CS 221 Project Final Report: Learning Predictive Models of Human Driving Behavior

Peggy (Yuchun) Wang, Khalid Ahmad, Hashem Elezabi

June 2018

1 Introduction

There has been extensive research on developing self-driving cars, and several companies have been working on deploying autonomous vehicles. Since the ultimate goal of developing these autonomous cars is for them to drive on public streets, they will have to interact with human-driven vehicles for the foreseeable future [4], which motivates the problem of predicting human driving behavior using effective models that work well in practice. When planning for this interaction, a major issue that arises is that human beings are unpredictable and do not behave optimally. The standard approach to interactions with human drivers is to treat human cars as moving obstacles [2], deriving autonomous driving strategies that are highly defensive. An unintended consequence of this approach is that human drivers do not anticipate such defensive driving because of its non-human-like and opaque nature [2], and in many cases respond even more unpredictably. Recent research [2] presented the key insight that an autonomous car's actions will affect what human drivers do in response, creating an opportunity for coordination. Specifically, the proposed idea is to use an approximate human reward function when deciding what actions the autonomous car should take, in addition to the reward function for the autonomous car itself, so that it can account for the effects its actions could have on the human driver's actions and choose its actions accordingly. This new model, which leads to improved coordination between autonomous cars and human cars, is a positive step towards a self-driving car that is better prepared for interactions with human drivers.

A key step towards this goal of improved coordination is learning a good human reward function. We focus on this task in this project. Briefly, we use maximum entropy inverse reinforcement learning (IRL) to learn a human reward function from driving demonstration data we generate using a driving simulator, and we evaluate the learned human model by running experiments comparing a real human driver with an autonomous driver with our learned human reward function as its reward function. We then work on improving the learned human reward function by adding new features, modifying existing features, creating new scenarios to get more diverse data and obtain a richer reward function, and creating a validation method to test different IRL algorithms and weights.

2 Related Work

There has been work on predicting human driving behavior using Bayesian Additive Regression Trees (BARTs) [4], where the authors focus on one specific scenario – whether a human driver would stop at an intersection before executing a left turn – and try to correctly predict the human action. The authors focus on one particular scenario because of the enormous complexity of the task of predicting human driving behavior, which they explain in the paper. This gives an indication that perhaps other tools can be more effective.

One appealing technique for predicting human driving behavior is to use inverse reinforcement learning (IRL) to learn a human reward function. This approach works well and provides long-term prediction of fine-grained driving behavior [3]. Recent work by Sadigh et. al [2] explored this idea. This project was inspired by that work, where the authors created a system where autonomous (robot) cars use IRL to approximate human reward function and add that into the robot reward function in order to have the robot collaborate and influence human driver behavior. Our work extends this research by refining the human reward function using IRL, designing and implementing new features, and designing additional world scenarios for the autonomous cars. The goal of this project is to demonstrate that it is possible to derive a reward function from human driver data that both 1) successfully characterizes what influences the behavior of a human driver under a given scenario and 2) allows us to reproduce human behaviors using a robotic agent placed in the same scenario from which the human driver data was collected.

3 Task Definition

3.1 Introduction

To build an autonomous car model that interacts effectively with human drivers under this new coordination paradigm while still being safe, we aimed to develop a good predictive model for human driving behavior. To learn these models, we ran IRL on data from human driving scenarios. This project entails running multiple scenarios in which, for each scenario, we measure driving statistics of a human agent, a baseline agent utilizing a simple reward function, and an agent utilizing our learned reward function from human driving data. We then compare the driving statistics of all three with the human data serving as our upper bound (oracle) for performance and our baseline data serving as our lower bound for performance.

3.2 Model

The problem is modeled as a linear dynamical system in a fully observable continuous state space, where the agents are the human driver and autonomous drivers. We are using a linear combination of features for the reward function, where we define the reward function R(s) in the following equation:

$$R(s_t) = \sum_{i=1}^{N} \alpha_i \phi_i(s_t)$$

where s_t, the current state, is a 3-tuple containing the current physical state x_t, the human control input h_t, and the robot control input r_t. Features in ϕ are clearly outlined under "Approach." We start with a simple initial configuration of weights qualitatively chosen, which serves as our baseline. Our goal is to optimize these weights to obtain a reward function that can be used to model human driving behavior, which serves as our oracle.

