Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Personalized Maneuver Prediction at Intersections

Viktor Losing, Barbara Hammer, Heiko Wersing

2017

Preprint:

This is an accepted article published in International Conference on Intelligent Transportation Systems. The final authenticated version is available online at: https://doi.org/10.1109/ITSC.2017.8317760

Personalized Maneuver Prediction at Intersections

Viktor Losing*†, Barbara Hammer* and Heiko Wersing†
*Bielefeld University, Universitätsstr. 25, 33615 Bielefeld
†HONDA Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach am Main

***Abstract*—We investigate a new approach towards maneuver prediction that is based on personalization and incremental learning. The prediction accuracy is continuously improved by incorporating only the individual driving history. The study is based on a collection of commuting drivers who recorded their daily routes with a standard smart phone and GPS receiver. Prediction target is the expected maneuver on the next intersection with three classes: stop, turn, or go straight. We show that a personalized prediction based on at least one experience of a certain intersection already improves the prediction performance over an average prediction model trained on all test driver commute routes. This performance gain increases further with more personal training data.**

I. Introduction

Current advanced driver assistance systems are designed to deliver robust performance over an average range of driving conditions and driver profiles. Consequently, drivers are often dissatisfied because the assistance offered does not match their expectations and preferred driving style. Additionally, the frequently reoccurring driving situations experienced by one particular individual driver constitute only a small fraction of all possible situations. Both factors, individual driver characteristics and reoccurring driving situations provide a great potential for an optimization of the assistance system from an average system to a better adapted, personalized one.

Personalization denotes the modification of a system towards the characteristics of an individual user. Two different modes of personalization have been distinguished [1]: i) Active customization by the user, e.g. by making selections and setting parameters and ii) adaptive systems where the usage history is employed to estimate user preferences and situation statistics to adjust parameters and behavior. In the automotive context, adaptive personalization based on offline estimation of an appropriate parametrized driver model has been recently considered for real-time route prediction [2], adaptive cruise control [3], [4], predictive Human Machine Interaction [5] and cooperative assistance-on-demand [6].

Recently, the application of more generic non-parametric machine learning models caught more interest in the context of advanced driver assistance systems (ADAS) and autonomous driving [7]. Their application is, however, often limited to cases of available big datasets necessary for training deep architectures. In this paper, we show that generic online learning architectures capable of incremental learning from few training data can be employed for efficient personalization of maneuver prediction as a subsystem of an integrated ADAS.

Tactical maneuver prediction with a horizon of about 2-5 seconds is a highly relevant sub-function for controlling warnings and active safety systems in a car (see [8] for a review). Approaches can be based on driver sensing and intention estimation [9] or just taking GPS traces for trajectory estimation [10], [11]. In [10] a parametric behavior model for curvature-dependent velocity profiles of straight driving and right turns is estimated based on intersection crossing training data (altogether 245 approaches). Klingelschmitt et al. [11] proposed the anticipated velocity at stop line (AVS) feature, defined as $AVS = v^2 + 2da$, where v denotes the velocity, a the acceleration and d the distance to the intersection. Using a small amount of data (34 approaches on seven intersections) they showed that this information alone is a strong indicator for the drivers intention approaching an intersection.

An online learning approach of feature-based maneuver prediction was proposed in [12] and applied in a limited setting of two intersections.

In this contribution, we propose a model-free data-driven approach to maneuver prediction, capable of incremental online learning. Compared to other feature-based contributions, our simple approach scales to distinctly more intersections (285 with 5043 approaches) and does not rely on specific filtering, nor on manual labeling of real-world data. Based on a previous analysis of incremental learning architectures [13], we choose an appropriate architecture and demonstrate the performance gain that can be obtained by personalized adaptation of the prediction. We also stress the benefits of personalized context features, which can be easily obtained in the personalized context and lead to a further performance boost.

