COMS 4701: Artificial Intelligence

Homework 3 Sample Solutions and Feedback

Problem 1

1. In most cases the third action value will converge (mean = 2). The others will not converge since we quickly discover that the third one gives the highest reward and we never try them again after one or two tries. Correspondingly, the UCB value of the third action decreases over time since we keep taking that action, while the UCB values of the other two will increase over time due to the logarithm term. The discrete jumps for these latter two values occur whenever these actions are actually taken, which only occurs a handful of times.

2. Generally the three action values will converge to the true means of 0, but sometimes one or two of them will be unlucky and not really converge at all. This occurs if the initial one or two tries of an action produces low rewards, causing us to think that it is objectively worse than the other two even though they are the same in reality. The distribution of "'N"' also varies quite a bit, though it is usually not uniform. If all action values are close to converging, the three actions will each be tried sufficiently, while non-convergence will correspond to an extremely unbalanced "'N"' distribution.

3. Convergence is much more consistent when $c = 5$. We are more willing to explore, which helps us realize that the three actions are distributed equally. So all three action values will generally approach the true means of 0, and the UCB values all go to 0 over time. "'N"' is also typically much closer to a uniform distribution.

Problem 2

1. With π_s representing the policy directing the car to go slow from both the cool and warm states,

$$V^{\pi_s}(\text{cool}) = 1(1 + 0.8V^{\pi_s}(\text{cool}))$$

and

$$V^{\pi_s}(\text{warm}) = 0.5[1 + 0.8V^{\pi_s}(\text{warm})] + 0.5[1 + 0.8V^{\pi_s}(\text{cool})]$$

The solution is

$$V^{\pi_s}(\text{cool}) = 5$$

$$V^{\pi_s}(\text{warm}) = 5$$

2. If we were to re-write the state values according to a policy π_f that directs the car to go fast when cool and warm, we would have the following:

$$V^{\pi_f}(\text{cool}) = 0.5[2 + 0.8V^{\pi}(\text{warm})] + 0.5[2 + 0.8V^{\pi}(\text{cool})]$$

$$V^{\pi_f}(\text{warm}) = -10 + 0.8V^{\pi}(\text{overheated})$$

Performing an iteration step evaluating policy π_f and fixing state values V^{π} to our results in part 1, we see that

$$V^{\pi_f}(\text{cool}) = 6$$

$$V^{\pi_f}(\text{warm}) = -10$$

Comparing these values for each state,

	$slow$	$fast$
cool	5	6
warm	5	-10

Since the value of the cool state is greater when going fast versus going slow, the policy will update to direct the car to move fast from the cool state, while still going slow from the warm state.

Cliffworld

- Value iteration updates should be implemented synchronously. There should be two sets of values in a given iteration, and the old values should be overwritten only after all new values are computed.
- For epsilon-greedy action selection, the optimal action *is included* in random action selection, which occurs with probability ε.
- The total probability for selecting the "greedy" action is thus $1 - \varepsilon + \frac{\varepsilon}{|A|}$
- One difference between SARSA and Q-learning is that Q-learning does **not** use the greedy action a' generated for the update as the next action. Q-learning should query the behavior policy π for the actual action taken in the next iteration.