Transportation Research Part C 69 (2016) 497–514

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Lane changing intention recognition based on speech recognition models [☆]

Keqiang Li [a], Xiao Wang [a,b], Youchun Xu [b], Jianqiang Wang [a,*]

[a] State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing 100084, China
[b] Department of Automobile Engineering, Military Transportation University, Tianjin 300161, China

ARTICLE INFO

Article history:
Received 28 April 2015
Received in revised form 4 November 2015
Accepted 15 November 2015
Available online 28 November 2015

Keywords:
Lane changing
Intention recognition
Hidden Markov models
Bayesian filtering
Speech recognition model
Driver assistance systems

ABSTRACT

Poor driving habits such as not using turn signals when changing lanes present a major challenge to advanced driver assistance systems that rely on turn signals. To address this problem, we propose a novel algorithm combining the hidden Markov model (HMM) and Bayesian filtering (BF) techniques to recognize a driver's lane changing intention. In the HMM component, the grammar definition is inspired by speech recognition models, and the output is a preliminary behavior classification. As for the BF component, the final behavior classification is produced based on the current and preceding outputs of the HMMs. A naturalistic data set is used to train and validate the proposed algorithm. The results reveal that the proposed HMM–BF framework can achieve a recognition accuracy of 93.5% and 90.3% for right and left lane changing, respectively, which is a significant improvement compared with the HMM-only algorithm. The recognition time results show that the proposed algorithm can recognize a behavior correctly at an early stage.

1. Introduction

Advanced Driver Assistance Systems (ADAS) have been receiving increasingly more attention in recent years. However, some common poor driving habits tend to interfere with the proper functioning of such systems. For example, blind-spot warning systems (BSWs) and lane departure warning systems (LDWs) are two typical assistance systems relying on turn signals because an alarm sounds, if BSWs or LDWs have already detected exceptional cases, only when the driver activates the turn signal. However, in the United States and China, the turn signal usage rates are approximately 44% (Lee et al., 2004) and less than 40% (Dang et al., 2013), respectively. Therefore, for most drivers, the two assistance systems can be seen as a useless technology. Consequently, it is important to develop an accurate and early lane changing (LC) intention recognition system for ADAS.

Driver behavior is a time-varying, continuous process and can be seen as being analogous to human spoken behavior. Fig. 1 shows driving and spoken behaviors on three levels of hierarchical structure (Boer et al., 1998). At the strategic level, needs are goals that humans try to satisfy and, therefore, they translate them into local goals; for example, route in driving or purpose in speech. At the tactical level, the task manager orchestrates the maneuvers or low-level tasks (e.g., lane changing or whisper) and the task scheduler communicates the performance-related information to the strategic level. At the two upper levels, the intentions of the two behaviors are inner mental states of humans, which cannot be observed directly. The operational level models the execution of the maneuvers that are learned as well as the automatic processes. At this

[☆] This article belongs to the Virtual Special Issue on: Modelling, calibrating and validating car following and lane changing behavior.

[*] Corresponding author.
E-mail address: wjqlws@tsinghua.edu.cn (J. Wang).

http://dx.doi.org/10.1016/j.trc.2015.11.007
0968-090X/© 2015 Elsevier Ltd. All rights reserved.

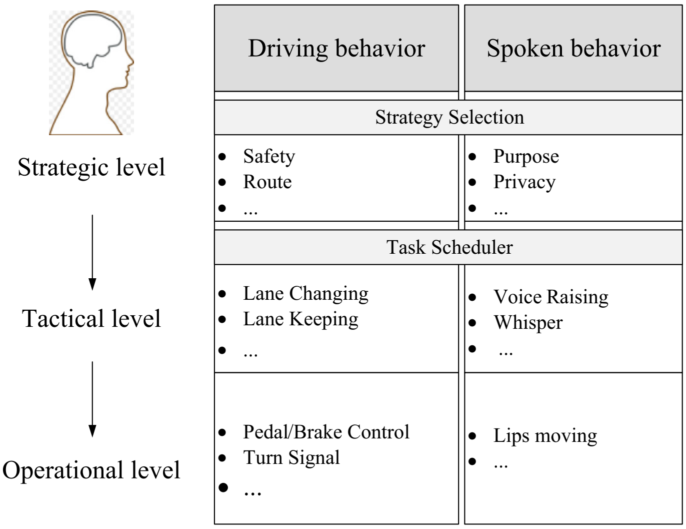


Fig. 1. Hierarchical driving and spoken behaviors. Performance is structured at three levels and the two behaviors can be seen to be analogous.

level, signals can be observed through vehicle movements or sound waves. In these ways, driving behavior is analogous to human spoken behavior on the three levels of a hierarchical structure, thus motivating us to expect that the study of LC intention recognition can learn from classical and mature speech recognition models.

In this study, a novel algorithm combining the hidden Markov model (HMM) and Bayesian filtering (BF) technique is proposed to improve the capability of drivers' LC intention recognition. In the HMM component, the definition of grammar is inspired by speech recognition models. Driving behavior is split into sub-behaviors, and corresponding sub-HMMs are developed. The input parameters are three signals from the controller area network bus (CAN bus), where the output is a preliminary behavior classification. To improve the recognition performance, a BF is used to produce the output of the final behavior classification based on the current and preceding outputs of the HMMs. Naturalistic data collected on various kinds of roads in Beijing, China are used to train and validate the proposed algorithm. The results reveal that the proposed algorithm can get good results both on recognition ratio and time.

The remainder of this paper is organized as follows. Related work from literature is introduced in Section 2, and Section 3 describes the principles of speech recognition models and explains how they can be applied to LC models. In Section 4, the system architecture and the theory of the HMM and BF are introduced. The data collection and post-processing will be discussed in Section 5. Section 6 presents the experimental results with the validation of the model, and finally, Section 7 draws out the conclusions of this study.

2. Related work

Substantial research has been conducted regarding LC models, with Gipps-type and Markov process-based models being representative (Zheng, 2014). Gipps (1986) conducted a pioneering work to introduce an LC structure in an urban street context. In that model, a driver's LC decision must consider the possibility, necessity, and desirability of changing lanes, with six factors and their effects being considered most important: safe gap distances, location of permanent obstructions, presence of transit lanes, driver's intended turning movement, presence of heavy vehicles, and speed. These factors are determined by a set of deterministic rules; thus, the driver behavior according to the Gipps's model is considered deterministic (Zheng, 2014).

Based on Gipps's study, Yang and Koutsopoulos (1996) developed an LC model and implemented it in a microscopic traffic simulator. In the model, an LC was classified as either mandatory or discretionary and consisted of three steps: checking if a change is necessary and defining the change type, selecting the target lane, and executing the lane change if the gap distances are acceptable. A difference between the models of Gipps (1986) and Yang and Koutsopoulos (1996) is that the latter model introduced an LC probability to replace the deterministic LC.

Another limitation of the work of Gipps (1986) is that the method ignores the interaction between the lane changer and the follower in the target lane, which is unrealistic when the traffic is heavy, congested, or impacted by an incident (Zheng et al., 2013). To address this limitation, Hidas (2002, 2005) proposed an improved model that classified an LC into three categories: free, cooperative, and forced. However, this LC model was only tested on two simple hypothetical road networks using simulation, and no detail was provided on how these parameters were calibrated.

Markov processes are another widely used method for modeling LC. Worrall et al. (1970) proposed a stochastic LC model that was developed as a homogeneous Markov chain and calibrated using naturalistic data collected on a section of a six-lane freeway in Chicago.

Extensive studies have been conducted on modeling driving behavior using HMMs at Massachusetts Institute of Technology and Nissan Cambridge Basic Research. Pentland and Liu (1999) modeled human driving behavior as a Markov chain of control states, with LC consisting of the following states: (1) preparatory centering of the car in the initial lane, (2) looking around to make sure the target lane is clear, (3) steering to initiate LC, (4) changing lanes, (5) steering to terminate the change, and (6) finally re-centering the car in the target lane. Experiments proved that human driving intentions could be recognized accurately soon after the beginning of the action. Kuge et al. (2000) proposed another LC intention recognition method using HMMs. Three HMM models were built to recognize discrete behaviors, including ordinary LC, emergency LC, and lane keeping. Several sub-HMMs were used to recognize continuous behaviors with each sub-HMM consisting of three states with a left-to-right structure that did not allow for skipping of states or backward state transition, with Gaussian mixture models (GMMs) being used as the observation probability distribution. The HMM input parameters were steering angle and angle velocity, with all data collection based on a driving simulator. Unlike Kuge's work, Mitrovic (2005) proposed discrete left-to-right HMMs to model the driving behavior. Five parameters, including average speed, average lateral acceleration, average longitudinal acceleration, lateral acceleration slope, and longitudinal acceleration slope were extracted as the input parameters. Except for the left turn, all of the other models had six-state structures. In experiments, data were collected based on a real vehicle in normal driving situations, and authors reported that only 1.7% of the test events were recognized incorrectly.

Other than the hidden state of the above HMMs, Zou and Levinson (2006) modeled hidden states as the attitudes of drivers, and vehicle dynamics data were taken as the observable states that are combinations of data regarding changes in speed. Toledo and Katz (2009) presented an LC model based on an integration of the HMM. The target lane choice was modeled as an HMM, with the gap acceptance model connecting the HMM and the observable outcome. Other factors, such as driving goals, driver characteristics, and surrounding traffic, were also considered. All model parameters were estimated using naturalistic data. In the test, the model with the HMM was compared with the model without it. In addition to the above work, Berndt et al. (2008) and Berndt and Dietmayer (2009) used HMMs to investigate early driver intention inference by observing easily accessible vehicle and environmental signals, the difference from the previous HMMs being that in his study, the best model consisted of nine states in a linear left-to-right structure, with a sub-model of the first three states being extracted to build a new HMM for continuous recognition. Aoude et al. (2011, 2012) introduced two new approaches based on support vector machines (SVMs) and HMMs to classify driver behaviors at road intersections. The two algorithms were validated successfully on more than 10,000 intersection approaches collected on real roads. Results illustrate that both the SVM and HMM based classifiers outperformed the traditional methods by a significant margin in each set of tests. Gadepally et al. (2014) presented a framework for the estimation of driver behaviors at intersections based on HMMs. In their framework, a driver-decision-vehicle dynamics coupling was encapsulated in a hybrid-state-system (HSS) consisting of a discrete-state system (DSS) at the higher level and a continuous-state system (CSS) at the lower level.