II. Framework

Our focus is the evaluation of off- and online models in the supervised classification setting [13]. The objective is to predict a target variable $y \in \{1, \ldots, c\}$ given a set of features $\mathbf{x} \in \mathbb{R}^n$. In our context, y is one out of c intersection maneuvers and $\mathbf{x}$ characterizes the ego vehicle state using attributes such as velocity, acceleration, GPS-coordinates etc. The difference between the off- and online learning schemes is described below.

A. Offline

In the offline learning setting an algorithm generates a model function $h : \mathbb{R}^n \mapsto \{1, \ldots, c\}$ based on a training set $D_{\text{train}} = \{(\mathbf{x}_i, y_i) \,|\, i \in \{1, \ldots, j\}\}$. In the subsequent test phase, the model is applied on another set $D_{\text{test}} = \{(\mathbf{x}_i, y_i) \,|\, i \in \{j+1, \ldots, k\}\}$, whose labels are kept hidden. The model predicts a label $\hat{y}_i = h(\mathbf{x}_i)$ for every point $x_i \in D_{\text{test}}$ and the 0-1 loss $\mathcal{L}(\hat{y}_i, y_i) = \mathbb{1}(\hat{y}_i \neq y_i)$ is calculated. The test error

$$E(D_{\text{test}}) = \frac{1}{k} \sum_{i=j+1}^{k} \mathcal{L}(h(\mathbf{x}_i), y_i) \tag{1}$$

is the commonly used performance metric.

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes,

creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. The final publication is available via https://doi.org/10.1109/ITSC.2017.8317760.

B. Online

The online learning setting is more challenging, since the data is accessed one by one in a predefined order and the algorithm has to provide a model after each datapoint. Therefore, online algorithms initially tend to deliver a lower performance compared to their offline counterparts. However, they provide the benefits of a lower time and space complexity during training, are able to process datasets of arbitrary sizes and allow particular tuning to a special problem domain.
Formally, a potentially infinite sequence $S_t = (s_1, s_2, \ldots, s_t)$ of tuples $s_i = (\mathbf{x}_i, y_i)$ arrives one after another. In contrast to the offline setting, a model function is generated after each tuple. As t represents the current time stamp, the prediction $\hat{y}_t = h_{t-1}(\mathbf{x}_t)$ is done according to the previously learned model h_{t-1}. After the true label y_t is revealed, the applied learning algorithm generates a new model $h_t = \text{train}(h_{t-1}, s_t)$ on the basis of the current tuple s_t and the previous model h_{t-1}. Usually, the interleaved test train error is used for performance evaluation and is defined as:

$$\hat{E}(S_t) = \frac{1}{t} \sum_{i=1}^{t} \mathcal{L}(h_{i-1}(x_i), y_i). \tag{2}$$

Please note, that we calculate both errors (off- and online) using the same data for testing, but the online algorithms continuously adapt their model h_i.
The principle difference between off- and online approaches is that the offline methods have generally a much larger set of training data available, whereas the online algorithms have the capability to adapt to the actual test data. The natural consequence is that online methods using few data are only applicable if the variation in the test condition is not too high. However, online algorithms are even able to adapt to non-stationary environments and efficient methods have been recently published [14], [15].

III. Dataset

Our dataset was extracted from recordings of the daily work-home commute of eleven different drivers. Personalization is particularly useful in this setting, since each driver takes daily an individual and usually similar route. This repeated pattern can be exploited by state-of-the-art machine learning methods and can provide a robust prediction function after only a few commutes. We used the mobile App TrackAddict[1] in combination with an IPhone 5 to record the data. The GPS trace itself was delivered by the Dual XGPS 160, which in contrast to the IPhone delivers the data at a rate of 10 Hz. TrackAddict provides additional raw data to each GPS coordinate such as time , velocity, gyroscope values as well as the corresponding video. However, we utilized only the time (s), velocity ($\mathrm{km\,h^{-1}}$) and the GPS coordinates (°).
The variance of the daily route varies from driver to driver as it is illustrated by Figure 1. Some drivers are taking only small detours, whereas others choose among several different alternatives.

