

Exploiting Maneuver Dependency for Personalization of a Driver Model *

Hien Dang and Johannes Fürnkranz

Knowledge Engineering Group, TU Darmstadt, Darmstadt, Germany
{hiendang, fuernkranz}@ke.tu-darmstadt.de

Abstract. Modeling the driver’s behavior is an importance task in the automotive field. Most data-driven approaches build a general model that works well for the majority of drivers in the training set. Personalization, on the other hand, addresses the problem of adapting the model on the driver. A personalized model has to be able to adjust to changing user preferences over time. In this work, we propose an approach to extract and incorporate information from previous maneuver executions to improve the performance of prediction tasks in impending maneuvers. We apply our proposed adaptation method to predict the gap taken at a left-turn scenario, and show that it can boost the prediction of a neural network.

Keywords: Personalization · LSTM · Driver Assistance System.

1 Introduction

It is often the case that users have to adapt themselves to a new system or function to be able to use it. When the users’ expectation and preference of a system are not met, their trust in the system will decrease and eventually they may ignore or turn it off. For a standard not-adapting system, there could be a gap between the user’s expectation and the outcome of the system. Personalization systems aim at closing this gap.

The problem of personalization is already addressed in different field studies and applications. In recommender systems, for example, the data are mostly sequences of actions [1] (i.e. “visiting a website” or “buying an item”...) The problem of personalization in this case is usually formulated as a prediction task of the next action B given the past action A_i or a sequence of the past actions $A_{0..i}$, i.e., $\pi : A_0 \dots A_i \rightarrow B$. In the automotive field, the sequence of maneuvers may by itself reveal very little about the driver. For example, if a driver wants to get to his travel destination, certain maneuvers have to be performed in any case (i.e. entering a roundabout, turning at an intersection...). However, different drivers vary in the way they perform these maneuvers. As a consequence, the problem of personalization must in this case be based on observing how a driver performs the driving maneuvers to infer the driver’s preference or to predict how s/he would perform the next maneuver or some specific maneuver of interest. Thus we need to learn a function which maps the driving style $f(\cdot)$ of previous maneuvers to the driving style of an upcoming maneuver:

$$\pi : f(A_0) \dots f(A_i) \rightarrow f(B) \quad (1)$$

* Supported by Continental AG. An extended version of this paper can be found at [2]

2 Approach on Personalization

Driving style is an abstract concept that is not directly observable from the data. What we can observe is the reflection of driving style on some measurable signals like speed, acceleration, etc. A survey conducted by Martinez et al. [5] shows that the choice of input signals to use for detecting driving style can be divided into three main groups: vehicle dynamics, energy consumption and personality traits. In this work we focus on the groups since it is more general and always available in all maneuver execution.

Problem Formulation. Given a situation S_i and a driver D_j , the driving maneuver at this situation can be formulated as a function of S_i and D_j

$$M_i = f(S_i, D_j + \epsilon_i) \quad (2)$$

where S_i represents all environmental factors, D_j constitutes the driver's dispositional factors. And ϵ_i is introduced to capture the fluctuations in the driver's behavior.

To personalize the predictions of y_0 , we additionally have to consider the influences of the driver $D_j + \epsilon_0$ in this situation, since y_0 depends on both factors, i.e., $y_0 \leftarrow S_0, D_j + \epsilon_0$. Now we have the same problem as in modeling maneuver execution, namely that the driver's dispositional factors are not directly observable. This makes it impossible to learn the individual impacts of driver D_j on y_0 .

As mentioned above, in real-world driving data, the drivers' factors are encoded in each maneuver execution M_i . We therefore propose that instead of learning y_0 as a function of S_0 and D_j , we can learn y_0 as a function of S_0 and M_i , where M_i is a recently performed maneuver. This temporal restriction of M_i allows us to approximate the current impact of the driver $D_j + \epsilon_0$ with $D_j + \epsilon_i$, which was captured in the M_i .

$$y_0 \leftarrow S_0, M_i = S_0, f(D_j + \epsilon_i, S_i) \approx S_0, f(D_j + \epsilon_0, S_i) \quad (3)$$

y_0 can then be written as a function G of S_0 , $D_j + \epsilon_0$ and S_i :

$$y_0 = g(S_0, f(D_j + \epsilon_0, S_i)) = G(S_0, D_j + \epsilon_0, S_i) \quad (4)$$

With the assumption that y_0 does not depend on other situations than the current S_0 , there should be no information of y_0 contained in S_i . G will be forced to learn to extract useful information about the driver from past maneuvers and then how they will affect the decision of the driver in the current situation.

