2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)
Windsor Oceanico Hotel, Rio de Janeiro, Brazil, November 1-4, 2016

Analysis of Microscopic Behavior Models for Probabilistic Modeling of Driver Behavior

Tim A. Wheeler[1], Philipp Robbel[2], and Mykel J. Kochenderfer[1]

Abstract— **The accurate simulation and prediction of human behavior is critical in many transportation applications, including safety and energy management systems. Construction of human driving models by hand is time-consuming and error-prone, and small modeling inaccuracies can have a significant impact on the estimated performance of a candidate system. This paper presents a comparative evaluation of several probabilistic microscopic human behavior models from the literature trained on naturalistic data for free-flow, car following, and lane change context-classes on highways. We propose several metrics to quantify model quality and use these metrics to demonstrate that a new class of Bayesian network models outperforms the state of the art.**

I. Introduction

Advanced active safety systems such as autonomous vehicles require new certification methods to determine their safety and performance over the full range of driving environments and traffic scenarios. The validation of a safety system often requires real-world driving tests, which are expensive, time consuming, and subject to safety constraints. Simulation, on the other hand, can significantly reduce cost, safety risk, and evaluation time, but requires the development of statistical models and principled methods that can accurately quantify performance.

Recent research has sought a unified statistical framework for constructing driver behavior models through machine learning on real-world data [1]–[5]. These models assume very little about the underlying form of the distribution and rely on large datasets to fit parametric distributions. Automated training of general models promises more accurate microscopic driver interactions and models that can extend across driving contexts and geographic locations.

This paper examines several recent models for human driving behavior and compares their performance in free-flow, car following, and lane change contexts on highways. Probabilistic microscopic driving models are reviewed in Section II, and several models from the literature are formulated under a common framework in Section III. A set of proposed validation metrics for probabilistic microscopic driving models are outlined in Section IV, which are compared in a series of experiments in Section VI. All associated software is publicly available to support automotive safety system development and analysis.

[1]Tim A. Wheeler and Mykel J. Kochenderfer are with Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA {wheelert, mykel}@stanford.edu

[2]Philipp Robbel is with Robert Bosch, LCC, Palo Alto, CA 94304, USA philipp.robbel@us.bosch.com

II. Probabilistic Microscopic Driving Models

Probabilistic microscopic driving models capture the interactive behavior between a vehicle and the local traffic context. They are critical components of traffic simulation models, and model variations can significantly impact the evaluation of system performance.

Let a scene s_t define the joint configuration of vehicles on a roadway at a particular time t and any past information necessary to leverage the first-order Markov assumption. A probabilistic microscopic driving model is a conditional distribution of the form $p(a_t^{(i)} \mid s_t)$, where $a^{(i)}$ is the action taken by the ith vehicle at time t. Sampling from the conditional distribution for each traffic participant and propagating each vehicle over a small time-step using a dynamics model leads to probabilistically valid successor scenes. The probabilistic microscopic driving models in this work leverage contextual information in the form of features to make their predictions. The models are reformulated as $p(a_t^{(i)} \mid \phi(i, s_t))$, where $\phi(i, s_t)$ extracts contextual features $f \in \mathbb{R}^m$ for the ith vehicle in scene s_t. Specific features and the action space are discussed in Section V.

An ideal driving model produces a realistic action distribution for any possible traffic scenario. Accomplishing this with a single model is difficult. Behaviors in different context classes are best described using different indicator features. Different models have traditionally been developed for different context classes [6]–[9]. We follow this approach and use a general framework that can fit any context class covered by the data set.

We restrict our focus to three scenarios that are particularly important for driving applications: free-flow, where the vehicle may drive unimpeded in its lane at its desired speed; car following, where the vehicle follows its lane while keeping pace with another vehicle; and lane change, where the vehicle selects and moves into another lane. Vehicle behaviors are identified at each time step so the appropriate model can be applied. Context class definitions are given in Table I and are defined to be mutually exclusive.

