Intention estimation and controllable behaviour models for traffic merges

Aditya Mahajan, Takayasu Kumano and Yuji Yasui
Honda R&D Co. Ltd., Tochigi, Japan

ABSTRACT
This work focuses on decision making for automated driving vehicles in interaction rich scenarios like traffic merges in a flexibly assertive yet safe manner. We propose a Q-learning based approach, that takes in active intention inferences as additional inputs besides the directly observed state inputs. The outputs of Q-function are processed to select a decision by a modulation function, which can control how assertively or defensively the agent behaves.

ARTICLE HISTORY
Received 22 October 2020
Revised 24 December 2020
Accepted 27 January 2021

KEYWORDS
Automated driving; interactive decision making; traffic merge; behaviour modelling

1. Introduction
One of the important tasks in automated driving (AD) is motion planning. Careful motion planning involves constant interaction with nearby vehicles and it is vital to execute it effectively and efficiently even in complex scenarios. One such case of high complexity is the situation of merging into traffic. Merge manoeuvres accounted for over 400,000 accidents in the United States in 2014 [1]. Thus it becomes a crucial task for AD systems to be able to handle. However, it is possible that the automated driving vehicles fail to act like an expert human would, thus leading to failure in merging or a significant delay in completing the assigned task. A possible reason might be the poor interaction with the nearby traffic to judge intentions and an overly defensive driving model. This highlights the two broad challenges for AD in a traffic merge: (1) To interact and perform the manoeuvre smoothly, (2) To act with a certain level assertiveness that can be tuned.

Motion planners comprise of two sub systems: a decision maker and a trajectory planner. The decision maker outputs a high-level action of whether to keep the lane or to change it. The command from the decision maker is passed onto the trajectory planner which lays out the path for the low-level controller to follow. Our work, focuses on developing the decision maker for the given scenario and tries to address the above mentioned challenges. The scenario can be treated as a time sequential problem and the timing of the decisions can affect the final results. Hence, we model the scenario as a reinforcement learning (RL) problem. Our model uses Q-learning [2] as the base method.

Our main contributions in this paper are: (a) intention inferring input features for the Q-function to enhance the interaction capability, (b) an assertiveness modulation functionality that can be tuned to output actions of varying degree of assertiveness while still providing safety and without the need of any retraining. Our model is designed to utilize minimal inputs to ensure realistic conditions. We generate scenarios of varying complexities in simulation and quantitatively analyze 4 versions of the algorithm: (A) No behaviour modulation and no intention estimation, (B) Assertive behaviour model with intention estimation, (C) Intermediate behaviour model with intention estimation and (D) Conservative behaviour model with intention estimation (Figure 1).

The remainder of the paper is organized as follows. A literature review is given in Section 2. Section 3 introduces the methodology and details of our algorithms. Evaluation and results are provided in Sections 4 and 5. Sections 6 and 7 include the discussions and concluding remarks, respectively.

2. Related work
2.1. Decision making
Prior work on decision making in traffic merges [3,4] have used predefined rules and explicit instructions to model the behaviour of the ego vehicle in a given traffic situation. In DARPA urban challenge, the Boss team [5] used a rule-based method to assess the feasibility of merging in a given gap, based on the modelled spacing requirements. Although decent reliability can be achieved from these hand-crafted approaches, their performance is limited to specific cases and may fail to capture the dynamic interactions with the other vehicles. Similarly, the numerical optimization approaches [6,7] don't majorly incorporate interaction between vehicles in the decision making problem.
The partially observable Markov decision process (POMDP) are formulated to handle decision making problems that involve uncertainty and dynamic environments but they are marred by intractability in many cases [8,9] and hence lack the scalability. Methods like the ones presented by Cunningham et al. [10] requires forward simulation. Hubmann et al. [11] used POMDP to perform decision making in intersection scenarios. Though the approach accounts for the effect on the movement of other vehicles, it can only postpone the decisions and perform a constant braking to handle emergencies, hence leading to a conservative ego car driving behaviour.