Extracting Driver's Information. A maneuver execution M_i with length n is characterized by a multivariate time series ($x_{i,t}$ is a vector of sensor values at time t):

$$M_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n}), \quad n > 0 \quad (5)$$

In this work, we used two approaches to extract a driver's information from the last executed maneuvers. The first approach uses the statistical information from each sensor (minimum, mean, maximum and standard deviation). The second approach makes use of recurrent network layers and takes the whole maneuver execution as an input. While the idea of the first approach is widely used in the literature for extracting driver's information, the second one does not make any assumption about the statistical values but learns to extract useful information directly from raw data.

3 Modeling Maneuver Dependency

We have now formulated the prediction of y_0 as a function of the current situation S_0 and the previous maneuver M_i . This can be extended by incorporating the last k maneuvers. Fig 1 shows an example of a network that uses $k = 3$ last maneuvers as features. To validate this concept we test it with $k = 1$ which is easier to train and requires less data.

Network Architectures. We design the network using two input layers separately, one takes the current situation (S_0) as input and one captures useful information from the past maneuver M_i .

Each of these two input layers are then followed by hidden layers and form two separate paths. For the ultimate purpose of predicting y_0 , these two paths are then combined in the deeper layers, followed by further hidden layers and lastly the output layer. The whole network is trained by back-propagating the classification error.

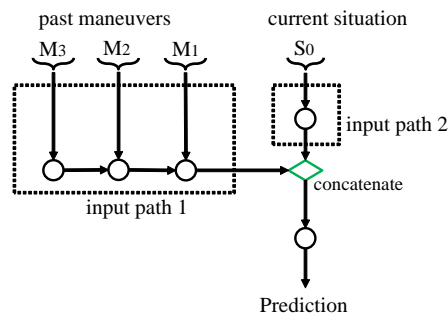


Fig. 1: Proposed architecture to capture the driver information from previous maneuver for predicting the current situation

Extracting Driver Information with Neural Network. As mentioned in Section 2, extracting driver information from M_i could be done in two different ways. We modeled the first approach by configuring the first input path using fully connected layers that take statistical values from the last maneuver execution as input. The function G could then be written as G_s for short. The second approach makes use of a recurrent layer to capture the maneuver execution. In particular, we use Long short-term memory networks (LSTM) [4] for capturing past maneuvers M_i . LSTM Networks have proven to be quite successful for sequence learning problems [3, 6]. The function G in this case is formalized as $G_{lstm}(S_0, M_i)$ for short.

4 Experiments

Data set. The data used in this work were collected from 32 drivers covering a wide range of ages and driving experiences. Each driver drove 30 rounds on a pre-defined route. The data set was collected in real-world traffic. The chosen route is a common urban route which requires the driver to perform different maneuvers roundabout, left turn, or intersection

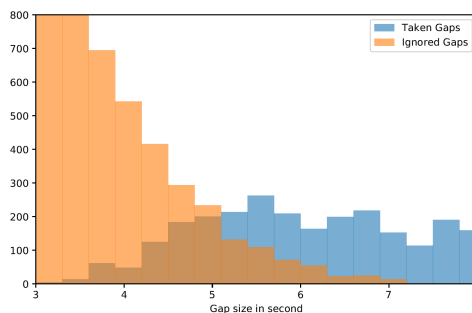


Fig. 2: Histograms of taken and ignored gaps

with right of way. For our experiments, we focused on the two most complex maneuvers, roundabout and left-turn.

For capturing the driving style we use four signals as features for describing a maneuver execution, which include speed, longitude and latitude acceleration and steering wheel speed. As features for describing the intersection situation we additionally use the position of the ego vehicle to determine the distance to the middle of the intersection.

Gap Acceptance at Left-Turn Maneuver. In an unsignalized intersection scenario, in which the incoming traffic has the right of way. We observe that the preference of the driver in choosing gaps differs from driver to driver. As shown in Fig. 2, gaps that are in range from approximately 3 to 7 seconds could either be taken or ignored. In such a scenario, the problem of predicting the gap acceptance of driver should not only consider the current traffic situation but also the driver’s individual preferences.