TABLE I: Context class definitions

Context Class	Description
free-flow	the front time gap is $\geq$ 2s and not in lane-change
car following	the front time gap is $<$ 2s and not in lane-change
lane change	continuous sets of frames up to $\pm$2s for which the absolute lateral velocity is $>$ 0.1m/s and a lane centerline switch occurs

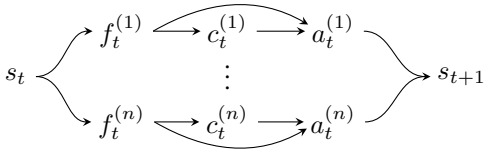

Fig. 1: Scene propagation using probabilistic microscopic driving models. For each vehicle in scene s_t, extract features $f_t^{(i)}$, identify context class $c_t^{(i)}$, and sample action $a_t^{(i)}$. Propagate all vehicles to obtain scene s_{t+1}.

Once a model is constructed, it may be sampled as many times as necessary for analysis. Contextual features f_t are extracted relative to each vehicle and are used to determine each vehicle's current context class c_t. Each context class has an associated action model. An action for each vehicle is sampled from the action model's distribution conditioned on the contextual features. These actions are used with a dynamics model to propagate each vehicle. Figure 1 outlines this scene propagation process.

III. Models

This section details the probabilistic microscopic driving models whose performance is compared in Section VI. The source code contains all details regarding model training and implementation. Note that in subsequent models $p(a_t^{(i)} \mid s_t) = p(a_t^{(i)} \mid \phi(i, s_t))$ is written $p(a \mid f)$ for brevity.

Context Classifier to Static Gaussian (SG)

An early example of a probabilistic microscopic driving model comes from Agamennoni, Nieto, and Nebot, who develop a classifier to identify scene context [1]. Each context class is tied to a Gaussian acceleration distribution:

$$p(a \mid c) = \mathcal{N}(\mu_c, \Sigma_c), \tag{1}$$

where c is a context class such as free-flow or car following.

In our work, context classification is provided, so the softmax model is equivalent to a static Gaussian (SG) distribution. Static Gaussian distributions were fit using maximum likelihood.

Linear Gaussian Regressor on Single Feature (LG)

A baseline model is included to capture the performance gained over the Static Gaussian model when using a single predictor. The Linear Gaussian (LG) model is an autoregressor on a single predictor with Gaussian error:

$$p(a \mid f) = \begin{cases} \mathcal{N}(w \cdot f + b, \sigma) & \text{for } f \text{ not missing} \\ \mathcal{N}(\mu, \sigma) & \text{otherwise} \end{cases} \tag{2}$$

where the autoregression terms are solved using ridge regression [10]. During training, all possible single-variable models are constructed and the one minimizing residual loss is selected. A separate mean is computed for cases where the predictor is missing.

Random Forest (RF)

A random forest is an ensemble of decision trees trained to minimize residual error [11]. Gindele, Brechtel, and Dillmann used random forests (RF) to model driver behavior [3]. The conditional distribution is realized by traversing the split decisions along the tree until reaching a leaf node. Each leaf contains a set of observations that are used to create a sample mean and covariance. The values from all active leaves are averaged when making a prediction.

The resulting RF model is a Gaussian distribution over acceleration and turn-rate, where the mean and covariance functions are each random forests over contextual features:

$$p(a \mid f) = \mathcal{N}(\mu \sim \textsc{Forest}(f),\ \Sigma \sim \textsc{Forest}(f)). \tag{3}$$

Dynamic Forest (DF)

The Dynamic Forest (DF) model trains an ensemble of autoregressive trees and was originally used for human motion prediction [4]. This model uses a single forest, whereas the RF model has separate forests for the action mean and covariance. The conditional distribution is obtained by traversing each tree to obtain the active leaves, and then creating a mixture model across the active leaves.

Each leaf contains a multivariate Gaussian distribution with a linearly regressed mean, $\mu = W \cdot f_l$, and a fixed covariance matrix fitted to the leaf's training samples, where f_l is the leaf's subset of observed features. The regression matrix is estimated using ridge regression [10] and the covariance matrices are approximated by sample covariances.

Mixture Regression (MR)

Mixture regression was used by Lefèvre, Sun, Bajcsy, *et al.* to develop a probability distribution over acceleration [2]. Their work used a history over headway, speed, relative speed, and acceleration to produce a maximum likelihood acceleration estimates. The model was used for model-predictive control and has been shown to work well in simulation and in real-world drive tests.

