MAE301: Applied Experimental Statistics Report

Name: Cody McMahon

Instructor: Yi Ren

Date: 2/25/19

Introduction:

This project will delve into the application of reinforcement learning in order to complete a puzzle in the minimum amount of moves. The puzzle that will be used is the Tower of Hanoi puzzle. The concept of the Tower of Hanoi puzzle is described as, "Given a stack of n disks arranged from largest on the bottom to smallest on top placed on a rod, together with two empty rods, the tower of Hanoi puzzle asks for the minimum number of moves required to move the stack from one rod to another, where moves are allowed only if they place smaller disks on top of larger disks" (Wolfram Alpha). In order to solve the puzzle, a Python program that performs reinforcement learning will run in order to converge into a puzzle-solving process that will be performed in the minimum amount of moves. As defined by an artificial intelligence wiki, "Reinforcement learning refers to goal-oriented algorithms, which learn how to attain a complex objective (goal) or maximize along a particular dimension over many steps" (Artificial Intelligence Wiki). In general, reinforcement learning as applied to a python program is a simulation of a situation going through trial and error for multiple trials until it reaches an optimal solution for the objective. The process for which an analysis is done to yield the results of an optimal solution is the Markov Decision Process. Using Markov Decision Process, the Tower of Hanoi puzzle will be solved in as little moves as possible based on various states, actions, and rewards that will be defined later. The minimum amount of moves to solve the Tower of Hanoi puzzle is "...finding the number of steps it takes to transfer n disks from post A to post B is: **2^n - 1**" (The Math Forum).

With reinforcement learning, the program is based on a system of rewarding the program for doing actions and the reward will vary based on the specific action taken. This can be modeled through the use of the Bellman Equation. The Bellman Equation is shown in Image 1 in this paper (Li, Johnson, and Yeung). It is the maximum value of the expectation of the discount factor (γ), reward (r), state (s), and action (a). This is where the Markov Decision Process occurs to idealize what defines a state, reward, and action. For instance, the reward in the Tower of Hanoi would be to place a disk on a different peg where the disk is not on top of a smaller disk. The state would be the position of a disk at a certain peg and the action would be the movement of a disk to another peg. With the second equation in Image 1, it represents that after many iterations, the program will choose the pattern that achieved the largest reward after trying multiple patterns. To find the optimal policy with Bellman Equation, Q-learning is applied to continuously update what is the optimal policy.

To conduct an analysis of the program solving the Tower of Hanoi puzzle, a program downloaded off a Github (RobertTLange) will be modified through its reward function in order to make the program explore the best way to solve the puzzle. Additionally, the report will state how many simulations were ran prior to reaching the minimum amount of moves to solve the puzzle.

References:

A Beginner's Guide to Deep Reinforcement Learning. (n.d.). Retrieved February 25, 2019, from https://skymind.ai/wiki/deep-reinforcement-learning

Li, F., Johnson, J., & Yeung, S. (2017, May 23). Lecture 14: Reinforcement Learning. Retrieved February 25, 2019, from http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture14.pdf

RobertTLange. (2019, January 09). RobertTLange/gym-hanoi. Retrieved February 25, 2019, from https://github.com/RobertTLange/gym-hanoi

Tower of Hanoi. (n.d.). Retrieved February 25, 2019, from http://mathworld.wolfram.com/TowerofHanoi.html

Tower of Hanoi. (n.d.). Retrieved February 25, 2019, from http://mathforum.org/dr.math/faq/faq.tower.hanoi.html