Introduction to Artificial Intelligence

Review RL Solutions

Q1. RL

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 $\gamma = 1$ and $\alpha = 0.5$.

(a) 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

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

(i) $Q(C, Stop) =$ 0.5

(ii) $Q(C, Go) =$ 1.5

For this, we only need to consider the following three samples.

$$Q(A, Go) \leftarrow (1-\alpha)Q(A, Go) + \alpha(r + \gamma \max_a Q(B, a)) = 0.5(0) + 0.5(2) = 1$$
$$Q(C, Stop) \leftarrow (1-\alpha)Q(C, Stop) + \alpha(r + \gamma \max_a Q(A, a)) = 0.5(0) + 0.5(1) = 0.5$$
$$Q(C, Go) \leftarrow (1-\alpha)Q(C, Go) + \alpha(r + \gamma \max_a Q(A, a)) = 0.5(0) + 0.5(3) = 1.5$$

(b) For this next part, we will switch to a feature based representation. We will use two features:

- $f_1(s, a) = 1$
- $f_2(s, a) = \begin{cases} 1 & a = \text{Go} \\ -1 & a = \text{Stop} \end{cases}$

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

What are the weights after the first update? (using the first sample)

(i) $w_1 =$ ___2___

(ii) $w_2 =$ ___2___

$$Q(A, Go) = w_1 f_1(A, Go) + w_2 f_2(A, Go) = 0$$
$$difference = [r + max_a Q(B, a)] - Q(A, Go) = 4$$
$$w_1 = w_1 + \alpha(difference) f_1 = 2$$
$$w_2 = w_2 + \alpha(difference) f_2 = 2$$

What are the weights after the second update? (using the second sample)

(iii) $w_1 =$ ___4___

(iv) $w_2 =$ ___0___

$$Q(B, Stop) = w_1 f_1(B, Stop) + w_2 f_2(B, Stop) = 2(1) + 2(-1) = 0$$
$$Q(A, Go) = w_1 f_1(A, Go) + w_2 f_2(A, Go) = 2(1) + 2(1) = 4$$
$$difference = [r + max_a Q(A, a)] - Q(B, Stop) = [0 + 4] - 0 = 4$$
$$w_1 = w_1 + \alpha(difference) f_1 = 4$$
$$w_2 = w_2 + \alpha(difference) f_2 = 0$$

Q2. Q-uagmire

Consider an unknown MDP with three states (A, B and C) and two actions ($\leftarrow$ and $\rightarrow$). Suppose the agent chooses actions according to some policy π in the unknown MDP, collecting a dataset consisting of samples (s, a, s', r) representing taking action a in state s resulting in a transition to state s' and a reward of r.

s	a	s'	r
A	$\rightarrow$	B	2
C	$\leftarrow$	B	2
B	$\rightarrow$	C	-2
A	$\rightarrow$	B	4

You may assume a discount factor of $\gamma = 1$.

(a) Recall the update function of Q-learning is:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha\left(r_t + \gamma \max_{a'} Q(s_{t+1}, a')\right)$$

Assume that all Q-values are initialized to 0, and use a learning rate of $\alpha = \frac{1}{2}$.

(i) Run Q-learning on the above experience table and fill in the following Q-values:

$Q(A, \rightarrow) =$ $5/2$ $\qquad Q(B, \rightarrow) =$ $-1/2$

$$Q_1(A, \rightarrow) = \frac{1}{2} \cdot Q_0(A, \rightarrow) + \frac{1}{2}\left(2 + \gamma \max_{a'} Q(B, a')\right) = 1$$

$$Q_1(C, \leftarrow) = 1$$

$$Q_1(B, \rightarrow) = \frac{1}{2}(-2+1) = -\frac{1}{2}$$

$$Q_2(A, \rightarrow) = \frac{1}{2} \cdot 1 + \frac{1}{2}\left(4 + \max_{a'} Q_1(B, a')\right)$$

$$= \frac{1}{2} + \frac{1}{2}(4+0) = \frac{5}{2}.$$

(ii) After running Q-learning and producing the above Q-values, you construct a policy π_Q that maximizes the Q-value in a given state:

$$\pi_Q(s) = \arg\max_a Q(s, a).$$

What are the actions chosen by the policy in states A and B?

$\pi_Q(A)$ is equal to:

- ○ $\pi_Q(A) = \leftarrow$.
- ● $\pi_Q(A) = \rightarrow$.
- ○ $\pi_Q(A) =$ Undefined.

$\pi_Q(B)$ is equal to:

- ● $\pi_Q(B) = \leftarrow$.
- ○ $\pi_Q(B) = \rightarrow$.
- ○ $\pi_Q(B) =$ Undefined.

Note that $Q(B, \leftarrow) = 0 > -\frac{1}{2} = Q(B, \rightarrow)$.