We apply their work to driving with acceleration and turnrate, where the MR model is a Gaussian mixture over the joint space of the target and predictor variables, trained using expectation maximization [12]. The conditional action distribution is the weighted combination of the conditional Gaussian components:

$$p(a \mid f) = \sum_{i=1}^{k} \beta_i(f) \cdot \mathcal{N}\left(\mu_{a|f} \mid \Sigma_{a|f}\right), \tag{4}$$

where $\beta_i(f)$ is the probability of the observed features belonging to the ith Gaussian. Greedy feature selection is used during training to select a subset of predictors up to a maximum feature count threshold while minimizing the Bayesian information criterion [13].

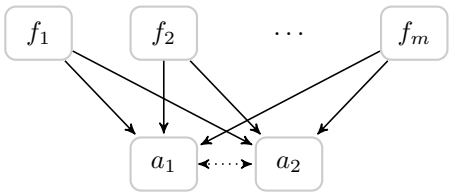

Fig. 2: Bayesian network model structure for probabilistic microscopic driving models with features $f_{1:m}$ and actions $a_{1:2}$.

Discrete Bayesian Network (DB)

Bayesian networks over discrete and discretized variables have also been used as probabilistic driver models [5]. A Bayesian network is a directed acyclic graph whose nodes are random variables and whose edges represent conditional dependence relations. Associated with each node is a conditional probability distribution (CPD) for the node's variable given the values of its parents in the network.

Discretization allows the model to match arbitrarily complex distributions with a sufficient number of intervals and allows for efficient structure learning using existing software packages. Predictors with missing values are naturally encoded as additional discrete states, and truncated values, whose infinitely-dense concentration at the truncation threshold can cause issues for Gaussian fitting, are naturally folded into the nearest interval. Naturally discrete variables were modeled using multinomial distributions. Continuous and hybrid variables were discretized into multinomial distributions over piecewise-uniform intervals.

Feature selection for the Discrete Bayesian model is conducted as part of the structure learning process [5]. General Bayesian network structure learning requires searching the full space of directed acyclic graphs [14]. Here the structures can be constrained to a two-layer form as shown in Fig. 2, for the features are observed, significantly reducing the search space. A graph-search algorithm with random initialization is used to traverse the search space over candidate predictors to find the structure with maximum likelihood [15]. We used the K2 parameter prior which assigns a baseline uniform distribution over conditional probability table statistics [14]. Only predictors that are parents of a target variable are retained.

Conditional Linear Gaussian Bayesian Network (LB)

A new model based on Bayesian networks was tested to avoid some of the limitations with using a piecewise-uniform representation. Conditional linear Gaussian distributions were used in each node:

$$p\left(x \mid \mathrm{pa}(x)\right) = \begin{cases} \mathcal{N}\left(w_1^T \, \mathrm{pa}_c(x) + b_1, \sigma_1\right) & \text{for } \mathrm{pa}_d^{(1)}(x) \\ \mathcal{N}\left(w_2^T \, \mathrm{pa}_c(x) + b_2, \sigma_2\right) & \text{for } \mathrm{pa}_d^{(2)}(x) \\ \vdots & \end{cases} \quad (5)$$

where $\mathrm{pa}_d^{(i)}(x)$ is the ith instantiation of the discrete parents of variable x and $\mathrm{pa}_c(x)$ is the vector of assignments to the continuous parents. Conditional Gaussian distributions often require fewer parameters than piecewise-uniform distributions, do not need to discretize naturally continuous variables, take advantage of naturally discrete variables, and can tightly regress to a continuous target. Variables that can be missing can be handled as hybrids, acting both as binary discrete variables indicating whether the variable was observed, and if it was, as a continuous variable in that branch's linear Gaussian. The conditional linear Gaussian is limited to producing a single-Gaussian component prediction. A graph-search algorithm with random initialization is used to traverse the search space over candidate predictors to find the model that maximizes the Bayesian information criterion [13].

IV. Validation Metrics

One must have confidence in the accuracy of a probabilistic microscopic driving model before deployment. This section proposes several validation metrics for probabilistic microscopic driving models that can be used to compare their ability to capture the behavior of human drivers.

A. Cross-Validated Likelihood

The first metric is the cross-validated likelihood of withheld data given the learned model. The cross-validated likelihood serves as the core performance metric in maximum likelihood model selection [12], assessing the model's ability to generalize across datasets. A good model will have a high cross-validated likelihood and will generally also have a high training likelihood.

