CS 234 Reinforcement Learning Project: Data-Driven Multi-agent Human Driver Modeling

Raunak Bhattacharyya [1] **Bernard Lange** [1] **Derek J. Phillips** [2]

Abstract

The validation of autonomous driving algorithms in simulation requires good models of human driving behavior. In this project, we explored GAIL based imitation learning algorithms and compared them with the rule based model. The NGSIM dataset was used which represented a freeway US 101. Results have shown that PS-GAIL performs significantly better and InfoRAIL performs worse over the domain examined. In some cases, InfoGAIL matches the performance of the PS-GAIL. However, we hypothesize that the training time was not long enough and that the implementation of InfoGAIL and InfoRAIL requires more iterations in comparison to PS-GAIL and RAIL as the expert policy is a mixture of numerous expert policies for different latent variables.

1. Introduction

One of the main challenges in the creation of autonomous vehicles is ensuring the safety of the designed system in simulation before any of the real world testing begins. Taking inspiration from the recent interest in robot learning from demonstrations (Argall et al., 2009), this project aims to use the different imitation learning approaches for the driver behaviour modelling. The intended contributions are twofold:

1. Compare different adversarial imitation learning algorithms in the case study of driver modelling and compare it with a rule-based method
2. Investigate whether the latent state inferencing can help imitation performance. Latent state refers to the underlying driving style that is not directly observable from the demonstration or policy rollout data.

Table 1. Features of NGSIM Dataset used in the project

COLUMN NAME	DESCRIPTION
VEHICLE ID	VEHICLE IDENTIFICATION NUMBER
FRAME ID	FRAME IDENTIFICATION NUMBER
TOTAL FRAMES	TOTAL NUMBER OF FRAMES
LOCAL X	VEHICLE LATERAL POSITION
LOCAL Y	VEHICLE LONGITUDINAL POSITION
V. LENGTH	VEHICLE LENGTH
V. WIDTH	VEHICLE WIDTH
V. VELOCITY	VEHICLE VELOCITY
V. ACCELERATION	VEHICLE ACCELERATION
LANE ID	CURRENT LANE POSITION

1.1. Background work

The first instance of adversarial learning in driving was using GAIL (Ho & Ermon, 2016) in single agent settings (Kuefler, 2017). Subsequently, there has been work on using GAIL in multi-agent driving situations (Bhattacharyya et al., 2018) and reward augmentation to provide domain knowledge to the learning agent (Bhattacharyya et al., 2019). However, there still remains room for improvement in driver modeling performance as we have not reached the point of perfect driving behavior as manifested by the presence of undesirable occurrences such as collisions and off the road driving in resulting driving.

At this point, the literature seems to be scattered in multiple directions with different approaches having been proposed. With this project, our goals are to collect these different approaches into one cohesive study with results on multi-agent situations benchmarked against rule based models. Simultaneously, we propose to investigate InfoGAIL (Li et al., 2017) in terms of imitation performance. Our hypothesis is that latent state inferencing provided by InfoGAIL will enable better imitation performance. This hypothesis rests on the assumption that the NGSIM dataset (see Dataset) captures enough driver variation for latent state inferencing to provide tangible benefits.

[*]Equal contribution [1]Department of Aeronautics and Astronautics, Stanford University, Stanford, California, USA. [2]Department of Computer Science, Stanford University, Stanford, California, USA. Correspondence to: Raunak <raunakbh@stanford.edu>, Bernard <blange@stanford.edu>, Derek <djp42@stanford.edu>.

[†]This work was a continuation of prior research conducted in the Stanford Intelligent Systems Laboratory by Raunak and Derek.

1.2. Dataset

The dataset used in this project originated from the Next Generation Simulation (NGSIM) program which collected vehicle trajectories and various supporting data through a network of synchronized digital video cameras (Colyar & Halkias, 2007). We have focused on dataset representing US 101 freeway (see Figure 1). The features included in the datasets range from the position and velocity to the type of the vehicle and the name of the freeway. Features used in this project are described in the Table 1.

