COMP3702/COMP7702 Artificial Intelligence
Semester 2, 2020
Tutorial 10

Notes:

- *Tutorial 10 covers multi-armed bandits only, which arise in Module 3 (MCTS) and Module 4 (exploration strategies for RL).*
- *To help you model the settings below, code has been provided at* `https://gitlab.com/3702-2020/multi-armed-bandits`

Exercises

Exercise 10.1. Consider a situation where an agent must decide between two actions, A_1 and A_2, but it does not know the distribution of rewards for either action. Unknown to the agent, selecting A_1 returns a random variable drawn from a normal distribution with mean $\mu = 3$ and standard deviation $\sigma = 1$; while selecting A_2 returns a random variable drawn from a Weibull distribution with shape parameter $a = 2$ and scale parameter $b = 2\sqrt{2}$.

a) In one instance of the MAB, the actions taken and rewards received for the first six trials are given in the table below:

Trial	Action	Reward
1	A_1	2.66
2	A_2	1.25
3	A_1	3.21
4	A_2	2.34
5	A_1	1.87
6	A_1	1.69

Using ϵ-greedy, which action is most likely to be chosen next?

b) Given the same sample information, now consider UCB1 with upper bounds given by:

$$UCB1_a = \hat{v}_a + \sqrt{\frac{C\ \ln(N)}{n_a}}$$

Set the tunable parameter C to 5.[1] Using this UCB algorithm, which action is chosen next?

c) Plot the distributions of rewards from each arm. *Note: the support provides classes for arms with normal and Weibull random rewards, which also show you how to sample from these distributions.* **Question**: If the agent wishes to maximise its cumulative reward over time and knew these distributions, which would be the optimal arm to pull?

d) Set up a MAB instance with two arms described above, and consider the ϵ-greedy exploration strategy with random sampling parameter set to $\epsilon = 0.1$, and the UCB bound as described in b) above. For each strategy, plot their cumulative rewards over 1000 arm trials in an MAB instance. **Questions**: Which performs better initially? Which performs better in the long run?

This tutorial worksheet is deliberately short, so that you have time to get Tutor support for Assignment 3.

[1] Nb. This form of the UCB1 bound has C inside the square-root, which is different from the lecture slides but equivalent up the transform; however, it easier to implement the default of $C = 2$ in code in the form above (even though we are not using the default value here).