B. Root-Weighted Square Error

The second validation metric, the root-weighted square error (RWSE), captures the deviation of a model's probability mass from real-world trajectories [16]. While the cross-validated likelihood measures the immediate probability of successor frames, the RWSE measures the expected square deviation of particular variables in successor traces, thereby assessing the probabilistic microscopic driving model's ability to act over longer time scales and with evolving traffic scenes. The RWSE is a natural extension to the root-mean square error, which is the mean deviation of a predicted trajectory from real-world examples. The models developed in this paper are distributions rather than maximum likelihood predictors, and it is important that the overall probability mass correctly reflect the true distribution over agents' actions. The RWSE for m trajectories for a predicted variable v at horizon H is:

$$RWSE_H = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\int_v p(v)\cdot\left(v_H^{(i)} - v\right)^2 dv}, \quad (6)$$

where $v_t^{(i)}$ is the true value in the ith trajectory at time t and $p(v)$ is the modeled probability density. Because the integral is difficult to evaluate directly, we used Monte Carlo integration [17] with $n = 50$ simulated traces per true trajectory:

$$RWSE_H = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(v_H^{(i)} - \hat{v}_H^{(i,j)}\right)^2}, \tag{7}$$

where $\hat{v}_H^{(i,j)}$ is the simulated variable under sample j for the ith trajectory at horizon H.

C. Emergent Variables and Smoothness

The final validation metric concerns the emergent behavior of the vehicles under a particular model. Likelihood alone is insufficient to capture the performance of a stochastic process. The probabilistic microscopic driving model must produce vehicle traces and overall driving behavior that is comparable to real-world driving. An emergent variable is a value extracted from a trace that is not explicitly fitted in the modeling process, such as collision frequencies and traffic flow rates. A distribution over emergent variables can be produced by simulation and compared to the distribution extracted from real world data. Matching distributions suggest model accuracy [18].

One such emergent variable is the comfort or smoothness of simulated trajectories, often described by the derivative of acceleration: $\text{jerk}(t) = j_t = \dot{a}_t$ [19]. Humans tend to drive smoothly, so accurate driving models should produce similarly smooth trajectories. We use the sum of the square jerk over the trajectory, $\sum_t j_t^2$. This is commonly used in the cost function for optimal trajectory generation in autonomous driving [19], [20]. Similarity to real-world driving can be objectively measured using the Kullback-Leibler divergence between the empirical distributions over the simulated and real-world trajectories [21]. We compute the divergence using piecewise-uniform distributions.

V. Dataset and Features

Models were constructed and evaluated using two hours of naturalistic highway driving data collected in the United States. The same dataset and the same preprocessing were used in a previous study [5]. All processing details are available in the source code, see Section VII.

A. Indicator Features

A set of candidate features was extracted for the ego-vehicle from lane-relative tracks. The set includes core features from ego dynamics, roadway features, relative features between vehicles, past states, and aggregate features over a vehicle's history. Core features describe the current state of the ego vehicle, including such features as velocity and turn-rate. Roadway features are measured with respect to the nearest center lane and require knowledge of the surrounding roadway structure. Relative features between vehicles include time and distance headways and other relative measurements required for interactive traffic prediction. Features dependent on past actions are included as well, and require that one record these values when propagating a vehicle in simulation. Aggregate features include the standard deviations, minima, and maxima for acceleration and turn-rate over a brief history.

This set of candidate features reflects those used in the driving literature for intention estimation [3], [22], [23]. For a complete list of candidate features and their definitions, see `github.com/sisl/2016_itsc_probdrive`.

B. Actions

Vehicle actions are the acceleration (a_{t+1}) and turn-rate ($\dot{\psi}_{t+1}$) over the next quarter-second time step corresponding to the traditional throttle/brake and steering wheel driving inputs. The mean acceleration over the following quarter-second horizon was extracted using the velocity difference, $a_{t+1} = (v_{t+1} - v_t)/\Delta T$.

Trajectories propagated directly with turn-rate exhibited high sensitivity when traveling at highway speeds, as small perturbations can lead to large lateral deviations. Performance improved by propagating vehicles using a turn-rate computed from a desired lane-relative heading, $\dot{\psi}_{t+1} = \psi_{\text{des}} - \psi$, where a desired lane-relative heading ψ_{des} of zero causes the vehicle to drive parallel to the nearest centerline. The final target variables are the future acceleration a_{t+1} and the constant desired lane-relative heading over the next time-step ψ_{des} extracted by solving the differential equation of motion assuming a constant turn-rate, $\psi_{\text{des}} = \left(\psi_{t+1} - \psi_t e^{-\Delta T}\right) / \left(1 - e^{-\Delta T}\right)$.

