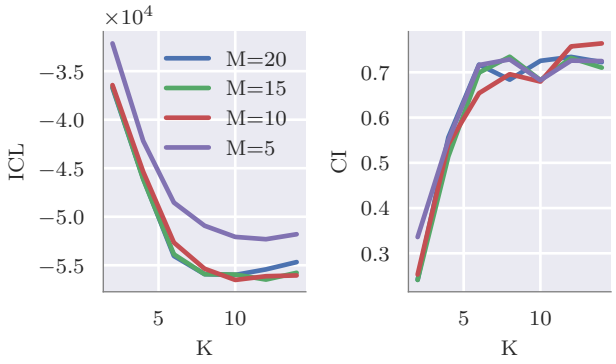

Fig. 2. The ICL and CI criteria for the model configuration search in D1F1. We have omitted $M = 2$ because it cluttered the figures and it is clearly ruled out by ICL.

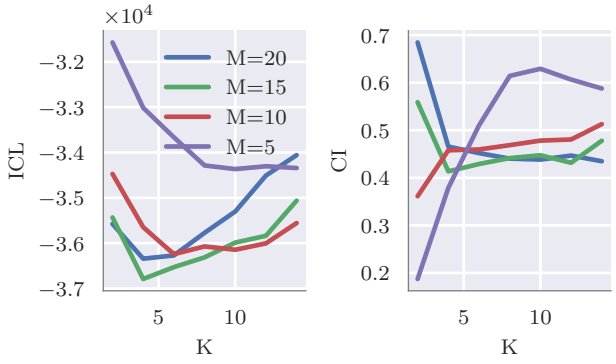

Fig. 3. The ICL and CI criteria for the model configuration search in D1F2. We have omitted $M = 2$ because it cluttered the figures and it is clearly ruled out by ICL.

A. Simulations on Data Set 1

We use two different feature sets when training on data set 1. Feature set 1 consists of the relative position between the host vehicle and the tracked vehicle (x_t, y_t), and feature set 2 consists of their relative velocity $(\dot{x}_t, \dot{y}_t)$. The two are called D1F1 and D1F2 respectively. We perform a model configuration search over the number of clusters $K = [2, 4, 6, 8, 10, 12, 14]$, and the number of hidden states $M = [2, 5, 10, 15, 20]$ in both D1F1 and D1F2.

We see in Figure 2 that according to ICL and CI we should choose the number of mixture components between 6 and 12, and the number of hidden states between 10 and 20 for D1F1. The estimated number of states is consistent with previous results suggesting that a good rule of thumb is to choose the number of states as a third of the average length of the observed sequences [9] which is roughly 20 for this data.

We see in Figure 3 that according to ICL we should choose the number of mixture components between 4 and 6, and the number of hidden states between 15 and 20 for D1F2. In this case, however, there is a disagreement between ICL and CI, and according to the CI criterion we should choose the number of mixture components to be 2, and the number of hidden states between 15 and 20. It also indicates a higher clustering consensus around $K = 10$ when $M = 5$, but it is clear by the ICL criterion that $M = 5$ is not a good choice, and it is thus ruled out. It is not trivial to resolve this disagreement between the two criteria, and ideally they should agree. Instead we have to use expert judgment and explore both alternatives.

The local optima for ICL are $M = 10$ and $K = 10$ for D1F1 and $M = 15$ and $K = 4$ for D1F2. However, we have also noted that there is a good consensus among the clusterings at $K = 2$ according to CI. Since we have access to ground truth labels in data set 1, and already know that $K = 2$ could potentially be the clustering with the cut-in and overtake maneuvers, we choose to ignore $K = 4$ suggested by ICL and continue with $K = 2$ for this analysis.

TABLE IV

THE ADJUSTED MUTUAL INFORMATION SCORE FOR THE HARD CLUSTERINGS DERIVED FROM THE TWO BEST MODELS ON DATA SET 1.