Reinforcement learning [12] is a popular approach that is suited for time sequential problems. It is being researched for application in motion planning for AD vehicles and has shown some promising results as well [13]. We use Q-learning as the base model for our approach.

2.2. Intention inference

Some methods use Bayesian approach [14] and probability graph models [15] to estimate whether the other vehicle intends to yield or not. These approaches however require a rich dataset to train from. Liebner et al. [16], on the other hand propose Intelligent Driver Model (IDM) [17] predictions to estimate the intention of vehicles approaching at an urban intersection. Vallon et al. [18] use a simple “Target-Ego Vehicle Interaction” module that outputs IDM prediction for the nearby target vehicle. Owing to the ease of use and reliability we incorporate the IDM in our approach for inferring intention inputs.

2.3. Behaviour models and controllability

Some approaches are data driven and use machine learning methods like Scalar Vector Machines (SVMs) to learn the merge timing and aggressiveness of the driver [18]. Though these methods provide personalization, they require personal data of every driver in order to match their desired level of aggressiveness. Sadigh et al. [19] use inverse reinforcement learning where they incorporate other vehicle’s reward function into ego vehicle’s reward function. This helps model assertive behaviour but assuming perfect anticipation of the other vehicle’s motion in response to ego motion can lead to overconfidence in actions and hence, unsafe manoeuvres. Moreover, aggressiveness modelled by rewarding the slowing down of other participants would mean that ego car could choose an action biased to obstructing the other vehicles rather than quickly completing task. Another approach uses multi-agent reinforcement learning but only models cooperative behaviour [20]. Although, Levine and Koltun [21] model the driving behaviours of aggressive, evasive and tailgating drivers, these can’t be tuned to a preferred level of assertive behaviour unless we have the corresponding driver style data.

Our proposed decision maker model improves the performance of the base Q-learning algorithm by the use of additional features that act as representatives of intention predictions, and a novel behaviour modulation functionality. These model improvements are not computation heavy and are aimed at providing greater customizability in the ego vehicle behaviour when implementing in situations of varying traffic characteristics. We will explain these in greater detail in the following sections.

3. Method

3.1. Algorithm structure

Figure 2 shows the algorithm layout. The state features, that is both the observable state features and the derived intention inference features, are sent to the Q function. The Q function outputs the Q-values for the available action choices corresponding to the given state features. These Q-values at the current time step along with those of the state action pairs in the previous time step, go as inputs to a behaviour modulator that selects the final decision based on the set control parameters configuration and sends it to the trajectory planner.

3.2. Action space

For the traffic merge scenario, the high-level decision maker outputs an action $a$ at a given time $t$.

$$a \in A, \text{ where } A = \{\text{Merge, No Merge}\}$$

The Merge action initiates lane change and the No Merge action instructs the vehicle to keep moving in the same lane. Once the ego car decides to merge, it
can't be revert back. This is done to avoid learning any sub-optimal behaviour.

### 3.3. State space

The state space \( S \) can be divided into observable state space \( S_{\text{obs}} \), which can be detected directly, and unobservable state space \( S_{\text{unobs}} \), which needs to be predicted using \( S_{\text{obs}} \).

#### 3.3.1. Observable state space

High dimensionality of inputs due to ignoring occlusion and treating every traffic participant as observable, leads to high computation cost and affects learning performance due to the presence of irrelevant traffic participants. Thus, to ensure robust performance and avoid superfluous information, we keep our observable state information minimal and close-to-real. Besides the ego car, we only consider the nearest vehicles in front and behind the ego car, that are in the target merge lane, for inputs to our algorithm. We refer to these two vehicles as front car and back car, respectively, hereafter. The observable state information of the front car, back car and ego car are denoted as \( s_{\text{obs}}^{\text{front}} \), \( s_{\text{obs}}^{\text{back}} \) and \( s_{\text{obs}}^{\text{ego}} \), respectively, and contain the following information:

\[
S_{\text{obs}} = (s_{\text{obs}}^{\text{front}} \cup s_{\text{obs}}^{\text{back}} \cup s_{\text{obs}}^{\text{ego}})
\]