A. Preprocessing

We removed too short streams (< 5 min) as well as those with a too low GPS rate resulting from failures of the GPS receiver. Only intersection approaches were extracted from the raw data. We used a maximum prediction horizon of four seconds leading to approximately 40 data points per approach. This time is sufficient to provide situation-dependent assistance at intersections [8].

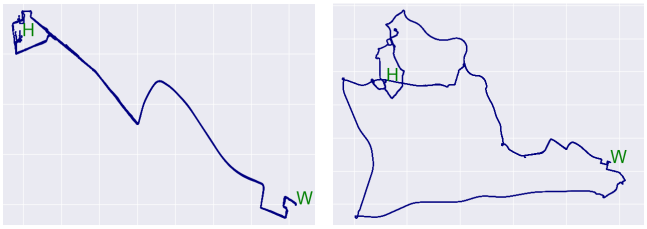


Fig. 1: All traces of two different drivers in the direction from home to work. The location of the drivers home as well as the working place are marked by "H" and "W" respectively. In contrast to the rather fix route on the left, the commute on the right incorporates multiple alternative routes.

1) Relevant intersections and potential stop points: A stream of driver d corresponding to a commute ride is given by a sequence $S^d = [p_1, \ldots, p_n]$ of n recorded tuples $p = (time, vel, long, lat)$. A potential stop point within such a given stream is determined by a measurement with velocity smaller than some predefined value α, whereby we always take the first such measurement in a row, and we make sure a certain velocity β is reached before the next stop can be encountered. That means, a potential stop is found at time step t_i if $vel_{t_i} \leq \alpha$, $vel_{t_i-1} > \alpha$, and for all i there exists some $t_i < t' < t_{i+1}$ such that $vel_{t'} \geq \beta$. This way, every commute ride S^d_r of a driver d yields a set of potential stop points $O^d_r := \{o_1, \ldots, o_{k_r}\}$. For every driver d, we collect all commute sequences $S^d_{\text{total}} := \bigcup_r S^d_r$ and the corresponding set of potential stop points $O^d_{\text{total}} := \{o_i \mid \exists r\, o_i \in O^d_r\}$.

Since we are interested in the prediction of driver behavior at *intersections*, we determine all intersections of the observed drives within the corresponding map area[2]. Thereby, intersections are simply represented by their GPS-coordinates $I = (long, lat)$. To reduce the amount of intersections and also to generate a challenging as well as balanced dataset, we consider only intersections at which the driver has stopped at least once. Further, we align a potential stop point o_j to an intersection I_j, provided the point o_j is the closest stop point to I_j and the distance is smaller than a predefined value γ, measured in the Euclidean distance of the GPS signals.[3] All potential stop points which are not aligned to an intersection are irrelevant.

B. Automatic labeling

For every commute, we identify the relevant intersections on its way. We identify the parts of the commute ride, which are within a distance of at most 20 m to the intersection. For these events, we distinguish the following classes:

- *straight* (cross without stopping)
- *stop*

[1] http://racerender.com/TrackAddict/Features.html

[2] We extracted the intersections from OpenStreetMap [16]

[3] We chose the parameter values $\alpha = 5\,\mathrm{km\,h^{-1}}$, $\beta = 20\,\mathrm{km\,h^{-1}}$ and $\gamma = 20$ m to generate the dataset.

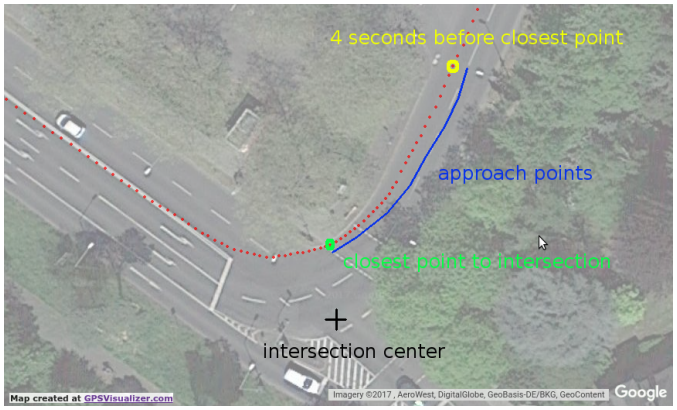


Fig. 2: A typical right turn approach. The approach data contains all points four seconds backwards from the closest point to the intersection.

