Machine Learning Project 5

Anthony Vasquez

Ph:313-920-8877

Email: avasque1@jh.edu

**Editor:** Anthony Vasquez

Abstract

This document investigates the implementation of three reinforcement learning algorithms.

**Keywords:** Reinforcement Learning (RL), Markov Decision Process (MDP), Value Iteration, state-action-reward-state-action (SARSA), Q-learning, On-policy, Off-policy, Action, State,

# Introduction

Reinforcement learning is used in many fields to tackle fast changing environments. Some of the use cases are self-driving cars, self-driving air vehicles, product ordering, automated forklifts, games like chess, and Go. These use cases are all challenging problems that require adaptive inferencing which is what RI tends to do better than supervised and unsupervised models. **Hypothesis**: The Q-learning method is likely to outperform SARSA and VI methods when comparing visual updates of the agent’s interactions with the environment.

## Data Used

The datasets used for experimentation are made up of three racetracks to include: R, O, and L configurations. The name of each of the racetracks is the configuration of each “#” and “.” Symbols. If the dataset is an R of example, then the racetrack is the shape of an “R” (**Figure 1**).

Figure 1. The R shaped racetrack below is an example of one of the datasets used in these experiments.

Background pattern

Description automatically generated

## Data Preprocessing

All datasets underwent some combination of adding and subtracting rows and/or columns to create a square matrix structure to make the datasets easier to handle during training. In addition, commas were added to the track to make it easier for pandas to read into a data frame.

# Algorithms

This section discusses the machine learning algorithms used throughout this experiment.

## Value Iteration

The value iteration estimates the optimal state values by iterating through each state of the environment and updating the value estimates.

Equation 1. Value Iteration Algorithm (Geron, 2019)

for all s

In Equation 1, is the transition probability of the agent act (a) from s to . is the reward obtained by the agent when acting (a) from s to . Finally, is the discount that dampens the effects of states that are less localized. The value of for this experiment was set to 0.9. A value of 1.0 means that each reward for taking an action (a) is equal no matter the distance, so multiplying a discount factor of 0.9 allows a value decay for further states.

## Q-Learning

The Q-learning algorithm can be used when the initial transition probabilities. The underlying technique used by Q-learning is to allow the agent to watch a random generator play while incrementally improving its policy with each time step. This algorithm is known as an off-policy method, since it does not use its current policy for best next action determination.

Equation 2. Q-Learning Algorithm (Alpaydin, 2020)

In Equation 2, is the previous optimal Q value, is the learning-rate, is the discount rate, is a decreasing coefficient, and is a sample of instances for each pair. The used for these experiments was 0.9 with an of 0.05.

## SARSA

The SARSA algorithm is an on-policy algorithm that uses the current policy for best next action determination. It uses the algorithm of Equation 2 with some modification seen in

Equation 3. The SARSA update algorithm (Alpaydin, 2020)

In Equation 3, is the updates Q value and can be seen on both sides of the equation. is a gradually decreasing coefficient, is the temporal error at time t, and is the eligibility decay that is one for all states except when and a = .

# Results

This section discusses the metrics used in obtaining the results for each RL network.

## Value Iteration Results

The value iteration results showed that for the L-track, the predetermined termination threshold of 0.005 was reached at 13 training iterations leading to a state value heatmap seen in Figure 2.

Figure 2. State value heatmap of L-track resulting from using value iteration algorithm after 13 train iterations.

Chart

Description automatically generated

The value states for the O-track with the same hyperparameter values converged at 12 training iterations. The state value heatmap can be seen in Figure 3.

Figure 3. State value heatmap of O-track resulting from using value iteration after 12 train iterations.

Chart

Description automatically generated

Figure 4. State value heatmap of R-track resulting from using value iteration after 13 train iterations.

Chart

Description automatically generated

## SARSA Results

This algorithm is mostly built and can be found in algorithms/sarsa.py, but I had a hard time with the implementation, so there are not results to show for SARSA.

## Q-Learning Results

This algorithm is also mostly built, more so than SARSA. It can be found in algorithms/qlearning.py, and like the previous algorithm, I had a hard time with the implementation, so there are no results to show for QLearning.

# Conclusion

Though I was not able to prove or disprove my hypothesis, I found the study of reinforcement learning to be the most challenging area of study of the semester. The update rules did not strike me as intuitive which leads me to believe that I need to spend much more time on the subject to fully grasp the concepts. With that said, I also found the project problems to be fun, which I was not expecting. My undergraduate degree was in physics so using the simplified kinematic functions was interesting to say the least. I look forward to learning more about reinforcement learning in the future. Thank you for the great semester!

# References

Alpaydin, E. (2020). In E. Alpaydin, *Introduction to Machine Learning.* Cambridge: The MIT Press.

Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow.* Canada: O'Reily.