3.3 Input/Output

Intended input is human driving data in the form of trajectories and velocity the driver takes recorded from simulated driving scenarios. This data is saved as a .pickle file, a python object serialization file. Desired output is a vector containing optimal weights that can then be applied to the human reward model that can be used to inform trajectories of a self-driving car. This output reward function is meant to succinctly characterize what influences the actions of a human driver so that an agent is able to choose actions that maximize the resulting reward function and thus, ideally, best match how a human driver would act in a given scenario. An example of input would be a human driver runs a simulation where they drive at a reasonable speed, avoid other agents, and pass them when needed. The resulting measurements taken from the scenario are then used to determine a reward function for a self-driving agent that then is able to maneuver around other agents carefully in a similar manner.

4 Infrastructure

Necessary infrastructure for this project primarily consists of the code base used for Professor Sadigh's paper [2] mentioned under "Related Work." Professor Sadigh's code base consists of code that allows for simulation of driving scenarios, a sample collection of data from driving scenarios run through simulation, and implementation of a driving agent that maximizes a given reward function. We designed driving scenarios to work with the simulator to use for evaluation of different agents in our experiments, which were not built in with Professor Sadigh's code base. Data collected from experiments consists of agent x and y positions, steer, heading, speed, and acceleration. Data is then stored as a pickle file that is used to run Inverse Reinforcement Learning to derive a reward function for features to be designed. We also created a script for parsing the pickle data file and plotting the data. Dynamics for driving agent such as steering controls that determine the path that an agent takes between two points are handled in the original code base and were used as is.

The code for our experiments, visualization, and IRL can be found here, which is a repository forked from the repository containing the original code base from Professor Sadigh's lab.

5 Approach

5.1 Proposed Methods

The main technique used for project is inverse reinforcement learning, used to learn a human reward function that predicts human behavior in driving scenarios. In IRL, instead of choosing what actions to take based on a given reward function, a reward function is derived from given behavior in some scenario based on representative features we design. We run inverse reinforcement learning on driving data that we have generated ourselves using our simulator.

After generating driving data from multiple scenarios, such as the one in Fig. 1, IRL was run to determine the reward function that best described the human driver's actions across the collected scenario data. The determined reward function was then used as the reward function employed by our computer driver. We then tested the abilities of the computer driver with the newly generated reward function to determine if there were any tangible improvements in its driving ability.

For our IRL algorithm, we apply the principle of maximum entropy to define a probability distribution over human demonstrations such that trajectories that have a higher reward are more probable. We achieve this by making the probability of the human action h proportional to the exponential of the rewards encountered along the trajectory, given by our parameterized reward function $R(s)$:

$$P(h \mid x_0, \alpha) = \frac{\exp(R(x_0, r, h))}{\int \exp(R(x_0, r, h'))dh'}$$

This setup makes good human expert actions (from our human driving data) closer to being deterministic, since their probability would be higher corresponding to their higher reward. At the same time, since our human driving data isn't perfectly optimal, this makes less optimal, noisy actions less probable, which partly solves the problem of non-optimal human behaviors affecting our learned human reward function. More details on this can be found here [1]. We then optimize by choosing the weights α_i in the reward function that make the human demonstrations the most likely: $\max_\alpha P(h \mid x_0, \alpha)$.

5.2 Features

Features for our reward function are as follows: ϕ_1 = sum of euclidean distances between the center point of the car to the boundaries of its lane, ϕ_2 = sum of euclidean distance between the center point of the car and the boundaries of the road, ϕ_3 =distance car has travelled in the y direction, ϕ_4 = speed at which car is travelling, and ϕ_5 = sum of euclidean distances between the center point of the car to the boundaries of other cars. ϕ_1 is meant to give a value to how centered the vehicle is in its respective lane. ϕ_2 is meant to give a value to the vehicle staying within the boundaries of the road. ϕ_3 is meant to give value to how far along the road the car has travelled. ϕ_4 is meant to give value to how fast the car is moving along the road. ϕ_5 is meant to give value to keeping a safe distance from other cars.