In addition to Gipps-type and Markov process-based models, game theoretic approaches are increasingly being used in LC studies. Wang et al. (2015) put forward a predictive approach for unified lane-changing and car-following control based on dynamic game theory for autonomous and connected vehicle systems, the output of the approach was a unique and continuous path that can be used by the automated vehicle actuators to follow, so their studies focused on the issue of when and where it is optimal to change lane, not LC behavior recognition. Talebpour et al. (2015) presented a LC model based on a game-theoretical approach that endogenously accounts for the flow of information in a connected vehicular environment, there were two game types used for modeling lane-changing behavior: a two-person non-zero-sum non-cooperative game under complete information in the presence of connected vehicle technology and a two-person non-zero-sum non-cooperative game under incomplete information in its absence. In their studies, a simplified version of the LC model was presented and calibrated and a simulation framework based on fictitious play was proposed and a segment was simulated based on this framework. However, this work aimed at traffic flow and designed for microscopic traffic simulations, while not vehicle safety.

Conclusions drawn from these related studies can be summarized as follows. First, the existing Gipps-type and Markov process LC models have different application objects. The Gipps-type LC models are proposed to facilitate traffic flow and designed for microscopic traffic simulations, while the majority of Markov process models focus on vehicle safety and are designed for ADAS. Furthermore, the HMMs are a naturally suitable tool to model LC because they can model the stochastic nature of the driving behavior and support the recognition of temporal data patterns. Finally, there are very few studies that combine HMMs and other models for more effective prediction outcomes.

The major contributions of this study are as follows. We propose novel LC intention recognition models combining the HMM and Bayesian filtering; the advantage of the proposed models is that they use an extra filter to improve the recognition performance compared with the HMM-only or SVM-only methods. Furthermore, the proposed models which learning from the speech recognition systems have a clear and formal model construction flow, and they are also easily extendable to predicting other driving behaviors such as car following, turning, and others, which is a big advantage over other current models. Finally, the models are trained and validated using naturalistic lane-changing data collected in Beijing, China, involving various road types such as highway and ring roads, and the real data used for model validation makes the model more robust and practical when compared with simulated data.

3. Recognition of LC intentions based on speech recognition models

Fig. 2 illustrates an isolated word recognition model that follows the usual pattern of machine learning methods. Assume that only three words are to be trained, and that different people collect each word in three utterances. In the model-training step, speech vectors are extracted from the speech waveforms and used to train the corresponding models, $M_1, M_2,$ and M_3. Then, to recognize an unknown new word, the likelihood of each model generating that word is calculated and the most likely model is used to identify the word.

If the hypothesis that different behaviors within a driving task correspond to different words within a sentence is true, for example, if lane changing left (LCL), lane changing right (LCR), and lane keeping (LK) are viewed as words 1, 2, and 3, respectively, then the speech recognition model and algorithm can be used directly for LC intention recognition. However, note that the words can be identified only after they have been spoken. For the task of continuous driver behavior recognition, we must produce recognition results continually rather than recognizing a result after an entire LC event has been completed. It is much more valuable to identify an LC at an early stage for ADAS. Another speech recognition model known as "continuous speech recognition" can be used to produce such continuous recognition results.

In the continuous speech recognition model, a word is considered as consisting of a sequence of sub-words or phonetics. For example, the word "ability" can be observed as a sequence of the following phonetic sounds.

"ability" /ax b ih1 l ix t iy/

This motivates us to consider a driving behavior as also being divisible into a sequence of sub-behaviors, with an LC being considered to be a sequence N_{aLC} of sub-LCs, where N_{aLC} is the number of sub-lane changing actions and the subscript " aLC " refers to "actions of lane changing".

lane changing $/lc_1, lc_2, \ldots, lc_{N_{aLC}}/$

where $lc_i,\ 1 \leqslant i \leqslant N_{aLC}$ is used to denote a sub-LC. As shown in Fig. 3, sub-LCs can be trained into independent models, $M_i,\ 1 \leqslant i \leqslant N_{aLC}$, and the input parameters are no longer from the entire driving event, but from a moving window, T_w, of fixed time-length, L_w. Let $\boldsymbol{O}_{T_W}$ represents the input features saved in T_w, the likelihood of each model generating the sub-LCs, $P(\boldsymbol{O}_{T_W}|M_i)$, is calculated, and the maximum from these is selected.

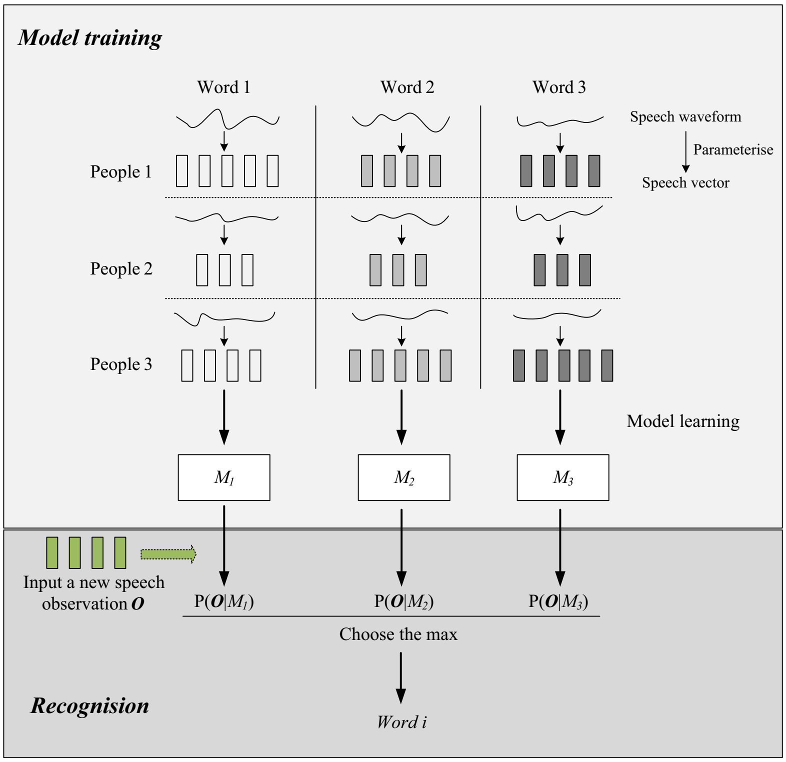


Fig. 2. HMMs-based isolated word recognition model. HMMs are trained using different words.

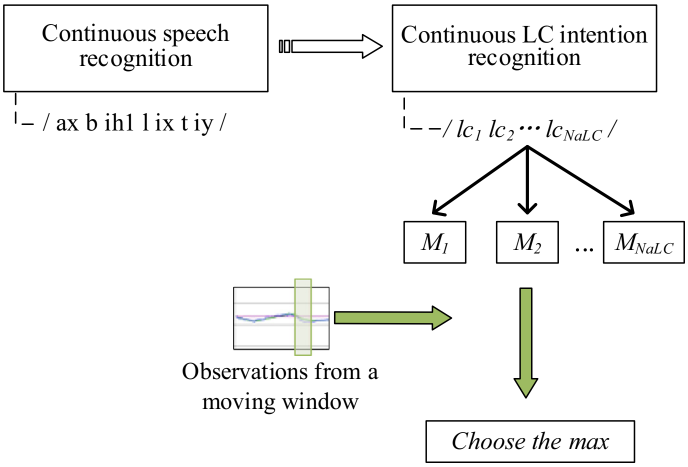

Continuous speech recognition
/ ax b ihl l ix t iy /
Continuous LC intention recognition
/ lc1 lc2 ··· lcNaLC /
M1
M2
...
MNaLC
Observations from a moving window
Choose the max

Fig. 3. The continuous LC intention recognition framework.

4. Algorithm

4.1. System architecture

The architecture of the proposed algorithm is shown in Fig. 4. The input features, $\boldsymbol{O}_{T_W}$, including the steering angle, lateral acceleration and yaw rate, are saved in the moving window, T_w. The HMM recognizer generates an output for a preliminary behavior classification (*lc* or *lk*) per computation cycle into a buffer of length, L_b. Here, *t* represents a discrete-time variable and set to be 1 when the HMM recognizer generates the first result, *t* will be subsequently auto-incremented after each computer cycle. When *t* is greater than $L_b - 1$, the BF begins to calculate the probability corresponding to each hypothetical preliminary classification and a threshold detector is used to produce the final behavior classification. Note that in the BF component, each element in the buffer can be expected to have different impacts on the final classification; hence, a weight function is designed to reflect the relationship between the elements and the final classification result.

4.2. The HMM component

HMMs are used extensively in behavioral modeling because they can model any time series, with classical applications including speech recognition (Rabiner and Juang, 1986; Rabiner, 1989; Gales and Young, 2007), handwriting recognition (Bunke et al., 1995; Chen et al., 1994), and DNA sequence analysis (Hughey and Krogh, 1996). In recent years, they have also been used to recognize human gestures (Brand et al., 1997; Bilal et al., 2013; Dittmar et al., 2015; Chen et al., 2003).

Assume that there are N_{aLC} sub-HMMs for lane changing sub-behaviors, and N_{aLK} sub-HMMs for lane keeping sub-behaviors. Each sub-HMM consists of a set of N_{sta} finite "hidden" states, $S_i,\ 1 \leqslant i \leqslant N_{sta}$, where N_{sta} is the total number of states with a set of *M* observable symbols per state. A generic left-to-right structured sub-HMM is shown in Fig. 5, where S_i represents the hidden state, and o_i represents the observation sequence. The $\boldsymbol{A}$ matrix represents the state transition probabilities, and the $\boldsymbol{B}$ matrix represents the observation probabilities. Therefore, a sub-HMM can be denoted by $\lambda_i^{behavior} = (\boldsymbol{A}, \boldsymbol{B}, \pi)$, where $behavior = \{LC, LK\}$, $1 \leqslant i \leqslant N_{aLC}$ or $1 \leqslant i \leqslant N_{aLK}$, and π represents the initial state distribution.

