Autonomous braking and effective warning systems

Daniel Sam Pete Thiyagu
College of Information and Computer Sciences
University of Massachusetts Amherst
{dthiyagu}@cs.umass.edu

I. Problem Statement

I looked at two specific problems as they were interlinked with one another. The first one is autonomous braking [Chae et al., 2017] where there are sensor information about the position of the obstacle ahead, as well as the system knows about the position and velocity of the car. I wanted to look at whether the agent(autonomous driver) will be able to learn how to accelerate and stop when needed.

The second problem relates to analyzing the effectiveness of warning systems modeled on users. The warning system is designed to analyze the user and is assigned a certain category: Cautious driver, Moderately Cautious driver, irresponsible driver. A cautious driver would want a warning as soon as the system knows of an obstacle. A moderately cautious driver would want a warning as it nears within a safe distance from the obstacle. An irresponsible driver would want it as it approaches really close to the obstacle. All these driver have different preference ranges for time to crash warning signals. If our warning system is modeled on such agents, would it be effective in a real world scenario. In particular we look at whether the driver(agent) is able to respond effectively to the alarm timings and achieve a good average reward over time and complete the episode in a small amount of time steps.

II. Related Work

[Chae et al., 2017] is related to designing brake control and searching for an optimal policy in Markov decision process (MDP) model where the state is given by the relative position of the obstacle and the vehicle's speed, and the action space contains no braking, weak, mid and strong braking actions. The policy used for brake control is learned through computer simulations using the deep reinforcement learning method called deep Q-network.

[Abe and Richardson, 2006] is related to a study that is related to the effect of alarm timing on driver trust and behavior with a Forward Collision Warning System (FCWS). In this driving simulator experiment three different kinds of alarm timing (late/middle/early) were compared with respect to driver braking strategy and driver trust.

III. Technical Approach and Model For Autonomous braking

A. MDP Model

Environment:

The Agent has to drive the car from start 0m to end 125m. It starts off at position 0. The episode ends when the car has crossed 125 m or when the number of timesteps has crossed 200 or when it has crashed. Each episode will contain of only one obstacle. The agent will only be able to view the obstacle if it comes within 40 m of the obstacle.

State : (position_pedestrian, position_car, velocity_car)
Actions : 1) Stay in same velocity 2) Accelerate 3) Brake. Acceleration is done at 20% of previous velocity, if there is no previous velocity it sets velocity to 10meter/s. It is capped at 30meter/s. Initial velocity is 10m/s.

Reward function : If the obstacle is visible and the car's position has crossed or is equal to the position of the obstacle, then the reward is -3000, ie. a crash happens. For each time step the reward is -1. If the agent(car) brakes when it sees the obstacle it achieves a reward of 10.

B. Experiments :

The following hyperparameters were fixed for all the below experiments .

The hyperparameters for QL are Fourier basis order=2, $\alpha = 0.05, \gamma = 1, \epsilon = 0.1$,eps_cnt=200, trials = 10 . The hyperparameters for SARSA are Fourier basis order=2, $\alpha = 0.05, \gamma = 1, \epsilon = 0.01$,eps_cnt=200, trials = 10 .

Experiment 1: For this experiment, the Obstacle doesn't move, so the agent has to learn to stop at the obstacle once it comes into view. Since a reward of 10 is given every time a break is applied when obstacle is in view, we get a positive out come, and SARSA works better. Q Learning and SARSA with Linear Function approximation were tried and the results are shown in 1 and 2. Since Qlearning takes more greedy actions than SARSA, the results were in favor of SARSA. SARSA does pretty well and achieves almost maximum reward.

Experiment 2: For this experiment, the Obstacle moves away as soon as the agent brakes. So the agent has to learn to stop at the obstacle once it comes into view. Obstacle is placed at 75m. Q Learning and SARSA with Linear Function approximation were tried and the results are shown in 3 and 4. They do pretty well and achieve almost maximum reward.

