8 Reinforcement Learning

8.1 Questions

Exercise 8.1

Tell if the following statements are true or false and provide the adequate motivations to your answer.

1. In RL we do not require to have the model of the environment;
2. In RL we do not represent the model of the environment;
3. We need to use data coming from the optimal policy if we want to learn it;
4. Since RL sequentially decide the action to play at each time point, we cannot use information provided by historical data;
5. We can manage continuous space with RL.

Exercise 8.2

Tell if the following properties hold for MC or TD and motivate your answers.

1. Can be applied to infinite horizon ML;
2. Can be applied to indefinite horizon ML;
3. Needs an entire episode;
4. Works step by step (online);
5. Applies bootstrap;
6. The number of samples depends on the dimension of the MDP;
7. The number of samples depends on the length of the episodes;
8. Solves the prediction problem;

9. Reuse the information learned from past learning steps;
10. Makes use of the Markov property of the MDP;
11. Has no bias;
12. Has some bias.

Exercise 8.3

Tell if the following statements are true or false and motivate your answers.

1. With MC estimation you can extract a number of samples for the value function equal to the length of the episode you consider for prediction;
2. Generally, every-visit estimation is better if you use a small amount of episodes;
3. Stochasticity in the rewards requires the use of a larger number of episode to have precise prediction of the MDP value in the case we use MC estimation;
4. MC estimation works better than TD if the problem is not Markovian.

Exercise 8.4

Tell if the following statements are true or false and motivate your answers.

1. To compute the value of a state TD uses an approach similar to the one used in the Policy Evaluation algorithm;
2. TD updates its prediction as soon as a new tuple (state, action, reward, next state) is available;
3. TD cannot be used in the case there is no terminal state in the original MDP;
4. Since with TD we use values computed by averaging, we introduce less variance in the estimation than MC.

Exercise 8.5

Evaluate the value for the MDP with three states $\mathcal{S} = \{A, B, C\}$ (C is terminal), two actions $\mathcal{A} = \{h, r\}$ given the policy π, given the following trajectories:

$$(A, h, 3) \to (B, r, 2) \to (B, h, 1) \to (C)$$
$$(A, h, 2) \to (A, h, 1) \to (C)$$
$$(B, r, 1) \to (A, h, 1) \to (C)$$

1. Can you tell without computing anything if by resorting to MC with every-visit and first-visit approach you will have different results?
2. Compute the values with the two aforementioned methods.
3. Assume to consider a discount factor $\gamma = 1$. Compute the values by resorting to TD? Assume to start from zero values for each state and $\alpha = 0.1$.

Exercise 8.6

Comment on the use of α in the stochastic approximation problem to estimate an average value:

$$\mu_i = (1 - \alpha_i)\mu_{i-1} + \alpha_i x_i$$

Is $\alpha_i = \frac{1}{i}$ a valid choice? Is $\alpha = \frac{1}{i^2}$ meaningful?

Exercise 8.7

Consider the following problems and tell when the optimal policy can be found by resorting to RL or DP techniques:

1. Maze Escape
2. Pole balancing problem
3. Ads displacement
4. Chess

Exercise 8.8

Tell if the following statements are true or false.

1. To converge to the optimal policy we can even use MC estimation and a greedy policy;
2. To ensure convergence we should ensure that all the states are visited during the learning process;

3. It is not possible to learn the optimal policy by running a different policy on an MDP;
4. Information gathered from previous experience can not be included in the RL learning process.

Provide adequate motivations for your answers.

Exercise 8.9

You want to apply RL to train an AI agent to play a single-player videogame. The state of the game is fully observable and, at each step, the agent has to select an action from a discrete set of possibilities. The interaction ends as soon as the agent reaches the end of the level or fails. To optimize the policy for your AI, you have a set of recorded trajectories (i.e., sequences of state, action, and reward) of the AI agent playing the game following a suboptimal policy. Unfortunately, most of these trajectories are not complete (i.e., they do not cover all the interactions from the beginning of the level to either the end, or to a game-over state).

Indicate if the following methods can be applied to this problem, motivating your answer.

1. Monte Carlo Policy Iteration;
2. Value Iteration;
3. Sarsa;
4. Q-Learning.

Exercise 8.10

Consider the following snippet of code and answers to the questions below providing adequate motivations.

```
while m < M:
  ns, r = env.transition_model(a)
  na = eps_greedy(s, Q, eps)
  Q[s, a] = Q[s, a] + alpha * (r + env.gamma * Q[ns, na] - Q[s, a])
  m = m + 1
  s = ns
  a = na
```

1. What algorithm is this code implementing? What kind of problem is it addressing?
2. Explain the operations performed by the `eps_greedy` function.