Gaussian mixture models (GMMs) are used to model the observation probabilities, *B*, which are of the form

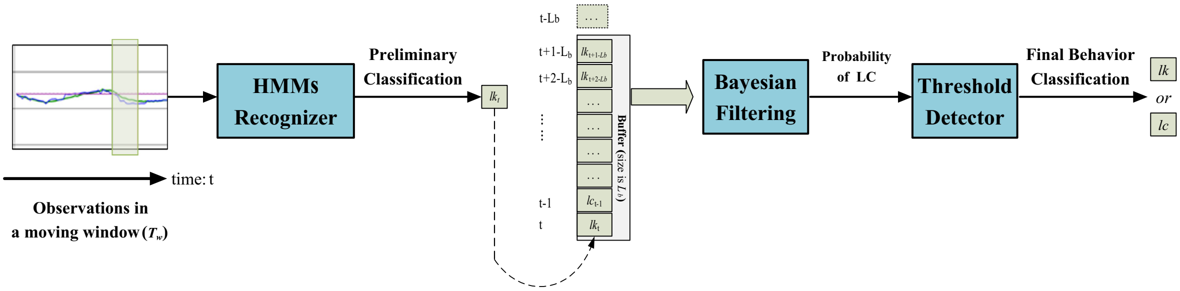


Fig. 4. System architecture.

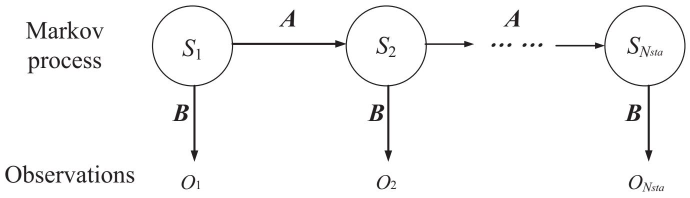


Fig. 5. Structure of a sub-HMM with N_{sta} states. A represents the state transition probabilities and B represents the observation probabilities.

$$\boldsymbol{B}(j) = b_j(o) = \sum_{k=1}^{M} c_{jk}\mathcal{N}\left(\boldsymbol{O}, \boldsymbol{\mu}_{jk}, \sum_{jk}\right), \quad 1 \leqslant j \leqslant N_{sta} \tag{1}$$

where c_{jk} is the mixture coefficient for the kth mixture in the jth state. $\mathcal{N}$ represents the probability density function of a multivariate Gaussian with mean, $\boldsymbol{\mu}$ and covariance matrix, $\sum$. The mixture coefficient c_{jk} satisfies the following constraints:

$$\sum_{k=1}^{M} c_{jk} = 1 \; c_{jk} > 0, \; 1 \leqslant j \leqslant N_{sta}, \; 1 \leqslant k \leqslant M \tag{2}$$

There are three basic HMM problems, as follows:

- **Evaluation problem:** Given the observation sequence, $\boldsymbol{O}$, and a sub-HMM model, $\lambda_i^{behavior}$, find $P(\boldsymbol{O}|\lambda_i^{behavior})$.
- **Decoding problem:** Given the observation sequence, $\boldsymbol{O}$, and a sub-HMM model, $\lambda_i^{behavior}$, find an optimal state sequence for the underlying Markov process.
- **Learning problem:** Given an observation sequence, $\boldsymbol{O}$, and the dimensions, N_{sta}, and M, find the model, $\lambda_i^{behavior} = (\boldsymbol{A}, \boldsymbol{B}, \pi)$ that maximizes the probability of $\boldsymbol{O}$.

As shown in Fig. 6, this study considers the learning and evaluation problems. Assume that we have trained N_{aLC} lane changing sub-HMMs, $\lambda_i^{LC}, \; 1 \leqslant i \leqslant N_{aLC}$, and N_{aLK} lane keeping sub-HMMs, $\lambda_j^{LK}, \; 1 \leqslant j \leqslant N_{aLK}$, using the Baum–Welch algorithm (Welch, 2003). For a given observation $\boldsymbol{O}$, the forward algorithm is used to find the posterior probabilities of observing that sequence given each model. First, $P(\boldsymbol{O}|\lambda_i^{LC}), \; 1 \leqslant i \leqslant N_{aLC}$, and $P(\boldsymbol{O}|\lambda_j^{LK}), \; 1 \leqslant j \leqslant N_{aLK}$ are calculated using the forward algorithm; then, the maximum values of $P(\boldsymbol{O}|\lambda_i^{LC})$ and $P(\boldsymbol{O}|\lambda_j^{LK})$ are chosen to calculate the ratio.

As shown in Eq. (3), the ratio determines whether the classification is more likely to be an lc or lk. A threshold τ can be selected to adjust how conservative the HMM classifier should be. The e term can be found in the research of (Van der Horst, 1990). If the ratio is greater than $e^{-\tau}$, then the output should be the lc which produces the maximum posterior, or the output should be the lk.

$$\frac{max(P(\boldsymbol{O}|\lambda_i^{LC}))}{max(P(\boldsymbol{O}|\lambda_j^{LK}))} > e^{-\tau}, \; 1 \leqslant i \leqslant N_{aLC}, 1 \leqslant j \leqslant N_{aLK} \tag{3}$$

4.3. The BF component

The task of a Bayesian filtering is to compute the probability corresponding to each hypothetical behavior, given a sequence of previous preliminary classifications produced by the HMMs. A Bayesian filtering regards the recognition results of the HMMs component as samples of a binary random variable $\{lc, lk\}$, where lc and lk represent arbitrary sub-action of LC and LK. The probability of $y = lc$ can be denoted by the parameter μ such that

$$p(y = lc|\mu) = \mu \tag{4}$$

where $0 \leqslant \mu \leqslant 1$. Hence, the probability distribution over y can be written in the *Bernoulli* distribution form:

$$\mathrm{Bern}(y|\mu) = \mu^y(1-\mu)^{1-y} = \begin{cases} \mu & \textit{if } y = lc, \\ 1-\mu & \textit{if } y = lk, \\ 0 & \textit{otherwise}. \end{cases} \tag{5}$$

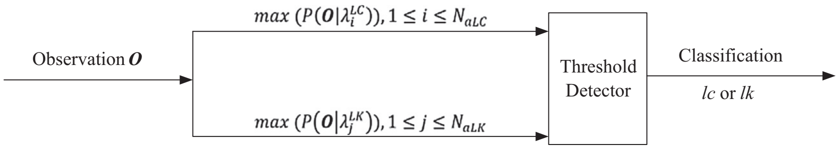

$max\,(P(\boldsymbol{O}|\lambda_i^{LC})), 1 \le i \le N_{aLC}$
Observation $\boldsymbol{O}$
Threshold Detector
Classification
lc or lk
$max\,(P(\boldsymbol{O}|\lambda_j^{LK})), 1 \le j \le N_{aLK}$

Fig. 6. The HMM-based classification architecture.

Given a sequence $\boldsymbol{y} = [y_1, \ldots, y_{L_b}]$ of preliminary classifications saved in the buffer, the distribution of the number m of $y = lc$ classifications can be written in the *binomial* distribution form:

$$\mathrm{Bin}(m|L_b, \mu) = \binom{L_b}{m} \mu^m (1-\mu)^{(L_b - m)} \tag{6}$$

where

$$\binom{L_b}{m} \equiv \frac{L_b!}{(L_b - m)!m!} \tag{7}$$

A standard Bayesian formulation (Bishop, 2006) is used to calculate the posterior distribution.

$$p(\mu|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mu)p(\mu)}{p(\boldsymbol{y})} \propto p(\boldsymbol{y}|\mu)p(\mu) \tag{8}$$

We already have the likelihood function, $\mathrm{Bin}(m|L_b, \mu)$, hence, a prior distribution for μ must be chosen. We note that the likelihood function takes the form $\mu^m(1-\mu)^{(1-m)}$. Hence, if a beta distribution is chosen as a prior, then the posterior has the same functional form as the prior (a so called conjugacy prior), thereby leading to a greatly simplified Bayesian analysis. The prior distribution for μ is given by:

$$p(\mu) = beta(\mu|a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \mu^{a-1} (1-\mu)^{b-1} \tag{9}$$

where $\Gamma(x)$ is the *gamma* function, and $a,\ b$ are the *hyperparameters* that control the distribution of μ. In this study, the parameters a and b represent the initial confidence of lane changing and lane keeping respectively.

The posterior distribution of μ can be calculated by multiplying the *beta* prior by the *binomial* likelihood and normalizing the product such that:

$$p(\mu|\boldsymbol{y}) = \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)\Gamma(l+b)} \mu^{(m+a-1)} (1-\mu)^{(l+b-1)} \tag{10}$$

where m and l represent the number of preliminary classifications corresponding to $y = lc$ and $y = lk$ respectively, and $N = m + l$.

Given the sequence $\boldsymbol{y} = [y_1, \ldots, y_{L_b}]$, the expected value of μ can be calculated as

$$E(\mu|\boldsymbol{y}) = \int_0^1 \mu p(\mu|\boldsymbol{y}) d\mu = \frac{m+a}{m+a+l+b} \tag{11}$$

Common sense tells us that the earlier behaviors in the sequence have less impact on the current behavior classification compared to the more recent ones. To improve classification accuracy, each element in the sequence must be assigned a different weight; hence, a sigmoid function is built to achieve the desired purpose:

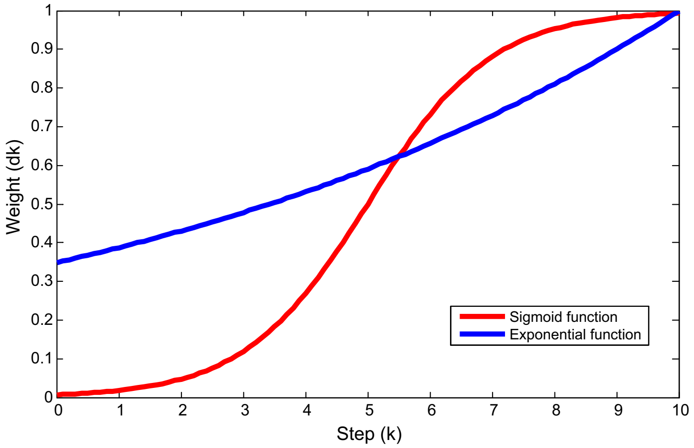


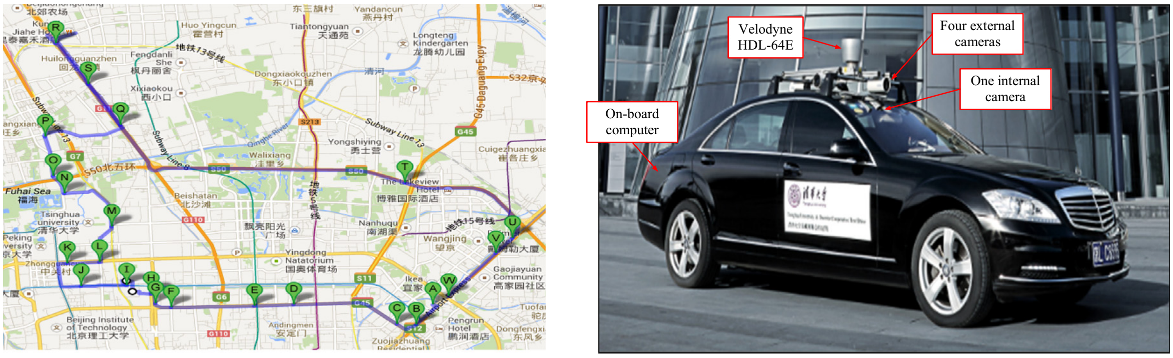
Fig. 7. Sigmoid function, $d_k = 1/\left(1 + e^{-r\left(\frac{20k}{L_b} - 10\right)}\right)$, and exponential function, $d_k = C^{(L_b - k)}$, where $r = 0.5,\ C = 0.9$ and $L_b = 10$. It can be seen that the blue line (exponential function) is approximately linear, assigning too high a weight to the earlier outputs and too low a weight to the later outputs. The red line (sigmoid function), in contrast, is more reasonable for practical applications. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$d_k = \frac{1}{1+e^{-r\left(\frac{20k}{L_b}-10\right)}}, \quad \text{with } d_0 = \frac{1}{1+e^{10r}} \tag{12}$$