Experiment 3: For this experiment, the only difference between experiment 2 and this is that the position of the Obstacle can be random and ranges from 45 to 105 meters. The agent has to learn to stop at the obstacle once it comes into view. It disappears if the agent breaks/stops. Then at the next time step it crosses and is no more in the visible region

of the agent. Q Learning and SARSA with Linear Function approximation were tried and the results are shown in 5 and 6. They do pretty well and achieve almost maximum reward. **Analytics:** You can view the results of how the agent is doing in certain trials in the link https://github.com/danielsamfdo/carSafetyStats/tree/master/notebooks/braking_system/experiments/ or in the folder braking_system.

You can also view graph plots in the Appendix - Graphs

Reproducing the Experiments: To reproduce the experiments please follow https://github.com/danielsamfdo/carSafetyStats/wiki/Experiments---Autonomous-Braking-Systems

IV. Technical Approach and Model For Analyzing Effectiveness of Warning Systems Modeled on drivers

Motivation: Given a user we try to determine whether he is a cautious driver, moderately cautious driver or an irresponsible driver. This can be determined based on alcohol intake/addict, drug intake/addict, age group, number of occupants(since if there are more people in the car, he is likely to be distracted), experience of the driver. Let us assume we can model the driver's preference to the warning system and the warning systems acts on drivers preference.

You can never guarantee that for a given type of driver, he can specify the requested time to crash for a warning system. He can not set his preferred "warn me when time to crash is x sec" as it is dependent on the distance and velocity and sensors are always based on distance, ie. You may be able to detect any collision object only within a particular distance. Each type of driver has a preferred distance metric, warn when distance is D. D varies for each type of driver. Therefore this models close to a real world scenario.

I have described types of drivers to be cautious, moderately cautious or irresponsible. If the user is cautious, the warning is signalled as soon as it is detected. For the purpose of these experiments, i have set the detection distance D of the system as 50 meters. The minimum velocity is 10m/s and max velocity is 30m/s. So for the cautious user, the ranges of time to crash will lie in the range (50/30, 50/10) ie. (1.6s,5s). For the moderately cautious user, the detection distance D is 30 m, the ranges of time to crash will lie in the range (30/30, 30/10) ie. (1s,3s). For the irresponsible user, the detection distance D is 10 m, the ranges of time to crash will lie in the range (10/30, 10/10) ie. (0.33s,1s).

A. MDP Model

Environment: The Agent has to drive the car from start 0m to end 125m. It starts off at position 0. The episode ends when the car has crossed 125 m or when the number of timesteps has crossed 200 or when it has crashed. Each episode will contain of only one obstacle. The agent will only be able to view the obstacle if it comes within a distance "D" of the obstacle. reward of 10 is given if car brakes for obstacle.

The state also additionally involves type of driver, which is named as trust.

State : (position_pedestrian, position_car, velocity_car, trust)
Actions : 1) Stay in same velocity 2) Accelerate 3) Brake.
Reward Function is similar to the previous setting.

We can model the driver based on certain attributes like experience, previous crashes, no of people in the car(likely to be disturbed), age group and find out what type of driver he is in general. This is the value of trust in our state.

B. Experiments:

Each episode ends at 200 timesteps or when the car crosses the position 125m or when the car crashes into the obstacle. We measure the effectiveness of the alarm signal based on the trust, and we do it on the basis of timesteps and rewards. We reward the agent +10 if agent stops when it sees the obstacle. We reward the agent -3000 if the agent hits the obstacle. The obstacle can be anywhere in the position between 45m to 105m. It disappears once vehicle stops. Each episode consists of one obstacle. Given attributes of the agent/driver, we model a trust setting.

The following hyperparameters were fixed for all the below experiments .

The hyperparameters for QL are Fourier basis order=2, $\alpha = 0.05, \gamma = 1, \epsilon = 0.1$,eps_cnt=300, trials = 10 . The hyperparameters for SARSA are Fourier basis order=2, $\alpha = 0.05, \gamma = 1, \epsilon = 0.01$,eps_cnt=300, trials = 10 . Qlearning Agent is set to take greedy actions 10% of the time, while the SARSA Agent is set to take greedy actions 1% of the time.