\[
s_{\text{obs}}^{\text{front}} = (x_{\text{front}}, y_{\text{front}}, \theta_{\text{front}}, v_{\text{front}}, w_{\text{front}})
\]

\[
s_{\text{obs}}^{\text{back}} = (x_{\text{back}}, y_{\text{back}}, \theta_{\text{back}}, v_{\text{back}}, w_{\text{back}})
\]

\[
s_{\text{obs}}^{\text{ego}} = (x_{\text{ego}}, y_{\text{ego}}, \theta_{\text{ego}}, l_{\text{ego}}, w_{\text{ego}})
\]

(\( x_{\text{ego}}, y_{\text{ego}}, \theta_{\text{ego}}, l_{\text{ego}}, w_{\text{ego}} \) represent the position in global x-y coordinates, speed, yaw angle, length and width of the ego car. \( x_{\text{back}}, y_{\text{back}}, \theta_{\text{back}}, v_{\text{back}}, w_{\text{back}} \) represent the relative position, relative speed, relative yaw angle of the back car with respect to the ego car, while \( b_{\text{back}} \) and \( w_{\text{back}} \) are length and width of the back car. The variables in \( s_{\text{obs}}^{\text{front}} \) have similar meanings to the ones in \( s_{\text{obs}}^{\text{back}} \) but for the front car. The visibility was limited to \( d_{\text{max}} \) ahead and behind the ego car depending on the desired sensor range.

#### 3.3.2. Unobservable state space

Features like intention can be treated as variables belonging to the unobservable state space \( S_{\text{unobs}} \) and cannot be accounted for directly. Hence, they need to be predicted using the observable state variables. The Intelligent Driver Model (IDM) is a car following model for urban traffic situations that predicts the acceleration response \( \dot{v}_a \) of a vehicle \( \alpha \) by combining a term \( \dot{v}_a^{\text{free}} \), for free road behaviour and a term \( \dot{v}_a^{\text{int}} \), for interaction behaviour with the vehicle that it should follow.

\[
\dot{v}_a^{\text{free}} = \dot{v}_a^{\text{int}} + \dot{v}_a^{\text{free}}(\alpha)
\]

\[
\dot{v}_a^{\text{free}}(\alpha) = a_r(\alpha) \left[ 1 - \left( \frac{v_a}{V_0^{(\alpha)}} \right)^\delta \right]
\]

\[
\dot{v}_a^{\text{int}} = -a_r(\alpha) \frac{s_0 + v_a T}{s_\alpha} + \frac{v_a \Delta v_a}{2\sqrt{abs_\alpha}}
\]

Here \( v_a \) is the speed of the vehicle \( \alpha \), \( v_0^{(\alpha)} \) is its desired speed and \( \Delta v_a \) is the velocity difference from the vehicle it follows. \( s_0 \) is the minimum spacing, \( T \) is the desired time headway, \( a \) is the maximum acceleration and \( b \) is the comfortable deceleration. At merges, the target lane vehicle's acceleration response could be affected by its choice to yield or not yield. While yielding, the interaction with the incoming vehicle from the next lane would heavily dominate its subsequent motion. As per IDM, it would treat this incoming vehicle that is about to merge, as the vehicle to follow. If the vehicle is not yielding, it would continue to use an acceleration response that helps it follow the car ahead in its own lane and remain close to the desired speed. Thus, we can get two possible IDM predictions for the acceleration response of the back car:

\[
\dot{v}_{\text{back/front}}^{\text{free}}: \text{front car treated as the vehicle to be followed.}
\]

\[
\dot{v}_{\text{back/ego}}^{\text{free}}: \text{ego car treated as the vehicle to be followed.}
\]

These predictions about the possible behaviours can act as meaningful signals of intention for our decision-making algorithm. Our proposed model, therefore uses these intention inferences to represent the unobservable state \( S_{\text{unobs}} \) as:

\[
S_{\text{unobs}} = (\dot{v}_{\text{back/front}}, \dot{v}_{\text{back/ego}})
\]