where $k = 1, \ldots, L_b$ is the index of the element in the sequence and r is the shape parameter. The reason for choosing the sigmoid function is illustrated in Fig. 7. In comparison with exponential functions, such as $d_k = C^{(L_b-k)}$ (Aoude et al., 2012), the sigmoid assigns greater weight to the middle and later elements and reduces the weight allocated to earlier elements, a more reasonable strategy in practical applications. Given the weight function, (11) can be rewritten as

$$E(\mu|\boldsymbol{y}) = \frac{\sum_{k=1}^{L_b} d_k m_k + d_0 a}{\sum_{k=1}^{L_b} d_k m_k + d_0 a + \sum_{k=1}^{L_b} d_k l_k + d_0 b} \tag{13}$$

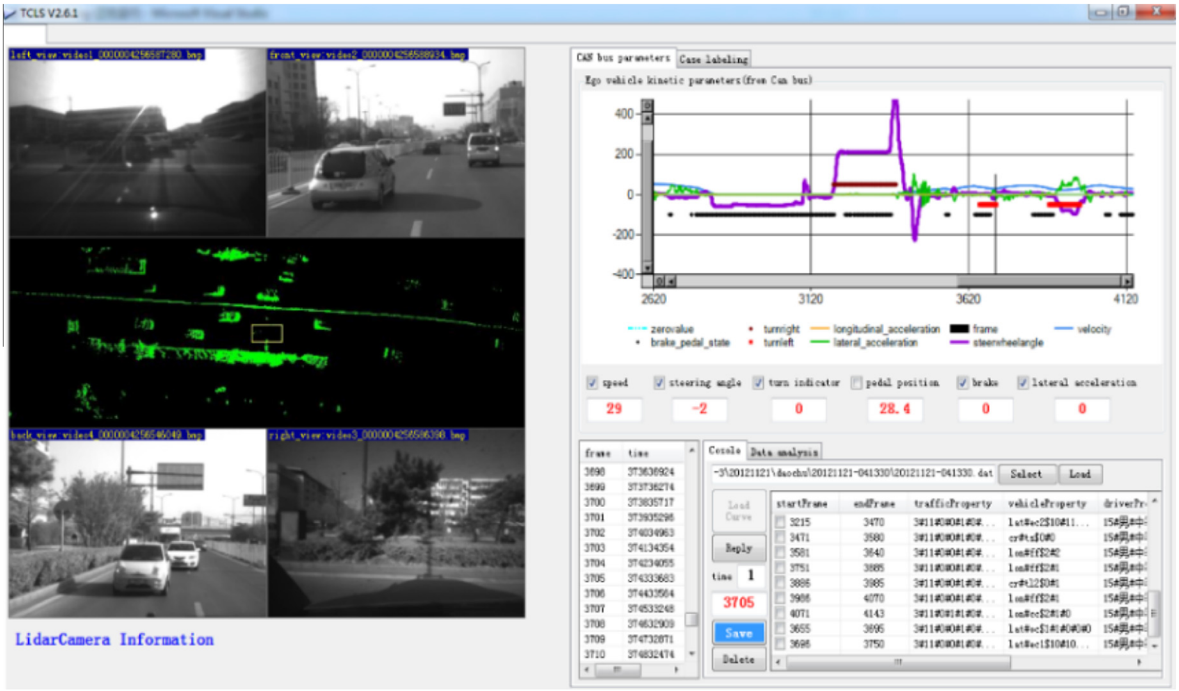
where m_k and l_k are the binary outputs at step k and $m_k + l_k = 1$. If the kth value of the sequence $\boldsymbol{y}$ is lc, then set $m_k = 1,\ l_k = 0$; otherwise, set $m_k = 0,\ l_k = 1$. The *hyperparameters a* and b are analyzed in Section 5.



(a) Test route-The highway, ring road, airport express, and normal city roads are approximately 2.1, 6, 2.6, and 7.1 kms. respectively.

(b) Mercedes S-Class platform and data collection equipment.

Fig. 8. Test route and data collection platform.



☑ speed 29
☑ steering angle -2
☑ turn indicator 0
□ pedal position 28.4
☑ brake 0
☑ lateral acceleration 0

frame	time
3698	3T3636924
3699	3T3736274
3700	3T3835717
3701	3T3935296
3702	3T4034963
3703	3T4134354
3704	3T4234055
3705	3T4333683
3706	3T4433564
3707	3T4533248
3708	3T4632909
3709	3T4732871
3710	3T4832474

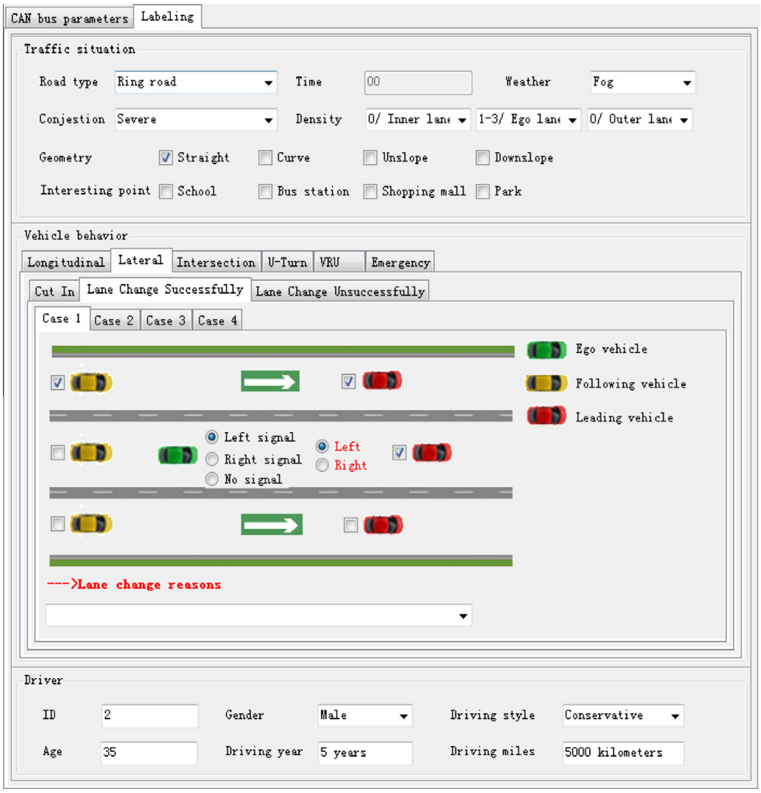
-3\20121121\daochu\20121121-041330\20121121-041330.dat

time 1

3705

	startFrame	endFrame	trafficProperty	vehicleProperty	driverPr
□	3215	3470	3#11#0#0#1#0#...	lat#ec2$10#11...	15#男#中
□	3471	3580	3#11#0#0#1#0#...	cr#ts$0#0	15#男#中
□	3581	3640	3#11#0#0#1#0#...	lon#ff$2#2	15#男#中
□	3751	3885	3#11#0#0#1#0#...	lon#ff$2#1	15#男#中
□	3886	3985	3#11#0#0#1#0#...	cr#tl2$0#1	15#男#中
□	3986	4070	3#11#0#0#1#0#...	lon#ff$2#1	15#男#中
□	4071	4143	3#11#0#1#1#0#...	lon#ec$2#1#0	15#男#中
□	3655	3695	3#11#0#0#1#0#...	lat#ec$1#1#0#0#0	15#男#中
□	3696	3750	3#11#0#0#1#0#...	lat#ec1$10#10...	15#男#中

Fig. 9. GUI of the TCLS developed specifically for manual labeling. The left area displays the real time LIDAR map and images of surrounding traffic situations. The top right area shows the CAN data curves.



Traffic situation

Road type: Ring road
Time: 00
Weather: Fog
Conjestion: Severe
Density: 0/ Inner lane; 1-3/ Ego lane; 0/ Outer lane
Geometry: ☑ Straight □ Curve □ Unslope □ Downslope
Interesting point: □ School □ Bus station □ Shopping mall □ Park

☑ ☑
□ ● Left signal ○ Right signal ○ No signal ● Left ○ Right ☑
□ □

--->Lane change reasons

Driver

ID: 2
Gender: Male
Driving style: Conservative
Age: 35
Driving year: 5 years
Driving miles: 5000 kilometers

Fig. 10. GUI for traffic case labeling, which includes traffic situation, vehicle behavior, and driver information. The traffic situation includes road type, weather, and collection time. Driver information includes gender, age, and driving experience (in years). The software designs a few vehicle behaviors, such as car following, lane change, intersection behavior, and U-turn behavior. In this study, only successful lane changes have been labeled.

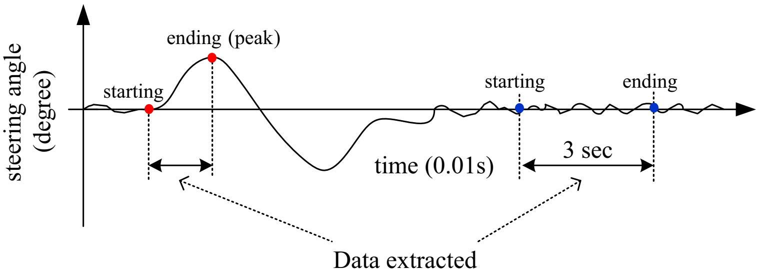


Fig. 11. LC and LK data extraction methods.

