INFO8006 Introduction to Artificial Intelligence

Exercises 9: Reinforcement Learning

Learning outcomes

At the end of this exercise session, you should be able to:

- Apply off-policy algorithms (direct evaluation, temporal difference learning) and on-policy algorithm (Q-learning).
- Learn Value-function and Q-function approximators with off-policy and and on-policy algorithms.

Exercise 1: Direct evaluation - Pacman[1]

Consider the grid-world given below and an agent who is trying to learn the optimal policy. Rewards are only awarded for taking the Exit action from one of the shaded states. Taking this action moves the agent to the Done state, and the MDP terminates. The agent starts from the top left corner and you are given Table 1 which contains episodes from runs of the agent through this grid-world. Each line in an Episode is a tuple containing (s, a, s0 , r). Assuming discount factor $\gamma = 1$, compute the following Q-values obtained from **direct evaluation** from the samples:

- $Q((3,2), N) =$
- $Q((3,2), S) =$
- $Q((2,2), E) =$

Here the reward is defined with respect to a pair of state-action.
Reminder: by definition, $V^\pi(s') = \mathbb{E}_{a' \sim \pi(s')}\left[Q^\pi(s', a')\right]$.
$Q^\pi(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^\pi(s') = P^\pi(a|s) R(s,a) + \gamma \sum_{s'} P(s'|s,a) \mathbb{E}_{a' \sim \pi(s')}\left[Q^\pi(s', a')\right]$.

- $Q((3,2), N) = 0 + \gamma \sum_{s'} P(s'|s,a) \mathbb{E}_{a' \sim \pi(s')}\left[Q^\pi(s', a')\right] = Q((3,3), Exit) = 50$
- $Q((3,2), S) = 0 + 1 \sum_{s' \in \{(3,1)\}} P(s'|(3,2), S) \mathbb{E}_{a' \sim \pi(s')}\left[Q^\pi(s', a')\right] = Q((3,1), Exit) = 30$
- $Q((2,2), E) = 0 + \mathbb{E}_{a' \sim \pi(s')}\left[Q^\pi((3,2), a')\right] = \frac{2}{4} Q((3,2), N) + \frac{2}{4} Q((3,2), E) = 40$

Episode 1	Episode 2	Episode 3	Episode 4	Episode 5
(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0
(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0
(2,2), E, (3,2), 0	(2,2), S, (2,1), 0	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0
(3,2), N, (3,3), 0	(2,1), Exit, D, -100	(3,2), S, (3,1), 0	(3,2), N, (3,3), 0	(3,2), S, (3,1), 0
(3,3), Exit, D, +50		(3,1), Exit, D, +30	(3,3), Exit, D, +50	(3,1), Exit, D, +30

Table 1: Observed episodes.

Exercise 2: Q-Learning - Pacman[2]

Pacman is in an unknown MDP where there are three states [A, B, C] and two actions [Stop, Go]. We are given the following samples generated from taking actions in the unknown MDP. For the following problems, assume the discount factor $\gamma = 1$ and the learning rate $\alpha = 0.25$.

1. We run Q-learning on the following samples:

s	a	s'	r
A	Go	B	2
C	Stop	A	0
B	Stop	A	-2
B	Go	C	-6
C	Go	A	2
A	Go	A	-2

[1] Source: `https://inst.eecs.berkeley.edu/~cs188/fa19/assets/section/section5_solutions.pdf`
[2] Source: `https://inst.eecs.berkeley.edu/~cs188/fa19/assets/section/examprep5_solutions.pdf`

What are the estimates for the following Q-values as obtained by Q-learning? All Q-values are initialized to 0.

(a) $Q(C, Stop) =$

(b) $Q(C, Go) =$

2. For this next part, we will switch to a feature based representation. The approximator $\tilde{Q}(s,a) = w_1 + w_2 f(s,a)$ will use one feature:

   - $f(s,a) = \begin{cases} 1 & a = Go \\ -1 & a = Stop \end{cases}$.

