Assignment3

Arya Ebrahimi
Reinforcement Learning
Ferdowsi University of Mashhad

Spring 2023

1 Question1

1.1 a

$$UCB_1(k,p) = \mathbb{E}[win|k,p] + c\sqrt{\frac{2\ln(n_{\text{parent}(k)})}{n_k}} \approx \frac{w_{k,p}}{n_k} + c\sqrt{\frac{2\ln(n_{\text{parent}(k)})}{n_k}} \tag{1}$$

Where k is a state, p is the corresponding player, n_k games that have state k and $w_{k,p}$ is number of games player p won.
From the root's children, it can be understood that a total number of 14 games are played which blue won 6, and red won 8. The UCB values for the states in the first level can be calculated using this knowledge.

- S1:

$$\frac{w_{S_1,R}}{n_{S_1}} = \frac{5}{7} \Rightarrow UCB_1(S_1,R) = \frac{5}{7} + 2\sqrt{\frac{2\ln(14)}{7}} \approx 2.45$$

- S2:

$$\frac{w_{S_2,R}}{n_{S_2}} = \frac{3}{4} \Rightarrow UCB_1(S_2,R) = \frac{3}{4} + 2\sqrt{\frac{2\ln(14)}{4}} \approx 3.04$$

- S3:

$$\frac{w_{S_3,R}}{n_{S_3}} = \frac{0}{3} \Rightarrow UCB_1(S_3,R) = \frac{0}{3} + 2\sqrt{\frac{2\ln(14)}{3}} \approx 2.65$$

Since the UCB_1 value for S_2 is higher than others, this node would be chosen. For the second level we have:

- S1:

$$\frac{w_{S_1,B}}{n_{S_1}} = \frac{1}{2} \Rightarrow UCB_1(S_1,B) = \frac{1}{2} + 2\sqrt{\frac{2\ln(4)}{2}} \approx 2.85$$

- S2:

$$\frac{w_{S_2,B}}{n_{S_2}} = \frac{1}{1} \Rightarrow UCB_1(S_2,B) = \frac{1}{1} + 2\sqrt{\frac{2\ln(4)}{1}} \approx 4.33$$

Thus, S_2 would be chosen because it has a higher UCB_1 value.

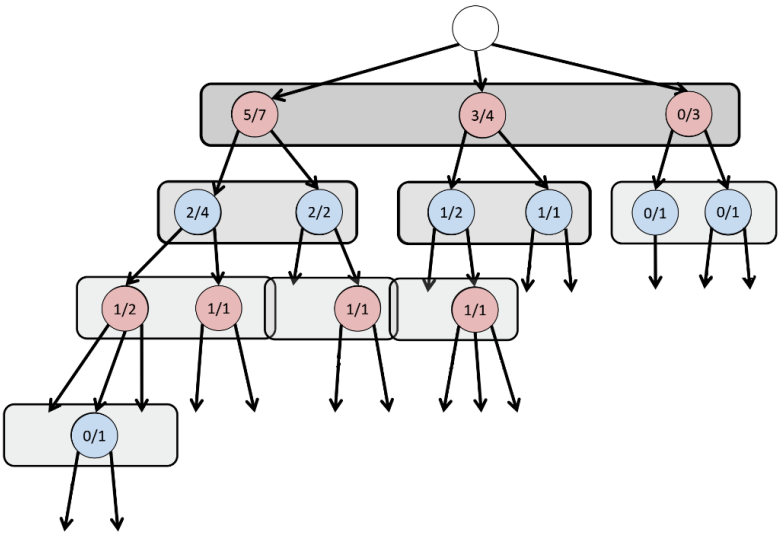

Figure 1: Question1 Image

1.2 b

- S1:

$$\frac{w_{S_1,R}}{n_{S_1}} = \frac{5}{7} \Rightarrow UCB_1(S_1, R) = \frac{5}{7} + 2\sqrt{\frac{2\ln(15)}{7}} \approx 2.47$$

- S2:

$$\frac{w_{S_2,R}}{n_{S_2}} = \frac{4}{5} \Rightarrow UCB_1(S_2, R) = \frac{4}{5} + 2\sqrt{\frac{2\ln(15)}{45}} \approx 2.88$$

- S3:

$$\frac{w_{S_3,R}}{n_{S_3}} = \frac{0}{3} \Rightarrow UCB_1(S_3, R) = \frac{0}{3} + 2\sqrt{\frac{2\ln(15)}{3}} \approx 2.68$$