#### 3.4. Reinforcement learning

Reinforcement learning involves an agent in an environment that takes an action \( \alpha_t \) according to policy \( \pi \) for a given state \( s_t \) at a given time instant \( t \). It consequently ends up in state \( s_{t+1} \) and receives a reward \( r_t \) for that given time step with a discount factor of \( \gamma \). Thus, from any time instant until the end of the task, the final return can be written as:

\[
R_t = \sum_{t'=t}^{t''} \gamma^{t'-t} r_t \quad \text{where} \quad \gamma \in [0, 1]
\]

One way to model the problem is in a form where the objective is to maximize this expected future return by learning the optimal policy \( \pi^* \). The expected future return for a given state and action combination is
Algorithm 1 Q-learning with Random Forest

1: Initialize experience buffer $\xi$, exploration rate $\epsilon_0$, decay rate $\zeta$, random forest $\mathcal{R}_0$.
2: for $i = 1$ to $N$ do
3:   Initialize state $s_0$, time $t = 0$
4:   while $s_t \notin \text{terminal AND } t < \text{limit}$ do
5:     $a_t = \begin{cases} 
        \text{random}(A), & \text{if random}(0, 1) \leq \epsilon_i \\
        \arg \max_a Q^{\mathcal{R}_i}(s_t, a), & \text{otherwise}
      \end{cases}$
6:     Execute $a_t$ to get $s_{t+1}, r_t$
7:     Record $(s_t, a_t, r_t, s_{t+1})$ to $\xi$
8:     $t \rightarrow t + 1$
9:   end while
10: Refit $\mathcal{R}_i$ on $(Q^{\mathcal{R}_i}(s_t, a_t) - r_t - \gamma \max_a Q^{\mathcal{R}_i}(s_{t+1}, a_{t+1}))^2$ over $\xi$ to get $\mathcal{R}_{i+1}$
11: $\epsilon_i \xi \rightarrow \epsilon_{i+1}$
12: end for
13: return $\mathcal{R}_N$

called the Q-value and can be represented as a Bellman equation (Equation (10)).

$$Q^\pi(s_i, a_i) = E \left[ r_i + \gamma \max_a Q^\pi(s_{i+1}, a_{i+1}) \right]$$  \hspace{1cm} (10)

This approach is referred to as Q-learning. As mentioned before, we use this as the underlying model and update the Q function at regular intervals. We use a random forest regressor as the Q-function approximator. Random forests inherently provide robustness against unbalanced data, have a low bias and a modest variance [22]. Algorithm 1 explains the learning process in detail.

3.5. Reward function

Our reward function is simple and sparse. It is as follows:

$$r = \begin{cases} 
    +1, & \text{if merge successful} \\
    -1, & \text{if bad merge/time - out} \\
    0, & \text{otherwise}
  \end{cases}$$  \hspace{1cm} (11)

Positive reward is for a successful lane change and negative reward for a failure in completing the task due to time-out or a bad merge. The discount factor $\gamma$ incentivizes the ego car to reach the target lane in minimum possible time steps.

3.6. Behaviour modulator

Using the Q-function that was previously learnt, we generate Q-values for all possible actions in the current state and the state at the previous time step and send these as inputs to a modulation function. This modulation function conducts two layers of checks, one for each of these two consecutive time steps to ensure temporal consistency and reliability. Each layer involves a 2-step threshold check of the Q-values. The formulation for the layer $i$ of the behaviour modulator is as follows: \forall i \in \{1, 2\}

$$e^{Q_{i-1}+1(s, \text{Merge})} - e^{Q_{i-1}+2(s, \text{No Merge})} > \theta_1$$  \hspace{1cm} (12)

$$e^{Q_{i-1}+1(s, \text{Merge})} > \theta_2$$  \hspace{1cm} (13)

where $Q_{t-1+i}(s, a)$ is the Q-value for a given state $s$ and an action $a$ at time $t-i+1$. $\theta_1, \theta_2$ are the tunable thresholds for the layer $i$ and can take values in the $[0, 1]$ and $[Q_{\min}, Q_{\max}]$ intervals, respectively. The choice of these thresholds allows assertiveness to be manipulated while concurrently balancing the safety.

