Modeling Mandatory Lane Changing Using Bayes Classifier and Decision Trees

Yi Hou, Praveen Edara, and Carlos Sun

***Abstract*—A lane changing assistance system that advises drivers of safe gaps for making mandatory lane changes at lane drops is developed. Bayes classifier and decision-tree methods were applied to model lane changes. Detailed vehicle trajectory data from the Next Generation Simulation (NGSIM) data set were used for model development (U.S. Highway 101) and testing (Interstate 80). The model predicts driver decisions on whether to merge or not as a function of certain input variables. The best results were obtained when both Bayes and decision-tree classifiers were combined into a single classifier using a majority voting principle. The prediction accuracy was 94.3% for nonmerge events and 79.3% for merge events. In a lane change assistance system, the accuracy of nonmerge events is more critical than merge events. Misclassifying a nonmerge event as a merge event could result in a traffic crash, whereas misclassifying a merge event as a nonmerge event would only result in a lost opportunity to merge. Sensitivity analysis performed by assigning higher misclassification cost for nonmerge events resulted in even higher accuracy for nonmerge events but lower accuracy for merge events.**

***Index Terms*—Bayesian methods, decision trees, driver behavior, intelligent transportation system, lane changing assistance.**

I. Introduction

With the increase in deployment of sensor technology in automobiles, driver assistance systems, such as adaptive cruise control, collision avoidance systems, and lane-departure warning systems, have become a reality in the recent years. In terms of lane changing assistance, the current technology focuses primarily on blind spot identification and warning. Limited research exists on other forms of lane changing assistance systems. In this paper, a lane changing assistance system that advises drivers of safe and unsafe gaps for making mandatory lane changes is developed.

Lane changing models describe drivers' lane changing behaviors under various traffic conditions. These models are an essential component of microscopic traffic simulation and have been extensively studied in the literature. Much of the literature on lane changing models is based on gap acceptance. A driver makes a lane change when both the lead and lag gaps in the target lane are acceptable. Given the assumption on the distribution of critical lead and lag gap lengths, various gap acceptance models were built in 1960s and 1970s. Herman and Weiss [1] assumed an exponential distribution for critical gaps, Drew *et al.* [2] assumed lognormal distribution, and Miller [3] assumed a normal distribution. Daganzo [4] modeled driver's merging from the minor leg of a stop-controlled T-intersection to the major leg using a probit model. Gipps [5] designed a lane changing model that was implemented in a microscopic traffic simulator. Kita [6] modeled driver's merging behavior from freeway on-ramp using a logit model for gap acceptance. Yang and Koutsopoulos [7] established a rule-based lane changing model that was incorporated into the microscopic simulator MITSIM. Ahmed *et al.* [8] developed a generic lane changing model that captures lane changing behavior under both mandatory and discretionary lane changes. Kita [9] also developed a game-theoretic lane changing model. A two-person nonzero noncooperative game was developed to model the interaction of drivers in the target lane and the merging lane. Hidas [10] used intelligent-agent-based techniques to model driver's lane changing behavior and implemented the model in the ARTEMiS traffic simulator. Toledo *et al.* [11] proposed an integrated driving behavior model that captures both lane changing and acceleration behaviors. Recently, Meng and Weng [12] used statistical methods such as the classification and regression tree to predict merging behavior near work zone tapers. In a recent study [13], Hou *et al.* have developed a genetic fuzzy model to predict merging behavior of drivers at lane drops.

In summary, several types of lane changing models have been proposed in the literature with the main goal of developing accurate traffic simulation models. However, none of these models were intended for use in a real-time lane changing assistance system that advises drivers of when it is safe or unsafe to merge. One main difference between the simulation and the lane change assistance system applications is the relative importance of misclassification between merge and nonmerge decisions. In a simulation model, the effect of a nonmerge event misclassified as a merge event only affects the mobility measures. The same misclassification, however, in a lane change assistance system could impact traffic safety significantly. Thus, any model of lane change targeted for use in vehicles as part of an assistance system must give more importance to not misclassifying nonmerge events as merge events. On the other hand, misclassifying a merge event as a nonmerge event would result in a lost opportunity to merge but would not have a negative safety effect. Many of the models proposed in the literature are not appropriate for this new application.

Manuscript received March 19, 2013; revised July 3, 2013 and August 22, 2013; accepted October 2, 2013. Date of publication October 25, 2013; date of current version March 28, 2014. This work was supported in part by the Mid-America Transportation Center, U.S. Department of Transportation Region VII University Transportation Center. The Associate Editor for this paper was S. Sun.

The authors are with the Department of Civil and Environmental Engineering, University of Missouri, Columbia, MO 65211 USA (e-mail: yhcz4@mail.missouri.edu; edarap@missouri.edu; csun@missouri.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2013.2285337

In this paper, Bayes classifier and decision-tree methods are applied to develop models for mandatory lane changes at lane drops. Both methods have been applied extensively in machine learning systems built for decision-making in many disciplines. They have several advantages for modeling lane changing. Both of them relax the assumptions of traditional lane changing model's mathematical forms and variable distributions. Therefore, they can mimic the complex nonlinear nature of driver's lane changing behavior more realistically. One additional advantage of a Bayes classifier is its ability to take into account the cost of misclassification. In a Bayes classifier, it is possible to assign a higher cost of misclassification to nonmerge events than to merge events.

Bayes classifier and decision-tree models were developed using the same training and validation data. Then, both classifiers were combined into a single hybrid classifier. The combined classifier, when tested on a new data set from a different highway segment, outperformed the individual classifiers in terms of the accuracy of nonmerge events. In this paper, mandatory lane changes at lane drops refer only to those executed by traffic entering from a ramp. The lane changes made by vehicles exiting the mainline, although also mandatory, were not included in this paper. Discretionary lane changes performed when drivers perceive the driving conditions in the target lanes to be better are also beyond the scope of this paper.