C. Truncated and Missing Features

Indicator features may be truncated or missing. A feature is truncated if it falls outside of an observable range and its value is clamped to the observation boundary. An example is distance to the lead vehicle, which is limited by the sensing range of the ego vehicle. Truncated features need to be handled carefully because of the repeated samples at the boundary. Some learning algorithms, such as decision trees, are robust to truncated features. Other learning algorithms must enforce smoothness constraints on the fitted distributions to prevent spikes at the observation boundaries or resort to expensive imputation.

Missing features also pose problems during the machine learning process. A feature is missing if it can be unobserved, either randomly or otherwise. An example is the velocity of the lead vehicle, which is unobserved when the lead vehicle is not present. Missing features are often set to average values or imputed in order to produce a fully continuous variable, and a second dummy feature is often included that indicates whether the feature was missing or not [12]. Setting the missing value to an average can lead to concentrated probability densities, data imputation is computationally expensive, and dummy features increase the feature space. The DB model includes missing and truncated values as additional discrete states and the LB model treats missing values as discrete-continuous hybrids to avoid these issues.

VI. Experiments

Model validation was conducted for lane change, free-flow, and car following. Each candidate model was trained using five-fold cross validation on the training set and then validated on the test set.

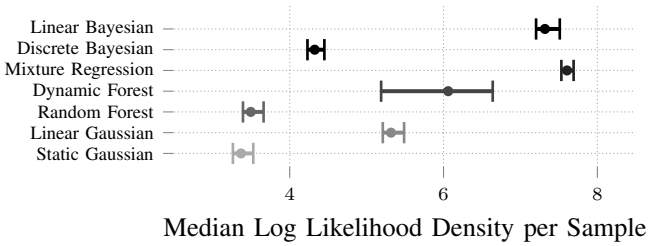

Fig. 3: The mean cross-validated log likelihood for each candidate model on the car following dataset. Error bars for training and test indicate maximum and minimum observed scores among five folds.

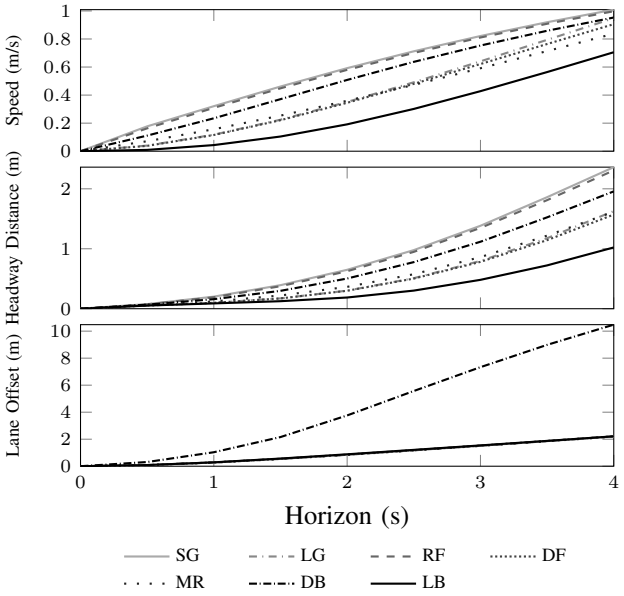

Fig. 4: The root weighted square error for each candidate model vs. prediction horizon on the car following dataset.

A. Context Classes

The results across context classes are summarized in Table II. The conditional linear Bayesian network has the highest performance in speed RWSE, headway RWSE, and smoothness. The mixture regression model has higher test likelihood, which emphasizes that likelihood only captures one-step predictions, and cannot measure a model's accumulated propagation error. The mixture regression model slightly outperforms in lane offset RWSE.

The performance of the linear Gaussian model is surprising. As a simple model it is less susceptible to being propagated to a state for which training data was not available. The random forest model hardly outperforms the static Gaussian and the discrete Bayesian network has poor emergent properties, particularly for lane following.