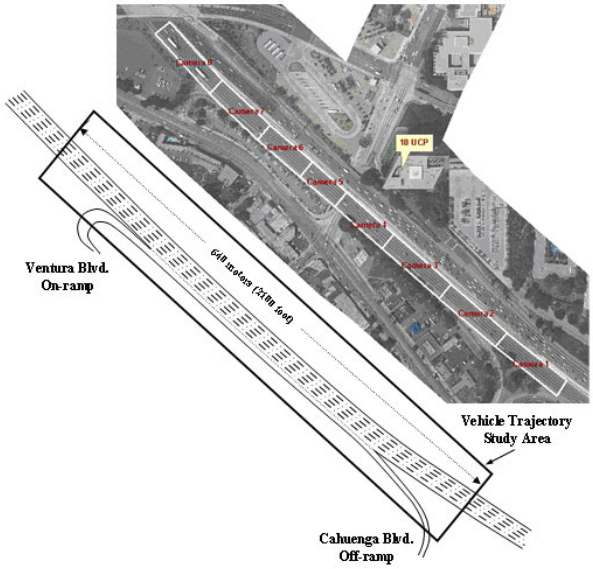

Figure 1. US 101 road from the NGSIM datasat. The length of the analyzed road is 640 meters and it consists of five lanes, an on-ramp and an off-ramp (Colyar & Halkias, 2007).

2. Approach

2.1. Formulation

We formulate highway driving as a sequential decision making task, in which the driver obeys a stochastic policy that maps observed road conditions to a probability distribution over driving actions (Kaelbling et al., 1998; Kochenderfer, 2015). Given a dataset consisting of a sequence of state-action tuples (s_t, a_t) demonstrating highway driving and a class of policies π_θ parameterized by θ, we adopt imitation learning to infer this policy.

We use the multi-agent extension of Markov decision processes adapted to the imitation learning framework (Littman, 1994). Suppose there are n agents. The state, action, and policy of agent i are denoted s_i, a_i, and π_i, respectively. The state, action, and policy of the multi-agent system are denoted $\mathbf{s} = [s_1, \ldots, s_n]$, $\mathbf{a} = [a_1, \ldots, a_n]$, and $\bar{\pi}(s_1, \ldots, s_n) = (\pi_1(s_1), \ldots, \pi_n(s_n))$. The state space and the action space of the multi-agent system are denoted $\mathcal{S}$ and $\mathcal{A}$, respectively. In the remainder of this paper, we use s and a without the subscripts to refer to the single agent scenario. We make some simplifying assumptions to the general Markov Games framework, which include agents being homogeneous (every agent has the same action and observation space), each agent getting independent rewards (as opposed to there being a joint reward function), and the reward function being the same for all the agents.

2.2. Problem Definition

We have compiled a set of Generative Adversarial Imitation Learning (GAIL) based algorithms we have investigated (GAIL, RAIL, InfoGAIL, InfoRAIL) and decided to use rule-based method as an benchmark. Each of the methods is described below.

2.3. Rule-Based Benchmark Implementation

We have implemented the IDM+MOBIL rule based driving controllers (Treiber et al., 2000). The driving behavior obtained using these rule based models will serve as a benchmark for for evaluating the performance of GAIL based algorithms in imitating human driving behavior. We expect the rule based models to perform well in terms of avoiding undesirable driving behavior but not perform well in imitating human driving behavior.

2.4. GAIL

Generative Adversarial Imitation Learning (GAIL) is a policy gradient method leveraging Generative Adversarial Networks. It is a useful method to train generative models by making them play a minimax game against a critic $D_\omega(s, a)$. The critic is trained to distinguish between trajectories coming from the demonstration dataset and those generated by the generative model. In this driving case study, our generative model is a driving policy $\pi_\theta(a|s)$ that outputs acceleration and turn rate based on the input features. To deal with the difficulty of training GAIL, we decided to use the Wasserstein distance metric (Arjovsky et al., 2017).

Algorithm 1 GAIL (Ho & Ermon, 2016)

for $i = 0, 1, 2, ...$ **do**
  Sample trajectories: $\tau_i \sim \pi_\theta$
  Update ω by ascending with gradients:
  $\Delta_\omega = \hat{\mathbb{E}}_{\tau_i}[\nabla_\omega log(D_\omega(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_\omega log(1 - D_\omega(s,a))]$
  Update θ using TRPO update rule with the following objective:
  $\hat{\mathbb{E}}_{\tau_i}[\nabla_\omega log \pi_\theta(a|s) Q(s,a)] - \lambda \nabla_\theta H(\pi_\theta)$
end for

2.5. RAIL

Reward Augmented Imitation Learning (RAIL) augments the learning agent with domain specific knowledge. The designer of the imitation learning agent can provide external reward signals. In this driving study, we penalized the learning agent for colliding and driving off the road. The rest of the algorithm proceeds similarly to GAIL with the augmented reward being added to the reward signal from the critic in the TRPO optimization process.