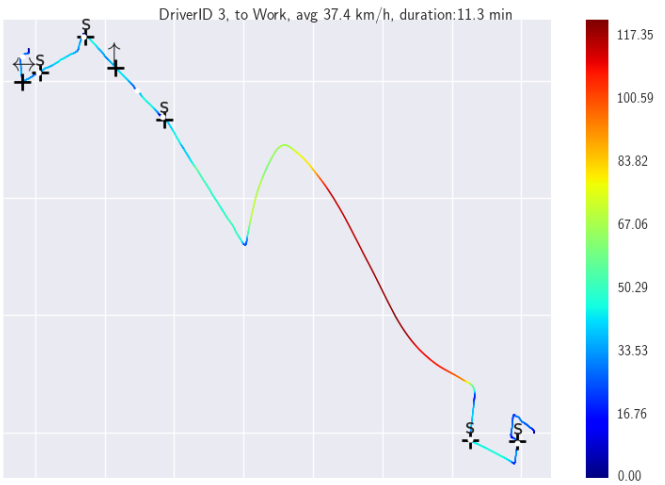


Fig. 3: A GPS trace from home to work. The crosses mark the relevant intersection for this trace, whereas white dots denote stopping positions. Please note that some intersections are excluded due to the fact that the driver did not stop at them on all streams in one direction. The labels of the approaches are given by ↑ = straight, ↔ = turn, s = stop. The color of the trace encodes the velocity in $\mathrm{km\,h^{-1}}$.

- *turn* (turn left or right without stopping)

All datapoints of one approach are labeled as the same class. An approach is labeled as *stop* provided it is contained in the list of stop points as described above. In this case, it is irrelevant whether the car goes straight or takes a turn after the initial stop. A *stop* approach contains the data sequence four seconds backwards from the stop point. Please note, that our learning task is not the same as learning the daily route of the driver. The car can stop at any day at any intersection caused by e.g. traffic lights, preceding cars or pedestrians, which makes the subsequently taken direction extraneous.
For the remaining events, we automatically determine the type based on the following geometric considerations. Two lines are fitted to the GPS trace, before and after the intersection. An approach is labeled as *turn* when the angle in between the two lined is larger than 30°. Otherwise, it is labeled as *straight*. The *turn* and *straight* data contain the data sequence four seconds backwards from the closest point to the intersection. Figure 2 illustrates an exemplary intersection *turn* approach.
A typical GPS stream with labeled intersection approaches is shown in Figure 3. Even though we optimized the parameters of the automatic labeling, our data driven approach comes at the cost of a minor amount of label noise. This mainly concerns approaches of the *straight* and *turn* class, which are sometimes hard to discriminate from each other, due to the arbitrary different layout of intersections. Furthermore, the imprecision of the GPS signal adds to the complexity.

C. Dataset characteristics

The main characteristics of the resulting datasets are given in Table I. The number of approaches of each driver varies naturally within the dataset, depending on the amount of recorded streams, the length of the commute as well as whether the corresponding route is located in a rural or urban area. Table II illustrates the largely similar class distribution of the approaches with *turn* having the smallest share.

IV. Experiments

We evaluate four different models on the dataset to investigate whether online personalized models perform better than offline average models. Table III depicts characteristics of the

TABLE I: The dataset broken down to the individual drivers, contributing to the dataset in different proportions. Driver ten has with 22.4% the highest share, whereas driver three with 2.7% the smallest one. Each approach is usually represented by 40 single datapoints.