II. Data

A. Data Reduction

In this paper, traffic data provided by the Federal Highway Administration's NGSIM project [14] was used to build the lane changing models. The NGSIM data set is an open-source data set that has been used in previous research for simulation model development and testing [15], [16]. NGSIM data include vehicle trajectories on a segment of southbound U.S. Highway 101 (Hollywood Freeway) in Los Angeles, CA, and a segment of Interstate 80 in San Francisco, CA. U.S. Highway 101 data were collected for 45 min from 7:50 A.M. to 8:35 A.M., on June 15, 2005. Interstate 80 data were also collected for 45 min from 4:00 P.M. to 4:15 P.M. and from 5:00 P.M. to 5:30 P.M., on April 13, 2005. Both data sets represent two traffic states: conditions when congestion was building up (period of the first 15 min), which are denoted as the transition period, and congested conditions (period of the remaining 30 min). Table I shows the aggregate speed and volume statistics of the NGSIM data set for every 15 min. For the congested period, the flows and speeds both decreased. As depicted in Fig. 1, the study segment of U.S. Highway 101 was located between an on-ramp and off-ramp and was 2100 ft long with five freeway lanes and an auxiliary lane. The study segment of Interstate 80 was 1650 ft in length and had five freeway lanes and an auxiliary lane, and one on-ramp.

Past research studies [18]–[21] have shown that NGSIM speed measurements exhibit noise (random errors). Data smoothing techniques such as moving average [21], Kalman filtering [22], and Kalman smoothing [23] have been used to

TABLE I
(a) Summary Statistics of U.S. Highway 101 Data Set [17].
(b) Summary Statistics of Interstate 80 Data Set [14]

(a)

Traffic condition	Time period	Flow (vph)	Time mean speed	
			m/s	km/h
Transition	7:50 a.m. – 8:05a.m.	8612	12.55	45.16
Congested	8:05 a.m. – 8:20a.m.	8016	11.10	39.96
	8:20 a.m. – 8:35a.m.	7604	9.74	35.05

(b)

Traffic condition	Time period	Flow (vph)	Time mean speed	
			m/s	km/h
Transition	4:00 p.m. – 4:15p.m.	8144	9.92	35.71
Congested	5:00 p.m. – 5:15p.m.	7288	8.34	30.13
	5:15 a.m. – 5:30a.m.	7048	7.78	28.00

improve speed data quality. In this paper, the moving average method was adopted to smooth the speed measurements.

The longitudinal and lateral coordinates, speed, acceleration, and headway for each vehicle were obtained from trajectory data at a resolution of 10 frames/s. Given the focus of this paper on mandatory lane changes, only trajectory data of vehicles in the auxiliary lane and the adjacent lane were used for model development. Hereafter, the auxiliary lane is referred to as the merge lane, and the adjacent lane is referred to as the target lane. The speed and position of each vehicle were identified in 1-s intervals. The 1-s interval produced data with comparable sample sizes for both lane changing and nonlane changing events. Other researchers [12] have also used a 1-s interval for analyzing the lane changing behavior of drivers. Since it is impossible to determine the intent of the driver using vehicle trajectory data alone, the observed behavior of drivers is modeled. During every 1-s interval, a driver's behavior is identified as either merge or no-merge. Merge events occurred when a vehicle's lateral coordinate began to shift toward the adjacent target lane direction without oscillations. Otherwise, it was deemed as a nonmerge event. A single driver could participate in several nonmerge events but could only participate in one merge event.

A total of 686 observations were obtained from U.S. Highway 101, 373 of them being nonmerge and 313 of them being merge events. As discussed in [24], there is no general rule on how many observations should be assigned to training and validation. To obtain high accuracy, a large training data size is required. Other studies have used 80% of the data set for training and 20% for validating the model [25], [26]. Base on these studies, the data set was divided into two groups: 80% of the observations were used for training and 20% were used for validation. The model was tested using the Interstate 80 data set consisting of 667 observations, 459 of them being nonmerge and 208 of them being merge events.

