Learning an Agent to Warn the Driver

Venkata Sai Kumar Gadde

Final project, Compsci 687, December 2017

1 Introduction

This project is an implementation of q_learning and sarsa using linear function approximation on the self driving car domain. The self driving car environment is modeled based on the discussions happened for the car safety project.

In brief, the aim of this project is to model a warning agent to warn the driver based on their age, sex, number of previous accidents, alcohol level etc. These parameters are not included in the model explicitly, but are considered implicitly in the model. I modeled an environment with the car and the surrounding agents. The driver navigates the environment with its own principles and rules. The warning agent is trained to understand the behavior of the driver and warn the driver when the driver is not driving according to his/her usual standard.

1.1 Problem Description:

Drivers drive good most of the time and they are bad only a few times. The problem statement considers the fact that all the drivers are good most of the time. This experiments main goal is to study whether an agent can learn the behavior of the driver when they drive better and warn them when they are bad according to their own standards.

Why this problem can be modelled as an MDP problem? To model the problem as an MDP, the state of the agent should follow the Markovian property. Markovian property is memory less, as in the future state depends upon only the current state and is independent of all the previous states. Velocity is Markovian as velocity tells both the direction and also the magnitude of the property and the future velocity is independent of the past stream of velocities given the present velocity. And also the relative position of the vehicle. The future position of the vehicle is independent of all the past stream of velocities given the present velocity and also the present state of the vehicle.

2 Environment

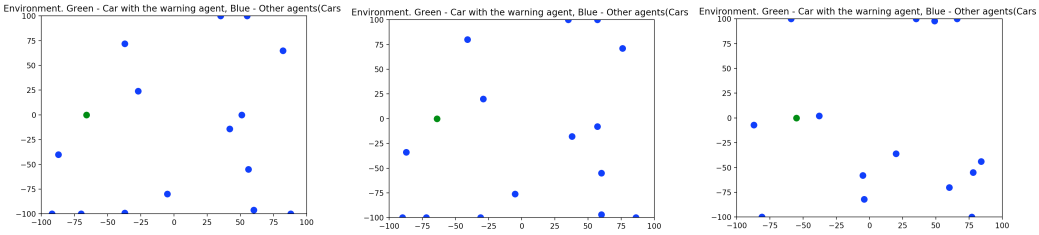


Figure 1. Pictures of the environment. "Green" circle is the car with the warning agent, and the blue dots are the other agents(cars) in the environment.

2.1 Driver behavior

As shown in the figure (1), the driver is the green circle and the driver is interacting with the other agents in the environment. A simple model is adopted for the driver in this model where the driver brakes the vehicle, when there is an agent which is nearer than the threshold value predefined for the driver. Various driver behaviors are experimented with. A very cautious driver is supposed to have higher threshold distance and a reckless driver is supposed to have lower threshold values. The agent is able to learn in all the scenarios as shown in the results section.

2.2 Actions:

The actions for the agent are

1. To not warn the driver
2. To warn the driver

2.3 Rewards:

There are 4 possible cases for the rewards to be given to the agent. They are:

- **Agent:** NotWarn and **Car is braked:** No: Reward is 0
- **Agent:** Warns and **Car is braked:** No: Reward is -1
- **Agent:** NotWarn and **Car is braked:** Yes: Reward is -1
- **Agent:** Warns and **Car is braked:** Yes: Reward is 1

2.4 Dimensions:

A 2 dimensional world is assumed for this model. The dimensions of the environment are 200 units in the X direction and 200 units in the Y direction. The environment is within the limits (-100,100) in both X and Y directions.The agent starts at $(X, Y) = (-100, 0)$ with constant velocity $V_x, V_y = (1, 0)$. For all the other agents, the starting positions are given at random and the velocities are chosen at random between $(-5, 5) units/time\_step$.

2.5 Mathematical model of the environment

Linear kinematic model is assumed for the environment. Assumptions include:

- Constant velocity for all the agents. Acceleration of all the vehicles is considered as zero.
- All the agents move in a straight line rather than the real life cases like turning or braking.
- The motions of all the agents in the environment are independent of each other.
- The car moves in a straight line and the other agents also move in a straight line, but are given velocities in 2 dimensions.

Mathematical kinematic equation for the motion model.

$$X_{t+1} = X_t + V_x * time\_step$$
$$Y_{t+1} = Y_t + V_y * time\_step$$
$$X_t = Position\_of\_the\_car$$
$$V_x = XComponentofthevelocity$$
$$V_y = YComponentofthevelocity$$

3 Learning

Function Approximation: For estimating the q state action value for a given state and action, linear approximation is used. The warning agent present in the car gets the data from the vehicle sensors, which are given by the environment in this model. Features for the warning agent are: relative positions and relative position of the other vehicles in the environment in the X and Y dimensions. As there are 2 actions that can be taken by the warning agent, the total number of features are:

(Number of actions)*length of (Number of features per agent in the environment)*(Number of agents)

Episode: An episode is from the start to the first time the agent brakes the vehicle. Once the driver brakes the vehicle, new episode is started. The maximum number of time steps in an episode are 200.