DriverID	#Streams	#Intersections	#Approaches	#Datapoints
1	31	35	809	32664
2	23	17	240	9722
3	14	27	260	5427
4	37	26	561	22702
5	12	32	211	8510
6	30	17	288	11674
7	35	16	414	16752
8	28	30	642	25814
9	24	13	379	15303
10	57	61	1103	44648
11	21	11	136	5482
$\sum$	312	285	5043	198698

different models and the used features. At each point in time of the approach towards the intersection the feature vector $\mathbf{x}_t$, possibly containing the velocity, acceleration, AVS [11] and intersection GPS coordinates, is used to compute a prediction $h_t(\mathbf{x}_t)$. We perform a leave-one-driver-out evaluation with the offline models. Precisely, they are tested with the data of one specific driver, whereas those of the remaining drivers is used for training. This is done repeatedly such that each driver is used for testing once. We mainly utilize on- and offline variants of the popular Random Forest (RF) [17] to enable a fair comparison. The RF is a well known state-of-the-art learning algorithm, delivering highly competitive results [13], [18] and is easy to apply out of the box. We additionally use

TABLE II: The class distribution of the dataset.

Class	#Approaches	#Datapoints	Proportion (%)
Straight	1819	72194	36.33
Stop	2062	80985	40.76
Turn	1162	45519	22.91

TABLE III: The evaluated models with corresponding feature sets. Vel = Velocity, Acc = Acceleration, Dist = Distance to intersection, I-Lat= Intersection-Latitude, I-Lon = Intersection-Longitude.

Abbreviation	Model	Learning	Features
LogReg_B	Logistic Regression	offline	AVS,Dist
RF_E	Random Forest	offline	AVS,Vel,Acc,Dist
ORF_E	Online Random Forest	online	AVS,Vel,Acc,Dist
$\text{ORF}_\text{E+C}$	Online Random Forest	online	AVS,Vel,Acc,Dist,I-Lat,I-Lon

as baseline a Logistic Regression [19] model with the AVS feature and the distance to the intersection as it was proposed by Klingelschmitt et al. [11]. The other offline model is a RF with an extended feature set including next to the AVS feature, also the velocity, acceleration as well as the distance to the intersection.

We conducted also experiments coupling the Logistic Regression model with the extended feature set, as well as the RF with the AVS feature only. However, the Logistic Regression model did not profit from the additional features and the RF performed on average about 5% worse than LogReg_B. Consequently, we omit these results in our analysis.

Both incremental models are instances of the Online Random Forest (ORF) [20]. One is using the same feature set as the RF, whereas the other additionally incorporates personalized context information in terms of the GPS coordinates of the corresponding intersection. Personalized context features have the advantage of being rather easily to obtain for a specific user and can substantially boost the individual prediction after only a few examples. In the context of the average user, however, they often have no specific meaning at all and can even deteriorate the performance or require a huge amount of training examples to be beneficial. In our case, for example, the intersection GPS coordinates would only boost the performance of the average model if it incorporates examples for each intersection approached from all directions. This requires a tremendous amount of data, by far more than contained in our dataset, even though the routes of the drivers are locally related.

Incremental models are usually evaluated in the online learning setting (see section II-B). The model predicts first the label of one sample and uses it afterward for model adaption. This is done for all samples in the dataset. However, the order of our dataset is predefined by the recording time, and therefore, there is a high degree of label autocorrelation, since each approach consists of around 40 samples. Using the ordinary online scheme in this case, is misleading because a naive classifier, simply predicting the previously seen label, achieves a very low error rate without learning anything. Therefore, we perform the online scheme approach-wise. Precisely, the model has to predict all samples of one approach, before the corresponding labels get revealed. We trained from the scratch one online model for every driver in single pass. Meaning, the online models are utilized without any form of pre-training and only access the data of one specific driver.

A. Results

Figure 4a shows the resulting error rates for different prediction horizons. Clearly, the prediction gets easier the closer the driver is to the intersection. The method RF_E performs on average similar to LogReg_B even though it uses additional features. However, the AVS feature is basically a compression of the velocity, acceleration and intersection distance for the purpose of intersection intent inference. Our experiments confirm its usefulness in this context.

