Assignment 3

Deep Reinforcement Learning

Spring 2022

Problem 1 - A Bandit Problem

Part 1

Consider the following situation. You start with a thousand dollars to invest in two different stocks, **Apple** and **Microsoft**, for 100 days. At the beginning of each day, you must pick a stock to invest in. At the end of each day, you observe the return, and decide which stock to invest in the next day. You don't know what the return will be the next day, but you know that the return (as a proportion) of **Apple** is normally distributed with $\mu = .05$ and $\sigma^2 = .1$. The return of **Microsoft** is normally distributed with $\mu = 0.1$ and $\sigma^2 = .3$.

Before doing any analyses, what do you expect to be the best strategy? Would you prefer to invest in **Apple** or **Microsoft**? Or would you vary your investment based off of your balance? (There is no right answer here, and grading will only be based off of a thoughtful answer that makes a justification by addressing the probability distributions)

Part 2

Using python or a language of your choice, write a program that simulates your strategy (a 100 day simulation starting with 1000 dollars). Repeat the simulation 100 times and report the mean and variance of your results.

Part 3

One approximation to the problem is known as the exp3 algorithm. It initially gives equal weights in deciding between both stocks, and based of results, re-weights the probability of choosing the next stock. An algorithm[1] is shown below, where you may set $\gamma = 1$, $T = 100$ (the number of days) and $i = 1$ represents **Apple** and $i = 2$ represents **Microsoft**.

Parameters: Real $\gamma \in (0, 1]$

Initialisation: $\omega_i(1) = 1$ for $i = 1, \ldots, K$

For each t = 1, 2, ..., T

1. Set $p_i(t) = (1 - \gamma)\dfrac{\omega_i(t)}{\sum_{j=1}^{K} \omega_j(t)} + \dfrac{\gamma}{K} \qquad i = 1, \ldots, K$
2. Draw i_t randomly according to the probabilities $p_1(t), \ldots, p_K(t)$
3. Receive reward $x_{i_t}(t) \in [0, 1]$
4. For $j = 1, \ldots, K$ set:

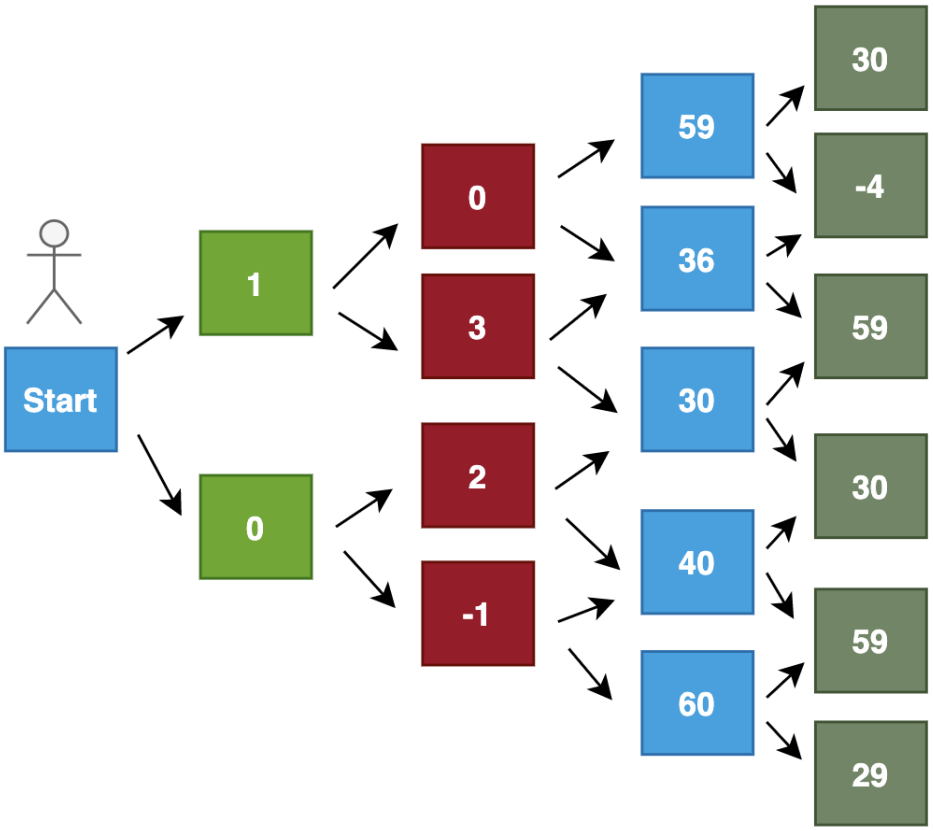
$$\hat{x}_j(t) = \begin{cases} x_j(t)/p_j(t) & \text{if } j = i_t \\ 0, & \text{otherwise} \end{cases}$$

$$\omega_j(t+1) = \omega_j(t) exp(\gamma \hat{x}_j(t)/K)$$

Implement this algorithm for the problem, and report the final return of the algorithm. How does it compare to your result in **Part 2**?

Problem 2 - Implementing Value Iteration

The diagram below represents a four period problem, where at each state (a box), you receive reward the number on the box. At each state, you must decide to move upwards or downwards to the next box.



Part 1

Consider the diagram above. What is the path(s) that maximizes your total reward (the sum of the values in the boxes)? In your notation, let U denote moving upwards and D denote moving downwards. Then the path U, U, U, D, for example, gives reward $r = 1 + 0 + 59 - 4 = 56$.

Part 2

Implement the value iteration algorithm for the problem above (by writing a program) with no more than ten iterations, and report your result. Are you able to achieve the optimal path?

References

[1] https://en.wikipedia.org/wiki/Multi-armed_bandit