B. Car Following

This section analyses model performance on the car following dataset. Figure 3 shows the log likelihood score for training and test for each candidate model, along with variations obtained from cross validation. Figure 4 shows the root-weighted square error versus prediction horizon.

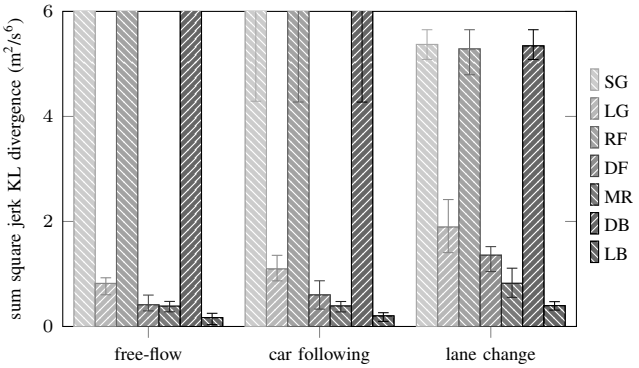

Fig. 5: The Kullback-Leibler divergence scores for sum square jerk. Error bars indicate maximum and minimum observed values over five folds. Lower values indicated a closer match to the withheld data.

Figure 5 compares the smoothness metric for all context classes.

The RWSE for speed and headway distance show significant variation in model performance, and the LB model accumulates the least error. Models perform roughly as well in speed as they do in headway distance. All models but the DB perform equally well on lane offset. The DB model cannot regress to the tight lateral actions necessary to stay oriented along the lane.

There is considerable variation between models in emergent smoothness. The SG, RF, and DB models produce non-smooth trajectories to such an extent so as to saturate the KL-divergence. The remaining models produce trajectories with similar smoothness to the real-world dataset. The LB model has the closest match across all context classes.

C. Important Features

Insight into what sort of information the models are using can be gained by looking at the features selected by each model during the training process. The top five important features, in the percentage of times chosen, were the current acceleration, the previous acceleration, the current turnrate, the previous lane-centerline offset, and the current speed. The relative velocity to the lead vehicle played an important role in car following models, whereas past desired angle, past turnrate, and mean desired angle over the last few frames played an important role in lane change models.

The single-feature linear Gaussian classifier always chose the current acceleration, as it has extremely high correlation with the next acceleration. The mixture regression and linear Bayesian network models chose similar features, though the latter also chose discrete or potentially missing features such as the number of lanes to the right. The discrete Bayesian network fluctuates the most across folds, and suggests that its optimal feature selection may be too aggressive. All models performed best when the maximum number of chosen features were limited, preventing overfitting.

VII. Conclusions

This paper conducted a qualitative and quantitative comparison of methods for learning human behavior models

TABLE II: Validation experiment results across context class. The best metric scores are bold.

	Context	Static Gaussian	Linear Gaussian	Random Forest	Dynamic Forest	Mixture Regression	Discrete Bayesian	Linear Bayesian
log-likelihood (test)	free-flow	3.40	5.58	3.51	6.51	**7.76**	4.26	7.53
	car following	3.36	5.31	3.49	6.04	**7.56**	4.30	7.30
	lane change	2.16	4.15	2.38	4.51	**8.00**	2.85	6.22
RWSE speed [m/s]	free-flow	1.17	0.89	1.16	0.90	0.95	1.25	**0.71**
	car following	1.01	0.95	1.00	0.91	0.84	0.95	**0.71**
	lane change	1.28	1.18	1.30	1.15	1.14	1.39	**0.84**
RWSE headway [m]	car following	2.36	1.63	2.31	1.57	1.62	1.96	**1.02**
RWSE lane offset [m]	free-flow	2.29	2.29	2.29	2.29	**2.28**	9.58	2.30
	car following	2.20	2.20	2.21	2.21	**2.19**	10.48	2.21
	lane change	3.41	3.42	3.49	3.49	**1.59**	18.73	3.68
KLdiv sum square jerk	free-flow	7.47	0.82	7.47	0.41	0.39	7.43	**0.17**
	car following	6.41	1.10	6.41	0.60	0.39	6.20	**0.20**
	lane change	5.37	1.89	5.29	1.36	0.82	5.34	**0.40**

from naturalistic driving data. These models were employed in collision risk estimation, policy optimization, and lane change prediction. Model parameters and structure were inferred from naturalistic driving data. The methodology presented in this paper was used to create a collection of human behavior models for different context classes: free-flow, car following, and lane change. All software is publicly available at `github.com/sisl/2016_itsc_probdrive`.

