CS 3600 Project 2 Wrapper

CS 3600 - Fall 2023

Due October 13th 2023 at 11:59pm EST via Gradescope

Introduction

This Project Wrapper is composed of 4 questions, each worth 1 point. Please limit your responses to a maximum of 200 words. The focus of this assignment is to train your ability to reason through the consequences and ethical implications of computational intelligence, therefore do not focus on getting "the right answer", but rather on demonstrating that you are able to consider the impacts of your designs.

Context

Reinforcement learning is a powerful technique for problem-solving in environments with stochastic actions. As with any Markov Decision Process, the reward function dictates what is considered optimal behavior by an agent. Since a reinforcement learning agent is trying to find a policy that maximizes expected future reward, changing when and how much reward the agent gets changes its policy.

However, if the reward function is not specified correctly (meaning rewards are not given for the appropriate actions in the appropriate states) the agent's behavior can differ from what is intended by the AI designer. Consider the boat racing game pictured above. The goal, as understood by people, is to quickly finish the race. Humans have no difficulty playing the game and driving the boat to the end of the course. However, when a reinforcement learning agent learns how to play the game, it never completes the course. In fact, it finds a spot and goes in circles until time runs out. You can see the RL agent in action in this video: https://youtu.be/tlOIHko8ySg.The agent's reward function is the score the player receives while playing the game. Score is given for collecting power-ups and doing tricks, but no points are given to players for completing the course.

Question 1

Watch the video and explain why the agent's policy has learned this circling behavior instead of progressing to the end of the course like we expect from a human player. Explain the behavior in terms of utility and reward.

Answer: From the video, we can see that every time the boat goes in a circle, it knocks down 3 green objects in the water which builds up points. Since the agent's reward function is based off of points alone (and not time or even completion of the race), it can just keep racking up points in this circle loop, and never actually complete the race. In this case, the utility would be this current position of these green objects and the wall that allows the boat to loop in a circle since this guarantees the highest reward to work ratio.

Question 2

When humans play, the rules for scoring are the same. Why do humans play differently then, always completing the course? Why don't humans circle in the same spot in the course endlessly if they are receiving the same score feedback as the agent?

Answer: There are a lot of reasons why a human wouldn't circle in the same spot for guaranteed points, but the main reason would be that it's not reasonably possible for a human to do it endlessly. Eventually, they would tire out and just finish the race or quit the game. The purpose of the game itself to finish the race, and that doesn't necessarily mean that higher points translate to a higher win, it's just supplementary. If the purpose was to get as much points as possible, then it's not a racing game anymore, which defeats the entire purpose of the game.

Question 3

The agent's original reward function is:

$$R(s_t, a) = game\_score(s_t) - game\_score(s_{t-1})$$

Describe in terms of utility, reward, and score **two** ways one could modify the reward function to get the agent to behave more like a human player. That is, what do we need to change to make the agent complete the course every single time? Assume the agent has access to state information such as the position and speed of the boat and all rival racers, but we cannot change how the game itself provides scores through the call $game\_score(s_t)$.

Answer: One way we could modify the reward function is considering the distance of the player boat to the goal position. The reward gets higher as the boat gets closer to the goal. This would motivate the agent to increase the speed and consider utility positions that would help the boat get closer to the goal. Another way to modify the reward function would be to take into account how many boats the player is ahead of during the race. In a racing game, you'd obviously want to be in first place, so we can look at this as how many boats we are ahead of. The more boats we're ahead of, the higher the reward. Thus, the agent is more inclined to speed up and consider utility positions that would put it ahead of other players. And the agent would naturally follow in the path of the goal since all the other racers are headed there.

Question 4

Self-driving cars do not use reinforcement learning for a variety of reasons, including the difficulty of teaching RL agents in the real world, and the dangers of a taxi accidentally learning undesired policies as we saw with the boat game example. Suppose however, that you tried to make a reinforcement learning agent that drove a taxi. The agent is given reward based on how much fare is paid for the ride, including tips given by the passenger. Describe a scenario in which, after the taxi agent has learned a policy, the autonomous car might choose to do an action that puts either the rider, pedestrians, or other drivers in danger.

Answer: If the agent's reward was based on factors like ride fare and tips, one potentially dangerous situation would be choosing the fastest and shortest distance path to the destination. This would be a path that does not take traffic rules and other cars into regard, so it would just speed as fast as possible in the direction of the destination, which more likely than not would be off road. Clearly, anything that is off road isn't meant to be driven on, and therefore poses a threat to whatever is there (people, buildings, structures, etc). But the agent doesn't care about this, since it thinks shortest time to get there = highest tips. So this clearly is dangerous for everyone involved (pedestrians, other drivers, even the rider themselves).