6 Evaluation

6.1 Baseline

Our baseline is a test driving agent with the same feature values as our agent following our learned human model but whose weight vector are a simple initial set of weights that has not been determined through reinforcement learning, but has instead been qualitatively chosen. We compare our agent's driving performance using our learned weights to the performance of the test agent with these simple weights to confirm that our learned weights are actually meaningful and are indeed responsible for the agent's improved driving ability. Our baseline agent was given the starting weights $\alpha = [1., 50., 10., 10., 60.]$ for its reward function R(s). These weights were chosen to highly reward maintaining distance from other vehicles and staying within the boundaries of the road.

6.2 Oracle

Our oracle is the expert human driver. Since inverse reinforcement learning is trying to imitate human behavior by maximizing the human reward function, we assume that the expert human driver is the ideal driver. In the simulation, our oracle is the data and trajectories collected from the human driver.

7 Results

Selected data from our baseline, oracle/human, and inverse reinforcement learning robot driver is shown below. Our baseline is shown in red, the human driver is shown in blue, and the inverse reinforcement learning robot driver is shown in green. A picture of the start of each world scenario is shown next to the corresponding plots of acceleration, heading, speed, steering, and trajectory. The x-axis unit for all the plots is each time step, except for the trajectory graph, which has the x coordinate on the x-axis and the y coordinate on the y-axis.

