Reinforcement Learning – Final exam

Nicolas Gast Panayotis Mertikopoulos

January 2022 (academic year 2021-2022)

Duration: 2 hours.

Please justify carefully your answer (try to be **concise** and **precise**). The grading scale is given as an indication.

Exercise 1 Bandit, 3 points

You consider a 3-arm Bernoulli bandit. We denote by p_i the probability that the arm i provides a reward 1 when chosen.

1. *(2 points)* You observe the episode:

Time	0	1	2	3	4	5	6	7	8	9
Choice	0	1	2	0	1	1	0	2	0	0
Reward	1	1	0	0	1	0	1	0	1	1

   a) Express the expected regret at time $t = 10$ as a function of the p_is.
   b) Suppose that you are using ε-greedy with $\varepsilon = 0.1$. What is the probability of choosing arms 0, 1 or 2 at time 10?
   c) Suppose that you are using UCB with a bonus $\sqrt{2 \ln t / N_i(t)}$, where $N_i(t)$ if the number of time that you choose this arm before time t. What will UCB choose at time 10? Indication: $\ln(10) \approx 2.3$.

2. *(1 point)* For both of the above policy, what is $\lim_{t\to\infty} N_i(t)/t$? Use it to explain the fundamental difference (in terms of regret) between UCB and ε-greedy.

Exercise 2 Black-jack, 7 points

We consider a simplified version of the Blackjack. You draw cards one by one from an infinite deck. The deck contains cards 2 to 10, J, Q, K and A. Each card is equally likely to be drawn each time. The "value" of a card is:

- The number shown on the card for cards 2 to 10.
- 10 for the cards J, Q and K.
- 11 for the cards A.

At each turn you may "hit" or "stay". If you hit, you draw a new card (and receive no immediate reward). If you stay, then you sum the values of your card. If this sum is 15, you earn 0. If it is

between 16 and 21, you earn 10. If it is (strictly) lower than 15 or (strictly) larger than 21, your reward is -10. Note that when the sum is larger than 21, the action "hit" is not available.

We propose to model the problem as a Markov decision process, with states $\mathcal{S} = \{0, 2, 3 \ldots, 21, \geq 21, \text{end}\}$, where end is a terminal state that you enter after choosing the action "stay". We consider a discount factor γ.

1. *(1 point)* What are the reward $R(s, a)$ and the transition probabilities $P(s'|s, a)$ for $s = 12$, $a \in \{hit, stay\}$ and $s' \in \mathcal{S}$?
2. *(2 points)* Suppose that you initialize your value function $V_0(s)$ with the following table (for $s \in \{12 \ldots 21, \geq 21, \text{end}\}$):

12	13	14	15	16	17	18	19	20	21	≥ 21	end
0	0	0	0	0	0	0	10	10	10	0	0

   a) What is the corresponding Q-table, assuming that $\gamma = 0.5$?
   b) You perform one iteration of value iteration. Write down the table of $V_1(s)$.
   c) What is the optimal policy? (please justify carefully). Note: you do not have to compute its value.
3. *(2 points)* We now suppose that you do not know the transition probabilities. You are using Q-learning with a learning rate α and a discount factor γ.
   a) Recall the Q-learning update equation, and use it to explain how does Q-learning work.

   At some point in time, your are in state 10, the Q-table has the following values, and you observe the end of the episode (which contains 3 steps):

s	19	20	21	≥ 21
stay	5	6	7	8
hit	-5	-6	-7	N/A

Q-table

S	A	R	S	A	R	S	A	R
19	hit	0	21	hit	0	≥ 21	stay	-10

Episode

   b) Assume that $\alpha = 0.2$, $\gamma = 0.5$. Write down the value of the Q-table at the end of the episode. Indicate only the values of the table that changed (use the value "–" to indicate a value that did not change).

* 4. *(2 points)* We now consider a two player games in which: you draw your cards as before and stop as before. The difference is that when you decide to stay, you do not immediately earn a reward but wait for the dealer to play. The later draws cards (the same way as you) and wins +10 if they obtain a score strictly higher than you. As before, if a player obtains a sum strictly larger than 21, then the game is lost for this player.
   a) Explain how to design an algorithm that computes the optimal solution, assuming to know the probability of drawing cards.
   b) If you do not know the probabilities, could you use a Q-learning approach? Explain how to do it.