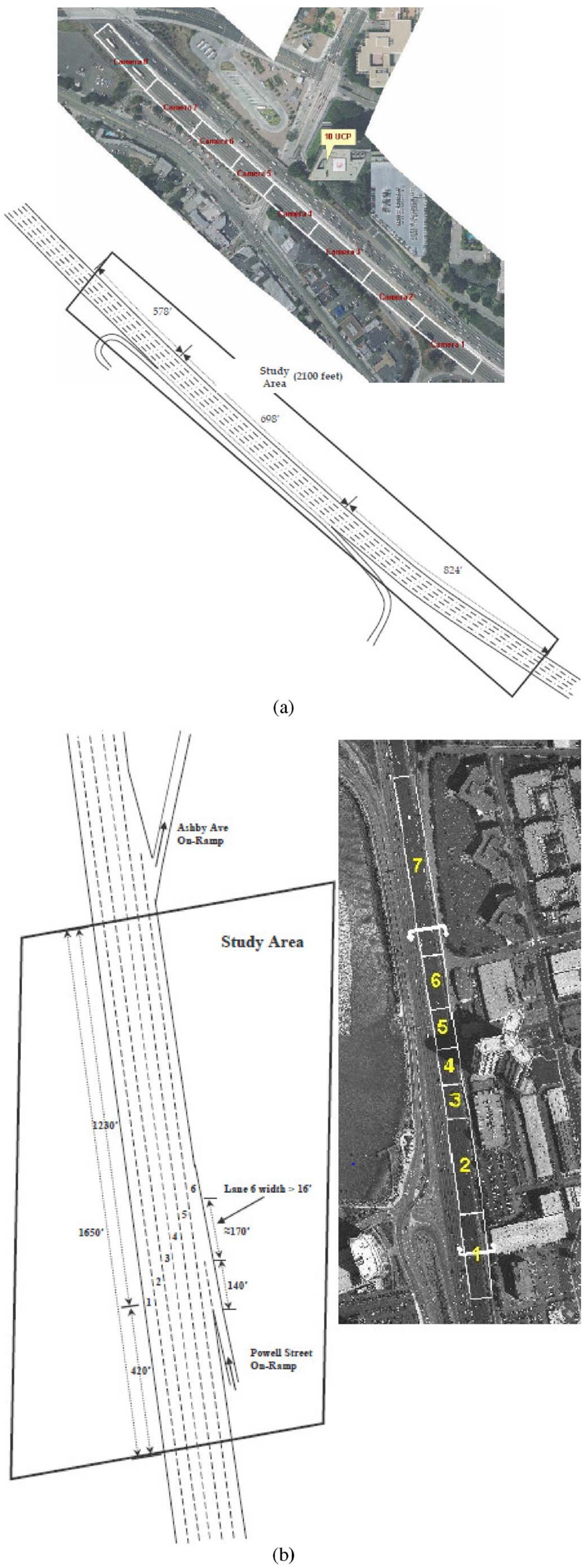


Fig. 1. (a) U.S. Highway 101 and (b) Interstate 80 study corridor from NGSIM [14].

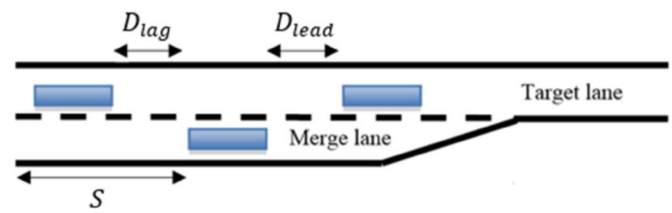


Fig. 2. Schematic illustrating input variables.

B. Input Variables

At any given instant, a driver traveling in the merge lane assesses traffic conditions in both the target lane and the merge lane to decide whether to merge or not. Several factors may affect a driver's lane changing decision. In this paper, five factors or dimensions that were found to affect a driver's merging decision in previous studies [8], [10] are considered input variables for the models. These factors are shown in Fig. 2 and defined as follows.

- ΔV_{lead}(m/s): The speed difference between the lead vehicle in the target lane and the merging vehicle, in feet per second. ΔV_{lead} can be expressed as

$$\Delta V_{\text{lead}} = V_{\text{lead}} - V_{\text{merge}}$$

  where V_{lead} is the speed of the lead vehicle, and V_{merge} is the speed of the merge vehicle.
- ΔV_{lag} (m/s) is the speed difference between the lag vehicle in the target lane and the merging vehicle (in feet per second). ΔV_{lag} can be expressed as

$$\Delta V_{\text{lag}} - V_{\text{lag}} - V_{\text{merge}}$$

  where V_{lag} is the speed of the lag vehicle.
- D_{lead} (m) is the gap distance between the lead vehicle in the target lane and the merging vehicle (in feet).
- D_{lag} (m) is the gap distance between the lag vehicle in the target lane and the merging vehicle (in feet).
- S (m) is the distance from the merging vehicle to the beginning of the merge lane.

III. METHODOLOGY

A. Bayes Classifier

1) Bayes Decision Theory: Let y_1 and y_2 denote the merge and nonmerge classes. According to the Bayesian classification rule [27]

$$P(y_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y_i)P(y_i)}{p(\boldsymbol{x})}, \qquad i = 1, 2 \tag{1}$$

where $\boldsymbol{x}$ is the input vector, $P(\cdot)$ is the probability, and $p(\cdot)$ is the probability density function. The Bayes classification rule [24] is stated as follows:

- If $P(y_1|\boldsymbol{x}) > P(y_2|\boldsymbol{x})$, then $\boldsymbol{x}$ is classified to y_1.
- If $P(y_1|\boldsymbol{x}) < P(y_2|\boldsymbol{x})$, then $\boldsymbol{x}$ is classified to y_2.
- If $P(y_1|\boldsymbol{x}) < P(y_2|\boldsymbol{x})$, then $\boldsymbol{x}$ can be assigned to either y_1 or y_2.

Using (1), the classification decision is equivalently based on the inequalities

$$p(\boldsymbol{x}|y_1)P(y_1) > (<) p(\boldsymbol{x}|y_2)P(y_2). \tag{2}$$

2) Risk of Misclassification: Risk considers both the likelihood of misclassification and the cost of the misclassification. A penalty term λ_{ki} denotes the cost of misclassifying $\boldsymbol{x}$ to a wrong class y_1 while belonging to class y_k [27]. To minimize the average risk, the classification decision inequalities (2) become

$$(\lambda_{12} - \lambda_{11})p(\boldsymbol{x}|y_1)P(y_1) > (<)(\lambda_{21} - \lambda_{22})p(\boldsymbol{x}|y_2)P(y_2). \tag{3}$$

Adopting the assumption that $\lambda_{ij} > \lambda_{ii}$ and $\lambda_{ii} = 0$, the Bayes classification rule becomes

$$\boldsymbol{x} \text{ belongs to } y_1(y_2) \text{ if } l_{12} = \frac{p(\boldsymbol{x}|y_1)}{p(\boldsymbol{x}|y_2)} > (<) \frac{p(y_2)\lambda_{21}}{p(y_1)\lambda_{12}} \tag{4}$$

where l_{12} is the likelihood ratio.

3) kNN Density Estimation: A driver's merging behavior can be predicted using the class-conditional probability density function $p(\boldsymbol{x}|y_i)$. In this paper, the k nearest neighbor (kNN) density estimation method [28] is used to estimate the class-conditional probability density functions. The kNN estimation method was chosen because, similar to kernel estimation, it is a nonparametric method; thus, there is no need to assume a distributional form unlike maximum likelihood. By using this method, the class-conditional probability density functions is estimated as

$$p(\boldsymbol{x}|y_i) = \frac{k}{N_i V_i}, \qquad i = 1, 2 \tag{5}$$

where N_i is the total number of training samples in class y_i, and V_i is the volume of the 5-D hypersphere (i.e., input data space) centered at $\boldsymbol{x}$ that contains k points from class y_i.