The softmax layer (Equation (12)) ensures a certain degree of relative significance of the Q-value for the Merge action over the No Merge action. The threshold $\theta_1$, sets the limit that the difference between the Q-values of the two action choices should be greater than. The second step (Equation (13)) ensures absolute significance of the Q-value for the Merge decision by checking if it is greater than a certain value. We generally set these values positive since they are meant to avoid instances with low expected rewards that might signal poor return for a decision. Further, we keep the threshold values $\theta_1 \geq \theta_2$ for all $j \in \{1, 2\}$ to accommodate and capture incremental movements in the Q-value as a positive signal.

Figure 3 shows an example of how this functionality improves the quality of decision making. Here point 2 has a high absolute Q-value for merge but a low softmax value and point 3 has a high softmax value for merge but a low Q-value. Such cases can cause ambiguity in judgement and lead to poor decision making. Our function minimizes the influence of such state conditions on the quality of decision making. The vector of thresholds $(\theta_{11}, \theta_{12}, \theta_{21}, \theta_{22})$ is denoted as $\Theta$. High thresholds imply stricter conditions for merging and lead to conservativeness. Low thresholds imply relaxed conditions and can enhance assertiveness.

In our experiment, to realize different degrees of assertiveness, we model three set of thresholds $-\Theta^B, \Theta^C, \Theta^D$ that satisfy the condition $\Theta^B \leq \Theta^C \leq \Theta^D, \Theta^B$ has the most relaxed conditions and hence is more

Figure 3. (a) Softmax Q-values vs. time (b) Q-value vs. time. Green colour stands for clearing the threshold $\theta_{11}$, red is for only clearing the threshold $\theta_{12}$ and black represents the rest. The blue circle is the set of points that satisfy all checks in both layers.
aggressive. On a similar grounds, $\Theta^D$ can be considered more conservative and $\Theta^C$ can be treated as intermediate or neutral. It must be noted, that beyond a certain range, these thresholds may induce extreme behaviours in the agent, like being too conservative and avoiding a merge or becoming too aggressive and recklessly tending to merge. This range is found out by trial and error and from within it, we pick any set of values that satisfy the above conditions. For $\Theta^B$, we set $\theta_1$ slightly lower than 0.5, with a low yet positive $\theta_2$. This implies that our decision maker would accept a Merge decision even if the Q-value for No Merge is slightly higher than that for Merge, provided that the Q-value for Merge represents a slightly favourable expected reward, which in our case would be inferred by it being positive. For modelling conservative behaviour, we raise all the thresholds and for the neutral behaviour we use an intermediate set of values. For the experiment, values that satisfied the above criteria and also gave distinguishable results, were chosen as $\Theta$ for each behaviour and are as follows: $[\Theta^B, \Theta^C, \Theta^D] = [(0.47, 0.3, 0.45, 0.3), (0.5, 0.3, 0.5, 0.3), (0.53, 0.4, 0.51, 0.4)]$.

The 4 behaviour models that we would analyze are: (A) No behaviour modulation and no intention estimation, (B) Assertive behaviour model with intention estimation which uses $\Theta^B$, (C) Intermediate behaviour model with intention estimation which uses $\Theta^C$ and (D) Conservative behaviour model with intention estimation which uses $\Theta^D$.

3.7. Speed and trajectory control

The longitudinal speed control of the ego car is carried out by a car following model similar to IDM. The free road desired velocity $v_{ego}^{des}$ for the ego car in IDM is defined as:

$$v_{ego}^{des} = \begin{cases} v_{ego}, & \text{if } |v_{ego} - v_{traffic}| \geq 0.5 \text{m/s} \\ v_{traffic} - 0.5, & \text{otherwise} \end{cases}$$

Here $v_{traffic}$ is the speed of traffic in the target lane. The output trajectories for executing the merge action come from a polynomial trajectory generator function.