Again, the UCB_1 value for S_2 is higher than others, this node would be chosen. For the second level we have:

- S1:

$$\frac{w_{S_1,B}}{n_{S_1}} = \frac{1}{2} \Rightarrow UCB_1(S_1, B) = \frac{1}{2} + 2\sqrt{\frac{2\ln(5)}{2}} \approx 3.03$$

- S2:

$$\frac{w_{S_2,B}}{n_{S_2}} = \frac{2}{2} \Rightarrow UCB_1(S_2, B) = \frac{2}{2} + 2\sqrt{\frac{2\ln(5)}{2}} \approx 3.53$$

Thus, S_2 would be chosen because it has a higher UCB_1 value, and this process continues.

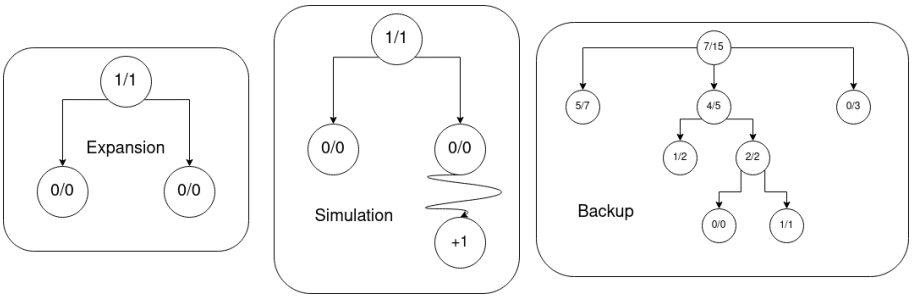

Figure 2: Expansion, Simulation and Backup process

2 Question2

2.1 a

To calculate $r(\pi)$, the following formula is used:

$$r(\pi) = \sum_{s} \mu_\pi(s) \sum_{a} \pi(a|s) \sum_{s',r} p(s', r|s, a)r \tag{2}$$

- Using only left action policy In this case, the agent can only be in three states. thus the $\mu_\pi(s)$ would be $\frac{1}{3}$. Since the agent only selects the left action $\pi(a|s)$ is 1, and the probability of getting a reward, in this case, is equal to 1, therefore $p(s', r|s, a)r = 0.6$ and we can write:

$$r(\pi)_{left} = \frac{1}{3} \times 0.6 = 0.2$$

  Since the reward for other transitions is zero, we only calculated the one resulting in a positive reward.

- Using only right action policy A similar approach to the previous part is taken. In this case, the only difference is that transition probabilities are not deterministic, and the likelihood of getting a +1 reward is 0.25 ($p(s', r|s, a)r = \frac{1}{4} \times 1$). Thus we have:

$$r(\pi)_{right} = \frac{1}{3} \times \frac{1}{4} \times 1 = \frac{1}{12} \approx 0.083$$

Therefore, the optimal policy is to take the left action since its average reward is higher.

2.2 b

There are three states, each having a 0.25 chance of getting a +1 reward. Thus the probability of at least one of them getting a reward is equal to the probability of their union:

$$P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$$
$$= \frac{1}{4} + \frac{1}{4} + \frac{1}{4} - \frac{3}{16} + \frac{1}{64} = \frac{37}{64} \approx 0.58$$

Because the probability of getting a reward using the right transitions is 0.58, an obtained policy might choose the right actions, which is not optimal. (the reward, in this case, is 0.6 for the left actions and about 0.58 for the right) Thus, using model-based or model-free approaches does not help without more

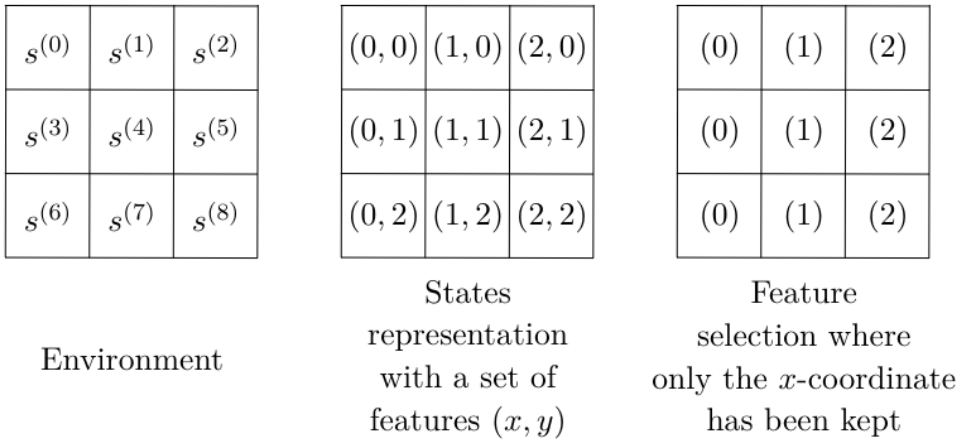

Figure 3: Proposed method to approximate states.

2.3 c

By using a linear function approximator, the problem mentioned in 2.b can be solved. Since the states in the same column are similar in the case of the reward they get and transitions, they can be represented together as a unified state shown in Figure 2.

$$f(x, y) = x \tag{3}$$

By calculating the average reward, it can be seen that like part a, the discrimination of policies would be better, and selecting the left action results in a higher average reward.

3 Question3

3.1 a

a. Assuming that the action Left is selected in the first $t-1$ steps, and the action Right is selected in the t-th step, a total of t steps are required to reach the State 2, where $t \in [1, \infty)$:

$$\begin{aligned} E_{S\to 2} &= 1 \cdot p^1 + 2 \cdot p^1(1-p)^1 + 3 \cdot p^1(1-p)^2 + ... \\ &= p \sum_{t=1}^{\infty} t(1-p)^{t-1} \\ &= \frac{p}{1-p} \sum_{t=1}^{\infty} t(1-p)^t = \frac{p}{1-p} \cdot \frac{1-p}{p^2} = \frac{1}{p} \end{aligned} \tag{4}$$

The transitiotn from state 2 to state 3 is divided into two cases, $S_2 \to S_3$, and $S_2 \to S \to S_2 \to S_3$.

$$\begin{aligned} E_{2\to 3} &= (1-p) + (2 + E_{S\to 2})p(1-p) + (3 + E_{S\to 2})p^2(1-p) + ... \\ &= (1-p) + (1-p) \sum_{t=1}^{\infty} [1 + t(1 + E_{S\to 2})]p^t \end{aligned} \tag{5}$$

The transitiotn from state 3 to state 4 is divided into two cases, $S_3 \to S_4$, and $S_3 \to S_2 \to S_3 \to S_4$.

$$\begin{aligned} E_{3\to 4} &= (1-p) + (2 + E_{S_2\to S_3})p(1-p) + (3 + E_{S_2\to S_3})p^2(1-p) + ... \\ &= (1-p) + (1-p) \sum_{t=1}^{\infty} [1 + t(1 + E_{S_2\to S_3})]p^t \end{aligned} \tag{6}$$

can be obtained in the same way:

$$\begin{aligned} E_{4\to G} &= p + (2 + E_{3\to 4})p(1-p) + (3 + 2E_{3\to 4})p(1-p)^2 + ... \\ &= p + p\sum_{t=1}^{\infty}[1 + t(1 + E_{3\to 4})](1-p)^t \end{aligned} \tag{7}$$

So, the total expectation value can be calculated as:

$$E = E_{S\to S_2} + E_{S_2\to S_3} + E_{S_3\to S_4} + E_{S_4\to G}$$

This process can be written as: (for more information see: first-link - second-link)

$$Q = \begin{bmatrix} 1-p & p & 0 & 0 \\ p & 0 & 1-p & 0 \\ 0 & p & 0 & 1-p \\ 0 & 0 & 1-p & 0 \end{bmatrix} \tag{8}$$

$$\begin{aligned} M = (I-Q)^{-1} &= \begin{bmatrix} p & -p & 0 & 0 \\ -p & 1 & p-1 & 0 \\ 0 & -p & 1 & p-1 \\ 0 & 0 & p-1 & 1 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} \frac{1}{p(1-p)^2} & \frac{-p+2}{-p+1} & \frac{1}{p(-p+1)} & \frac{1}{p} \\ \frac{-p+2}{-p+1} & \frac{-p+2}{-p+1} & \frac{1}{p(-p+1)} & \frac{1}{p} \\ \frac{-1}{(-p+1)(p-1)} & \frac{-1}{(-p+1)(p-1)} & \frac{1}{p(-p+1)} & \frac{1}{p} \\ \frac{-1}{p-1} & \frac{-1}{p-1} & \frac{1}{p} & \frac{1}{p} \end{bmatrix} \end{aligned} \tag{9}$$

Using this method explained in the first-link, we can calculate the expected steps between two states which is the sum of the first row of the matrix M.

$$E[t] = \frac{p^3 - 2p^2 - p + 3}{p(1-p)^2} \tag{10}$$

Since the reward is -1 per step we have:

$$J(p) = -E[t] = -(\frac{p^3 - 2p^2 - p + 3}{p(1-p)^2}) \tag{11}$$

To maximize the expectation, $\frac{dJ}{dp}$ should be zero.

$$\begin{aligned} &\frac{d}{dp}(-\frac{p^3 - 2p^2 - p + 3}{p(1-p)^2}) = 0 \\ &\Rightarrow p = \frac{9 - \sqrt{33}}{8} \approx 0.406 \end{aligned} \tag{12}$$

3.2 b

ϵ-greedy action selection is forced to choose between just two policies: ϵ-greedy right or left. The stochastic policy with the probability of the right action equal to 0.4 performs better than the two ϵ-greedy methods since acting in the S_2 and S_3 results in different behavior. Consider the right ϵ-greedy case; after starting from S and taking the right action, the agent transits to S_2. If $\epsilon = 0.1$, there is a 95 percent chance that with the next right action, the agent travels to the S. (because of the ϵ-greedy, there is a 5 percent chance to take the left action). Since the environment has two reverse states, an agent must select the left action twice in a row, which has a low probability (around 0.0025). The same explanation for the left ϵ-greedy can be considered. Thus, in this case, using a stochastic policy is better than a deterministic one.

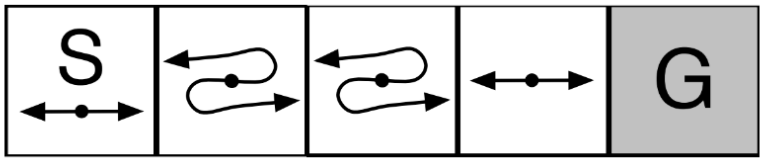

Figure 4: Question3 Image

3.3 c

Considering $p = 0.5$, the probability of being in each state would be equal, resulting in $\mu_\pi(s) = \frac{1}{5}$. (if $p \neq 0.5$, states would have different $\mu_\pi(s)$ and hard to calculate) if $p = 0.5$, the probability of selecting each action in states would also be equal, and $p(s', r|s, a)$ is 1 for all actions taken in states. Thus:

$$r(\pi) = \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) r$$

$$\begin{aligned} r(\pi) &= \frac{3}{5}\left(\frac{1}{2}(-1) + \frac{1}{2}(-1)\right) + \frac{1}{5}\left(\frac{1}{2}(-1) + \frac{1}{2}(0)\right) + \frac{1}{5}(20) \\ &= -0.6 - 0.1 + 4 = 3.3 \end{aligned}$$

Differential value function can be formulated as:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{r,s'} p(s', r|s, a)[r - r(\pi) + v_\pi(s')] \tag{13}$$

$$\begin{aligned} v_\pi(S) &= \frac{1}{2}\big(-1 - 3.3 + v_\pi(S)\big) + \frac{1}{2}\big(-1 - 3.3 + v_\pi(S_2)\big) \\ v_\pi(S_2) &= \frac{1}{2}\big(-1 - 3.3 + v_\pi(S)\big) + \frac{1}{2}\big(-1 - 3.3 + v_\pi(S_3)\big) \\ v_\pi(S_3) &= \frac{1}{2}\big(-1 - 3.3 + v_\pi(S_2)\big) + \frac{1}{2}\big(-1 - 3.3 + v_\pi(S_4)\big) \\ v_\pi(S_4) &= \frac{1}{2}\big(-1 - 3.3 + v_\pi(S_3)\big) + \frac{1}{2}\big(-1 - 3.3 + v_\pi(G)\big) \\ v_\pi(G) &= \big(+20 - 3.3 + v_\pi(S)\big) \end{aligned}$$

3.4 d

- By directly parametrizing policies, it can approach a deterministic policy, whereas, with epsilon-greedy action selection over action values, random actions are selected with a probability of epsilon.

- In tabular cases, there always exists an optimal deterministic policy, but by using function approximation methods, these optimal deterministic policies might not be found, resulting in the optimal policy being stochastic like the example in previous parts. One can find a stochastic policy because of the softmax layer using policy gradient methods.

- Sometimes directly learning a policy is more accessible than the value of each action. For example, one might know what action is good or bad but might not know the exact measure of how good or bad that is.