2.6. InfoGAIL and InfoRAIL

The application of Generative Adversarial Imitation Learning (GAIL) provided the tool to learn the policy from expert demonstrations. However, one of the major drawbacks of GAIL is its dependence on the quality of the expert demonstrations which can vary, as in our case drivers skills and driving conditions does. Hence, there is a need to disentangle these states called latent factors. The approach which addresses this issue is called InfoGAIL (Li et al., 2017). It learns the latent variables depending on the expert trajectories and the policy depending on those variables. To enforce the disentangling of the trajectories based on the latent variable, information-theoretic regularization ($L_I(\pi_\theta, Q_{\psi_{i+1}})$ is used to maximize mutual information between trajectories and latent variable, where $Q(c|\tau)$ provides a posterior approximation $P(c|\tau)$ (Li et al., 2017). Our final policy is then a mixture of expert policies for different latent variables. The algorithm is outlined in Algorithm 2.

Algorithm 2 InfoGAIL (Li et al., 2017)

for $i = 0, 1, 2, ...$ **do**
  Sample a batch of latent codes:$c_i \sim p(c)$
  Sample trajectories: $\tau_i \sim \pi_\theta(c_i)$
  Sample state-action pairs: $\chi_i \sim \tau_i, \chi_E \sim \tau_E$
  Update ω by ascending with gradients:
  $\Delta_{\omega_i} = \hat{\mathbb{E}}_{\chi_i}[\nabla_{\omega_i} log(D_{\omega_i}(s,a))] + \hat{\mathbb{E}}_{\chi_E}[\nabla_{\omega_i} log(1 - D_{\omega_i}(s,a))]$
  Update ψ by descending with gradients:
  $\Delta_{\psi_i} = -\lambda_1 \hat{\mathbb{E}}_{\chi_i}[\nabla_{\psi_i} log Q_{\psi_i}(c|s,a)]$
  Update θ using the TRPO update rule with the following objective: $\hat{\mathbb{E}}_{\chi_i}[\nabla_{\omega_i} log D_{\omega_{i+1}}(s,a)] - \lambda_1 L_I(\pi_\theta, Q_{\psi_{i+1}}) - \lambda_2 H(\pi_\theta)$
end for

2.7. Imitation Learning Implementation

We have implemented the GAIL (Ho & Ermon, 2016) and InfoGAIL (Li et al., 2017) using the rllab framework (Duan et al., 2016). We have integrated the algorithm with our driving simulator wherein the states and action demonstrations are provided from the NGSIM driving data and the resulting learned policies from imitation learning are fed back into the driving simulator for validation.

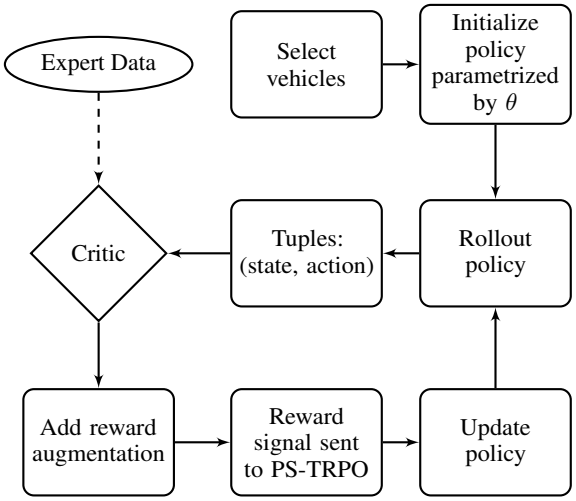

Figure 2. The high level system diagram of GAIL using parameter sharing TRPO and reward augmentation from external domain knowledge.