Table 1
Labeled samples' detailed information.

Object	Number	Distribution		
Lane changing left (LCL)	188	Highway: Ringroad: Airport express: Normal city road:	26% 42% 9% 23%	Male:63% Female:37%
Lane changing right (LCR)	212	Highway: Ringroad: Airport express: Normal city road:	29% 39% 7% 25%	Male:72% Female:28%
Lane keeping (LK)	242	Highway: Ringroad: Airport express: Normal city road:	32% 30% 21% 17%	Male:58% Female:42%

4.4. Threshold detector

Given $E(\mu|\mathbf{y})$ and time t, the HMM–BF algorithm generates an output of the final behavior classification based on the threshold, τ_{bf}. The behavior at time t is classified as an *lc* if $E(\mu|\mathbf{y}) > \tau_{bf}$; otherwise it is classified as an *lk*. The value to be chosen for τ_{bf} will be analyzed in Section 5.

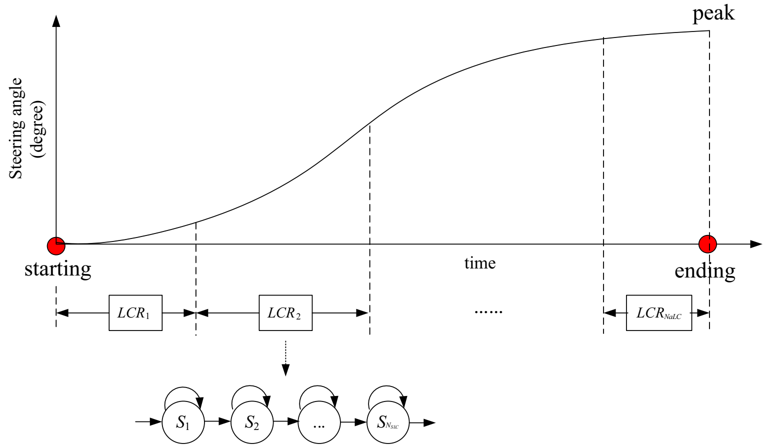


Fig. 12. Grammar corresponding to an LCR. We suppose that this event can be split into N_{aLC} sub-LCs, and that each HMM was built with a left–right structure, consisting of N_{sLC} states.

Table 2
Key parameters in the HMM–BF system framework.

Parameter	Definition	Considered values
N_{aLC}	Number of sub-LCs (sub-LCRs and sub-LCLs)	N_{aLC} ranges from 2 to 4
N_{aLK}	Number of sub-LKs	$N_{aLK} = 1$
N_{sLC}	States number of per sub-HMM of LC (LCR and LCL)	N_{sLC} is from 3–6
N_{sLK}	States number of per sub-HMM of LK	$N_{sLK} = 1$
L_w	Length of the moving window T_w, the subscript is a label	5, 10, and 15 frames
L_b	Buffer size for Bayesian filtering, the subscript is a label	5 and 10 frames
τ	Threshold for the HMMs output	From 0 to 100, the interval is 0.5
τ_{bf}	Threshold for the Bayesian filtering output, the subscript is a label	0.7, 0.8 and 0.9
a, b	Hyperparameters of gamma function. These values reflect a bias toward one driving behavior based on given prior knowledge. $a = b = 0.5$ means no bias toward either behavior	*a* = 0.5 *b* = 0.5
r	Sigmoid function parameter	0.5
n_{mix}	Number of mixture components in GMM, the subscript is a label	1

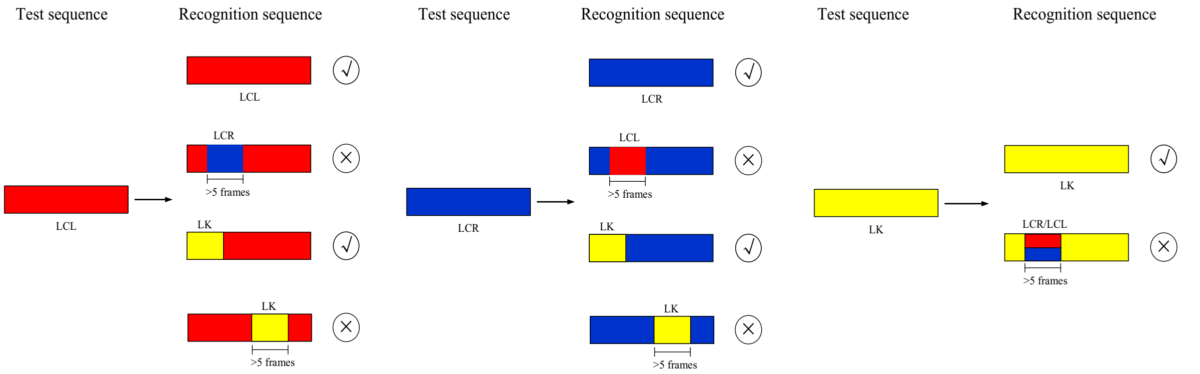


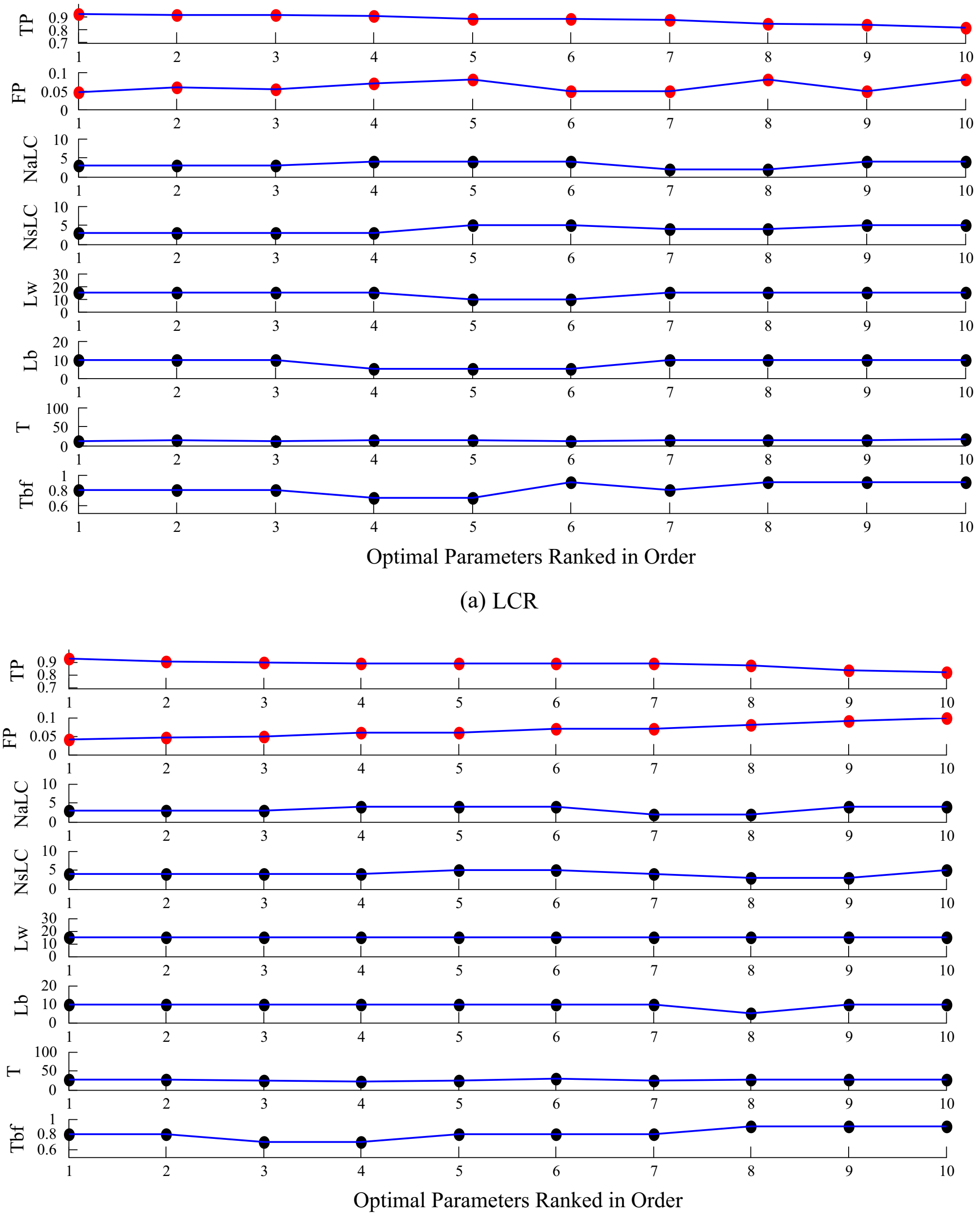
Fig. 13. Rules for judging whether a test sequence recognition result is correct or incorrect.

5. Data preparation

5.1. Data collection

Real and accurate traffic data are essential for model building. Fifty participants were recruited to drive a vehicle equipped with sensors around the roads in Beijing, China. Fig. 8(a) shows the test route, which includes four road types—a highway, ring road, airport express lane, and a normal city road of lengths approximately 2.1, 6, 2.6, and 7.1 km, respectively. A Mercedes S-Class vehicle (see Fig. 8(b)) equipped with the following sensors was used for data collection.

- **Velodyne HDL-64E scanning LIDAR.** This sensor is mounted on the roof of the vehicle and used to detect the surrounding objects. It has 64 lasers, a 360° horizontal field of view, a 26.8° vertical field of view, and a 0.09° angular resolution. In this study, this sensor is used to collect information regarding the surrounding vehicles.



(a) LCR

(b) LCL

Fig. 14. Parameter combinations of the ten best recognition results.

- **CAN bus.** This sensor provides kinetic parameters of the vehicle including yaw rate, lateral acceleration, speed, steering angle, brake pressure, throttle pedal position, and turn signals.
- **Cameras.** All cameras are black and white and used for documentation. One internal camera provides driver's information, and four external cameras provide road and traffic information.

None of the 50 participants are professional drivers; all are common drivers. In order to collect real driving and traffic data, participants were asked to maintain their usual driving style.