   Starting from initial weights of 0, compute the updated weights after observing the following samples:

s	a	s'	r
A	Go	B	4
B	Stop	A	0

   (a) What are the weights after the first update (using the first sample) ?
   $\tilde{Q}(A, Go) = w_1 + w_2 f(A, Go) = 0$
   $Q(A, Go) := [r(A, Go) + \gamma \max_a \tilde{Q}(B, a)] = 4 + 0 = 8$
   $\Delta = Q(A, Go) - \tilde{Q}(A, Go) = 4$

$$w_1 \leftarrow w_1 - \alpha \frac{\partial(\Delta)^2}{\partial w_1}$$
$$w_1 \leftarrow w_1 + 2\alpha\Delta$$
$$w_1 \leftarrow 0 + 2 \times 0.25 \times 4 = 2$$

$$w_2 \leftarrow w_2 - \alpha \frac{\partial(\Delta)^2}{\partial w_2}$$
$$w_2 \leftarrow w_2 + 2\alpha\Delta f(A, Go)$$
$$w_2 \leftarrow 0 + 2 \times 0.25 \times 4 \times 1 = 2$$

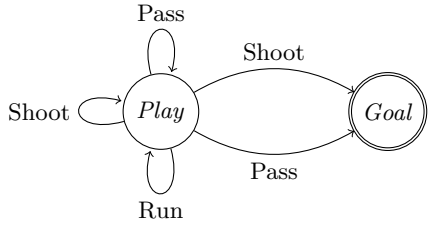
   (b) What are the weights after the second update (using the second sample) ?
   $\tilde{Q}(B, Stop) = w_1 + w_2 f(B, Stop) = 2 + 2 \times -1 = 0$
   $Q(B, Stop) := [r(B, Stop) + \gamma \max_a \tilde{Q}(B, a)] = 0 + 4 = 4$
   $\Delta = Q(B, Stop) - \tilde{Q}(B, Stop) = 4$

$$w_1 \leftarrow w_1 - \alpha \frac{\partial(\Delta)^2}{\partial w_1}$$
$$w_1 \leftarrow w_1 + 2\alpha\Delta$$
$$w_1 \leftarrow 2 + 2 \times 0.25 \times 4 = 4$$

$$w_2 \leftarrow w_2 - \alpha \frac{\partial(\Delta)^2}{\partial w_2}$$
$$w_2 \leftarrow w_2 + 2\alpha\Delta f(B, Stop)$$
$$w_2 \leftarrow 2 + 2 \times 0.25 \times 4 \times -1 = 0$$

Exercise 3: Temporal difference learning - Football[3]

Standard de Liège football team is playing against Anderlecht for the big homecoming game Saturday night. With a lot of losses in the season so far, Liège needs to switch up their strategy to get any hope of winning this game. Luckily, one of the midfielders Midlane (Mehdi) is a star student in INFO-8006 and has decided to model the game as a Markov Decision Process. There are only two states – the *Play* state (shown as the field in the diagram) and the *Goal* state. Although the connectivity of the states is known, the transition probabilities are not. There are no actions available from the *Goal* state – the game simply ends following the golden goal rule.



[3]Source: `https://inst.eecs.berkeley.edu/~cs188/fa19/assets/section/examprep5_solutions.pdf`

From the *Play* state there are three actions, *Run*, *Pass*, *Shoot*. The connectivity of each action to the two state is shown above.

Rewards:

s	a	s'	R(s,a,s')
Play	Run	Play	2
Play	Pass	Play	4
Play	Pass	Win	10
Play	Shoot	Play	0
Play	Shoot	Win	100

1. Mehdi wants to learn the value of the play state so he can estimate the outcome of the game. He uses a discount factor $\gamma = 0.5$ for all questions below.