$P(y_i)$ is easily estimated from the observations as follows:

$$P(y_i) = \frac{N_i}{N}, \qquad i = 1, 2 \tag{6}$$

where N_i is the total number of training samples in class y_i, and N is the total number of training samples.

By substituting (5) and (6) into (4), the Bayes classification rule is equivalent to

$$\boldsymbol{x} \text{ belongs to } y_1(y_2) \text{ if } l_{12} = \frac{V_2}{V_1} > (<) \frac{\lambda_{21}}{\lambda_{12}}. \tag{7}$$

Let r_i denote the radius of the hypersphere centered at $\boldsymbol{x}$ that contains k points from class y_i. Since the hypersphere dimension in this paper is five (the total number of input variables), the likelihood ratio can be computed as

$$l_{12} = \frac{V_2}{V_1} = \left(\frac{r_2}{r_1}\right)^5 \text{ [29]}. \tag{8}$$

4) Distance Measurement: The hypersphere radius r_i can be easily obtained by searching for the kth nearest distance from all the training vectors of class y_i. The weighted distance measure was used to calculate hypersphere radius. Let $\boldsymbol{x_j}$ and $\boldsymbol{x_k}$ denote two vectors of l features. The weighted distance is

$$D(\boldsymbol{x_j}, \boldsymbol{x_k}) = \sqrt{\sum_{i=1}^{l} W_i (x_{ji} - x_{ki})^2} \tag{9}$$

where w_i is the weights associated with features. The maximum margin decision boundary established by support vector machines (SVMs) is used to determine the weights [30]. Let $\boldsymbol{q}$ be the query point whose class label is to be predicted. The SVM classifier gives decision hyperplane $g(\boldsymbol{x})$. Let $\boldsymbol{p}$ be the point with the closest Euclidean distance to $\boldsymbol{q}$ on decision hyperplane $g(\boldsymbol{x})$. $R(\boldsymbol{q})_j$ is defined as

$$R(\boldsymbol{q})_j = \left| e_j^T \nabla g(\boldsymbol{p}) \right| \tag{10}$$

where $\boldsymbol{e}_j$ denote the canonical unit vector along input feature j. The weights are given by

$$w(\boldsymbol{q})_j = \frac{\left(R(\boldsymbol{q})_j\right)^t}{\sum_{i=1}^{l} \left(R(\boldsymbol{q})_i\right)^t} \tag{11}$$

where t is a positive integer. In this paper, t values ranging from 1 to 4 were applied, and $t = 2$ produced the best model performance. In this case, the SVM decision hyperplane is in linear form, i.e., $g(\boldsymbol{x}) = \boldsymbol{b}^T \boldsymbol{x} + b_0 = 0$. Thus, $R(\boldsymbol{q})_j \equiv b_j$.

B. Decision-Tree Model

A decision tree achieves a classification decision by performing a sequence of tests on feature vectors along a path of nodes [31]. Each internal node in the tree provides the following question. "Is feature $x_i \geq a$?", where a is a threshold value. The binary answer to the question corresponds to a descendant node. At the end, each terminal node returns a class label. The size of a decision tree is the key factor in developing the decision-tree model. If the size of a tree is too small, the tree results in high misclassification rates. On the other hand, if a tree grows too large, it could overfit the training data and perform poorly on testing data. Therefore, the suggested approach is to grow a tree with a large enough size and then prune branches according to a set of pruning rules.

1) Node Splitting: To construct a decision tree, the set of questions at the tree nodes are to be determined. Each node t is associated with a subset of training set $\boldsymbol{X}_t$. The root node is assigned with the entire training set. The goal of the binary split at each node is to produce subsets that are more homogeneous or purer than the parent subset. In this model, Shannon's information theory [32] is adopted to measure the impurity of subset $\boldsymbol{X}_t$, which is also known as node impurity.

Let y_1 and y_2 denote the two classes: merge and nonmerge. Let $p(y_i|t)$ denote the probability that a sample in subset $\boldsymbol{X}_t$ belongs to class y_i, $i = 1, 2$. Node impurity is then defined as

$$I(t) = -\sum_{i=2}^{2} P(y_i|t) \log_2 P(y_i|t). \tag{12}$$

$P(y_i|t)$ can be easily estimated by N_t^i/N_t, where N_t^i is the number of vectors in subset $\boldsymbol{X}_t$ that belongs to class y_i, and N_t is the total number of vectors in subset $\boldsymbol{X}_t$. After performing a binary split at node t, a subset $\boldsymbol{X}_{tY}$ with an answer "Yes" is assigned to node t_Y, and a subset $\boldsymbol{X}_{tN}$ with an answer "No" is assigned to node t_N. The decrease in node impurity $\Delta I(t)$ is given by

$$\Delta I(t) = I(t) - \frac{N_{tY}}{N_t} I(t_Y) - \frac{N_{tN}}{N_t} I(t_N) \tag{13}$$

where N_{tY} and N_{tN} are the numbers of vectors in subsets $\boldsymbol{X}_{tY}$ and $\boldsymbol{X}_{tN}$. By exhaustively searching for all candidate questions, the one that leads to the maximum impurity decrease is chosen.

