9/30/2021

# Assignment 5: Reinforcement Learning (RL)

## The instructions—about this Word file template, the evaluation matrix, no more than 4 pages, and appendices—are as before.

## Provide figures if doing so contributes to clarity and demonstrating your understanding.

## What AItic Will Learn

Suppose that you want to build an AI tic-tac-toe player (we’ll call it AItic) which knows only the rules of the game but needs to learn how to win via RL. You plan to carry this out by repeatedly playing one copy of AItic against another.

In one or two sentences, describe concretely what kind of thing AItic would learn.

The AItic agent will learn by considering its opponent to be a part of the larger environment it is interacting with because, unlike min max problems were opponents make the most optimal movements when we integrate alpha-beta pruning, RL does not assume the opponent will make optimal moves. Of the two types of rewards mentioned in lecture, immediate vs end-state, this agent will learn through end state reward, and for my specific implementation it will use e-Greedy policy.

## AItic Code Format

Provide a Python code fragment showing the data structure for your answer in Part 1. For example, *this list:* or *this class: …*, etc.

.Text

Description automatically generated

Here, the AITicPlayer class houses a unique player ID, their exp rate (defaults to 0.4), the states and their values in a list and a dictionary, and the player’s learning rate, which is 0.2 as that is what I have found was most widely used.

## The RL Process for AItac

In 3 sentences excluding figures, and as concretely as you can, describe the reinforcement learning process for AItic.

As the game progresses, the state of the board will constantly be changing with player’s ids populating the squares in the tic tac toe board in place of Xs and Os. While the value of particular states update throughout the game, the reward will not be delivered until the end of the game when a winner is determined: 1 to the winner and 0 to the loser or 0 to both in the case of a draw. The default exp-rate is set to 0.4, meaning the players will take the greediest option 60% of the time when deciding where to move next.

Continue with Parts 4 and 5 below.

(If you are *very* ambitious and are willing to take *a lot* of time by the assignment deadline, you have the option to implement 1 through 3 above instead.)

## An Application Described

In 4 sentences or fewer, excluding figures, concretely describe an application not in the literature that you would solve with reinforcement learning. Point out what would be learned and why RL would be appropriate.

In my Data Science in Python course here, I worked with recommendation systems, which seems like a great application for reinforcement learning: the content the user consumes is the state, the action is the next piece of content a user is recommended to consume, and the reward is some measure of customer/user satisfaction (ex: a like or a positive review). If Markov decision process properties are incorporated, then each step in the recommendation chain can be completed independently. This aids in the RL advantage of exploration which is not seen in other learning strategies, since the program could occasionally try out new content for the user to test if they enjoy it or not. The has plenty of business applications as well since companies could learn what users like most on their platform and can help increase user engagement and satisfaction.

## The RL Process for Your Application

In 3 sentences or fewer, excluding figures, describe the reinforcement learning process for your application described in Part 4, as concretely as you can but not using code. You may want to consider increasing levels of learning sophistication.

The learning process would be straightforward: the users are part of the environment who interact with the platform’s content. When users interact with this content, they change their state and the program recommends new content, which the user can engage with in some positive, negative, or neutral capacity. When something positive occurs, like a purchase, a good review, or long time of user engagement, then the agent which contains the RL program, receives a reward from the environment, enabling it to continue to learn and provide users with good recommendations (while occasionally including exploration and providing new recommendations to test).

# Evaluation



# Appendix

References:

* <https://www.cs.dartmouth.edu/~lorenzo/teaching/cs134/Archive/Spring2009/final/PengTao/final_report.pdf>
* <https://towardsdatascience.com/recommendation-system-with-reinforcement-learning-3362cb4422c8>
* <https://medium.com/nerd-for-tech/how-will-reinforcement-learning-based-recommendation-system-be-in-the-future-part-1-recommender-34ab562ab257>