10

RL: Assignment #1

Due on Sunday, April 23, 2023

Well done !
– nice format !!!

:)

Abu El Komboz, Tareq 3405686 | Jain, Likhit 3678905 | Wurm, Marcel 3695946

Task 1

Multi-armed Bandits (4 Points)

(a) Consider ϵ-greedy action selection for a bandit with two actions ($k = 2$) and $\epsilon = 0.5$. What is the probability that the greedy action is selected? (2P)

> The probability that the greedy action is selected is calculated by the probability that the greedy action is chosen plus the probability that the greedy action is chosen by random: $(1 - \epsilon) + \epsilon \cdot \frac{1}{k} = (1 - 0.5) + 0.5 \cdot \frac{1}{2} = 0.75$

(b) Consider a k-armed bandit problem with $k = 4$ actions, denoted 1, 2, 3, and 4. Consider applying to this problem a bandit algorithm using ϵ-greedy action selection, sample-average action-value estimates, and initial estimates of $Q_1(a) = 0$, for all a. Suppose, you observe the following sequence of actions and rewards: $(A_1 = 1, R_2 = 1), (A_2 = 2, R_3 = 1), (A_3 = 2, R_4 = 2), (A_4 = 2, R_5 = 2), (A_5 = 3, R_6 = 0)$. On some of these time steps the ϵ case may have occurred, causing an action to be selected at random. (2P)

1. On which time steps did this definitely occur?

   > On timesteps $t = 2$ and $t = 5$ the actions were definitely selected at random because there were other actions with higher current value estimates for these timesteps.

2. On which time steps could this possibly have occurred?

   > This (random choice) could have occurred on all possible time steps as even the selection of the action that would have been chosen by acting greedy might have been chosen by chance.

Task 2

Action Selection Strategies (6 points)

(a) Implement the greedy action selection strategy in the function greedy. Initialize the values by playing each arm once. (3P)

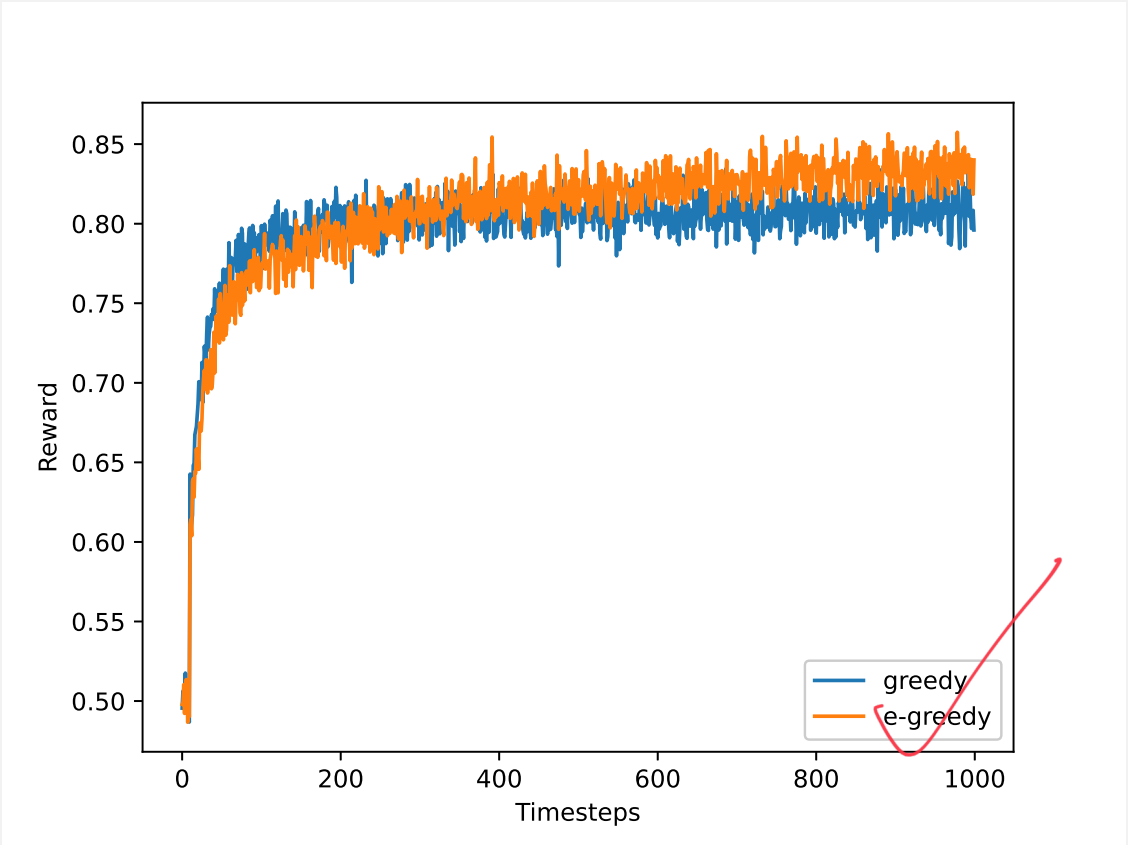
> See .py file for code!

(b) Implement the ϵ-greedy strategy in the function epsilon greedy. Use $\epsilon = 0.1$. (1P)

> See .py file for code!

Task 2 continued on next page...

(c) In the main function set $n_{episodes} = 10000$ to create a plot with less noise (this might take some time). The code template stores it as an eps file. Which of the 2 methods performs better, why? (1P)



The greedy method performs better initially because it always exploits the most promising option immediately. On the other hand, the greedy method often sticks with suboptimal actions in the end.
The ϵ-greedy method outperfoms the greedy method in the long run, as it discovers more optinos due to higher exploration. ϵ-greedy converges to the optimal Q-value and thus perfoms better.

Task 2 continued on next page...

(d) Think about possible ways to improve the implemented methods. What changes could you make to the strategies in order to improve them? (1P)

To improve data efficiency and runtime one could use incremental action-value estimates instead of saving the entire history.
To improve the performance, there are many possible ways:

- Use softmax activation function instead of random action in the ϵ-case
- Decaying ϵ so that it converges to greedy
- Ond could give higher value to rarely visited states (e.g. UCB)
- Select very bad states less likely
- Use prior information over reward distribution (if available), for example optimistic initialization could encourage exploration