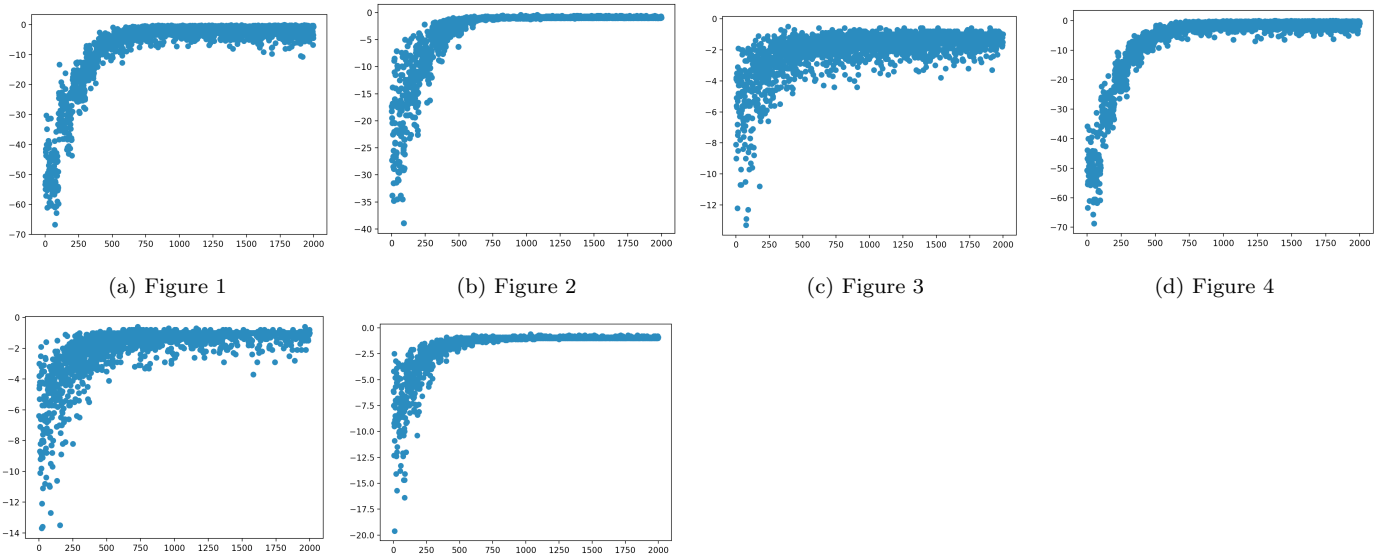
4 Hyper parameters

The model has multiple hyperparameters starting from the environment variables to the learning hyper parameters. Hyper parameters include: There are multiple sets of hyper parameters in the model. One set is within the simulation of the learning environment like 1. Number of agents 2. Kinematic model of the environment etc.
The other set of hyper parameters include the parameters related to the learning model. 1. Learning rate, epsilon, epsilon-decay.

5 Experiments and Observations

The experiments and observations include the learning of the warning agent in the environment using SARSA and Q_Learning. Number of trails is equal to 10 and the number of episodes in each trial is equal to 2000.

Fig.No	Number of agents	Learning Rate	Method	Driver threshold	Converged
1	15	1e-3	SARSA/Q_learning	2	Yes
2	15	1e-3	SARSA/Q_learning	10	Yes
3	15	1e-3	SARSA/Q_learning	15	Yes
4	2	1e-3	SARSA/Q_learning	15	Yes
5	30	1e-3	SARSA/Q_learning	15	Yes



(a) Figure 1 (b) Figure 2 (c) Figure 3 (d) Figure 4

(e) Figure 5 (f) Figure 6

Figure 1: Results for application of SARSA on the environment with parameters mentioned in the table. Figure 6 is the result of q_learning for illustration.

6 Observations

1. It is observed that the model converges for all kinds of drivers (different driver thresholds).
2. It is observed that the model converges slower in the case of larger number of surrounding agents. As the number of surrounding agents are more, the episodes are shorter and therefore lead to slower convergence.
3. It is observed that the Markovian assumption for relative velocities and relative position features is reasonable.

7 Conclusion

Reinforcement learning can be a very good application in the self driving cars for motion planning. It is assumed that the driver uses a very simple rule to drive the car(distance). But the agent is able to learn the complex patterns from the relative positions and the relative velocities. I would like to explore application of reinforcement learning in the self driving car domain using simulated environments (for example in ROS).