Analyzing Effectiveness: I defined failure to be the number of instances in 3000 episodes, where you have hit an obstacle or where the time steps to end is greater than 50 timesteps. The max time steps possible is 200, but since the track is only 125m, it should be able to comfortably finish the track in less than 10s. I also define crash percentage to be the percentage of crashes in 3000 episodes. We assume the agent will be able to effectively learn how to complete the track with a lower failure percentage as well as low crash percentage. Since SARSA takes greedy actions less frequently than the QLearning agent, we would get better performance. This was done just to do contrast analysis.

If the agent is able to learn effectively with low failure and crash rates if the warning system is based on user preferences, then we can assume such a system would work well in a real world scenario. The following experiments are designed to test out the above hypothesis.

Experiment 4: Trust of all episodes is 0, which means it's a cautious driver.

As you can see in this experiment in SARSA in Table II, where it takes greedy actions less frequently than QLearning(Table I), the warning system based on user preferences if applied on cautious users has a lower crash rate and failure rate. This is what was expected. We expect this in a real world scenario when timings of alarms is well ahead which gives a safe cushion to the cautious driver, crash rates will be lower.

TABLE I: Avg Reward and Time Step for QLearning

	Avg Reward	Avg Time	Crash	Failure
Cautious Agent	-47.92	38.85	0.6%	21%
Moderately Cautious Agent	-101	31.64	2.6%	17%
Irresponsible Agent	-339	69.08	9.2%	52.3%
Model trained on all Agents	-187	57.02	4.6%	38%

TABLE II: Avg Reward and Time Step for SARSA

	Avg Reward	Avg Time	Crash	Failure
Cautious Agent	-1.77	12.77	0%	0.6%
Moderately Cautious Agent	-20.39	21.28	0.3%	3.6%
Irresponsible Agent	-281.0	118.32	5.6%	63%
Model trained on all Agents	-81.97	70.74	0.6%	32%

Experiment 5: Trust of all episodes is 1, which means it's a moderately cautious agent.

As you can see in this experiment in SARSA in Table II, where it takes greedy actions less frequently than QLearning(Table I), the warning system based on user preferences if applied on cautious users has a lower crash rate and failure rate. We can see the increase int time steps and decrease in average reward from cautious driver which was expected. We expect this in a real world scenario when the timings of alarms are moderate, crash rates will be lower around 0.3% in Table II.

Experiment 6: Trust of all episodes is 2, which means it's a irresponsible agent. Qlearning Agent is set to take greedy actions 10% of the time, while the SARSA Agent is set to take greedy actions 1% of the time. The other hyperparameters are Fourier Basis order$= 2, \alpha = 0.05, \gamma = 1, \epsilon = 0.01$, eps_cnt$= 300$ trials = 10.

As you can see in this experiment in SARSA in Table II, and QLearning(Table I), the warning system based on user preferences has a high crash rate and high failure rate. The increase in time steps and decrease in average reward are an indicator of that, and we can expect that it will be the case in a real world scenario, if there are untimely warnings, and a warning system designed for irresponsible drivers should not be deployed in a real world setting.

Experiment 7: Trust of all episodes is randomly chosen between 0,1,2, which means it can be any type of driver.

As you can see in this experiment in SARSA in Table II, and QLearning(Table I), the warning system based on user preferences has a high failure rate. The model is preferring to do nothing and take a longer time to finish the episode. If there are a mix of timely and untimely warnings based off the preference of users, the driver/agent who are irresponsible might end up in more failures/crashes. So the results seem to indicate that a warning system based on irresponsible drivers seem to increase failure by a huge margin.

Analytics: You can view the results of how the agent is doing in certain trials in the link https://github.com/danielsamfdo/carSafetyStats/tree/master/notebooks/warning_system/experiments/ or in the folder warning_system

You can also view graph plots in the Appendix - Graphs - Warning System

Reproducing the Experiments: To reproduce the experiments please follow https://github.com/danielsamfdo/carSafetyStats/wiki/Experiments---Based-on-Trust-of-User

C. Conclusion:

The results seems to conclude that warning systems can be based off preferences of users/drivers like cautious and moderately cautious drivers, but the warning system should not take into account irresponsible drivers in which case the system would be worse, with a high increase in crash rates.