Simulation	Label Set 1	Label Set 2
D1F1 (M=10, K=10)	0.84	0.27
D1F2 (M=15, K=2)	0.32	1.0

In Table IV we see the adjusted mutual information [18] score for the partitions found by each estimated model. In D1F1 with $M = 10$ and $K = 10$ we find a clustering which agrees with label set 1, that is, the specific maneuvers. And in D1F2 with $M = 15$ and $K = 2$ we find a clustering which agrees with label set 2, that is, the more generic cut-in and overtake maneuvers. In fact, it finds the ground truth clustering, suggesting the importance of the feature vector choice, and the criterion choice. Using CI as a complement to ICL we were able to detect structure in the data which we might otherwise have missed.

Figure 4 shows the clusters found in D1F2. The relative position of the host vehicle and the tracked car is scatter plotted where the color gradient, which changes from dark ($t = 0$) to bright ($t = L_i$), indicates time.

B. Simulations on Data Set 2

From the results on data set 1 we saw that the relative velocity is an important feature for finding the types of maneuvers we are looking for, but we also want to emphasize that lane changes are an important part of the maneuver. Therefore, in the simulation on data set 2 we use the feature vector $(x_t, \dot{x}_t, \dot{y}_t)$, which consists of the relative lateral position and the relative velocity. We call this D2F3.

We assume $M = 15$ since the sequences in this data set are roughly of the same length as in data set 1,

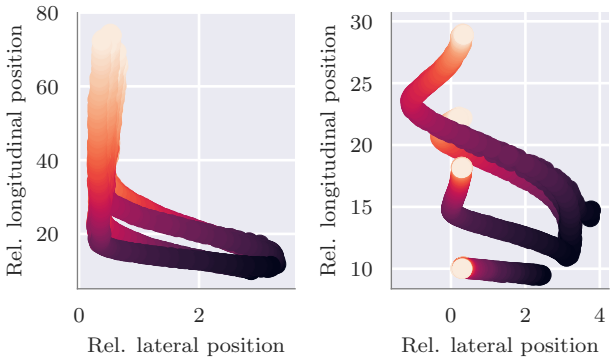

Fig. 4. The figure show the two clusters found in the Astra Zero test track data in D1F2. In the left plot we see the cluster of cut-ins, where the tracked car cuts in front of the host car from the right lane, and in the right plot we see the cluster of overtakes, where the host car changes into the tracked car's lane, and the tracked car starts to give way for the host car.

and perform a model configuration search over $K = [5, 10, 20, 40, 60, 65, 70, 75, 80, 85, 90, 100, 150]$.

According to ICL, an optimum exists at $K = 85$. The CI criterion was not used in this analysis since it takes a lot of time to fit each model on this larger data set, and the CI criterion requires us to fit many models for each parameter configuration. The model configuration search can easily be run in parallel, so a possible solution to this problem is more CPU cores.

We trained five models, each with a left-to-right topology, using $K = 85$ mixture components and $M = 15$ hidden states. Each of the five runs was randomly initialized and we set the early stopping threshold to 50 to let the final models train longer than during model selection. The best performing model according to ICL out of the five was chosen as the final model, and used to cluster the data set.

In Figure 5 we show four examples of clusters found in data set 2 using the feature vector $(x_t, \dot{x}_t, \dot{y}_t)$. The dashed red marks indicate the assumed lane widths. Note that these are not the real lane markings, since everything is relative to the host car, but under the additional assumption that either the host car or the tracked car tends to be at the center of its lane, these markings can be thought of as representing the true lane markings which helps us interpret the clusters.

First, in the top-left plot of Figure 5, we see that the tracked car and the host vehicle keeps roughly the same direction, and the tracked car tends to start close to the host vehicle and end up farther away, which can be explained by the tracked car driving past the host vehicle. We also note an anomaly in the cluster showing that the clusters are not perfect. A close inspection shows that the number of initial samples that lie on the horizontal line in the anomlay are much fewer than the ones which agree with the rest of the cluster, which may be the reason for the obtained clustering.

Second, in the top-right plot of Figure 5 we see that the two vehicles again keep roughly the same direction, but that the tracked car tends to start farther away from the host vehicle, and end up closer to it. The cluster can be explained

by the host vehicle driving past the tracked vehicle.

Third, in the bottom-left plot of Figure 5 we see that the tracked vehicle moves into the same lane as the host vehicle while driving past the host vehicle, which can be explained by a cut-in from the left lane, or the host vehicle moving into the same lane as the tracked vehicle while the tracked vehicle drives faster than the host vehicle. The maneuver is ambiguous because we use the relative motion of the two cars, meaning that the movement of either car could result in the observed change in relative position.