3. What conditions do we need on `alpha` and `eps` to make the algorithm converge to a desirable solution?

4. How can we modify Line 4 to make the algorithm work off-policy?

Exercise 8.11

Consider the following episode obtained by an agent interacting with an MDP having two states $\mathcal{S} = \{A, B\}$ and two actions $\mathcal{A} = \{l, r\}$,

$$(A, l, 1) \to (A, l, 1) \to (A, r, 0) \to (B, r, 10) \to (B, l, 0) \to (A, r, 0) \to (B, l, 0) \to (A).$$

Answer to the following questions providing adequate motivations.

1. Execute the *Q-learning* algorithm on the given episode considering initial state-action values $Q(S, a) = 0$ for every state-action pair, learning rate $\alpha = 0.5$, and discount factor $\gamma = 1$.

2. Provide the best policy according to the output of *Q-learning*.

3. Do you think that the agent fully exploited the policy learned in the episode above? Make a consistent guess with the available information.

Exercise 8.12

We are given an Heating, Ventilation, and Air Conditioning (HVAC) in which the states are cold (c), medium (m), warm (w) temperature. We can perform three actions: heat (h), refrigerate (r), and do nothing (d). Assume to have the following partial episodes for the HVAC functioning.

$$(c, d, 0) \to (c, h, 1) \to (m, h, 1) \to (m, h, -1) \to (w, r, 1) \to (m, \cdot, \cdot) \to \dots$$
$$(m, r, -2) \to (c, h, -2) \to (c, h, 1) \to (m, h, 1) \to (m, h, 1) \to (w, \cdot, \cdot) \to \dots$$

where a tuple (S, A, R) correspond to the State, Action, and Reward at a specific time.

1. Model it as an MDP and draw the corresponding graphical representation, specifying the transition probabilities and rewards (estimated from the episodes) for each transition.

2. Can you tell if the reward of this process is stochastic or deterministic? And what about the transitions?

3. Assuming we want to evaluate the performance of the HVAC, tell which kind of problem we are in and suggest a technique to solve it.

Solutions

Answer of exercise 8.1

1. TRUE This is the difference from dynamic programming approaches;
2. FALSE In model based RL we do not have a known model of the environment, but we built an approximation of the model of the environment by basing on the data. This statement is true for model free RL approaches, which does not learn an MDP explicitly;
3. FALSE It is possible to learn also from the non optimal policies (off-policy RL);
4. FALSE For instance, in the case of model-free prediction we might use data coming from previously executed policies and process them as a single batch;
5. TRUE We might resort to function approximation if the state space is continuous or if the complexity of the problem does not allow us to solve the problem.

Answer of exercise 8.2

1. TD since it might operate even without having complete episodes;
2. MC since is able to use episodes with different length and TD since it might be updated after each transition;
3. MC (and TD(∞)) since it needs the overall reward at the end of the episode;
4. TD since it updates the value function after each instantaneous reward;
5. TD since it makes computes the estimates by using the information learned so far. This also applies to MC in its incremental version with $\alpha \neq \frac{1}{n}$, n being the number of states processed so far;
6. MC (first-visit) since you will get only one sample per state in each episode (assuming to have fewer states than transitions in an episode);
7. TD and MC (every-visit) since the more the episodes are long the more the sample we have;
8. MC and TD since both returns the value for each state given the observed policy;
9. TD since it uses a bootstrap strategy (and MC with $\alpha \neq \frac{1}{n}$);
10. TD since it explicitly uses it to compute the state value function;

11. MC (first-visit) since the computed value is an unbiased estimator of the state value function;

12. TD and MC (every-visit) since they make use of biased state values and dependent samples, respectively.

Answer of exercise 8.3

1. TRUE/FALSE With every-visit we have a sample per states in the episode, while if we consider first-visit we will count the occurrence of a state only once, thus, we have at most a number of samples equal to the number of states;

2. TRUE Even if it is a biased estimator for the expected value of a state, by resorting to every-visit MC we are gathering more samples, thus, we reduce the variance, which in the case we have only few samples is crucial;

3. TRUE The presence of stochasticity increases the variance of the estimates, thus, the estimated values are more and more uncertain;

4. TRUE MC does not use any information about the transition model of the MDP, thus no assumption on the transition model is considered when using MC. Instead, TD explicitly makes use of the Markovian property of MDPs.