Both online models substantially outperform their offline counterparts, underlining the benefits of a personalized prediction within this setting. This is particularly remarkable considering the severely smaller amount of available training data as well as the usual performance advantage of offline models. The error rate may even decrease further with larger training sets. Moreover, it is shown that the addition of the intersection GPS coordinates boosts the performance throughout the whole prediction horizon.

The advantage of the online models is even more pronounced if their performance is measured after a certain amount of training data has been seen. Figure 4b contrasts the mean error rate to those achieved on the second half of each drivers data. The error rate is distinctly lower for the second half of the data, due to the naturally higher amount of mistakes done at the beginning of learning.

The learning curve of the online models is shown in Figure 4c. Precisely, it depicts the error rate depending on the number of trained approaches for a specific intersection. The personalized predictors require only a small amount of training data to compete with the average ones. In fact, they are already more accurate at the second time they approach the same intersection. Hence, only one commute tour is sufficient to gain an advantage with the personalized models in our scenario. ORF_E seems to be converged after approximately four approaches, whereas $\text{ORF}_\text{E+C}$ keeps improving with additional data.

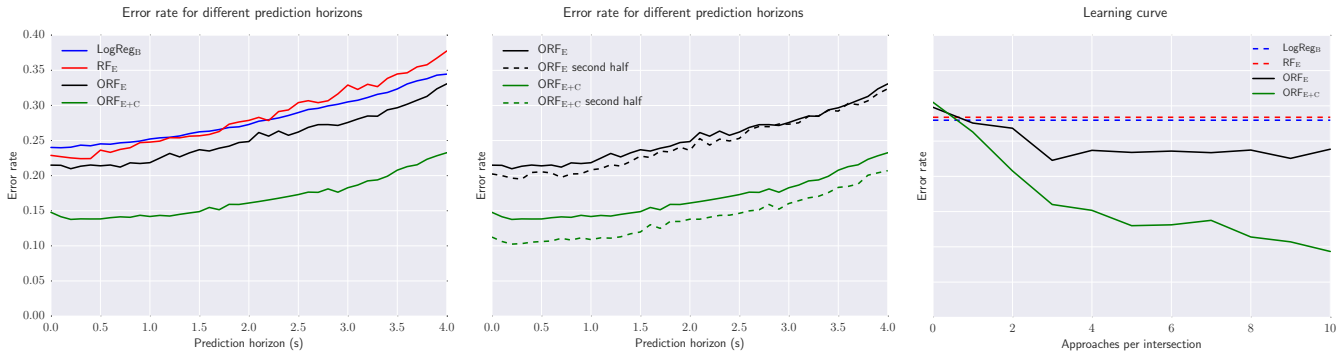
Figure 5 depicts the receiver operating characteristic (ROC) curves for all classes and Figure 6 shows the confusion matrices of the models. Most confusions occur between the *straight* and *turn* class, whereas the *stop* approaches are classified with high accuracy. LogReg_B has the lowest error rate for the *straight* and *stop* class. However, particularly the ROC curve for the *straight* class illustrates that it is not necessarily the best model. Rather, these low error rates come at the cost of a poor accuracy for the minority class *turn*, constantly predicted with a low confidence, and therefore, rarely used. Nonetheless, the ROC curve for the *turn* class shows that LogReg_B is quite able to reasonably predict this class with an appropriate tuning of the confidence thresholds.

The personalized models perform particularly well for the *straight* and *turn* class because of several reasons. Clearly, they profit from the fact that the class distribution of a specific driver is often different than those of the average driver. Furthermore, a driver may approach intersections in a specific way, facilitating the personalized classification.

$\text{ORF}_\text{E+C}$ is by far the best model in general. It is able to implicitly generate an intersection specific prediction model, due to its access to the intersection GPS coordinates.