   (a) Mehdi first uses temporal difference value learning to learn the value of the play state. After initializing his beliefs to 0, he sees two episodes ([{Play, Run, Play}] and [{Play, Shoot, Play}]) while in tape review. With a learning rate $\alpha = 0.5$ what value of the state play does he learn? TDL: $V^\pi(s) = (1-\alpha)V^\pi(s) + \alpha(r + \gamma V^\pi(s'))$

   $$V^\pi(Play) = (1-0.5)0 + 0.5(2+0.50) = 1 V^\pi(Play) = (1-0.5)1 + 0.5(0+0.51) = 0.75 \quad (1)$$

   (b) Coach Montanier decides to give Mehdi a fixed policy instead: $\pi(s) = Run$. What value for the state *Play* would Mehdi calculate if he ran value iteration until convergence? As long as the team's policy is to Run the game will stay in Play state and thus: $V(Play) = 2 + \gamma V(Play) = 2 + 0.5V(Play) \iff V(Play) = 4$

2. Mehdi now wants to use Q-learning to compute his optimal strategy. He sees three episodes ([{Play, Run, Play}], [{Play, Pass, Play}], [{Play, Pass, Win}]) during the first quarter. Update the Q node values after processing each episode (in order). Use a learning rate $\alpha = 0.5$ and a discount factor $\gamma = 0.5$. We start with $Q(s,a) = 0 \quad \forall s, a$.

   (a) After {Play, Run, Play}:

   $$Q(Play, Run) \leftarrow (1-\alpha)Q(Play, Run) + \alpha\left[(R(Play, Run, Play) + \gamma \max_a Q(Play, a)) - Q(Play, Run)\right]$$
   $$Q(Play, Run) \leftarrow 0.5 \times 0 + 0.5\left[2 + 0.5 \times 0 - 0\right] = 1$$

   (b) After {Play, Shoot, Play}:

   $$Q(Play, Shoot) \leftarrow (1-\alpha)Q(Play, Shoot) + \alpha\left[(R(Play, Shoot, Play) + \gamma \max_a Q(Play, a)) - Q(Play, Shoot)\right]$$
   $$Q(Play, Shoot) \leftarrow 0.5 \times 0 + 0.5\left[0 + 0.5 \times 1 - 0\right] = 0.25$$

   (c) After {Play, Pass, Win}:

   $$Q(Play, Pass) \leftarrow (1-\alpha)Q(Play, Pass) + \alpha\left[(R(Play, Pass, Win) - Q(Play, Pass)\right]$$
   $$Q(Play, Pass) \leftarrow 0.5 \times 0 + 0.5\left[10 - 0\right] = 5$$

3. Q learning is going well, but it's taking too much time. Thankfully Montanier shows up with some special information – he has watched so many games that he knows the true transition probabilities! Here they are:

s	a	s'	R(s,a,s')	P(s'|s, a)
Play	Run	Play	2	1
Play	Pass	Play	4	0.5
Play	Pass	Win	10	0.5
Play	Shoot	Play	0	0.9
Play	Shoot	Win	100	0.1

   (a) Now with these probabilities, what is the optimal policy when there is one time step left? The value? With only one step left, there is no need to take into account future rewards, choosing the action that maximizes the average immediate reward is sufficient. We have, in this setting:

   $$Q(Play, Run) = 2$$
   $$Q(Play, Pass) = 0.5 \times 4 + 0.5 \times 10 = 7$$
   $$Q(Play, Shoot) = 0.9 \times 0 + 0.1 \times 100 = 10$$

   And so the best action is to shoot.

   (b) For two time steps left, what is the optimal policy with discount factor 0.5? Hint: you can use your value above to aid in this computation. Keeping Bellman's equations spirit, we now that the optimal move 2 time steps ahead only depends on the immediate reward and the Value functions one step ahead. Thus we have:

   $$Q(Play, Run) = 1(2 + \gamma 10) = 7$$
   $$Q(Play, Pass) = 0.5(4 + \gamma V(Play)) + 0.5(10 + \gamma V(Play)) = 12$$
   $$Q(Play, Shoot) = 0.1(100 + \gamma V(Win)) + 0.9(0 + \gamma V(Play)) = 10 + 0.9 \times 0.5 \times 10 = 14.5$$

   And so the best action is to shoot.