Probabilistic action models based on supervised learning to obtain the relationship between states and human actions is reaching its limit. While conceptually sound [24], small inaccuracies of the learned model compound over time, and can lead to situations not encountered during training [25]. Future work will investigate inverse reinforcement learning and generative adversarial networks, which are two of the most successful approaches to imitation learning. Contexts can be generalized to capture other aspects such as driver drowsiness or aggressiveness.

ACKNOWLEDGMENTS

This work was sponsored by Robert Bosch LLC. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by Bosch. The authors appreciate the support and assistance provided by the Automated Driving team at Bosch. The authors gratefully acknowledge the helpful comments received from anonymous reviewers.

REFERENCES

[1] G. Agamennoni, J. Nieto, and E. Nebot, "Estimation of multivehicle dynamics by considering contextual information," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 855–870, Aug. 2012.

[2] S. Lefèvre, C. Sun, R. Bajcsy, and C. Laugier, "Comparison of parametric and non-parametric approaches for vehicle speed prediction," *American Control Conference (ACC)*, pp. 3494–3499, Jun. 2014.

[3] T. Gindele, S. Brechtel, and R. Dillmann, "Learning context sensitive behavior models from observations for predicting traffic situations," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2013, pp. 1764–1771.

[4] A. M. Lehrmann, P. V. Gehler, and S. Nowozin, "Efficient nonlinear Markov models for human motion," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[5] T. Wheeler, P. Robbel, and M. J. Kochenderfer, "A probabilistic framework for microscopic traffic propagation," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Sep. 2015, pp. 262–267.

[6] R. E. Chandler, R. Herman, and E. W. Montroll, "Traffic dynamics: Studies in car following," *Operations Research*, vol. 6, no. 2, pp. 165–184, 1958.

[7] Q. Yang and H. N. Koutsopoulos, "A microscopic traffic simulator for evaluation of dynamic traffic management systems," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 3, pp. 113–129, 1996.

[8] D. McLean, T. P. Newcomb, and R. T. Spurr, "Simulation of driver behaviour during braking," in *International Mechanical Engineers Conference on Braking of Road Vehicles*, 1976.

[9] K. L. Ahmed, M. Ben-Akiva, H. Koutsopoulos, and R. Mishalani, "Models of freeway lane changing and gap acceptance behavior," *Transportation and Traffic Theory*, vol. 13, pp. 501–515, 1996.

[10] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[11] L. Breiman, "Random forests," *Journal of Machine Learning*, vol. 45, no. 1, pp. 5–53, 2001.

[12] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer, 2001, vol. 1.

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

[14] D. Koller and N. Friedman, *Probabilistic graphical models: Principles and techniques*. Cambridge, Massachusetts: MIT press, 2009.

[15] R. E. Neapolitan, *Learning Bayesian networks*. Upper Saddle River, NJ: Prentice Hall, 2004.

[16] J. A. Cox and M. J. Kochenderfer, "Probabilistic airport acceptance rate prediction," in *AIAA Modeling and Simulation Technologies Conference*, 2016.

[17] R. E. Caflisch, "Monte Carlo and quasi-Monte Carlo methods," *Acta Numerica*, vol. 7, pp. 1–49, 1998.

[18] M. J. Kochenderfer, M. W. Edwards, L. P. Espindle, J. K. Kuchar, and D. J. Griffith, "Airspace encounter models for estimating collision risk," *AIAA Journal on Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 487–499, 2010.

[19] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 163–168.

[20] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 987–993.

[21] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.

[22] D. Kasper, G. Weidl, T. Dang, G. Breuel, A. Tamke, A. Wedel, and W. Rosenstiel, "Object-oriented Bayesian networks for detection of lane change maneuvers," *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, vol. 4, no. 3, pp. 19–31, 2012.

[23] J. Schlechtriemen, A. Wedel, J. Hillenbrand, G. Breuel, and K.-D. Kuhnert, "A lane change detection approach using feature ranking with maximized predictive power," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 108–114.

[24] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2007, pp. 1449–1456.

[25] S. Ross and J. Bagnell, "Efficient reductions for imitation learning," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 661–668.