Answer of exercise 8.4

1. TRUE The estimated return in the TD equation is nothing else but the right hand side of the Bellman expectation equation in the case we have a single action and we know the transition we perform;

2. TRUE It requires only a single transition to update the value of a state, while for instance MC needs to have a complete episode before updating the estimation of the values of the MDP;

3. FALSE Since its update are performed at each transition, we do not require to have finite episodes. This is different from MC for which we need to wait for the end of the episode;

4. TRUE By considering TD we are introducing some bias (in the estimates of the state values), but the fact that they are computed by averaging more transitions (i.e., we are considering multiple state, action, reward tuples) decreases the variance of the estimates. Moreover, we know that the estimate is consistent, thus, if we consider an infinite number of transitions it becomes unbiased.

Answer of exercise 8.5

1. Since we have multiple instances of the same state in a single episode and their value is not the same, the two approach will provide different results.

2. Every-visit MC:

$$V(A) = \frac{6+3+1+1}{4} = \frac{11}{4}$$
$$V(B) = \frac{3+1+2}{3} = 2$$

   First-visit MC:

$$V(A) = \frac{6+3+1}{3} = \frac{10}{3}$$
$$V(B) = \frac{3+2}{2} = \frac{5}{2}$$

3. TD:

$$V(A) \leftarrow V(A) + 0.1(3 + V(B) - V(A)) = 0 + 0.1(3 + 0 - 0) = 0.3$$
$$V(B) \leftarrow V(B) + 0.1(2 + V(B) - V(B)) = 0 + 0.1(2 + 0 - 0) = 0.2$$
$$V(B) \leftarrow V(B) + 0.1(1 + V(C) - V(B)) = 0.2 + 0.1(1 + 0 - 0.2) = 0.28$$
$$V(A) \leftarrow V(A) + 0.1(2 + V(A) - V(A)) = 0.3 + 0.1(2 + 0.3 - 0.3) = 0.5$$
$$V(A) \leftarrow V(A) + 0.1(1 + V(C) - V(A)) = 0.5 + 0.1(1 + 0 - 0.5) = 0.55$$
$$V(B) \leftarrow V(B) + 0.1(1 + V(A) - V(B)) = 0.28 + 0.1(1 + 0.55 - 0.28) = 0.407$$
$$V(A) \leftarrow V(A) + 0.1(1 + V(C) - V(A)) = 0.55 + 0.1(1 + 0 - 0.55) = 0.595$$

Answer of exercise 8.6

We know that if $\sum_{i \geq 0} \alpha_i = \infty$ and $\sum_{i \geq 0} \alpha_i^2 < \infty$ the estimator μ_i is consistent, thus, it converges to the real mean of the approximating problem when $i \to \infty$.

Thus the former choice $\alpha_i = \frac{1}{i}$ will provide a consistent estimator, while the latter $\alpha_i = \frac{1}{i^2}$ will be not guarantee to converge.

Answer of exercise 8.7

Any time you are able to use DP for solving a problem you might also consider to find the corresponding approximate solution with RL, thus, DP $\Rightarrow$ RL.

1. RL: we do not have the complete knowledge of the state we are in, thus we can not resort to DP.

2. DP and RL: we have complete information about the transition model and of the reward function. We might resort to RL if we consider continuous action space, which is usually difficult to handle with DP.

3. RL: in this case the transition is known (single state), but we do not know the reward associated to each action. In the specific, we might use MAB techniques to solve it.

4. RL: we have complete information, given a fixed strategy of the opponent, but usually we do not resort to DP for computational complexity issues.

Answer of exercise 8.8

1. FALSE If we use a greedy policy it might happen that we are not exploring some actions at all, while they are the optimal ones. This is especially true if the reward you gain from the states are stochastic.

2. TRUE The GLIE property requires that if we are considering an arbitrarily long learning process it will visit all the states an infinite number of times:

3. FALSE If we consider an off-policy learning method we are able to converge to the optimal policy even if we do not run the learned policy;

4. FALSE If we consider an off-policy learning method we might extract information about the optimal policy even if we consider data coming from sub-optimal ones.