References

Genya Abe and John Richardson. The influence of alarm timing on driver response to collision warning systems following system failure. *Behaviour & Information Technology*, 25(5):443–452, 2006. doi: 10.1080/01449290500167824. URL https://doi.org/10.1080/01449290500167824.

Hyunmin Chae, Chang Mook Kang, Byeoungdo Kim, Jaekyum Kim, Chung Choo Chung, and Jun Won Choi. Autonomous braking system via deep reinforcement learning. *CoRR*, abs/1702.02302, 2017. URL http://arxiv.org/abs/1702.02302.

D. APPENDIX - GRAPHS

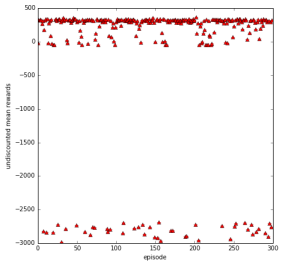

Fig. 1: Exp 1 - Qlearning

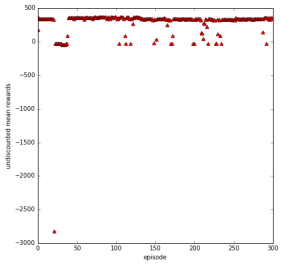

Fig. 2: Exp 1 - SARSA

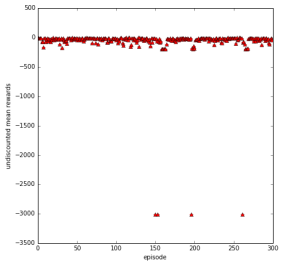

Fig. 3: Exp 2 - Qlearning

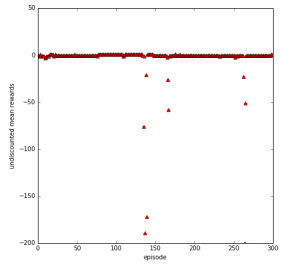

Fig. 4: Exp 2 - SARSA

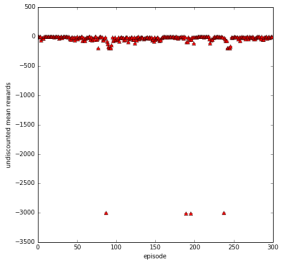

Fig. 5: Exp 3 - Qlearning

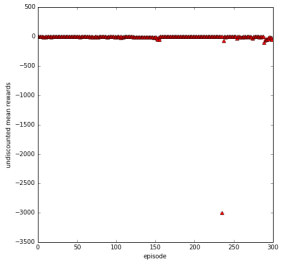

Fig. 6: Exp 3 - SARSA

E. APPENDIX - GRAPHS - WARNING SYSTEM

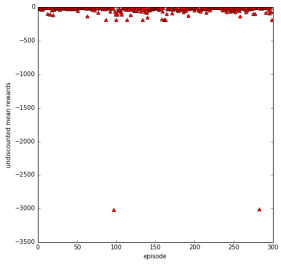

Fig. 7: Exp 4 - Qlearning Reward

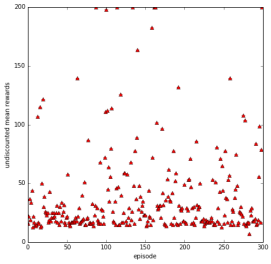

Fig. 8: Exp 4 - Qlearning - Time Steps on Y axis