2) Stop-Splitting Criteria and Class Assignment: Threshold probability value P_0 is necessary to stop the node splitting process at any node. Splitting stops when more than $P_0 \times 100\%$ of vectors in the subset belong to any one single class, i.e., $\max_{\mathrm{i}} P(y_i|t) > P_0$. In this model, 0.9 is selected to be the threshold value as this value will also ensure that the tree grows large enough for pruning. Once a terminal node is determined, the class label is given by y_j, where

$$j = \arg\max_i P(y_i|t). \tag{14}$$

3) Tree Pruning: Minimal cost–complexity pruning [33] is employed as the pruning rule in this paper. Due to its computational efficiency, the minimal cost–complexity pruning is one of the most common methods to prune a decision tree. The sequence of subtrees generated by this pruning process is nested, meaning that the nodes that were previously cut off will not reappear in subsequent subtrees. The cost–complexity measure $R_\alpha(T)$ of decision tree T is defined as

$$R_a(T) = R(T) + \alpha|\tilde{T}| \tag{15}$$

where $R(T)$ is the substitution estimate for the overall misclassification rate of tree T, $\alpha \geq 0$ is the complexity parameter, and $|\tilde{T}|$ is the total number of terminal nodes in tree T. Each value of α is associated with subtree $T(\alpha)$ that minimizes $R_\alpha(T)$. As α increases from 0 to a sufficiently large number, the size of $T(\alpha)$ decrease from its largest size to the smallest size (only for the root node). If a subtree that minimizes $R_\alpha(T)$ for a given value of α, it will remain minimizing $R_\alpha(T)$ until α increases to a jump point. Let $\{\alpha_k\}$ be the increasing sequence of the jump points. For any $\alpha_k \leq \alpha \leq \alpha_{k+1}$, $T(\alpha) = T(\alpha_k) = T_k$. Finally, a sequence of minimal cost–complexity trees $\{T_k\}$ is generated. The right sized tree T^* can be selected by test sample estimates

$$R^{\mathrm{ts}}(T^*) = \min_k R^{\mathrm{ts}}(T_k) \tag{16}$$

where $R^{\mathrm{ts}}(\cdot)$ denotes the misclassification rate for test samples.

C. Combining Classifiers

Majority voting rule [27] is used as the combination rule to combine both Bayes classifier and decision-tree methods. The majority voting rule was chosen due to its robust performance in previous research. The majority voting rule is simple to apply and belongs to the class of hard-type combination rules. Let L denote the number of classifiers; the majority voting rule is stated as follows.

- If L is odd, the unknown pattern is classified to a class when at least $(L+1)/2$ of classifiers agree on the class label.
- If L is even, the unknown pattern is classified to a class when at least $(L/2)+1$ of classifiers agree on the class label.

In this paper, a vehicle will merge (class) only if both the Bayes classifier and the decision tree agree on the decision to merge (same class label). Thus, the combined classifier is more conservative than either of the individual classifiers that it is constructed from. This is a valuable attribute for safety applications such as the lane change assistance system where nonmerge decisions are more critical and error due to conservativeness is safer.

TABLE II
WEIGHT GIVEN BY SVMs

Variables	Weights
ΔV_{lead}	0.7243
ΔV_{lag}	0.2449
D_{lead}	0.0292
D_{lag}	0.0012
S	0.0005

IV. Results

A. Bayes Classifier

For the Bayes classifier, the weights were first estimated using SVMs. The estimated weights shown in Table II reveals that ΔV_{lead} has the largest weight, which indicates that ΔV_{lead} is the most relevant feature in classifying the merge and nonmerge events, and that a slight change in ΔV_{lead} may greatly change the distance. Speed differences ΔV_{lead} and ΔV_{lag} are more relevant than lead gap D_{lead} and lag gap D_{lag}. The distance from the beginning of the merge (auxiliary) lane S turned out to be the least relevant feature.

A Bayes classifier was developed from $k = 3$ and $(\lambda_{21}/\lambda_{12}) = 1$. The model's prediction accuracy on validation and test data for merge and nonmerge events is shown in Table III. For the test data, the accuracy of merge events was high (92.3%), but it was only 79.3% for the critical nonmerge events.

B. Decision Tree Model

A decision tree with 62 terminal nodes was constructed using training data before pruning. After applying the pruning rules, a sequence of 16 minimal cost–complexity trees was generated. The total numbers of terminal nodes $|\widetilde{T_k}|$ are shown in Table IV.

TABLE III
ACCURACY RATES OF DIFFERENT MODELS

Decision	Validation data			Test data			
		Bayes Classifier	Decision Tree		Bayes Classifier	Decision Tree	Combined Classifier
	Observations	Accuracy	Accuracy	Observations	Accuracy	Accuracy	Accuracy
Non-merge	73	82.2%	89.0%	459	79.5%	84.3%	94.3%
Merge	56	91.1%	85.7%	208	92.3%	80.8%	79.3%

TABLE IV
NUMBER OF TERMINAL NODES IN MINIMAL COST–COMPLEXITY TREES

Tree	$\lvert\widetilde{T_k}\rvert$
T_1	62
T_2	58
T_3	53
T_4	46
T_5	29
T_6	27
T_7	22
T_8	18
T_9	12
T_{10}	10
T_{11}	9
T_{12}	7
T_{13}	5
T_{14}	3
T_{15}	2
T_{16}	1

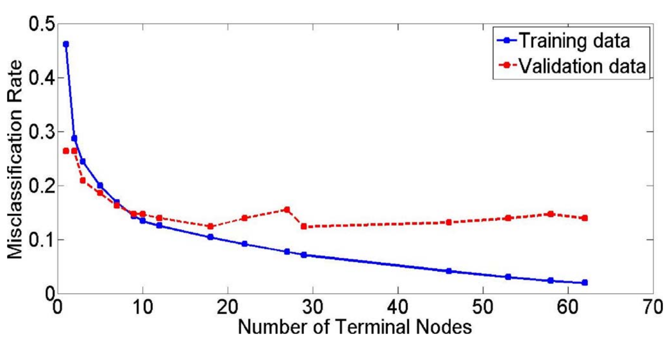


Fig. 3. Relationship between the total number of terminal nodes and misclassification rate.