4. Evaluation

We train and test the models in our own simulation software designed to resemble the pattern of the traffic flow. The car-following model used for all vehicles is IDM. To test robustness of our proposed approach, we ensure that the IDM model used for intention inference is different from the one used by the traffic in the simulator. While the intent inference outputs predictions based on a fixed minimum spacing value, desired speed, maximum acceleration and comfortable deceleration, the simulator generates scenarios by treating these values as variables. The traffic speed varies between 1 to 6 m/s. The traffic density was modelled as high and medium, implying a minimum spacing of less than 6 m, and greater than 6 m, respectively. The length of agent was set to 4.5 m and the simulator’s sampling rate was 10 Hz.

The traffic composition was modelled as friendly (all drivers are friendly), mixed (each driver has an equal probability of being friendly or unfriendly) and aggressive (all drivers are unfriendly). Vehicles assigned a friendly behavior allow the ego car to merge in front of them by opening the gap. However, this gap is only opened up for a randomly assigned time window of 0.4 to 1.0 s and varies with every vehicle. Hence within the friendly vehicles there is a varying degree of patience. After the time window is passed, vehicles change their behavior to unfriendly. Vehicles that are unfriendly, do not respond significantly to the presence of the ego car. It is assumed that for a randomly assigned value from $\{0, 0.1, 0.2, 0.3\}$ s, they decelerate and then accelerate back to resume following the car in front of them unless the ego car comes in their way, at least partially. They have a higher maximum acceleration of $4 \text{m/s}^2$, compared to the friendly cars' $2 \text{m/s}^2$. The deceleration response limit was set to $-2 \text{m/s}^2$ for all traffic complexity is higher in situations vehicles.

The discount factor $\gamma$ was set to 0.6. The visibility, $d_{max}$ was set to 50 m. For simplicity, no cars except the ego car change lanes. The models were trained for 400 episodes. The random forest had 10 trees and a maximum depth of 40. The exploration decay rate $\xi$ was set to 0.99.

We classify successful merges into two categories. The cases where the inter-vehicle gap of the ego car with both front and back car at the time of the completion of merge is greater than 1 m, are called clear merges. Otherwise, they are considered as tight merges. There are 2 kinds of failure cases: time-outs and bad merges. We compare the performance of our proposed model with the vanilla model in terms of the percentage cases of clear merges, tight merges, bad merges, time-outs and the time to completion of successful merges over a test run of 200 episodes.

5. Result

Figure 4 compares the four different methods for different metrics in different scenarios. In general, all methods seem to show an increase in the time to completion, percentage of tight merges and the failure cases, that is, instances of time-outs and bad merges, both as the traffic density increases and as the nature of target lane vehicles transitions from friendly to aggressive. The percentage of clear merges consequently decreases while the time to completion increases with above mentioned transitions in traffic trends. These are
Figure 4. Comparison of different models. The models are denoted as: A – No uior modulation and no intention estimation, B – Assertive behaviour model with intention estimation, C – Intermediate behaviour model with intention estimation and D – Conservative behaviour model with intention estimation. (a) Time to successful completion (sec) for different traffic situations. (b) Performance in dense traffic and (c) Performance in mild traffic.

The intention inputs clearly help generate a better performance over the vanilla methods due to their quite intuitive, as the traffic complexity is higher in situations that are denser and have a higher number of aggressive traffic participants to interact with.

Our method using IDM predictions performs better than the vanilla Q-learning approach in most cases. The benefit of including intention inputs reflects in the occurrence of relatively lower number of failure cases (bad merge+time-out) for most situations by models B, C and D, in comparison to the model A that does not use intentions, as can be seen from Figure 4(b,c). Though the model A demonstrates higher successful tight merges than other methods when the traffic is more friendly, it also generates higher number of bad merges in aggressive traffic conditions, highlighting the inability to assess nature of traffic while generating a response.

