Worksheet 9

1. An agent observes the following two episodes from an MDP,

$$S_0 = 0, A_0 = 1, R_1 = 1, S_1 = 1, A_1 = 1, R_2 = 1$$

$$S_0 = 0, A_0 = 0, R_1 = 0, S_1 = 0, A_1 = 1, R_2 = 1, S_2 = 1, A_2 = 1, R_3 = 1$$

and updates its deterministic model accordingly. What would the model output for the following queries:

(a) Model($S = 0, A = 0$):

(b) Model($S = 0, A = 1$):

(c) Model($S = 1, A = 0$):

(d) Model($S = 1, A = 1$):

Answer:

(a) Model($S = 0, A = 0$): 0, 0

(b) Model($S = 0, A = 1$): 1, 1

(c) Model($S = 1, A = 0$): None

(d) Model($S = 1, A = 1$): 1, terminal

2. An agent is in a 4-state MDP, $\mathcal{S} = \{1, 2, 3, 4\}$, where each state has two actions $\mathcal{A} = \{1, 2\}$. Assume the agent saw the following trajectory,

$$\begin{aligned}
&S_0 = 1, A_0 = 2, R_1 = -1,\\
&S_1 = 1, A_1 = 1, R_2 = 1,\\
&S_2 = 2, A_2 = 2, R_3 = -1,\\
&S_3 = 2, A_3 = 1, R_4 = 1,\\
&S_4 = 3, A_4 = 1, R_5 = 100,\\
&S_5 = 4
\end{aligned}$$

and uses Tabular Dyna-Q with 5 planning steps for each interaction with the environment.

(a) Once the agent sees S_5, how many Q-learning updates has it done with **real experience**? How many updates has it done with **simulated experience**?

(b) Which of the following are possible (or not possible) simulated transitions $\{S, A, R, S'\}$ given the above observed trajectory with a deterministic model and random search control?

i. $\{S = 1, A = 1, R = 1, S' = 2\}$

ii. $\{S = 2, A = 1, R = -1, S' = 3\}$

iii. $\{S = 2, A = 2, R = -1, S' = 2\}$

iv. $\{S=1, A=2, R=-1, S'=1\}$
v. $\{S=3, A=1, R=100, S'=5\}$

Answer:

(a) Once the agent sees S_5, how many Q-learning updates has it done with **real experience**? How many updates has it done with **simulated experience**?

5 update with real experience and 25 updated with simulated experience

(b) Which of the following are possible (or not possible) simulated transitions $\{S, A, R, S'\}$ given the above observed trajectory with a deterministic model and random search control?

i. $\{S=1, A=1, R=1, S'=2\}$: Possible
ii. $\{S=2, A=1, R=-1, S'=3\}$: Not Possible
iii. $\{S=2, A=2, R=-1, S'=2\}$: Possible
iv. $\{S=1, A=2, R=-1, S'=1\}$: Possible
v. $\{S=3, A=1, R=100, S'=5\}$: Not possible

Worksheet 9

3. Modify the Tabular Dyna-Q algorithm so that it uses Expected Sarsa instead of Q-learning. Assume that the target policy is ϵ-greedy. What should we call this algorithm?

Tabular Dyna-Q

Initialize $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$
Loop forever:
(a) $S \leftarrow$ current (nonterminal) state
(b) $A \leftarrow \varepsilon$-greedy(S, Q)
(c) Take action A; observe resultant reward, R, and state, S'
(d) $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
(e) $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
(f) Loop repeat n times:
$S \leftarrow$ random previously observed state
$A \leftarrow$ random action previously taken in S
$R, S' \leftarrow Model(S,A)$
$Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$

Answer:

I'm not sure what we should call this algorithm, maybe Dyna-Expected-Sarsa.

To make the algorithm use Expected Sarsa instead of Q-learning, we should change the updates made both using real experience and using simulated experience as shown below:

$$Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma \sum_a \pi(a|S')Q(S',a) - Q(S,A)]$$

4. Consider an MDP with two states $\{1, 2\}$ and two possible actions: $\{\text{stay}, \text{switch}\}$. The state transitions are deterministic, the state does not change if the action is "stay" and the state switches if the action is "switch". However, rewards are randomly distributed:

$$P(R\,|S=1, A=\text{stay}) = \begin{cases} 0 & \text{w.p. } 0.4 \\ 1 & \text{w.p. } 0.6 \end{cases}, \quad P(R\,|S=1, A=\text{switch}) = \begin{cases} 0 & \text{w.p. } 0.5 \\ 1 & \text{w.p. } 0.5 \end{cases}$$

$$P(R\,|S=2, A=\text{stay}) = \begin{cases} 0 & \text{w.p. } 0.6 \\ 1 & \text{w.p. } 0.4 \end{cases}, \quad P(R\,|S=2, A=\text{switch}) = \begin{cases} 0 & \text{w.p. } 0.5 \\ 1 & \text{w.p. } 0.5 \end{cases}$$

(a) How might you learn the reward model? Hint: think about how probabilities are estimated. For example, what if you were to estimate the probability of a coin landing on heads? If you observed 10 coin flips with 8 heads and 2 tails, then you can estimate the probabilities by counting: $p(\text{heads}) = \frac{8}{10} = 0.8$ and $p(\text{tails}) = \frac{2}{10} = 0.2$.

(b) Modify the tabular Dyna-Q algorithm to handle this MDP with stochastic rewards.

Answer:

(a) We can estimate $P(R|S=s, A=a)$ by keeping counts of each event.

Worksheet 9

(b)
In the beginning, initialize counts of size $(|\mathcal{S}| \times |\mathcal{A}|, |\mathcal{R}|)$ to 0.