5.2. Data post-processing

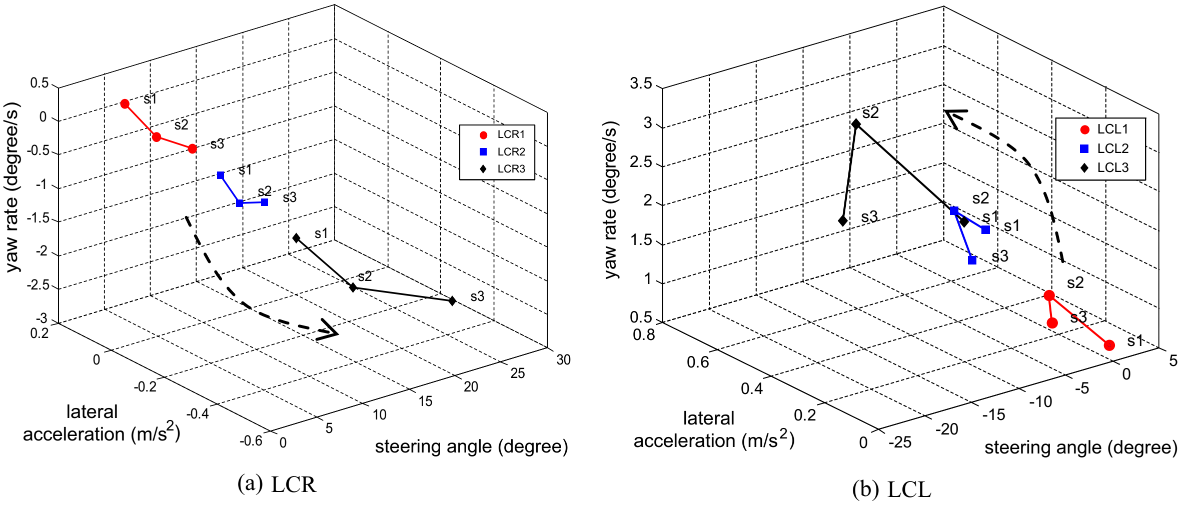
The goal of data post-processing is to extract the LC and LK events from the collected data and label corresponding properties, such as starting and ending point of an LC, road type, and weather and driver information. Fig. 9 shows a graphical user interface (GUI) of the traffic case labeling system (TCLS), developed for manual labeling and parameter calculation. The primary functions include:

- **Display of LIDAR map and photos.** A point cloud collected by the LIDAR in each frame is drawn by OpenGL. Photos are from surrounding traffic situations collected by the four external cameras. A time synchronization algorithm is used to fuse LIDAR and camera data.
- **Display of CAN bus data.** CAN bus signals are represented by curves of different colors, with a signal collection frequency of 100 Hz.
- **Maneuver labeling.** The maneuver type is selected and marked with the starting and ending times, and the traffic situation and driver information are also labeled. Fig. 10 shows the GUI.
- **Parameter calculation.** Based on the LIDAR data, the target vehicles (a leading vehicle in the original lane and a leading and real vehicle in the target lane) are detected using segmentation and classification algorithms.

There are two labeling criteria: first, for LCs and LKs, the velocity at the starting point should be greater than 30 km/h; second, for LCs, the duration should be lower than 8 s. The procedures for labeling an LC are as follows. First, the labeling operators quickly browse the data using the LIDAR map and images, and mark the rough position of an LC. Next, if the velocity and duration meet the criteria, the steering angle is used as a guide to mark an accurate LC start and end. Normally, since the steering angle curve of a lane change can be expected to resemble a cosine or sine curve, the starting time is set to be the

Table 3
Best parameter combinations.

LCR	LCL
$N_{aLC} = 3$	$N_{aLC} = 3$
$N_{sLC} = 3$	$N_{sLC} = 3$
$L_w = 15$	$L_w = 15$
$L_b = 10$	$L_b = 10$
$\tau = 12.5$	$\tau = 26$
$\tau_{bf} = 0.8$	$\tau_{bf} = 0.8$



(a) LCR (b) LCL

Fig. 15. Learning performance of HMMs.

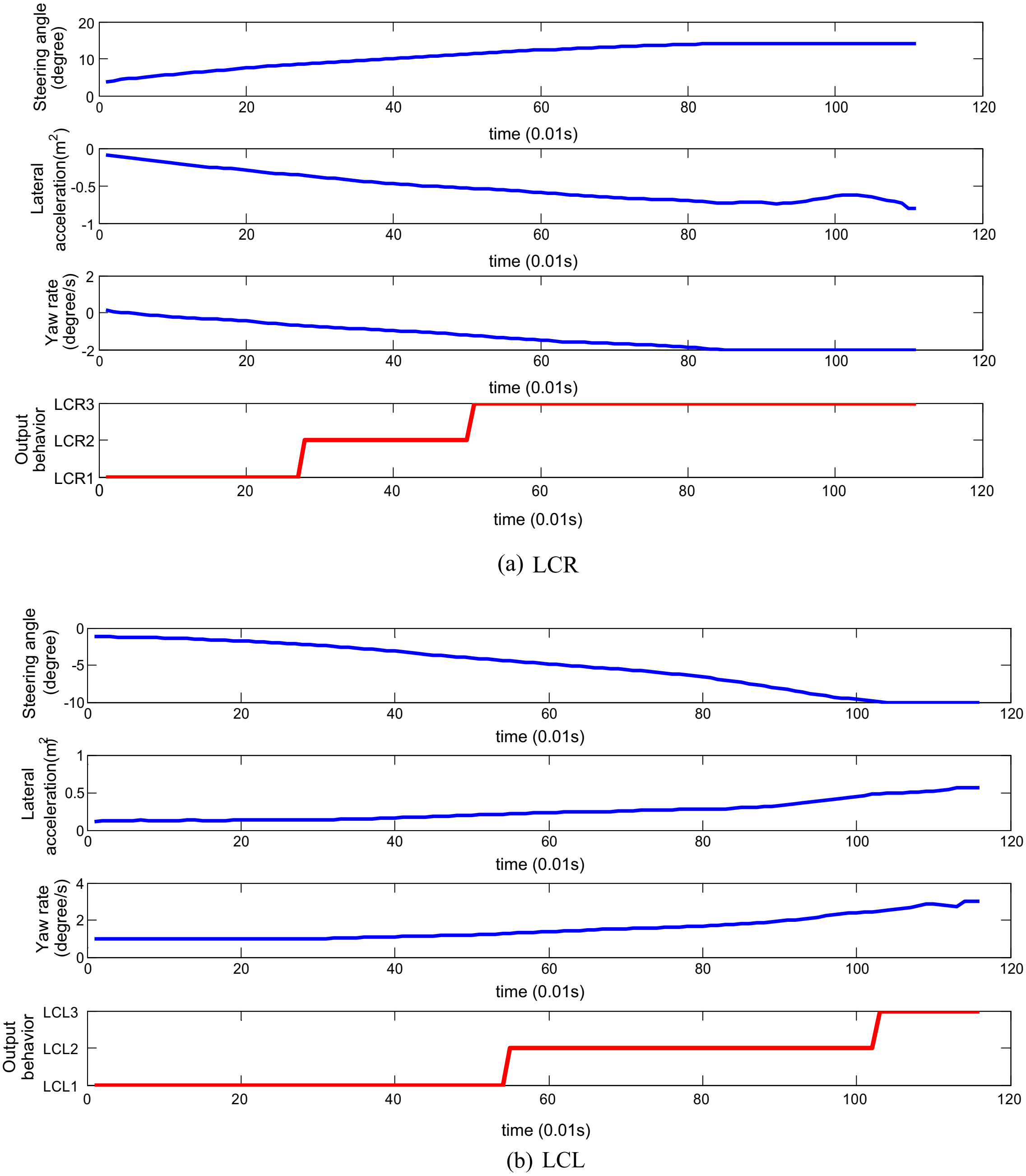
initial time of a cosine or sine curve, and the ending time is set to be the first peak (Kuge et al., 2000) (see Fig. 11). For an LK instance, data which meet the criteria are extracted for a 3 s interval. Finally, the extracted data are labeled with other associated properties (e.g., traffic situation and driver information).

A total of 642 samples were extracted, with the detailed information shown in Table 1.

6. Experiments

6.1. HMM grammar

As described in Section 2, an LC can be considered as a sequence of N_{aLC} sub-LCs, and each sub-LC can be trained with an independent HMM. Suppose that each HMM consists of N_{sLC} states, and that a left–right structure is adopted. Fig. 12 shows



(a) LCR

(b) LCL

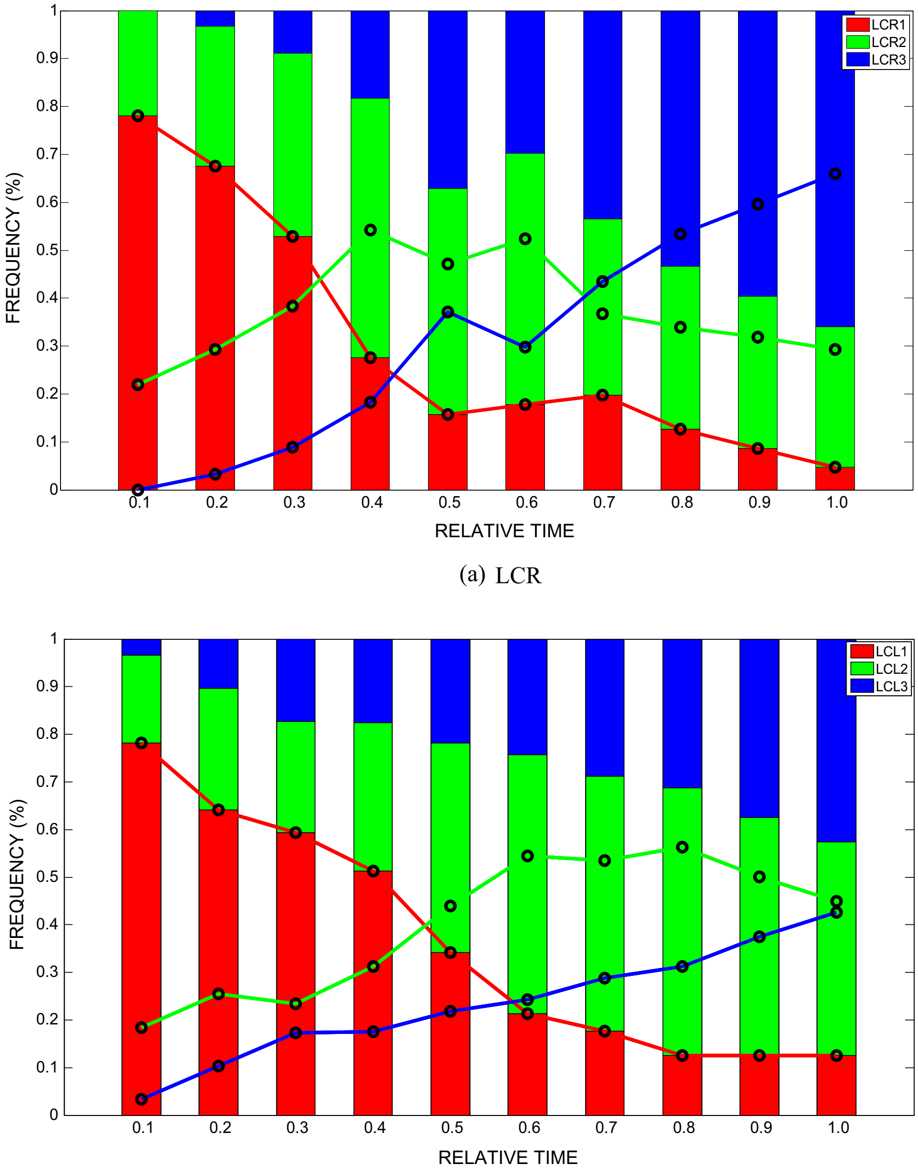
Fig. 16. Distribution of the sub-behavior recognition results.