The relationship between the total number of terminal nodes $|\widetilde{T_k}|$ and the estimated misclassification rate for both training and testing data is presented in Fig. 3. In Fig. 3, the estimated misclassification rate for training data $R(T_k)$ decreases sharply as the tree initially increases in size and then decreases slowly. The estimated misclassification rate for testing data $R^{\mathrm{ts}}(T_k)$ also decreases sharply initially, but after reaching its minimum value at 18 terminal nodes, the rate begins to climb as tree size grows. Thus, the tree T_8 with 18 terminal nodes was selected as the right-size decision-tree model for predicting merge and nonmerge events.

The tree structure is presented in Fig. 4, where terminal nodes are represented by shaded squares, and decision nodes are represented by circles. A number of observations, class labels, and prediction accuracy rates for terminal nodes are displayed beneath them. Node 1 was first split by using the relative speed between the lead and merging vehicles ΔV_{lead}. This result further supports the finding from the Bayes classification model that ΔV_{lead} is the most relevant driver feature in making merging decisions. The decision-making process of the decision-tree model is intuitive. For example, as shown by terminal node t_8, a driver merges if the merging vehicle is slower ($\Delta V_{\mathrm{lead}} \geq 0$ m/s) or slightly faster ($0 > \Delta V_{\mathrm{lead}} \geq -2.7$ m/s) than the lead vehicle and both the lead and lag gap is large ($D_{\mathrm{lag}} \geq 2.4$ m, $D_{\mathrm{lead}} \geq 7.6$ m). In contrast, terminal node t_7 is interpreted in natural language as follows: If the merging vehicle is much faster ($\Delta V_{\mathrm{lead}} < -2.7$ m/s) than the lead vehicle and the lead gap is small ($D_{\mathrm{lead}} < 8.9$ m), then the driver does not merge. For terminal node t_{14}, if the merging vehicle speed is much greater ($\Delta V_{\mathrm{lead}} \geq -2.7$ m/s) than the lead vehicle, the lead gap is large ($D_{\mathrm{lead}} \geq 8.9$ m), the distance from the beginning of the merge lane is far ($S \geq 138.7$ m), and even the lag gap is not too large ($D_{\mathrm{lag}} \geq 0.76$ m), the driver decides to merge because, as driver approaches the end of a merge lane, his or her merge behavior become more aggressive. These rules generated by the decision tree are representative of everyday driving experiences. The prediction results of the decision tree are presented in Table III. The accuracy of both merge and nonmerge events for test data was above 80%. However, higher accuracy for nonmerge events is desirable for a lane changing assistance system.

C. Combining Classifiers

The Bayes classifier and decision-tree models were combined using the majority voting rule. The resulting model was tested using the test data, and the results are shown in Table III. The accuracy for nonmerge improved to 94.3%, whereas the accuracy for merge events dropped slightly to 79.3%. As previously discussed, the accuracy of nonmerge events is more critical than the merge events for a safety application like the lane change assistance system. Misclassifying a merge event as a nonmerge event would result in a lost opportunity to merge but would not have a negative safety effect. The misclassification parameter $(\lambda_{21}/\lambda_{12})$ can be adjusted to give greater weight to the prediction accuracy of nonmerge events. For illustration, the model performance for $(\lambda_{21}/\lambda_{12}) = 2$ and 5 is shown in Table V. The prediction accuracy of nonmerge events increased

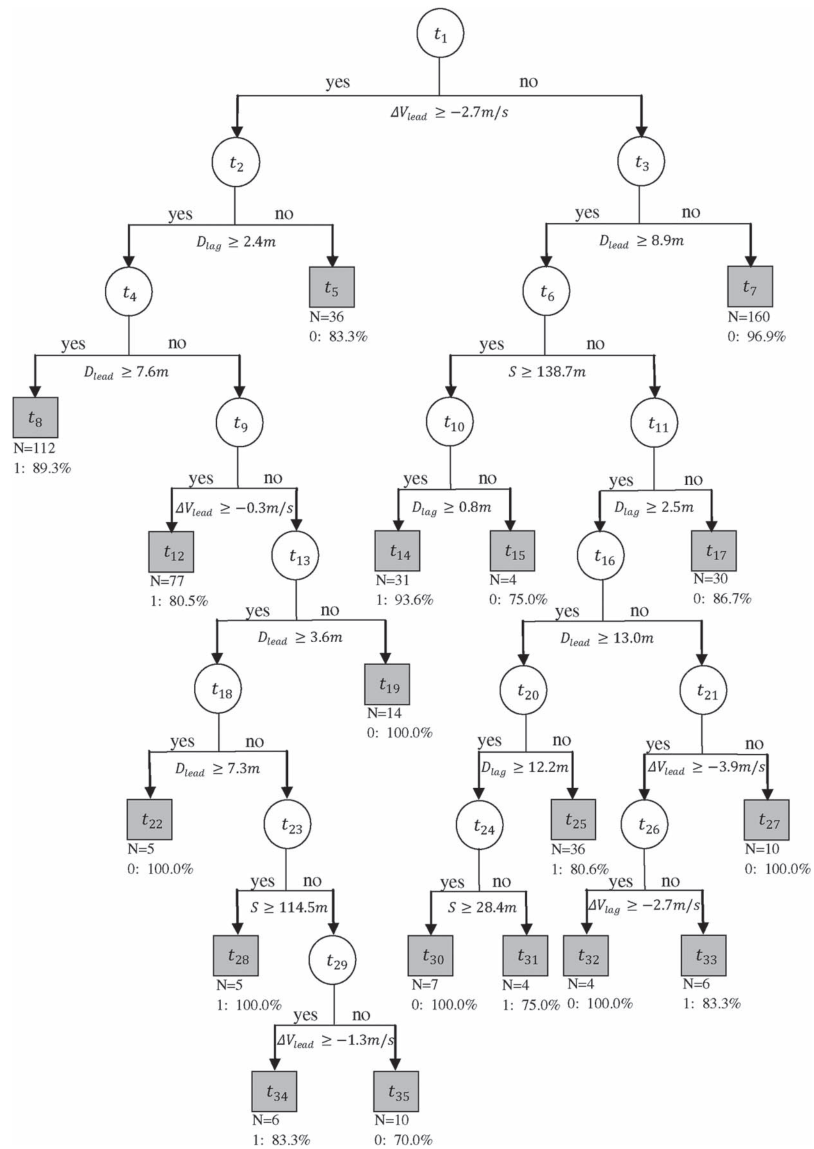


Fig. 4. Decision-tree model structure.