(b) Use the empirical frequency count model-based reinforcement learning method described in lectures to estimate the transition function $\hat{T}(s, a, s')$ and reward function $\hat{R}(s, a, s')$. (Do not use pseudocounts; if a transition is not observed, it has a count of 0.)

Write down the following quantities. You may write N/A for undefined quantities.

$\hat{T}(A, \rightarrow, B) =$ 1 $\quad \hat{R}(A, \rightarrow, B) =$ 3

$\hat{T}(B, \rightarrow, A) =$ 0 $\quad \hat{R}(B, \rightarrow, A) =$ N/A

$\hat{T}(B, \leftarrow, A) =$ N/A $\quad \hat{R}(B, \leftarrow, A) =$ N/A

(c) This question considers properties of reinforcement learning algorithms for *arbitrary* discrete MDPs; you do not need to refer to the MDP considered in the previous parts.

(i) Which of the following methods, at convergence, provide enough information to obtain an optimal policy? (Assume adequate exploration.)

- [x] Model-based learning of $T(s, a, s')$ and $R(s, a, s')$.
- [ ] Direct Evaluation to estimate $V(s)$.
- [ ] Temporal Difference learning to estimate $V(s)$.
- [x] Q-Learning to estimate $Q(s, a)$. Given enough data, model-based learning will get arbitrarily close to the true model of the environment, at which point planning (e.g. value iteration) can be used to find an optimal policy. Q-learning is similarly guaranteed to converge to the optimal Q-values of the optimal policy, at which point the optimal policy can be recovered by $\pi^*(s) = \arg\max_a Q(s, a)$. Direct evaluation and temporal difference learning both only recover a value function $V(s)$, which is insufficient to choose between actions without knowledge of the transition probabilities.

(ii) In the limit of infinite timesteps, under which of the following exploration policies is Q-learning guaranteed to converge to the optimal Q-values for all state? (You may assume the learning rate α is chosen appropriately, and that the MDP is ergodic: i.e., every state is reachable from every other state with non-zero probability.)

- [x] A fixed policy taking actions uniformly at random.
- [ ] A greedy policy.
- [x] An ϵ-greedy policy
- [ ] A fixed optimal policy. For Q-learning to converge, every state-action pair (s, a) must occur infinitely often. A uniform random policy will achieve this in an ergodic MDP. A fixed optimal policy will not take any suboptimal actions and so will not explore enough. Similarly a greedy policy will stop taking actions the current Q-values suggest are suboptimal, and so will never update the Q-values for supposedly suboptimal actions. (This is problematic if, for example, an action most of the time yields no reward but occasionally yields very high reward. After observing no reward a few times, Q-learning with a greedy policy would stop taking that action, never obtaining the high reward needed to update it to its true value.)

Q3. Reinforcement Learning

Imagine an unknown environments with four states (A, B, C, and X), two actions (← and →). An agent acting in this environment has recorded the following episode:

s	a	s'	r	Q-learning iteration numbers (for part b)
A	→	B	0	1, 10, 19, ...
B	→	C	0	2, 11, 20, ...
C	←	B	0	3, 12, 21, ...
B	←	A	0	4, 13, 22, ...
A	→	B	0	5, 14, 23, ...
B	→	A	0	6, 15, 24, ...
A	→	B	0	7, 16, 25, ...
B	→	C	0	8, 17, 26, ...
C	→	X	1	9, 18, 27, ...

(a) Consider running model-based reinforcement learning based on the episode above. Calculate the following quantities:

$\hat{T}(B, \rightarrow, C) =$ $\frac{2}{3}$

$\hat{R}(C, \rightarrow, X) =$ 1

(b) Now consider running Q-learning, repeating the above series of transitions in an infinite sequence. Each transition is seen at multiple iterations of Q-learning, with iteration numbers shown in the table above.

After which iteration of Q-learning do the following quantities first become nonzero? (If they always remain zero, write *never*).

$Q(A, \rightarrow)$? 14

$Q(B, \leftarrow)$? 22

(c) True/False: For each question, you will get positive points for correct answers, zero for blanks, and negative points for incorrect answers. Circle your answer **clearly**, or it will be considered incorrect.

(i) [*true* or *false*] In Q-learning, you do not learn the model.
Q learning is model-free, you learn the optimal policy explicitly, and the model itself implicitly.

(ii) [*true* or *false*] For TD Learning, if I multiply all the rewards in my update by some nonzero scalar p, the algorithm is still guaranteed to find the optimal policy.
If p is positive then yes, the discounted values, relative to each other, are just scaled. But if p is negative, you will be computing negating the values for the states, but the policy is still chosen on the max values.

(iii) [*true* or *false*] In Direct Evaluation, you recalculate state values after each transition you experience.

In order to estimate state values, you calculate state values from episodes of training, not single transitions.

(iv) [*true* or *false*] Q-learning requires that all samples must be from the optimal policy to find optimal q-values.
Q-learning is off-policy, you can still learn the optimal values even if you act suboptimally sometimes.