3. Experiments

We use the results from Rules based as a baseline to compare against the PS-GAIL, InfoGAIL, RAIL and InfoRAIL algorithms by learning policies and calculating specific metrics, as described in (Bhattacharyya et al., 2018). We train a policy for each set of parameters that we want to compare

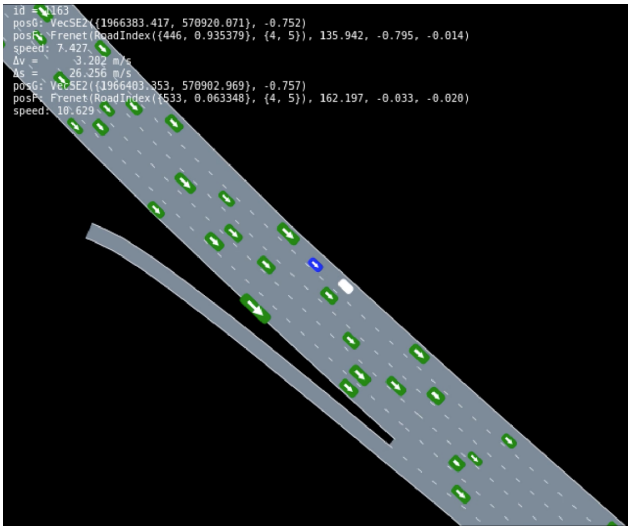

Figure 3. The green vehicles are expert demonstrations from the NGSIM dataset. The blue vehicle is controlled by our policy. The white object is where the controlled agent was in the demonstration data.

in the 10-agent training environment. The results presented in section 4 are extracted by evaluating our policies in the same manner, but on scenes sampled from the held-out testing dataset.

3.1. Experimental Setup

We evaluate our algorithm using the same simulator as is used in the development of PS-GAIL (Bhattacharyya et al., 2018). The simulator allows us to sample initial scenes from real traffic data and then simulate for 5 s at 10 Hz. The most important feature of this simulator is that expert vehicles observed in the real data can be replaced with policy controlled agents, crucial to both learning a good policy and evaluating final policies. We replace 10 vehicles from the initial scene with vehicles driven by the learned policy. Another crucial component of the simulator is the extraction of features from the environment which are then fed into the policy controller as observations. The agent's decisions are translated into actions, which the simulator uses to determine the next state.

3.2. Coding Environment

Moreover, we have been working on updating the entire system (environment, imitation learning training framework, data visualization, etc.) to work with a more up-to-date version of Julia (the programming language). This is not reflected in any other section of the milestone, but we do believe it consists of working on the implementation of our algorithm.

3.3. Evaluation Methodology

We have written scripts to evaluate the imitation performance in terms of local and global properties. Further, we have scripts to evaluate the resulting driving behavior for undesirable metrics as well as emergent properties. These scripts act on the output of simulating data from initial scenes that are sampled from the NGSIM dataset. We calculate the RMSE of the generated trajectories compared to the demonstrations, as well as emergent values such as the number of collisions or total off-road duration. The procedure to generate the necessary simulated trajectories is described here.

First, we sample a random scene from the dataset. We then sample 10 vehicles from the scene, which we will control with the desired policy. If there are less than 10 vehicles present, we pick a new random scene from the dataset. With the 10 policy controlled vehicles, termed agents, we simulate the scene forward, interacting with the environment and expert demonstrations to generate observations for our policy. Each agent then responds with its own action. We repeat this process for the duration of the simulation, saving the trajectory at the end. We also have a visualization script, which generates videos of the policy interacting with the environment. This allows us to qualitatively compare the different policies. By controlling the random seed of the initial scene, we can force the policies to start from the same state. We include a screenshot from the environment in fig. 3.

4. Results

4.1. State and Trajectory Comparison

This section describes the local imitation performance in terms of how well the vehicles driven using our driving models imitate the original demonstration data. fig. 4 shows the root mean square error between the position of the original vehicles in the demonstration data and the the position of the vehicles in the rollouts generated using our trained driving policies. Similarly, fig. 5 describes the imitation performance in terms of the lane relative heading.

In general InfoRAIL performs the worst when it comes to local imitation, and InfoGAIL does slightly better. However, PS-GAIL performs the best with the lowest root mean square error both in terms of value and the rate of growth with time. As expected, the rules-based model does not provide good imitation performance.

