ASEN 5519-003 Decision Making under Uncertainty
Homework 4: Tabular Reinforcement Learning

February 24, 2021

1 Conceptual Questions

Question 1. (30 pts) Consider a 3-armed Bernoulli Bandit with payoff probabilities $\theta = [0.2, 0.3, 0.7]$.

a) After a very large number of pulls, what is the expected payoff per pull of an ϵ-greedy policy with $\epsilon = 0.1$ (and no decay)?
b) After a very large number of pulls, what is the probability of selecting arm 3 when using a softmax policy with $\lambda = 10$ (and a "precision factor" of 1.0)?
c) Suppose that you are maintaining a Bayesian belief over the parameters θ starting with initial prior of Beta(1,1). Plot or sketch[1] the pdfs of the posterior probability distributions for each θ assuming the following numbers of wins and losses for each arm: $w = [0, 1, 3]$, $l = [1, 0, 2]$.
d) Given the situation in (c), describe one iteration of Thompson sampling. What quantities are sampled from what distributions? Choose some plausible values for the random samples and indicate which arm will be pulled.

2 Exercises

Question 2. (70 pts) Implement **two** different tabular or deep learning algorithms to learn a policy for the `DMUStudent.HW4.gw` grid world environment. *At most one* of these algorithms may be copied from the course notebooks or from any other reinforcement learning library you can find online, and at least one must be implemented by you from scratch or substantially modified from the notebooks. Some traditional algorithms to consider implementing are:

- Policy Gradient
- Dyna
- Max-Likelihood Model Based RL [with Prioritized Sweeping]
- [Double] Q-Learning
- Sarsa [λ]

Use only functions from `CommonRLInterface` to interact with the environment, and use the `HW4.render` function if you want to render the environment.

For each of the algorithms you implement, **plot** two learning curves. The first should have the number of samples or steps (calls to `act!`) that have been taken in the environment on the x axis. The second should have the wall clock elapsed time since training began on the x axis. Make sure to evaluate based on the best learned policy (and not the exploration policy), and give the plots for both algorithms the same x limits so that they are visually comparable. **Write** a short paragraph describing the relative strengths of the algorithms. Which one has higher sample complexity? Which one learns faster in terms of wall clock time?

[1] You may wish to use `https://homepage.divms.uiowa.edu/~mbognar/applets/beta.html` for this.