TABLE V
SENSITIVITY OF THE COMBINED CLASSIFIER TO MISCLASSIFICATION WEIGHTS

Decision	Observations	$\frac{\lambda_{21}}{\lambda_{12}} = 2$	$\frac{\lambda_{21}}{\lambda_{12}} = 5$
		Accuracy	Accuracy
Non-merge	459	95.4%	96.7%
Merge	208	73.6%	49.5%

TABLE VI
COEFFICIENTS OF THE BINARY LOGIT MODEL

Variable	Coefficient	p-value
ΔV_{lead}(m/s)	0.163	<.0001
ΔV_{lag}(m/s)	0.070	0.0043
D_{lead}(m)	0.061	<.0001
D_{lag} (m)	0.003	0.4721
S(m)	-0.004	<.0001
Intercept	1.967	<.0001

* Not significant at 0.05 significance level

while the accuracy of merge events decreased since the model becomes more conservative.

D. Performance of Other Models

Two models from the literature, the genetic fuzzy model and the binary logit model, were also evaluated. They were estimated using the same data set and the same set of variables. The coefficients of the binary logit model are presented in Table VI. For the genetic fuzzy system, a total of 120 rules were generated from the training data. The performance of the two models for the test data is shown in Table VII. Both models performed poorly compared with the classifier models developed in this paper. The low accuracy of nonmerge events is a concern for their use in real-time lane changing assistance systems. However, it is noted that the estimated logit model is

TABLE VII
PREDICTED RESULTS FOR GENETIC FUZZY AND BINARY LOGIT MODELS

Decision	Observed	Accuracy	
		Genetic Fuzzy	Binary Logit
Non-merge	459	73.6%	20.9%
Merge	208	71.6%	95.7%

a binary logit model and is based on existing research. In the future, advanced discrete choice models could be developed to increase the prediction accuracy rates.

V. CONCLUSION AND CONTRIBUTIONS

In this paper, models for mandatory lane changes at lane drops were developed using Bayes classifier and decision-tree methods. The publicly available NGSIM vehicle trajectory data set that consists of traffic conditions approaching congestion and congested conditions was used for model development and testing. The model employed factors, such as the vehicle speeds relative to the lead and lag vehicles in the target lane, the lead and lag gap distances, and the distance from the beginning of merge lane. Previous research focused on developing models for use in microscopic simulation, whereas this paper has focused on the design of a lane changing assistance system. One main difference between the simulation and the lane change assistance system applications is the relative importance of misclassification between merge and nonmerge decisions. In a simulation model, the effect of a nonmerge event misclassified as a merge event only affects the mobility measures. The same misclassification, however, in a lane change assistance system would likely result in a traffic crash.

The combined classifier that merges both Bayes classifier and decision-tree models generated high prediction accuracy for the critical nonmerge events. By assigning values of 1, 2, and 5 to the cost of misclassification, the classifier produced accuracy rates of 94.3%, 95.4%, and 96.7% for nonmerge events, and 79.3%, 73.6%, and 49.5% for merge events. The cost of misclassification can be treated as a surrogate to driver conservativeness. The greater the cost, the more conservative or less aggressive a driver is in working toward the gap to change lane. As the cost of misclassification increases, the accuracy for nonmerge events also increases, but the accuracy for merge events decreases. Although this paper has illustrated the performance of two other models from literature, i.e., the genetic fuzzy system and the binary logit model, these models, as proposed in the literature, were targeted at microscopic simulation. In future research, other modeling techniques can be applied for the design of lane changing assistance system and the accuracy rates compared with the combined classifier accuracy rates obtained in this paper.

REFERENCES

[1] R. Herman and G. H. Weiss, "Comments on the highway crossing problem," *Oper. Res.*, vol. 9, no. 6, pp. 828–840, Nov./Dec. 1961.

[2] D. R. Drew, L. R. LaMotte, J. H. Buhr, and J. A. Wattleworth, "Gap acceptance in the freeway merging process," Texas Transp. Inst., College Station, TX, USA, 1967.

[3] A. J. Miller, "Nine estimators of gap acceptance parameters," in *Proc. 5th Int. Symp. Theory Traffic Flow Transp.*, 1972, pp. 215–235.

[4] C. F. Daganzo, "Estimation of gap acceptance parameters within and across the population from direct roadside observation," *Transp. Res. B*, vol. 15B, no. 1, pp. 1–15, Feb. 1981.

[5] P. G. Gipps, "A model for the structure of lane changing decisions," *Transp. Res.*, vol. 20B, no. 5, pp. 403–414, Oct. 1986.

[6] H. Kita, "Effect of merging lane length on the merging behavior at expressway on-ramps," in *Proc. 12th Int. Symp. Theory Traffic Flow Transportation*, 1993, pp. 37–51.

[7] Q. Yang and H. N. Koutsopoulos, "A microscopic traffic simulator for evaluation of dynamic traffic management systems," *Transp. Res. C*, vol. 4, no. 3, pp. 113–129, Jun. 1996.

[8] K. I. Ahmed, M. E. Ben-Akiva, H. N. Koutsopoulos, and R. G. Mishalani, "Models of freeway lane changing and gap acceptance behavior," in *Proc. 13th Int. Symp. Transp. Traffic Theory*, 1996, pp. 501–515.

[9] H. Kita, "A merging-giveway interaction model of cars in a merging section: A game theoretic analysis," *Transp. Res. A*, vol. 33, no. 3/4, pp. 305–312, Apr. 1999.