5 Evaluation

Evaluation Setup. We evaluate our method using 10-fold cross-validation to construct training and testing sets. The data are split into disjoint folds according to the driver ID. In total we evaluate four models. As the baseline model, we estimate the best possible threshold that separates the taken and ignored gaps in the training set.

$$\hat{\theta} = \arg \min_{\theta} L(y, t_i > \theta) \quad (6)$$

where t_i is the time available for turning left if gap i is taken. L is the loss function that computes the error rate for a given label y and the predictions based on threshold θ . The baseline is then compared to the three models described in the previous sections.

Learning from previous maneuvers. The average F1 score and the accuracy of all four models on the validation sets are show in Table 1. Both variants that extract information from M_i show significant improvements. The overall best score is obtained by G_{lstm} , which uses an LSTM layer for capturing M_i . Various network configuration are tested with different regularization parameters applied to each input path individually. In all settings, using extra information from M_i always leads to improvement of the prediction performance.

Reviewing the cross-validation results, we observe the improvement in terms of F1 score and accuracy of G_s and G_{lstm} over $G(S_0)$ in eight out of ten folds. The maximal improvement reaches 18.5% in F1 score which translates to 9.4% accuracy. Fig 3a shows the validation score of all three models in one of these eight folds. Here we can see that $G(S_0)$ gets stuck and its best score is quite low, whereas both, the G_s and G_{lstm} models benefit from the extra input M_i and reach much higher accuracy.

Table 1: Mean of F1 and accuracy score on validation set over 10-fold cross-validation

Model	F1 Score	Accuracy
Threshold	81.9 %	89.8 %
$G(S_0)$	86.8 %	92.6 %
$G(S_0, h_s(M_i))$	89.9 %	94.1 %
$G(S_0, h_{lstm}(M_i))$	91.7 %	95.1 %

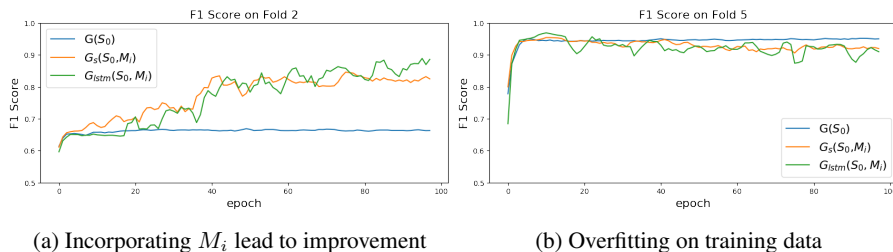


Fig. 3: Learning curve on different validation folds

In Fig 3b we observe the first slight effect of overfitting as the score of G_{lstm} and G_s keeps fluctuating and overfit on training data after 40 epochs. To deal with such effect, early stopping was also used for training the final model.

6 Conclusion

In this work we proposed a new approach to learn the dependency between maneuver execution, namely to extract the information about driver behavior and style, and use it to improve the performance of the prediction task in driver assistance systems. We implement our approach using neural networks as building blocks and empirically evaluate the model in a left-turn situation using on-road data. The results show that the model is able to extract the driver impact from the past maneuver executions and can use it improve the prediction quality by almost 10 % in terms of F1 score. We compared two approaches for extracting driver’s information from maneuver execution, which make use of statistical features and an LSTM layer. The LSTM model shows improvement over a model that only uses statistical features and fully connected layers.

References

1. Aggarwal, C.C.: Recommender Systems: The Textbook. Springer-Verlag (2016)
2. Dang, H.Q., Fürnkranz, J.: Using Past Maneuver Executions for Personalization of a Driver Model. In: Proceedings of the IEEE 21th International Conference on Intelligent Transportation Systems (ITSC-18). IEEE (Nov 2018), to appear
3. Dang, H.Q., Fürnkranz, J., Biedermann, A., Hoepfl, M.: Time-to-lane-change prediction with deep learning. In: Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC-17). pp. 1–7. IEEE (oct 2017)
4. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. *Neural Computation* **9**(8), 1735–1780 (nov 1997)
5. Martinez, C.M., Heucke, M., Wang, F.Y., Gao, B., Cao, D.: Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey. *IEEE Transactions on Intelligent Transportation Systems* **19**(3), 666–676 (mar 2018)
6. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to Sequence Learning with Neural Networks. In: Advances in Neural Information Processing Systems. pp. 3104–3112 (2014)