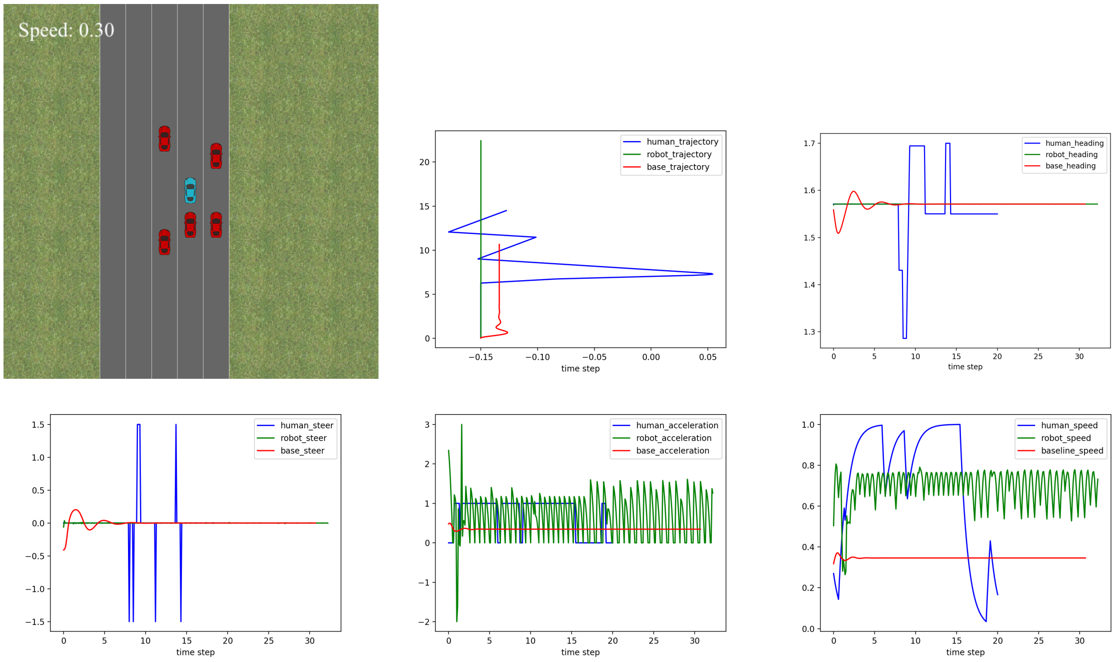

Figure 1: Scenario 1 - Vehicle on 5 lane highway shared with 5 other vehicles

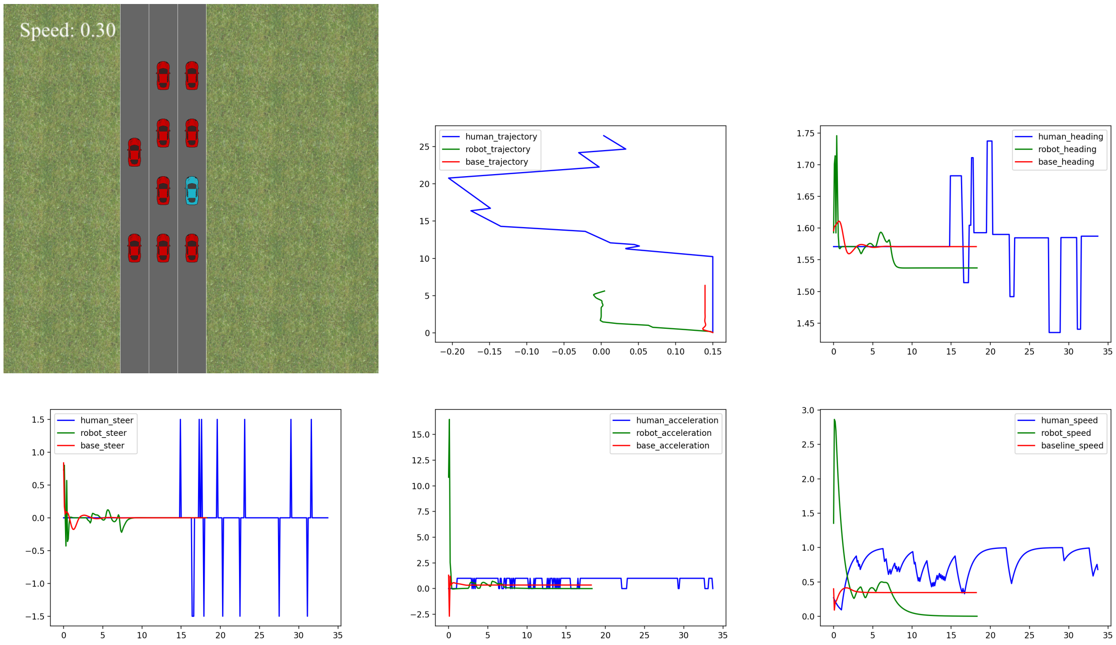

Figure 2: Scenario 2 - Vehicle on 3 lane highway shared with 9 other vehicles

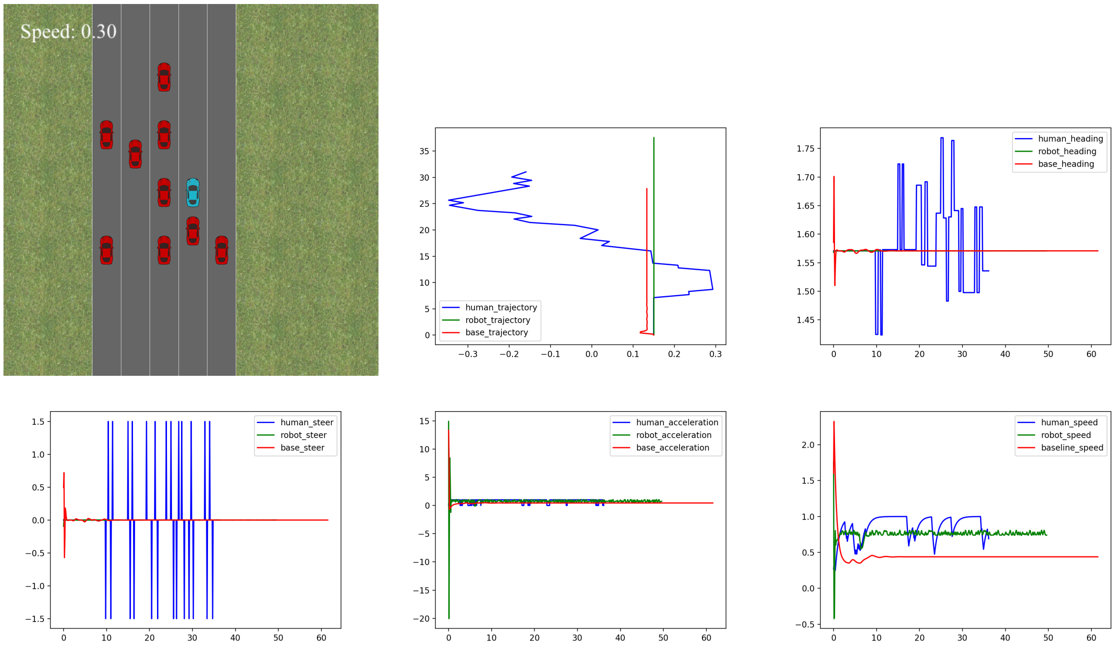

Figure 3: Scenario 3 - Vehicle on 5 lane highway shared with 9 other vehicles

8 Analysis

We qualitatively evaluated the performance of our autonomous agent across three driving scenarios by comparing the graphs of each of its driving statistics (trajectory, speed, steer, acceleration, and heading) to those of our baseline agent and of our human driver.

Our learned model yields behavior different from the baseline agent's, but does not exhibit many similarities with our oracle agent. With values such as speed and acceleration, our learned agent appears to take patterns in the human driver's behavior to the extreme, as can be seen in the plots for scenario 1 and scenario 2. In scenario 2, learned model yields behavior closest overall to that of oracle, exhibiting a similar trajectory, heading, and speed, although curves of learned agent appear to be smoother versions of curves of human agent. In scenario 3, learned model yields behavior closest overall to that of baseline agent, nearly matching the baseline agent's trajectory, heading, and steer.

A qualitative evaluation of the learned behavior shows more human-like characteristics in driving of learned agent compared to that of baseline agent but not significant enough to suggest that learned agent in fact utilizes a human reward function. Stability in learned agent's driving can likely be attributed to that fact that vehicle steering controls, which control how an agent moves from one point to another, are built into simulator and held constant across both the learned agent and baseline agent. Since the human agent does not utilize the simulation's steering controls, it is able to move from lane to lane differently than either robot agent and there is no way for our current system to capture this for our learned agent to then use for itself. Taking this into account, we conclude that our reward function was not enough to capture the complexity of human driving, likely due to its low number of features and lack of complex features.