Answer of exercise 8.9

To provide a solution for the described scenario we need to solve a control problem. We need to use an online method since we do not have trajectories which are complete. Moreover, one might not resort to a dynamic programming approach, since we do not have a full description of the environment, but only trajectories. Finally, we need to have an off-policy method since we can only rely on trajectories collected in the past. Therefore:

1. Not a viable option since Monte Carlo requires complete trajectories;

2. Not a viable option since we would require a complete description of the environment;

3. Not a viable option since it requires to follow the policy provided by it. (Using importance sampling might be a solution)

4. Viable option since it is off-policy, online and uses only trajectories.

Answer of exercise 8.10

1. This snippet of code is implementing the main loop of the `SARSA` algorithm, which tackles the Reinforcement Learning control problem.

2. The `eps_greedy` function is implementing an epsilon greedy policy. Thus, it returns the action that maximizes the `Q` value in `s` with probability 1 - `eps`, or a random action with probability `eps`.

3. To ensure that the algorithm will eventually converge to the optimal policy, one should take `eps` that goes to zero, and a learning rate `alpha` that follows Robbins-Monro conditions.

4. We could change the update rule to implement the `Q-learning` algorithm, i.e.,

   ```
   Q[s, a] = Q[s, a] + alpha * (r + env.gamma * np.max(Q[ns, :])
   - Q[s, a]),
   ```

   or apply the importance sampling correction to the samples we have.

Answer of exercise 8.11

1. We start with the initial values $Q(A,l) = Q(A,r) = Q(B,l) = Q(A,r) = 0$. Then, we compute the *Q-learning* update $Q(S_t,a_t) = (1-\alpha)Q(S_t,a_t) + \alpha(R_t + \gamma \max_{a\in\{l,r\}} Q(S_{t+1},a))$ for every step t in the episode:

   a) $Q(A,l) \leftarrow 0.5Q(A,l) + 0.5(1 + Q(A,\cdot)) = 0.5\cdot 0 + 0.5\cdot 1 = 0.5$

   b) $Q(A,l) \leftarrow 0.5Q(A,l) + 0.5(1 + Q(A,l)) = 0.5\cdot 0.5 + 0.5\cdot 1.5 = 1$

   c) $Q(A,r) \leftarrow 0.5Q(A,r) + 0.5(0 + Q(B,\cdot)) = 0.5\cdot 0 + 0.5\cdot 0 = 0$

   d) $Q(B,r) \leftarrow 0.5Q(B,r) + 0.5(10 + Q(B,\cdot)) = 0.5\cdot 0 + 0.5\cdot 10 = 5$

   e) $Q(B,l) \leftarrow 0.5Q(B,l) + 0.5(0 + Q(A,l)) = 0.5\cdot 0 + 0.5\cdot 1 = 0.5$

   f) $Q(A,r) \leftarrow 0.5Q(A,r) + 0.5(0 + Q(B,r)) = 0.5\cdot 0 + 0.5\cdot 5 = 2.5$

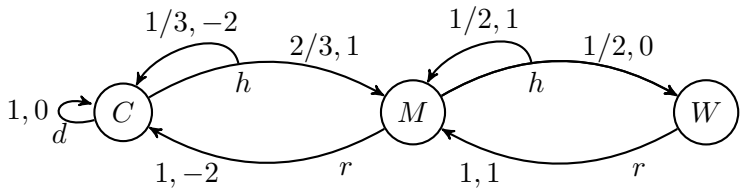
   g) $Q(B,l) \leftarrow 0.5Q(B,l) + 0.5(0 + Q(A,l)) = 0.5\cdot 0.5 + 0.5\cdot 1 = 0.75$

   which results in $Q(A,l) = 1, Q(A,r) = 2.5, Q(B,l) = 0.75, Q(B,r) = 5$.

2. We select the greedy policy $\pi(S) \in \arg\max_{a\in\{l,r\}} Q(S,a)$ w.r.t. the state-action values obtained with *Q-learning*, which gives $\pi(A) = r, \pi(B) = r$.

3. Since at the steps $(c), (e), (f), (g)$ the agent is not choosing the action that is maximizing the current estimate of the Q values, we can infer that the episode does involve some exploration.

Answer of exercise 8.12

1. 

2. The transition are stochastic since some of the actions are leading to two different new states, e.g., the action heat (h) in the state cold (c). The reward is stochastic as well, since heating in the medium state provided once the reward 1 and once -1.

3. This setting suggests it is an MDP prediction problem, either using directly the original episodes (using MC or TD) or one might use the estimated model and use DP techniques to solve it in an exact way, due to the limited dimension of the problem.