Last, in the bottom-right plot of Figure 5 we see that the tracked vehicle moves one lane width to the right while the host vehicle drives past the tracked vehicle, which can be explained by the tracked vehicle changing lane to allow way for the host vehicle, or the host vehicle changing lane to drive past the tracked vehicle.

The plots show only a small subset of the clusters found. Other vehicle maneuvers such as cut-in from the other lane, the host vehicle and the tracked vehicle passing each other in different lanes, the host vehicle following the tracked vehicle and so on are also observed in the clusters. We do detect some duplicate clusters with maneuvers that an expert would probably classify as the same maneuver, but where the clustering algorithm finds structure that is not immediately clear to the human observer. This can be an advantage if maneuvers have previously been defined heuristically, and the clusters are used to verify that these maneuver definitions cover the maneuvers that actually occur on the road. However, it could also be a disadvantage if the learned clusters are used as ground truth labels when training scenario-specific models. A compromise should probably be used where the clusters are used as a guide during both the labeling and maneuver verification process. The method also finds smaller clusters, accounting for a small proportion of the data, between 10 and 15 percent, which are not easily explained by a driver maneuver. These may be less frequently occurring complex maneuvers that we have not been able to interpret.

C. Summary of the Results

We have used three different feature sets for the clusterings. The first was the relative position (x_t, y_t) of the host vehicle and the tracked car, the second was their relative velocity $(\dot{x}_t, \dot{y}_t)$, and the third was a combination of the two. When using the relative position we are asking the question "What are the typical relative motions between the host vehicle and the tracked cars, and where, in relation to the host vehicle, do they typically occur?", and using the second feature set we are simply asking "What are the typical relative motions between the host vehicle and the tracked cars?". Depending on which feature set we use, we expect different clusters to be found.

Clustering using the relative position in data set 1 results in very specific maneuvers, and clustering using the relative velocity results in more generic maneuvers such as overtakes and cut-ins. It suggests that it is important to choose the features with care so that the clustering of those features represent relevant behaviors. In this case we wanted to find typical driver maneuvers with an emphasis on lane changes, and thus used the relative lateral position and the relative velocity of the host car and the tracked cars for the final clustering of the large data set.

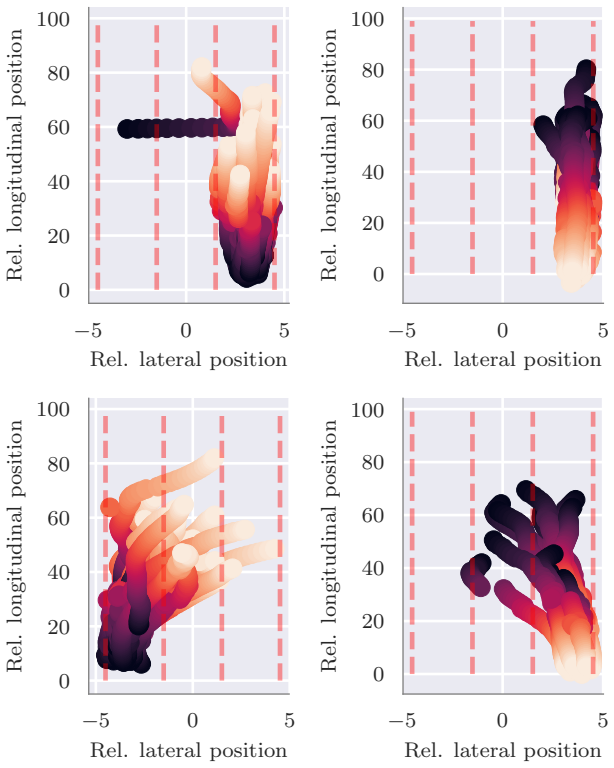

Fig. 5. The figure shows four different motion patterns found when clustering the expedition data: The vehicles keep roughly the same direction while the tracked vehicle drives past the host vehicle (top-left). The vehicles keeps roughly the same direction while the host vehicle drives past the tracked vehicle (top-right). The tracked vehicle moves into the same lane as the host vehicle while driving past it (bottom-left). The tracked vehicle moves one lane width to the right while the host vehicle drives past it (bottom-right).