B. Generalization of personalized models

One interesting question is whether the personalized models are able to learn a driver specific way of approaching intersections in general. Therefore, we analyze the personalized model in a leave-one-intersection-out as well as in a leave-one-approach-out experiment and compare its performance with those of an average model. For a fair comparison we use for both models the extended feature set (AVS,Vel,Acc,Dist). The



(a) (b) (c)

Fig. 4: (a) The mean error rate of the evaluated models depending on the prediction horizon. (b) The mean error rate as well as the one considering only the second half of each drivers data, achieved by both online models. The error rate is distinctly lower for the second half, because more mistakes are done at the beginning of learning. (c) Learning curve of the online personalized models. The average error rate is given depending on the number of experienced approaches for the specific intersection. The personalized predictors are on average more accurate as soon as they encounter an intersection they have already seen. The average performance of the offline models is given via dotted lines.

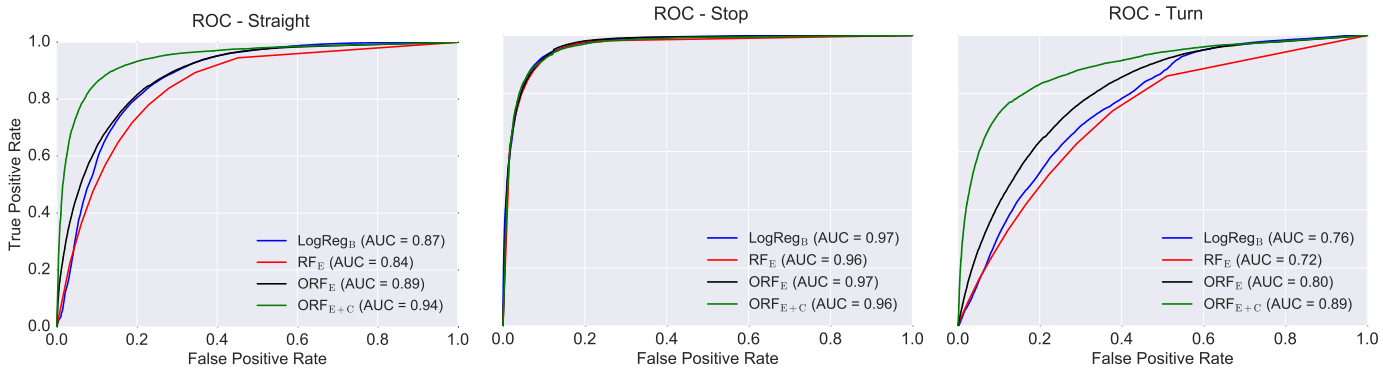


Fig. 5: ROC curves of the models for all classes. The online models are superior for the *straight* and *turn* class, whereas all models have a similar ROC for the *stop* class.



Fig. 6: Confusion matrices of the models. Most confusions occur between the *straight* and *turn* class. However, the context features distinctly enhance the discrimination. LogReg$_B$ has constantly a low confidence for the *turn* class as it is rarely used for prediction.

average model achieves in both experiments an error rate of 0.29, whereas the personalized model performs slightly worse in the leave-one-intersection-out experiment (error: 0.30), but clearly better in the leave-one-approach-out setting (error: 0.24). Therefore, it is not confirmed that the personalized model learns a generic driver specific way of approaching intersections. Rather, we can conclude that the advantage of the personalized models is mainly based on already seen intersections approached by the specific driver.

V. Conclusion

In this paper, we showed the benefits of a personalized incremental learning approach in the setting of intersection maneuver prediction. A dataset containing the GPS traces of the daily work-home commute, driven by eleven different drivers, was recorded, covering altogether 285 intersections with corresponding 5043 approaches. Our dataset is severely larger and more diverse than those in comparable contributions. We applied a simple, model-free as well as data-driven approach, providing an accurate maneuver prediction. In contrast to state-of-the art techniques, it does not rely on explicit and sophisticated ego vehicle modeling, nor on manually labeled data. Precisely, we compare the prediction error rate of offline models trained in a leave-one-driver-out scheme with incremental personalized models, trained in online fashion exclusively with the data of one specific driver. The efficient personalized models turned out to be more accurate after only a small amount of training data than their offline counterparts. Precisely, they are already more accurate at the second time they approach the same intersection. Hence, only one commute tour is sufficient to gain an advantage with the personalized models in our scenario. Furthermore, we highlighted that personalized context data such as the intersection GPS coordinates, often only viable and useful for the specific user, additionally increase the performance.
Our simple, model-free as well as data-driven approach leads to an accurate maneuver prediction and, in comparison to state-of-the art techniques, does neither rely on explicit and sophisticated ego vehicle modeling nor on manually labeled data. However, in the case of available precise lane-level maps, an explicit modeling of intersections may have advantages with respect to a better generalization to areas driven for the first time [10]. In subsequent experiments, we showed that the advantage of the personalized models is not due to a generic driver specific way of approaching intersections, but rather based on already seen intersections approached by the specific driver.

