Reinforcement Learning

Tutorial-1 Solutions

October 11, 2010

I. Tic-Tac-Toe Problem Formulation

Assume that the RL agent is going to play the X's.

- Actions: in which cell to put the next X
- States: the state of the grid
- Reward:
  - Positive reward for actions leading to winning states (grids with 3 X's in a row, column, or diagonally)
  - Negative reward for actions leading to losing states
  - 0 reward for every other action
- State transition: After the agent chooses where to put the X, the opponent (as part of the environments) puts an O. That means after taking any action (deterministic), the world evolves into one of different possible configurations (stochastic). The state transition $P^a_{ss'}$ should capture the probabilities of these transitions. If we do not know how the opponent is going to play, we don't know the state transition map.

Exercise 1.1

What would happen is that the agent will learn how to play optimally against itself. So, if we have two copies of the agent adapting to each other, we will end up with an agent that is capable of winning, generally speaking, against other opponents. That, however, does not mean that we cannot design an opponent that can exploit and beat our agent.

Exercise 1.2

Using symmetries we can reduce the size of the state space; i.e. less memory and less learning time. However, if the opponent does not take advantage of these symmetries and acts differently in symmetrical states, we will be better off learning for individual states, since these symmetrical states would have different values, given the opponent strategy.

Exercise 1.3

A greedy player exploits what it knows, but ignores what it does not know. A non-greedy player, on the other hand, keeps discovering and updating its state value estimates. The non-greedy style of play, sometimes, ends up playing better.

Exercise 1.4

The probabilities learnt when not considering exploration are the probabilities of winning when playing the actions of policyπ. On the other hand, the probabilities learnt when considering exploration moves are the probabilities of winning when playing the actions of policyπ and jumping randomly from time to time. The latter would value more the states that are safe; that is, always lead to winning even after some random exploratory jumps.

If our agent plans to keep exploring, due to a non-stationary environment for example, then it's more useful to learn the values of states with exploratory moves considered.

Exercise 1.5

- using smarter state representations to speed learning; e.g., symmetries in states.
- using prior knowledge of the problem for initial values; e.g. the centre cell is more valuable than other cells.
- learning the strategy that the opponent uses in playing (the environment dynamics).
- ...

II. Understanding Value Functions

Exercise 3.9

$$
\begin{aligned}
V^{\pi}(s) &= \sum_{a} \pi(s,a) \sum_{s'} P^{a}_{ss'}[R^{a}_{ss'} + \gamma V^{\pi}(s')] \\
&= \pi(s,up) \sum_{s'} P^{up}_{ss'}[R^{up}_{ss'} + \gamma V^{\pi}(s')] + \\
&\quad \pi(s,down) \sum_{s'} P^{down}_{ss'}[R^{down}_{ss'} + \gamma V^{\pi}(s')] + \\
&\quad \pi(s,right) \sum_{s'} P^{right}_{ss'}[R^{right}_{ss'} + \gamma V^{\pi}(s')] + \\
&\quad \pi(s,left) \sum_{s'} P^{left}_{ss'}[R^{left}_{ss'} + \gamma V^{\pi}(s')]
\end{aligned}
$$

$$
\begin{aligned}
= \quad & \pi(s, up)[R^{up}_{ss'} + \gamma V^{\pi}(s')] + \\
& \pi(s, down)[R^{down}_{ss'} + \gamma V^{\pi}(s')] + \\
& \pi(s, right)[R^{right}_{ss'} + \gamma V^{\pi}(s')] + \\
& \pi(s, left)[R^{left}_{ss'} + \gamma V^{\pi}(s')] \qquad \text{(deterministic actions)} \\
= \quad & 0.25 \times (0 + 0.9 \times 2.3) + 0.25 \times (0 + 0.9 \times (-0.4)) + \\
& 0.25 \times (0 + 0.9 \times 0.4) + 0.25 \times (0 + 0.9 \times 0.7) \\
\approx \quad & 0.7
\end{aligned}
$$

Exercise 3.10

Value of a state is the expected reward starting from this state and using policyπ:

$$
\begin{aligned}
R_t &= r_{t+1} + \gamma r_{t+2} + \gamma_2 r_{t+3} + \ldots \\
&= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}
\end{aligned}
$$

If we add a constant, C, to all immediate rewards r:

$$
\begin{aligned}
R'_t &= \sum_{k=0}^{\infty} \gamma^k (r_{t+k+1} + C) \\
&= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} + \sum_{k=0}^{\infty} \gamma^k C \\
&= R_t + \sum_{k=0}^{\infty} \gamma^k C
\end{aligned}
$$

The additional term,

$$
K = \sum_{k=0}^{\infty} \gamma^k C = \frac{C}{1-\gamma} \qquad \text{when } \gamma < 1
$$

is added to every state value. That means, by adding a constant to the immediate rewards, all the states are shifted equally in value, and hence, the semantics of the value function does not change (the policies extracted from the value function depends on the difference between the values, not their absolute values.)

Exercise 3.11

In an episodic task immediate rewards are summed to the end of the episode (no need for a discounting rate, or equivalently, $\gamma = 1$) :

$$R_t = \sum_{k=0}^{T} r_{t+k+1}$$

where T is the length of the episode, which may be variable.
By adding a constant, C, to the immediate rewards in an episodic task:

$$\begin{aligned} R'_t &= \sum_{k=0}^{T} (r_{t+k+1} + C) \\ &= \sum_{k=0}^{T} r_{t+k+1} + \sum_{k=0}^{T} C \\ &= R_t + T \cdot C \end{aligned}$$

The value of every state would be shifted with a value that is related to the length of the episode. For this, the agent will prefer the states that lead to longer episodes rather than the shorter ones. For example, an agent would prefer to stay in a maze, or it would prefer Tic-Tac-Toe plays which lead to winning states after the maximum number of turns.

III. Policy Evaluation

Exercise 4.1

$$\begin{aligned} Q^\pi(s,a) &= \sum_{s'} P^a_{ss'}[R^a_{ss'} + V^\pi(s')] \qquad \text{(episodic task)} \\ &= R^a_{ss'} + V^\pi(s') \\ Q^\pi(11, down) &= 1 \times (-1 + 0) = -1 \\ Q^\pi(7, down) &= 1 \times (-1 + (-14)) = -15 \end{aligned}$$

Exercise 4.2

Since the transitions of the original states are unchanged, no change would occur to any of the original states.

$$\begin{aligned} V^\pi(15) &= 0.25 \times (-1 + V^\pi(13)) + 0.25 \times (-1 + V^\pi(14)) + \\ &\quad 0.25 \times (-1 + V^\pi(12)) + 0.25 \times (-1 + V^\pi(15)) \\ &= -15 - 0.25 \times V^\pi(15) \\ V^\pi(15) &= -20 \end{aligned}$$

When the transitions are updated and state 13 starts leading to state 15, its value should change accordingly (and then the values of the neighbours of

13 , and then every single state.) Fortunately, because action *down* in state 13 used to lead to a state with value-20 (state 13 itself), which is equal to the new value of state 15, the value of that action would not change, and hence nothing have to change and the values are stable.

Exercise 4.3

$$\begin{aligned} Q^{\pi}(s,a) &= \sum_{s'} P^{a}_{ss'}[R^{a}_{ss'} + \gamma \sum_{a'} \pi(s',a') Q^{\pi}(s',a')] \\ Q_{k+1}(s,a) &= \sum_{s'} P^{a}_{ss'}[R^{a}_{ss'} + \gamma \sum_{a'} \pi(s',a') Q_{k}(s',a')] \end{aligned}$$

IV. Grid world example

See MATLAB code for this question.