9 Future Work

While this project has demonstrated that there is potential for deriving a meaningful reward function from human driving data from our given simulation scenarios, there are many opportunities for improving methods for testing our thesis.

Firstly, work can be done to expand our features to include more complex features that allow us to capture finer properties of human driving behavior in our reward function. Our current set of features only allows us to characterize basic driver traits, such as how much a driver values staying within the bounds of a lane, but does not capture, for example, what speed a driver prefers to be travelling when in a specific lane or how fast a driver prefers to be moving relative to the speeds of other vehicles on the road at the same time.

Secondly, our current driving scenarios are not complex enough representations of real world driving scenarios. Use of the RTI automotive simulator to collect human driving data for tests would allow for more detailed driving scenarios to be run for testing and for more data to be collected over a longer period of time. A larger quantity of higher quality driving data should improve chances of being able to derive a reward function through IRL that better characterizes the reward function for human driving behavior.

References

[1] Sergey Levine and Vladlen Koltun. "Continuous Inverse Optimal Control with Locally Optimal Examples". In: *ICML '12: Proceedings of the 29th International Conference on Machine Learning*. 2012.

[2] Dorsa Sadigh et al. "Planning for cars that coordinate with people: leveraging effects on human actions for planning and active information gathering over human internal state". In: (May 2018).

[3] Masamichi Shimosaka et al. "Predicting driving behavior using inverse reinforcement learning with multiple reward functions towards environmental diversity". In: *2015 IEEE Intelligent Vehicles Symposium (IV)* (2015), pp. 567–572.

[4] Y. V. Tan, C. A. C. Flannagan, and M. R. Elliott. "Predicting human-driving behavior to help driverless vehicles drive: random intercept Bayesian Additive Regression Trees". In: *ArXiv e-prints* (Sept. 2016). arXiv: 1609.07464 `[stat.AP]`.