[10] P. Hidas, "Modeling vehicle interactions in microscopic simulation of merging and weaving," *Transp. Res. C*, vol. 13, no. 1, pp. 37–62, Feb. 2005.

[11] T. Toledo, H. N. Koutsopoulos, and M. Ben-Akiva, "Integrated driving behavior modeling," *Transp. Res. C*, vol. 15, no. 2, pp. 96–112, Apr. 2007.

[12] Q. Meng and J. Weng, "A classification and regression tree approach for predicting drivers' merging behavior in short-term work zone merging areas," *J. Transp. Eng., ASCE*, vol. 138, no. 8, pp. 1062–1070, Aug. 2012.

[13] Y. Hou, P. Edara, and C. Sun, "A genetic fuzzy system for modeling mandatory lane changing," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, 2012, pp. 1044–1048.

[14] Next Generation Simulation Fact Sheet, Washington, DC, USA. [Online]. Available: ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

[15] C. Choudhury, V. Ramanujam, and M. Ben-Akiva, "Modeling acceleration decisions for freeway merges," *Transp. Res. Rec.*, vol. 2124, pp. 45–57, 2009.

[16] H. Yeo, A. Skabardonis, J. Halkias, J. Colyar, and V. Alexiadis, "Oversaturated freeway flow algorithm for use in next generation simulation," *Transp. Res. Rec.*, vol. 2088, pp. 68–79, 2008.

[17] "NGSIM U.S. 101 Data Analysis Summary Report," Federal Highway Admin., Washington, DC, USA, 2005.

[18] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "Estimation of vehicle trajectories from observed discrete positions and next-generation simulation program (NGSIM) data," presented at the Proc. 88th Annu. Meeting Transportation Research Board, Washington, DC, USA, 2009.

[19] A. Kesting and M. Treiber, "Calibrating car-following models using trajectory data: a methodological study," presented at the Proc. 87th Annu. Meeting Transportation Research Board, Washington, DC, USA, 2008.

[20] A. Duret, C. Buisson, and N. Chiabaut, "Estimating individual speed-spacing relationship and assessing the Newell's car-following model ability to reproduce trajectories," presented at the Proc. 87th Annu. Meeting Transportation Research Board, Washington, DC, USA, 2008.

[21] S. Ossen and S. P. Hoogendoorn, "Validity of trajectory-based calibration approach of car-following models in the presence of measurement errors," presented at the Proc. 87th Annu. Meeting Transportation Research Board, Washington, DC, USA, 2008.

[22] V. Punzo, D. J. Formisano, and V. Torrieri, "Nonstationary kalman filter for estimation of accurate and consistent car-following data," *Transp. Res. Rec.*, vol. 1934, pp. 3–12, 2005.

[23] X. Ma and I. Andreasson, "Statistical analysis of driver behavior data in different regimes of the car-following stage," *Transp. Res. Rec.*, vol. 2018, pp. 87–96, 2007.

[24] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York, NY, USA: Springer-Verlag, 2001.

[25] T. Martin, P. Harten, D. Young, E. Muratov, A. Golbraikh, H. Zhu, and A. Tropsha, "Does rational selection of training and test sets improve the outcome of QSAR modeling?" *J. Chem. Inform. Model.*, vol. 52, no. 10, pp. 2570–2578, 2012.

[26] P. Edara, D. Teodorovic, and H. Baik, "Using neural networks to model intercity mode choice," in *Intelligent Engineering Systems through Artificial Neural Networks*. New York, NY, USA: ASME, 2007, pp. 143–148.

[27] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 3rd ed. New York, NY, USA: Academic, 2006.

[28] L. J. Buturovic, "Improving k-nearest neighbor density and error estimates," *Pattern Recog.*, vol. 26, no. 4, pp. 611–616, Apr. 1993.

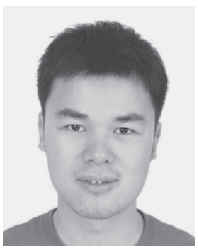
[29] C. R. F. Maunder, *Algebraic Topology*. New York, NY, USA: Dover, 1997.

[30] C. Domeniconi, D. Gunopoulos, and J. Peng, “Large margin nearest neighbor classifiers,” *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 899–909, Jul. 2005.

[31] S. Russel and P. Norvig, *Artificial Intelligence A Modern Approach*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2003, pp. 653–663.

[32] C. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL, USA: Univ. Illinois Press, 1949.

[33] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Tree*. Belmont, CA, USA: Wadsworth, 1984, pp. 66–75.



Yi Hou received the B.S. degree in civil engineering from Southwest Jiaotong University, Chengdu, China, in 2004 and the M.S. degree in civil engineering from the University of Missouri, Columbia, MO, USA, in 2011. He is currently working toward the Ph.D. degree in civil engineering in the Department of Civil and Environmental Engineering, University of Missouri.

His research interests include intelligent transportation systems, traffic modeling and simulation, traffic safety, machine learning, and data mining.

Praveen Edara received the Ph.D. degree in transportation systems from Virginia Polytechnic Institute and State University, Blacksburg, VA, USA.

He was a Research Scientist with the Virginia Transportation Research Council and a Research Contractor with the Turner Fairbank Highway Research Center of the Federal Highway Administration. Since 2007, he has been a faculty member with the Department of Civil and Environmental Engineering, University of Missouri, Columbia, MO, USA. His current research interests include traffic modeling, simulation, and intelligent transportation systems.

Carlos Sun received the B.S. degree in electrical engineering and the M.S. and Ph.D. degrees in civil engineering from the University of California, Irvine, CA, USA, and the J.D. degree from the University of Missouri, Columbia, MO, USA.

He is currently with the Department of Civil and Environmental Engineering, University of Missouri. His research interests include intelligent transporation systems, safety, work zones, and legal issues.

Dr. Sun is a Professional Engineer and an Attorney.