4.2. Undesirable Phenomena

Figure 7 describes the undesirable metrics of driving such as collisions, hard deceleration, and driving off the road. In terms of collision instances, the driving policy generated using InfoRAIL performs the worst while PS-GAIL performs

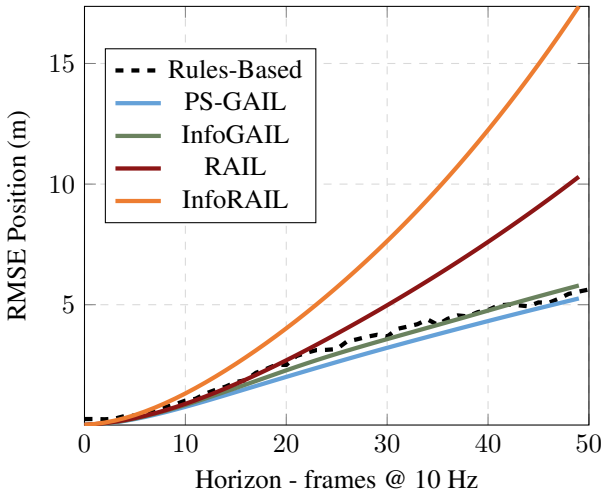

Figure 4. The root mean squared error of the position for 100 trajectory rollouts over 5 seconds. The error is calculated relative to the expert vehicle replaced by our policy-controlled agent.

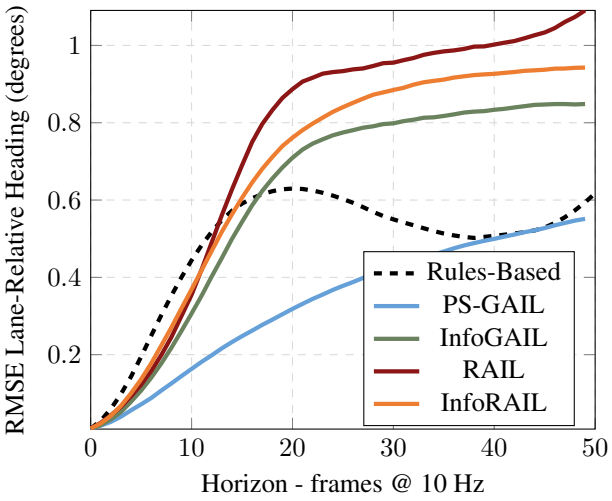

Figure 5. The root mean squared error of the lane relative heading over 100 trajectory rollouts of 5 seconds. The error is calculated relative to the expert vehicle that we replace with our policy-controlled agent.

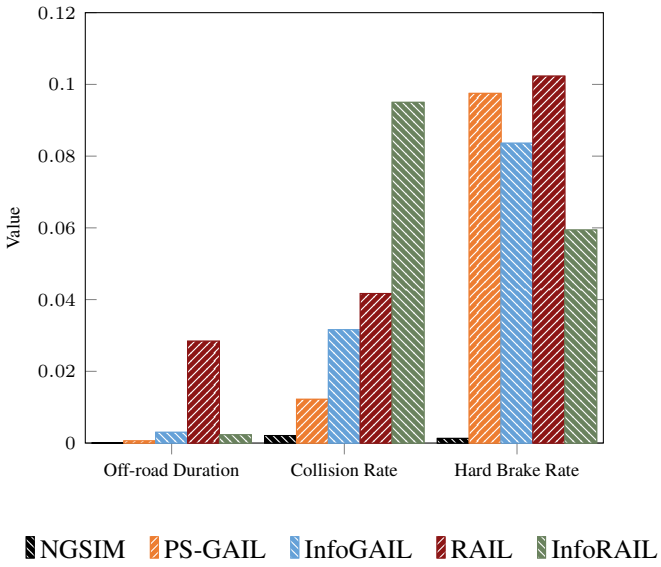

Figure 7. Prevalence of undesirable characteristics that emerge from driving simulations. Lower values are generally better, where a perfect model would exactly match the values from the expert (NGSIM) demonstrations.

the best.

However, InfoRAIL performs the best when it comes to hard-deceleration while RAIL performs the worst. We note that all the values are high as compared to the demonstration data and the rule-based baseline. RAIL again performs the worst in terms of driving off the road while PS-GAIL performs the best. Adding the latent state information does not really help with the undesirable metrics. It is important to note that these are preliminary results based on training using 10 agents for 200 iterations, which is a remarkably short duration. We hypothesize that more training time will improve these metrics and we leave it for future work.