The choice of the model configuration (K, M) could possibly be done in a more principled way using, for example, a countably infinite mixture model [19]. However, such methods are computationally expensive which was a limiting factor in this work.

VII. Conclusions and Future Work

We have shown that mixtures of hidden Markov models can be used to find typical patterns in relative motion trajectories of driver maneuvers on highways and country roads. Our results suggest that the choice of the feature vector affects the type of clusters found. In particular, the relative velocity between the host vehicle and other tracked vehicles on the road seems like a good feature choice when looking for typical driver maneuvers.

A natural next step is to segment the maneuvers into "partial-maneuvers" such as left-turn, right-turn, keeping straight, and so on using a MHMM fitted to windowed

sequences of the data. The clustering can be used to segment the maneuvers into partial-maneuver categories, and the learned segments can be used to train a HMM for maneuver generation. This has previously been used for segmentation of video sequences [20]. We have implemented a proof of concept of such a segmentation model with promising preliminary results.

Other potential application areas for the method are complex driving behaviors in other kinds of traffic scenarios, for example, free, congested, and saturated traffic, and driving behaviors at signalized intersections; all potential research directions for future work.

ACKNOWLEDGMENT

All authors would like to acknowledge funding from Vinnova through the FFI Grant: Big Automotive Data Analytics: Sensor Modeling and Performance Analysis (BADA-SEMPA). We would like to thank the Volvo Car Corporation for providing the data necessary for this work, and the BADA-SEMPA group for their support.

REFERENCES

[1] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[2] W. Damm and R. Galbas, "Exploiting learning and scenario-based specification languages for the verification and validation of highly automated driving," in *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, ser. SEFAIS '18. Gothenburg, Sweden: ACM, 2018, pp. 39–46.

[3] J. Florbäck, L. Tornberg, and N. Mohammadiha, "Offline object matching and evaluation process for verification of autonomous driving," in *Proc. of IEEE International Conference on Intelligent Transportation Systems*, Nov 2016, pp. 107–112.

[4] E. L. Zec, N. Mohammadiha, and A. Schliep, "Statistical Sensor Modelling for Autonomous Driving Using Autoregressive Input-Output HMMs," in *Proc. of IEEE International Conference on Intelligent Transportation Systems*, 2018.

[5] N. Hirsenkorn, T. Hanke, A. Rauch, B. Dehlink, R. Rasshofer, and E. Biebl, "Virtual sensor models for real-time applications," *Advances in Radio Science*, vol. 14, pp. 31–37, 2016.

[6] P. Smyth, "Clustering sequences with hidden markov models," in *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. MIT Press, 1997, pp. 648–654.

[7] C. Li and G. Biswas, "A bayesian approach to temporal data clustering using hidden markov models," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 543–550.

[8] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 1, June 2003, pp. I–I.

[9] A. Schliep, I. G. Costa, C. Steinhoff, and A. Schonhuth, "Analyzing gene expression time-courses," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 2, no. 3, pp. 179–193, Jul. 2005.

[10] V. Romero-Cano, J. I. Nieto, and G. Agamennoni, "Unsupervised motion learning from a moving platform," in *2013 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, June 2013, pp. 104–108.

[11] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[12] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *The Computer Journal*, vol. 41, no. 8, pp. 578–588, Jan 1998.

[13] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[14] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 719–725, July 2000.

[15] N. X. Vinh and J. Epps, "A novel approach for automatic number of clusters detection in microarray data based on consensus clustering," in *2009 Ninth IEEE International Conference on Bioinformatics and BioEngineering*, June 2009, pp. 84–91.

[16] I. Costa and A. Schliep, "On external indices for mixtures: validating mixtures of genes." Berlin, Heidelberg: Springer, 2005, pp. 662–669.

[17] J. Schreiber, "Pomegranate: fast and flexible probabilistic modeling in python," *Journal of Machine Learning Research*, vol. 18, pp. 1–6, 2018.

[18] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Dec. 2010.

[19] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[20] J. S. Boreczky and L. D. Wilcox, "A hidden markov model framework for video segmentation using audio and image features," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 6, May 1998, pp. 3741–3744 vol.6.