Acknowledgment

We gratefully thank Benedict Flade and Julian Eggert from the Honda Research Institute in Offenbach for providing the OpenStreetMap intersection data. We also thank Martin Heckmann for discussions on the role of a generic versus an intersection specific personalization.

References

[1] G. Fischer, "User modeling in human-computer interaction," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1-2, pp. 65–86, 2001.

[2] F. D. N. Neto, C. de Souza Baptista, and C. E. Campelo, "A user-personalized model for real time destination and route prediction," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 401–407.

[3] G. N. Bifulco, L. Pariota, F. Simonelli, and R. Di Pace, "Development and testing of a fully adaptive cruise control system," *Transportation Research Part C: Emerging Technologies*, vol. 29, pp. 156–170, 2013.

[4] V. Butakov and P. Ioannou, "Personalized driver/vehicle lane change models for ADAS," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4422–4431, October 2015.

[5] B. Harsham, S. Watanabe, A. Esenther, J. Hershey, J. Le Roux, Y. Luan, D. Nikovski, and V. Potluru, "Driver prediction to improve interaction with in-vehicle hmi," in *Proc. Workshop on Digital Signal Processing for In-Vehicle Systems (DSP)*, 2015.

[6] D. Orth, D. Kolossa, M. Sarria Paja, K. Schaller, A. Pech, and M. Heckmann, "A maximum likelihood method for driver-specific critical-gap estimation," in *accepted at: 2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[7] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," *arXiv preprint arXiv:1701.06699*, 2017.

[8] A. Doshi and M. M. Trivedi, "Tactical driver behavior prediction and intent inference: A review," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1892–1897.

[9] C. Rodemerk, H. Winner, and R. Kastner, "Predicting the driver's turn intentions at urban intersections using context-based indicators," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 964–969.

[10] M. Liebner, F. Klanner, M. Baumann, C. Ruhhammer, and C. Stiller, "Velocity-based driver intent inference at urban intersections in the presence of preceding vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 2, pp. 10–21, 2013.

[11] S. Klingelschmitt, M. Platho, H.-M. Groß, V. Willert, and J. Eggert, "Combining behavior and situation information for reliably estimating multiple intentions," in *IEEE Intelligent Vehicles Symposium (IV)*, July 2014. [Online]. Available: http://tubiblio.ulb.tu-darmstadt.de/65706/

[12] J. Wiest, M. Karg, F. Kunz, S. Reuter, U. Kreßel, and K. Dietmayer, "A probabilistic maneuver prediction framework for self-learning vehicles with application to intersections," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 349–355.

[13] V. Losing, B. Hammer, and H. Wersing, "Choosing the Best Algorithm for an Incremental Learning Task," in *European Symposium on Artificial Neural Networks (ESANN)*, 2016.

[14] V. Losing, B. Hammer, and H. Wersing, "KNN Classifier with Self Adjusting Memory for Heterogeneous Concept Drift," in *16th International Conference on Data Mining (ICDM)*. IEEE, 2016.

[15] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," *Machine Learning and Knowledge Discovery in Databases*, pp. 135–150, 2010.

[16] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.

[17] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems," *J. Mach. Learn. Res*, vol. 15, no. 1, pp. 3133–3181, 2014.

[19] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[20] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1393–1400.