6. Discussion

The tuning of thresholds in the behaviour modulator function leads to varying behaviours. Lowering the thresholds slightly, increases the vehicle’s tendency of accepting risk and hence acting assertively. This could be seen in the Figure 4(b,c), where the number of tight merges increases in the case of model B compared to model D. Model B consequently exhibits the minimum instances of time-outs and produces quicker merges than the other methods. It is thus able to perform better than the other models for dense and aggressive traffic situations.

When we alter the same thresholds to a higher value of model D, we tend to see an increase in conservative behaviour, such as lower number of tight merges. Also, Figure 4(a) clearly shows that within the behaviour models, the time to completion is low for the assertive setting in model B and high for the conservative setting in model D. This is quite intuitive, as stricter thresholds lead to stricter preference and consequently, slower merges and longer wait times. Though model D tends to generate more time-out type failure cases, it does nearly eliminate the cases of bad merges, due to strict preference for Merge action Q-values with higher absolute and relative significance. The neutral parameter setting of model C generates an averaged behaviour of the assertive and conservative model settings.

Comparing Figure 5(a,b) we can observe the behavioural difference for the assertive and conservative models, respectively, in similar traffic settings. Model B’s assertiveness can be noticed from its response to merge early even when the gap available in the target lane is narrow. On the other hand, model D acts a bit more cautiously by waiting for the gap to become wider until there is sufficient confidence for the merge action to be adjudged safe.
ability to produce the possible future motion predictions of the back car for a given situation and hence incorporate a factor of interaction. The Q-function benefits from these inference patterns. Our novel behaviour modulator feature helps model the different ego vehicle behaviours without the requirement of re-learning. The various settings of parameters demonstrate different degrees of assertiveness, in terms of the time taken to complete the task and in terms of tendency to perform tight merges. Tuning of the behaviour modulator parameter settings within an acceptable range can help alter the ego car behaviour according to preference with minimal impact on the safety.

Our method not only helps the ego car perform decision making in interaction rich scenarios with greater customizability of behaviour, it also helps provide this without a high computation cost and with minimal input requirements from the observable state space. However, it must be noted that the degree of tunability and range of parameter choices, can be affected by many factors. For example, a complex reward function including rewards of varying weights can affect the shape of the Q-function and hence, have a non-linear impact on the consequent threshold choices, meant to generate the intended behaviour.

In future works, we would explore the possibility of improving the assertiveness and interaction in traffic situations where the vehicles in other lanes also try to merge to the ego car’s lane. The subsequent steps would also involve testing against a human driver in a simulator environment to analyze the response of test subjects to the ego vehicle’s behaviour.

7. Conclusion

We proposed a method that is targeted to help solve the decision making problem in a scenario which involves constant interaction with nearby traffic participants. Our approach uses IDM predictions as representative for intention, a random forest based Q-function to output Q-values for state-action pairs and a behaviour modulator for regulating the assertiveness of the final decision. In most cases, the use of intention, seems to successfully reduce the failure rates and the behaviour modulator adds the ability to tune the vehicle’s behaviour as per preference in terms of the quickness of manoeuvres and the handling of merging tasks, even in tight spaces. Further research in developing such features could be of utility for implementing interactive and customizable AD systems, while ensuring safety.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Aditya Mahajan He received his BTech degree from the Indian Institute of Technology Delhi, India, in 2017. He subsequently joined Honda R&D Co., Ltd. He is currently a Staff Engineer in the Computer Science Domain at Honda’s Centre for Innovative Research Excellence. His research interests include machine learning and decision making.

Takayasu Kumano He received his BS, MS, and PhD degrees from Tohoku University, Japan, in 2003, 2006, and 2009, respectively. After NASA post-doc program fellow and data analytics consultant, he joined Honda R&D Co., Ltd in 2017. He has researched AI-based path planning and decision-making technologies for automated driving. He is currently an assistant chief engineer in the Computer Science Domain at Honda’s Center for Innovative Research Excellence.