the grammar corresponding to an LCR; the parameters N_{aLC} and N_{sLC} will be determined later. The HTK toolkit, which is a powerful tool for speech recognition, is used to develop the HMM-based behavior recognition system (Entropic Inc.©).

6.2. Parameters

The key parameters listed in Table 2 must be extracted from the HMM–BF system framework. A combination test method is used to search for the best parameters because there are no analytical methods for calculating the optimal values. To reduce the complexity, some parameters are confined to a certain range and the remaining are determined according to experience and based on previous studies (Kuge et al., 2000; Aoude et al., 2011, 2012).

A 5-fold cross validation and the receiver operation characteristic (ROC) curve are used to find the best parameters. In the ROC curve, the decision threshold, τ, is selected as the beta parameter in the signal detection theory (SDT) (McNicol, 2005). The rules for judging, whether or not the recognition result of a sequence is correct or incorrect, are shown in Fig. 13.



(a) LCR

(b) LCL

Fig. 17. Statistical distributions of the sub-LC recognition results. The horizontal axis represents the normalized relative time from 0.1 to 1, and the vertical axis represents the frequency of each sub-LC.

For a test sequence of LCL (correspondingly LCR), if a recognition result in frame *f* (*f* locates an arbitrary position of the sequence) is sub-LCR (correspondingly sub-LCL) and if this result remains unchanged continuously for more than five frames, then the recognition is considered to have failed. If a recognition result is sub-LK in frame *f* (*f* locates the starting point), then the recognition continues; otherwise, if *f* locates the non-starting point, and there are no sub-LK results prior to *f*, the recognition is considered to have failed.

For a test sequence of LK, if a recognition result is sub-LCR or sub-LCL and if this result is maintained continuously for more than five frames, then the recognition is considered to have failed.

According to the rules, for LCR and LCL models, all combinations of these parameters were calculated and ten combinations of the highest rates of the true positive (TP) rate value, while maintaining a low false positive (FP) rate, are shown in Fig. 14.

The best parameter combinations are listed in Table 3.

6.3. Results

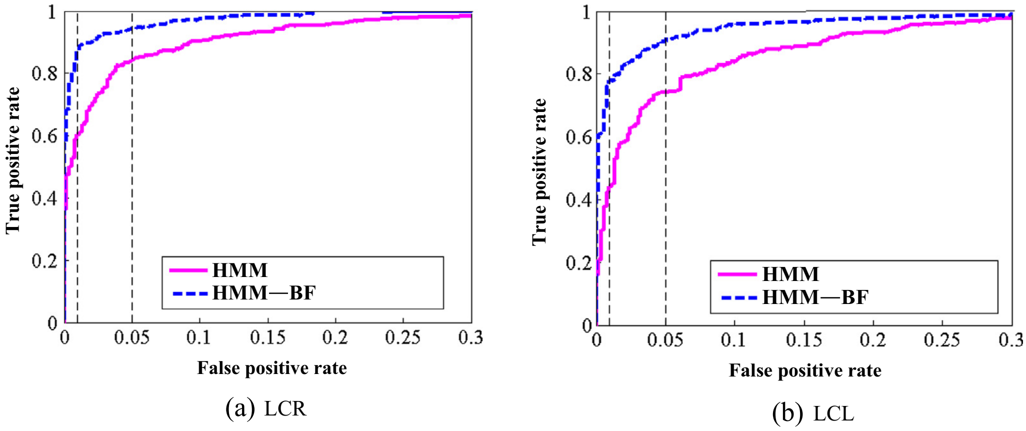
6.3.1. Learning performance of HMMs

In this study, the output distribution is modeled using Gaussian mixture densities; hence, the change in the average values of the output distribution accompanying a state transition can be used to evaluate the performance of the HMMs. Fig. 15 illustrates the learning performance of LCR and LCL, with the states of each sub-LCR or sub-LCL represented by the average values of the steering angle, lateral acceleration, and yaw rate. For LCR (see Fig. 15(a)), we observe that the changes in the steering angle, lateral acceleration and yaw rate correspond well to a typical lane changing right. To be specific, when a right lane change begins, the steering angle gradually increases until it comes to the first peak, and the other two parameters also change accordingly. The same distribution can be found in LCL (see Fig. 15(b)). Thus, we can conclude that the trained sub-HMMs of LCR and LCL learned the data well, the characteristics of a temporal change in a lane changing scenario, and can be utilized to recognize the intention of an LC.

6.3.2. Distribution of sub-LCs

Fig. 16 shows the sub-LCs of an LCR and LCL. In Fig. 16(a), the horizontal axis represents the absolute time and the four vertical axes represent the steering angle, lateral acceleration, yaw rate, and sub-LCs respectively. For LCR (see Fig. 16(a)), we observe that LCR1 appeared almost simultaneously with the starting time, followed by LCR2 and LCR3. The durations of the three sub-LCRs were about 0.28, 0.22, and 0.65 s, respectively. For LCL (see Fig. 16(b)), we could also see that as the steering angle decreased, LCL1, LCL2, and LCL3 appeared one by one and were stable.

Fig. 17 shows the statistical distributions of all the recognition results; the results were calculated based on a normalized relative time from 0.1 to 1 because the duration of LC samples are different. The vertical axis represents the frequency of each sub-LC.



(a) LCR

(b) LCL

Fig. 18. ROC curves for LCR and LCL based on the HMM and HMM–BF algorithms.

Table 4
Percentage detected at 1% and 5% FP rate, the bold numbers are recognition rate at 5% FP.

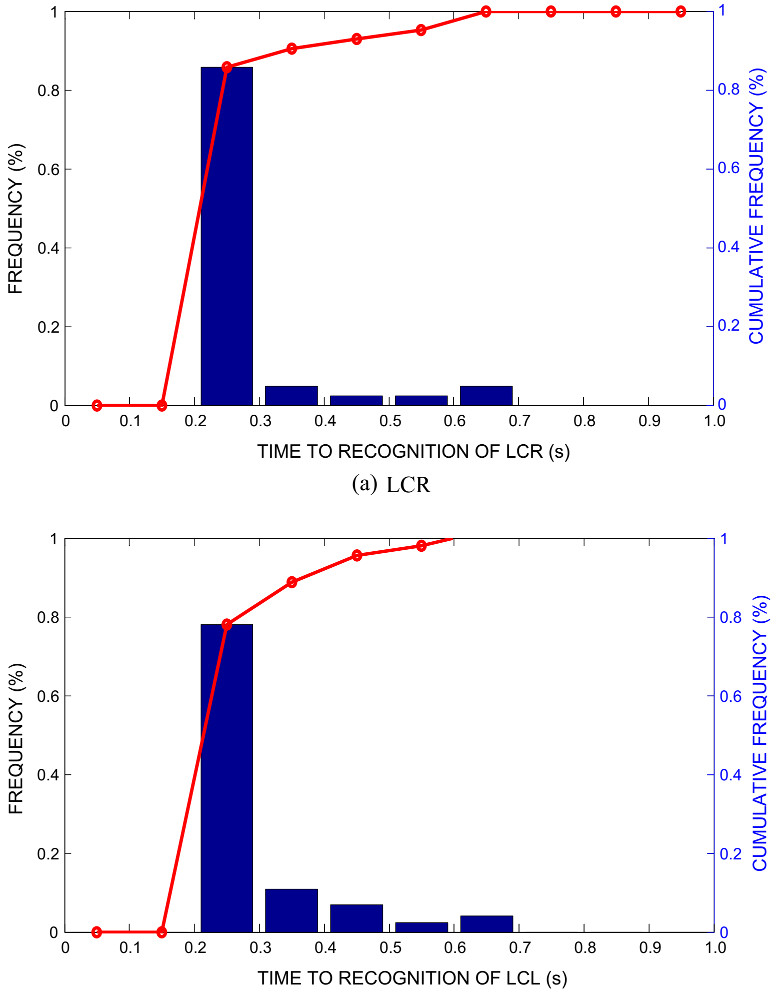
	LCR		LCL	
	1% FP	5% FP	1%FP	5%FP
HMM	60.4%	81.7%	44.3%	73.1%
HMM-BF	85.8%	**93.5%**	78.4%	**90.3%**

For LCR (see Fig. 17(a)), we observe that LCR1 occupied the greatest frequency for a relative time of 0.1–0.3; especially at 0.1 relative time, LCR1 accounted for approximately 80%. For relative time between 0.4 and 0.6, the frequency of LCR2 gradually exceeded LCR1, and for the remaining stages, LCR3 showed the greatest frequency. Note that, strangely, at 0.1 relative time, LCR2 accounted for nearly 20%, and at 1.0 relative time, LCR1 accounted for nearly 6% and LCR2 even accounted for nearly 24%. One could wonder why LCR1 did not account for 100% at the start and LCR3 did not account for 100% towards the end. The reason is that under normal driving conditions, the sub-behaviors would be expected to appear in an order, akin to the order of LCR1, LCR2, and LCR3 as observed in Fig. 16(a). Nevertheless, for emergent LCs, the initial results may correspond to LCR2 or LCR3, and no LCR1 events were recognized. This is because when we perform an emergency lane change, the wheel is turned with a wide steering angle within a short time during the initial stage compared with the scenario of a normal lane change. Consequently, the current input features have a higher likelihood of LCR2 or LCR3 than LCR1. For normal LCs which have a long duration and a small steering angle peak, the result may entirely show up to be LCR1 or LCR2, with no occurrence of LCR3. The same conclusion can be drawn from LCL in Fig. 17(b). From these results, we can conclude that the LC behavior can be recognized clearly and hierarchically based on the proposed algorithm.