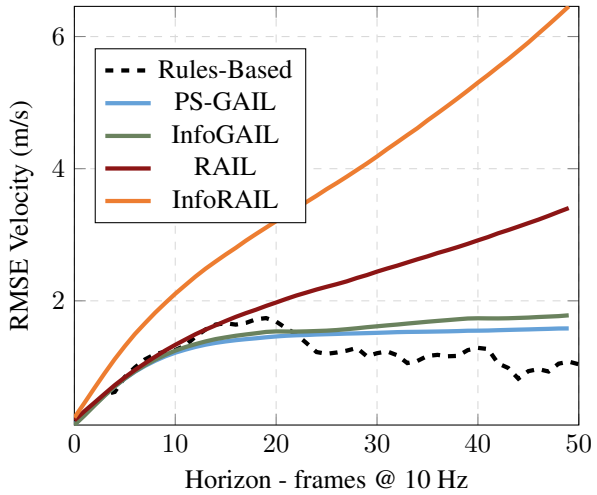

Figure 6. The root mean squared error of the velocity of policy-controlled agents with respect to the expert demonstration that they replaced, over 100 trajectory rollouts to 5 seconds.

5. Conclusions

This project has only scratched the surface of this problem. We have compared four GAIL-based algorithms with a Rules-Based. We believe that significantly more training is needed to validate our conclusions. We also believe that InfoGAIL and InfoRAIL should be trained for longer to fairly compare them against PS-GAIL and RAIL as the expert policy is a mixture of policies defined for different latent factors.

6. Future Work

First of all, it is necessary to train all models for significantly longer to validate the results of this project. We have trained all models for 10 agents. However, we noticed that the training with more agents would improve the performance of our models and should be explored in the future work. Secondly, a hybrid between Rules-based model and GAIL-based model would be interesting to explore. If we are able to differentiate between regular driving and emergent maneuvers that we would be able to use different models depending on the scenario, e.g. use rules-based in regular, calm driving and GAIL in emergency situations.

Moreover, the intention as outlined in our proposal was to

use two datasets, NGSIM US 101 and NGSIM Lankershim Boulevard in Los Angeles. The datasets capture two different driving scenarios: highway traffic and urban traffic, respectively. While vehicle trajectories and the road were successfully imported into our coding environment, we ran into many issues with the traffic signal representation for Lankershim Boulevard.

As a result of these issues, we reached out to the dataset providers to get their assistance and guidance. They, while helpful, described a situation that seems unlikely to succeed for the purposes of this project: we would have to find a "traffic signal guru from the Department of Transportation," who would ideally be able to decode the traffic signal timing sheets included in the NGSIM dataset. Due to the time constraints associated with a class project, we have decided to focus on the NGSIM US 101 dataset for our experiments. However, we hope to make progress towards eventual integration of the Lankershim data for external research.

References

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning (ICML)*, pp. 214–223, 2017.

Bhattacharyya, R. P., Phillips, D. J., Wulfe, B., Morton, J., Kuefler, A., and Kochenderfer, M. J. Multi-agent imitation learning for driving simulation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1534–1539. IEEE, 2018.

Bhattacharyya, R. P., Phillips, D. J., Liu, C., Gupta, J. K., Driggs-Campbell, K., and Kochenderfer, M. J. Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning. *arXiv preprint arXiv:1903.05766*, 2019.

Colyar, J. and Halkias, J. US highway 101 dataset. Technical Report FHWA-HRT-07-030, January 2007.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. Benchmarking deep reinforcement learning for continuous control. *arXiv preprint arXiv:1604.06778*, 2016.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 4565–4573, 2016.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

Kochenderfer, M. J. *Decision Making Under Uncertainty: Theory and Application*. MIT Press, 2015.

Kuefler, Alex, e. a. Imitating driver behavior with generative adversarial networks. 2017.

Li, Y., Song, J., and Ermon, S. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3815–3825, 2017.

Littman, M. L. Markov games as a framework for multi-agent reinforcement learning. In *icml*, pp. 157–163, 1994.

Treiber, M., Hennecke, A., and Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805, 2000.

7. Appendix: Supplementary materials

Rllab environment for learning human driver models with imitation learning we have been working on during this project with video examples:

`https://github.com/sisl/ngsim_env`

Julia package for working with the NGSIM dataset:

`https://github.com/sisl/NGSIM.jl`

NGSIM dataset:

`https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm`