CS5500: Reinforcement Learning

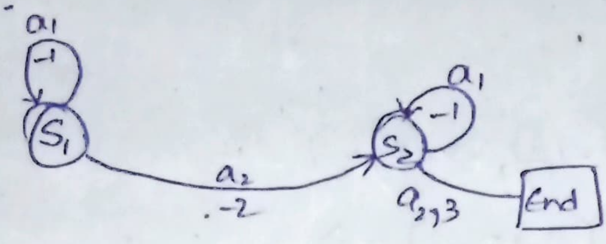
Final Exam

-Abburi Venkata Sai Mahesh

-CS18BTECH11001

Problem 1.

a) MDP:



b) from the above figure we can see that

$\pi^*(s_1) = a_2$, $\pi^*(s_2) = a_2$

So $V^*(s_1) = -2+3 = 1$

$V^*(s_2) = 3$

Given to consider initial value function zero

Let us assume $\gamma = 1$

$V_0(s_1) = 0$, $V_0(s_2) = 0$

1st iteration:

$$V_1(s_1) = \max_a \left[\sum_{s' \in S} P^a_{ss'} (R^a_{ss'} + \gamma V_k(s')) \right]$$

$$= \max [-1+0, -2+0]$$

$$= -1$$

$$V_1(s_2) = \max_a \left[\sum_{s' \in S} P_{ss'} (R_{ss'} + \gamma V_k(s')) \right]$$

$$= \max_a [-1+0, 3]$$

$$= 3$$

2nd iteration

$V_2(s_1) = \max[-1+(-1), -2+3]$

$= 1$

$V_2(s_2) = \max[-1+3, 3]$

$= 3$

3rd iteration:

$V_3(s_1) = \max[-1+1, -2+3] = 1$

$V_3(s_2) = \max[-1+3, 3] = 3$

We can observe that the value iteration has converged after 2nd iteration and even converged to the optimal value function V^* as introduced in the start.

c) If the successive value function doesn't iterate monotonic, it would contradict the fact that Bellman optimality operator is a contraction as the contraction mapping the inequality is only one way.

$\therefore$ The successive value function should be monotonic.

d) The total discounted return of an episode is given by

$$G = \sum_{t=0}^{\infty} \gamma^t r_{t+1}$$

As we use the $\gamma \in (0,1)$ which in return get $\gamma^t \to 0$ when $t \to \infty$, we get the bounded returns for the finite state, actions and bounded rewards.

e) for the shortest path problem using MDP. all the steps are equally important as reducing the step count at any point is the same. So the discount factor (γ) should be '1' for optimal choice.

2) a In the online Q-learning algorithm the Q(s,a) function is used for estimation of Q values which canges for all Q(s,a) pairs and suffers from moving target problem. whereas in watkins tabular Q-learning the Q(s,a) pairs will not be updated even on updating one of them and thus doesn't suffer from moving target problem.

b. In the stachastic batch update, the approximation function Q_ϕ learns Q-values using the reply buffer available which inturns get updated and as we collect more reply samples using expected Q function, our network target keeps on moving.

3) a given sofmax fuction for discrete action spaces as

$$\pi_\theta(a|s) = \frac{e^{\phi(s,a)^T\theta}}{\sum_{k=1}^{N} e^{\phi(s,a_k)^T\theta}}.$$

score function is

$$\nabla_\theta \log \pi_\theta(a|s) = \nabla_\theta \left[\phi(s,a)^T\theta - \log \sum_{a'} e^{\phi(s,a')^T\theta}\right]$$

$$= \phi(s,a) - \frac{1}{\sum_{a'} e^{\phi(s,a')^T\theta}} \nabla \sum_{a'} e^{\phi(s,a')^T\theta}$$

$$= \phi(s,a) - \frac{1}{\sum_{a'} e^{\phi(s,a')^T\theta}} \cdot \sum_{a'} \phi(s,a') e^{\phi(s,a')^T\theta}.$$

$$\cancel{= \phi(s,a) \frac{1}{\sum_{a''} e^{\phi(s,a'')^T\theta}}}$$

$$= \phi(s,a) - \sum_{a'} \phi(s,a') \frac{e^{\phi(s,a')^T\theta}}{\sum_{a''} e^{\phi(s,a'')^T\theta}}$$

$$= \phi(s,a) - \sum_{a'} \phi(s,a') \pi_\theta(a'|s)$$

$$\boxed{\nabla_\theta \log \pi_\theta(a|s) = \phi(s,a) - E_{a' \sim \pi_\theta(a'|s)}[\phi(s,a')]}$$

b) Using the derivation similar to below we get

$$\nabla_\theta \log \pi_\theta(a/s) = \frac{(a - \phi(s)^T\theta)\,\phi(s)}{\sigma}$$

where $\phi(s)^T\theta$ is given mean

σ is variance.

4) a. MAB framework can be used to solve single state MDP problem, Managing of exploration - exploitation trade-off problem and has major applications in online learning, dynamic pricing, music recommendation system etc.

b. MAB is a special case of RL where we have to make a single decision and will get a single reward where as in full RL decisions, states, rewards can vary with time. i.e., state space and action space, episode horizon can be finite or infinite.

c. In the naive exploration approach, we try to explore first and then we choose the estimates from the before exploration using any greedy approach. Where as in optimistic face of uncertainity we donot explore first but we are optimistic that the unexplored actions might turn out to be good and give more priority to unexplored actions with algorithms like UCB.

d. In UCB1 algorithm using optimism in face of uncertainity principle, we select the action using

$$a_t = \arg\max_a \Big[\underbrace{\hat{Q}_t(a)}_{\text{Exploitation}} + \underbrace{\sqrt{\frac{2 \ln t}{N_t(a)}}}_{\text{Exploration}} \Big]$$

Here we are adding a positive term of Exploration to the estimated Q values and the considering the maximum. which determines the usage of optimism in the face of uncertainity principle.

f. In UCB flavour we only estimate a single value $\hat{Q}(a)$ as a mean distribution for arm a where as in thompson sampling algorithm, for an arm 'a' we directly model the reward distribution as a beta distibution with parameter α_t^a, γ_t^a. In UCB, we estimate mean of distribution where as in Thompson Sampling, we estimate posterior distribution itself.

f) Let us calculate upper confidence bound for the 4 arms. for the next action.

$$Q_1 + \sqrt{\frac{2\ln 12}{n_1}} = 0.55 + \sqrt{\frac{2\ln 12}{3}} = 1.837$$

$$Q_2 + \sqrt{\frac{2\ln 12}{n_2}} = 0.63 + \sqrt{\frac{2\ln 12}{4}} = 1.745$$

$$Q_3 + \sqrt{\frac{2\ln 12}{n_3}} = 0.61 + \sqrt{\frac{2\ln 12}{3}} = 1.897$$

$$Q_4 + \sqrt{\frac{2\ln 12}{n_4}} = 0.4 + \sqrt{\frac{2\ln 12}{2}} = 1.976$$

As upper confidence bound of 4th arm is highest, it is played next

g) In thompson algorithm we first sample the distribution and pick the max from these sample which has a non-zero chance of picking any of the arm. Therefore the inner for loop ensures exploration

$$Q_t^a = \beta_{\alpha_t^a, r_t^a}$$

$$a^* = \arg\max_a Q_t^a$$