6.3.3. ROC curves

Fig. 18 presents the ROC curves for the HMM–BF and HMM-only algorithms. The algorithms were tested with 5-fold cross validation as previously described, and the threshold, τ, was selected as the beta parameter.

The ROC curves clearly demonstrate the superiority of the HMM–BF against the HMM-only algorithm. Table 4 demonstrates that based on the HMM–BF, the TP rates for LCR and LCL were at 5%, while the FP rates were 93.5% and



(a) LCR

(b) LCL

Fig. 19. Statistical distribution of the time to produce the results. We can see that the recognition times are concentrated in the region between 0.2 and 0.3 s for both, LCR and LCL.

90.3% respectively, which was 12 and 17 percentage points higher than the TP rates based on the HMM-only algorithm. At 1% FP rate, the HMM–BF was far more superior to the HMM-only algorithm, as shown in Table 4. We conclude that compared with the HMM-only algorithm, the HMM–BF framework does improve the performance of true LC detection significantly.

6.3.4. Recognition time

Recognition time is critical for ADAS and the sooner they predict the lane changes, the better the system gets to be. Fig. 19 demonstrates the recognition time for LCR and LCL with the blue bars representing the frequency and the red dotted lines corresponding to the cumulative frequency. The time is calculated from the start of an LC until the first correct sample is recognized; note that all the test sequences used for time elapsed analysis should be correctly recognized first according to the rules listed in Fig. 13. We can see that the recognition times for LCR and LCL were concentrated in the region between 0.2 and 0.3 s. These results indicate that the sub-HMMs can classify the behavior correctly at an early enough stage.

7. Conclusion

We proposed a novel lane changing intention recognition algorithm combining the HMM and BF models. In the HMM component, learning from speech recognition models, we split a driving behavior into sub-behaviors and then built the corresponding sub-HMMs. The input parameters consisted of three signals from CAN bus (steering angle, lateral acceleration, and yaw rate), and the output was a preliminary behavior classification. The BF was then used to improve the recognition performance; in this component, a sigmoid function was used to assign different weights to preliminary outputs of the HMMs, and a threshold detector was used to produce the final behavior classification. Models were developed using a naturalistic data set collected in Beijing, China, where parameters are optimized using experimental methods. The results reveal that the HMM–BF can achieve a recognition accuracy of 93.5% and 90.3% for lane changing left and right, respectively, showcasing a significant improvement when compared to the HMM-only algorithm. The recognition time results show that the proposed algorithm can also recognize a behavior correctly at a relatively early stage.

Our future study will focus on building an improved driving behavior dictionary that includes not only lane changing and lane keeping but also turns, U-turns, and other common driving behaviors. In the driving behavior dictionary, behaviors are decomposed into finite and standard sub-actions, whose relationships are just like that between words and phonetics in a language dictionary. This study can be expected to produce a deeper understanding of driving behaviors and provide a standardized means of recognizing driving intention. Moreover, certain environmental features, such as the distance between ego and target vehicles, and the flow and density of each lane, will also be incorporated into the model to enhance its robustness and practicality.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 51175290 and 51475254) and in part by the joint research project of Tsinghua University and Daimler®. The authors would like to thank Prof. Dariu M. Gavrila (Dept. of Environment Perception, Daimler Research & Development) for valuable discussions. They would also like to thank the reviewers for their time and suggestions.

References

Aoude, G.S., Desaraju, V.R., Stephens, L.H., How, J.P., 2011. Behavior classification algorithms at intersections and validation using naturalistic data. In: IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, pp. 601–606.

Aoude, G.S., Desaraju, V.R., Stephens, L.H., How, J.P., 2012. Driver behavior classification at intersections and validation on large naturalistic data set. IEEE Trans Intell Transport Syst 13 (2), 724–736.

Berndt, H., Dietmayer, K., 2009. Driver intention inference with vehicle onboard sensors. In: IEEE Vehicular Electronics and Safety (ICVES), Pune, India, pp. 102–107.

Berndt, H., Emmert, J., Dietmayer, K., 2008. Continuous driver intention recognition with hidden Markov models. In: IEEE Intelligent Transportation Systems (ITSC), Beijing, China, pp. 1189–1194.

Bilal, S., Akmeliawati, R., Shafie, A.A., Salami, M.J.E., 2013. Hidden Markov model for human to computer interaction: a study on human hand gesture recognition. Artif Intell Rev 40 (4), 495–516.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.

Boer, E.R., Hildreth, E.C., Goodrich, M.A., 1998. A driver model of attention management and task scheduling: satisficing decision making with dynamic mental models. In: Proceedings of the XVIIth European Annual Conference on Human Decision making and Manual Control, Valenciennes, France, pp. 14–16.

Brand, M., Oliver, N., Pentland, A., 1997. Coupled hidden Markov models for complex action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, pp. 994–999.

Bunke, H., Roth, M., Schukat-Talamazzini, E.G., 1995. Off-line cursive handwriting recognition using hidden Markov models. Pattern Recogn. 28 (9), 1399–1413.

Chen, F.-S., Fu, C.-M., Huang, C.-L., 2003. Hand gesture recognition using a real-time tracking method and hidden Markov models. Image Vis. Comput. 21 (8), 745–758.

Chen, M.-Y., Kundu, A., Zhou, J., 1994. Off-line handwritten word recognition using a hidden Markov model type stochastic network. IEEE Trans. Pattern Anal. Mach. Intell. 16 (5), 481–496.

Dang, R., Zhang, F., Wang, J., Yi, S., Li, K., 2013. Analysis of Chinese driver's lane change characteristic based on real vehicle tests in highway. In: IEEE Intelligent Transportation Systems (ITSC), The Hague, the Netherlands, pp. 1917–1922.

Dittmar, T., Krull, C., Horton, G., 2015. A new approach for touch gesture recognition: conversive hidden non-Markovian models. J. Comput. Sci. 10, 66–76.

Entropic Cambridge Research Laboratory Inc., 2001. <http://www/entropic.com/>.

Gadepally, V., Krishnamurthy, A., Ozguner, U., 2014. A framework for estimating driver decisions near intersections. IEEE Trans. Intell. Transport. Syst. 15 (2), 637–646.

Gales, M.J.F., Young, S.J., 2007. The application of hidden Markov models in speech recognition. Found. Trends Signal Process. 1 (3), 195–304.

Gipps, P.G., 1986. A model for the structure of lane-changing decisions. Transport. Res. Part B: Methodol. 20 (5), 403–414.

Hidas, P., 2002. Modelling lane changing and merging in microscopic traffic simulation. Transport. Res. Part C: Emerg. Technol. 10 (5), 351–371.

Hidas, P., 2005. Modelling vehicle interactions in microscopic simulation of merging and weaving. Transport. Res. Part C: Emerg. Technol. 13 (1), 37–62.

Hughey, R., Krogh, A., 1996. Hidden Markov models for sequence analysis: extension and analysis of the basic method. Comput. Appl. Biosci.: CABIOS 12 (2), 95–107.

Kuge, N., Yamamura, T., Shimoyama, O., Liu, A., 2000. A Driver Behavior Recognition Method Based on a Driver Model Framework. SAE Technical Paper.

Lee, S.E., Olsen, E.C., Wierwille, W.W., 2004. A Comprehensive Examination of Naturalistic Lane-Changes. Nat. Highway Traffic Safety Admin., U.S. Dept. Transp., Washington, DC, Rep. DOT HS 809702.

McNicol, D., 2005. A Primer of Signal Detection Theory. Allen & Unwin, Sydney.

Mitrovic, D., 2005. Reliable method for driving events recognition. IEEE Trans. Intell. Transport. Syst. 6 (2), 198–205.

Pentland, A., Liu, A., 1999. Modeling and prediction of human behavior. Neural Comput. 11 (1), 229–242.

Rabiner, L., Juang, B.H., 1986. An introduction to hidden Markov models. ASSP Mag. 3 (1), 4–16.

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. In: Proc. IEEE, vol. 77, pp. 257–286.

Talebpour, A., Mahmassani, H.S., Hamdar, S.H., 2015. Modeling lane-changing behavior in a connected environment: a game theory approach. Transport. Res. Part C: Emerg. Technol. 59, 216–232.

Toledo, T., Katz, R., 2009. State dependence in lane-changing models. Transport. Res. Rec.: J. Transport. Res. Board 2124, 81–88.

Van der Horst, A.R.A., 1990. A time-based analysis of road user behaviour in normal and critical encounters. Dissertation Delft University of Technology, The Netherlands.

Wang, M., Hoogendoorn, S.P., Daamen, W., van Arem, B., Happee, R., 2015. Game theoretic approach for predictive lane-changing and car-following control. Transport. Res. Part C: Emerg. Technol. 58, 73–92.

Welch, L.R., 2003. Hidden Markov models and the Baum–Welch algorithm. IEEE Inform. Theory Soc. Newsletter 53 (4), 10–13.

Worrall, R.D., Bullen, A.G.R., Gur, Y., 1970. An elementary stochastic model of lane-changing on a multilane highway. Highway Res. Rec. 308, 1–12.

Yang, Q., Koutsopoulos, H.N., 1996. A microscopic traffic simulator for evaluation of dynamic traffic management systems. Transport. Res. Part C: Emerg. Technol. 4 (3), 113–129.

Zheng, Z., 2014. Recent developments and research needs in modeling lane changing. Transport. Res. Part B: Methodol. 60, 16–32.

Zheng, Z., Ahn, S., Chen, D., Laval, J., 2013. The effects of lane-changing on the immediate follower: anticipation, relaxation, and change in driver characteristics. Transport. Res. Part C: Emerg. Technol. 26, 367–379.

Zou, X., Levinson, D., 2006. Modeling pipeline driving behaviors: hidden markov model approach. Transport. Res. Rec.: J. Transport. Res. Board 1980, 16–23.