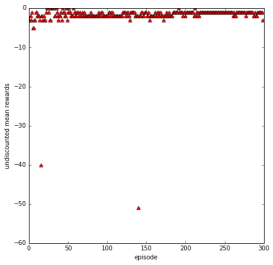

Fig. 9: Exp 4 - SARSA Reward

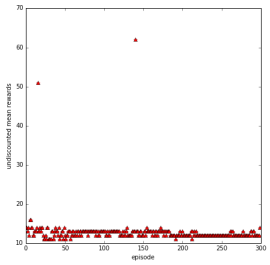

Fig. 10: Exp 4 - SARSA - Time Steps on Y axis

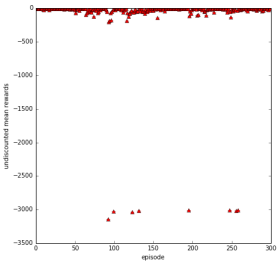

Fig. 11: Exp 5 - Qlearning Reward

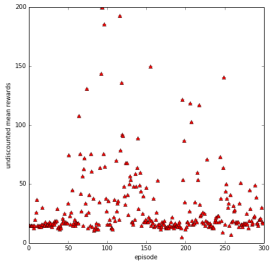

Fig. 12: Exp 5 - Qlearning - Time Steps on Y axis

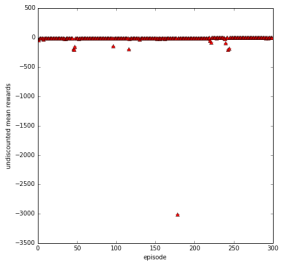

Fig. 13: Exp 5 - SARSA Reward

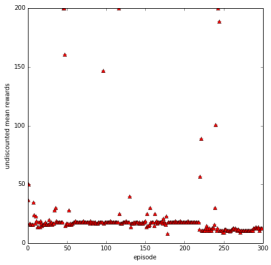

Fig. 14: Exp 5 - SARSA - Time Steps on Y axis

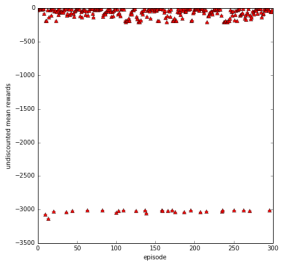

Fig. 15: Exp 6 - Qlearning Reward

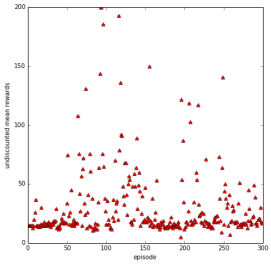

Fig. 16: Exp 6 - Qlearning - Time Steps on Y axis

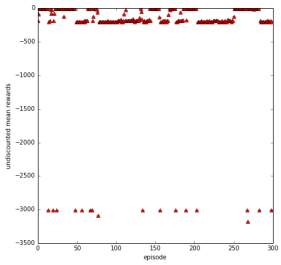

Fig. 17: Exp 6 - SARSA Reward

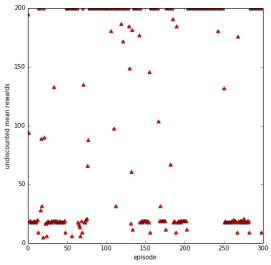

Fig. 18: Exp 6 - SARSA - Time Steps on Y axis

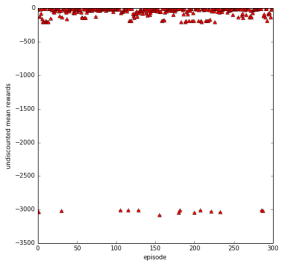

Fig. 19: Exp 7 - Qlearning Reward

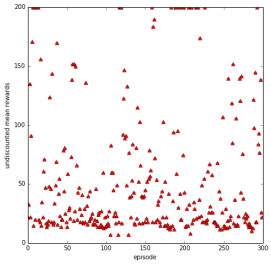

Fig. 20: Exp 7 - Qlearning - Time Steps on Y axis

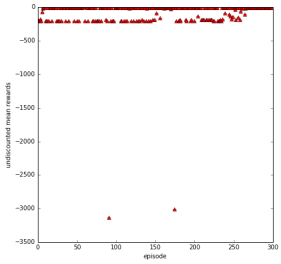

Fig. 21: Exp 7 - SARSA Reward

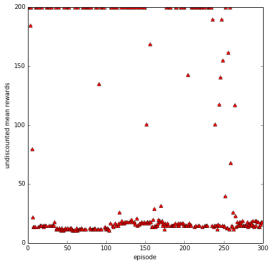

Fig. 22: Exp 7 - SARSA - Time Steps on Y axis