Yuji Yasui He received the BE and ME degrees in mechanical engineering from Tokyo University of Science, Japan, in 1992 and 1994, and the PhD degree from Sophia University, Japan, in 2012. He joined Honda R&D Co., Ltd in 1994 and has researched powertrain control for low-emission vehicles and HEVs, traction control for F-1 racing car, transmission control and device control by using adaptive control, model predictive control, neural network, etc. He is currently an executive chief engineer in a research group for automated driving and advanced driving assistance system using AI and advanced control technologies.

References

[1] National Highway Traffic Safety Administration. Traffic safety facts 2014: a compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system. p. 87.
[2] Watkins CJCH, Dayan P. Q-learning. Mach Learn. 1992;8(3–4):279–292.
[3] Urmson C, Anhalt J, Bagnell D, et al. Autonomous driving in urban environments: boss and the urban challenge. J Field Robot. 2008;25(8):425–466.
[4] Montemerlo M, Becker J, Bhat S, et al. Junior: the stanford entry in the urban challenge. J Field Robot. 2008;25(9):569–597.
[5] Baker CR, Dolan JM. Traffic interaction in the urban challenge: putting boss on its best behavior. IEEE/RSJ International Conference on Intelligent Robots and Systems; 2008, Nice, France.
[6] Ferguson D, Howard TM, Likhachev M. Motion planning in urban environments. J Field Robot. 2008;25(11–12):939–960.
[7] Hardy J, Campbell M. Contingency planning over probabilistic obstacle predictions for autonomous road vehicles. IEEE Trans Robot. 2013;29(4):913–929.
[8] Madani O, Hanks S, Condon A. On the undecidability of probabilistic planning and related stochastic optimization problems. Artif Intell. 2003;147(1–2):5–34.
[9] Bernstein DS, Givan R, Immerman N, et al. The complexity of decentralized control of Markov decision processes. Math Oper Res. 2002;27(4):819–840.

[10] Cunningham AG, Galceran E, Eustice RM, et al. MPDM: multipolicy decision-making in dynamic, uncertain environments for autonomous driving. 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, Washington, USA.

[11] Hubmann C, Becker M, Althoff D, et al. Decision making for autonomous driving considering interaction and uncertain prediction of surrounding vehicles. 2017 IEEE Intelligent Vehicles Symposium, Milan, Italy.

[12] Sutton R, Barto A. Reinforcement learning: an introduction. Cambridge, Massachusetts: MIT Press; 1998.

[13] Mukadam M, Cosgun A, Nakhaei A, et al. Tactical decision making for lane changing with deep reinforcement learning; 2017.

[14] Dong C, Dolan JM, Litkouhi B. Intention estimation for ramp merging control in autonomous driving. 2017 IEEE Intelligent Vehicles Symposium, Milan, Italy.

[15] Forbes J, Huang T, Kanazawa K, et al. The batmobile: towards a Bayesian automated taxi. IJCAI, Montreal, Quebec, Canada; Vol. 95; 1995.

[16] Liebner M, Baumann M, Klanner F, et al. Driver intent inference at urban intersections using the intelligent driver model. 2012 IEEE Intelligent Vehicles Symposium, Alcala de Henares, Spain.

[17] Treiber M, Hennecke A, Helbing D. Congested traffic states in empirical observations and microscopic simulations. Phys Rev E. 2000;62(2):1805–1832.

[18] Vallon C, Ercan Z, Carvalho A, et al. A machine learning approach for personalized autonomous lane change initiation and control. 2017 IEEE Intelligent Vehicles Symposium, Milan, Italy.

[19] Sadigh D, Sastry S, Seshia SA, et al. Planning for autonomous cars that leverage effects on human actions. Robot Sci Syst. 2016;2:Chap 29.

[20] Yang J, Nakhaei A, Isele D, et al. Cm3: cooperative multi-goal multi-stage multi-agent reinforcement learning. Preprint; 2018. arXiv:1809.05188.

[21] Levine S, Koltun V. Continuous inverse optimal control with locally optimal examples. Preprint; 2012. arXiv:1206.4617.

[22] Breiman L. Random forests. Mach Learn. 2001;45(1):5–32.