When updating the model (line e):
1- update the counts for R, S': counts$[S \times |\mathcal{A}| + A, R] + +$
2- update the model: Model$(S, A) \leftarrow \frac{\text{counts}[S \times |\mathcal{A}| + A, R]}{\sum \text{counts}[S \times |\mathcal{A}| + A, :]}, S'$

When sampling from the model, sample the reward using the estimated probability.

Worksheet 9

5. **Challenge Question:** Consider an MDP with three states $\mathcal{S} = \{1, 2, 3\}$, where each state has two possible actions $\mathcal{A} = \{1, 2\}$ and a discount rate $\gamma = 0.5$. Suppose estimates of $Q(S, A)$ are initialized to 0 and you observed the following episode according to an unknown behaviour policy where S_3 is the terminal state.

$$S_0 = 1, A_0 = 1, R_1 = -7, S_1 = 3, A_1 = 2, R_2 = 5, S_2 = 1, A_2 = 1, R_3 = 10$$

(a) Suppose you used Q-learning with the above trajectory to estimate $Q(S, A)$, what are your new estimates for $Q(S = 1, A = 1)$ using $\alpha = 0.1$?

(b) What is one possible model for this environment? Is the model stochastic or deterministic?

(c) Suppose in the planning loop, after search control, we would like to update $Q(S = 1, A = 1)$ with Q-planning. What are the possible outputs of Model$(S = 1, A = 1)$?

(d) If your model outputs $R = R_3$ and $S' = S_3$, what is $Q(S = 1, A = 1)$ after one Q-planning update? Use the estimates of $Q(S, A)$ from before.

Answer:

(a)
The updates for the first transition:

$$Q(1, 1) \leftarrow Q(1, 1) + 0.1[-7 + 0.5\text{max}_a Q(3, a) - Q(1, 1)] = -0.7$$

The updates for the second transition:

$$Q(1, 1) \leftarrow Q(1, 1) + 0.1[10 + 0.5 \times 0 - Q(1, 1)] = -0.7 + 1 + 0.07 = 0.37$$

(b)
Any model that is consistent with the trajectory. The probability for the transitions in the trajectory should not be zero.
The model is stochastic since in state 1 with action 1, we can transition to both state 3 or the terminal state.

(c)
The possible outputs of the model are (-7, 3) and (10, terminal)

(d)

$$Q(1, 1) \leftarrow Q(1, 1) + 0.1[10 + 0.5 \times 0 - Q(1, 1)] = 0.37 + 1 - 0.037 = 1.333$$

Worksheet 9

6. (*Exercise 8.2 S&B*) Why did the Dyna agent with exploration bonus, Dyna-Q+, perform better in the first phase as well as in the second phase of the blocking experiment in Figure 8.4?

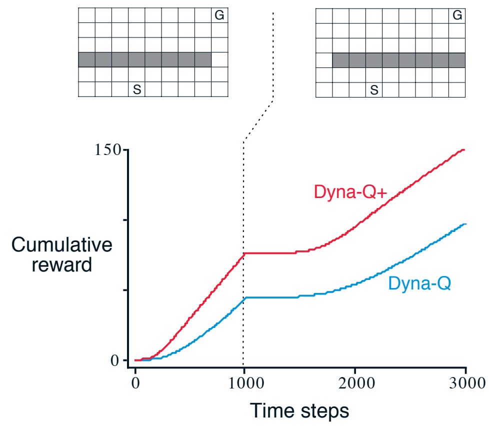

Figure 8.4: Average performance of Dyna agents on a blocking task. The left environment was used for the first 1000 steps, the right environment for the rest. Dyna-Q+ is Dyna-Q with an exploration bonus that encourages exploration. ■

Answer: In the maze, the agent receives a non-zero reward only when visiting the goal state. Therefore, the state-action values are pretty similar for many state-action pairs in the beginning. This causes the Dyna-Q algorithm to have a random policy in the beginning.The Dyna-Q+ algorithm, however, has an exploration bonus encouraging the agent to visit the less visited state-action pairs. Visiting the less explored part of the maze increases the chance of the agent to stumble upon the goal state (or states with non-zero values) compared to the random policy initially used by Dyna-Q.

7. (*Exercise 8.3 S&B*) **Challenge Question:** Careful inspection of Figure 8.5 reveals that the difference between Dyna-Q+ and Dyna-Q narrowed slightly over the first part of the experiment. What is the reason for this?

Answer:

After finding the optimal path to the goal, the exploratory policy of Dyna-Q+ is no more beneficial and results in Dyna-Q outperforming Dyna-Q+ since Dyna-Q+ sometimes do not follow the optimal path.

Worksheet 9

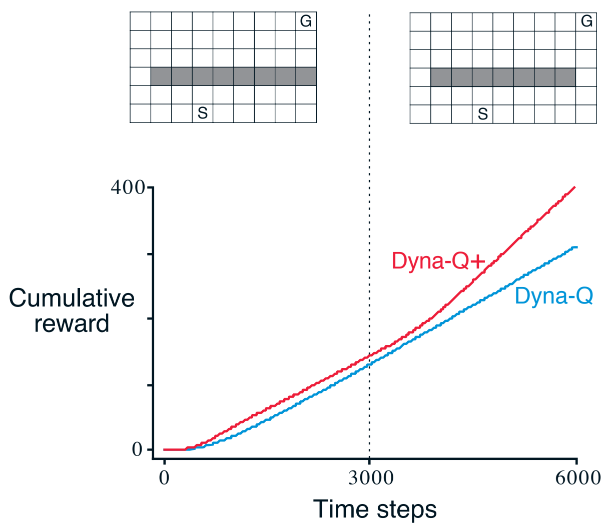

Figure 8.5: Average performance of Dyna agents on a shortcut task. The left environment was used for the first 3000 steps, the right environment for the rest.

■