Universitat Politècnica de Catalunya
Computer Science Dept.
Master in Artificial Intelligence

Reinforcement Learning Quiz (ATCI course - MAI)

Mario Martin

Last Revision Date: April 12, 2024
mmartin@cs.upc.edu

Instructions:

1. **Make sure that you are answering the correct file**, it has to correspond with the identification number you gave when you enrolled the course.
2. **Enter your name** in the box at the beginning of the quiz.
3. You have to open this file using Acrobat Reader in order to fill the questionnaire. Click on 'Begin Quiz' to initialize the quiz. When finished, click on 'End Quiz' **and save the file**. Opening the saved file at any moment **should** maintain your previous answers (try this in your software before starting to work hard on the quiz). You can send me this file with answers filled (check always, before sending the file, that your answers are there!). *I recommend you always have a copy on paper of your solutions because errors can happen at any moment and you could lose all the work done.*

   Alternatively, you can edit the PDF file using your favourite PDF editor, but make sure that the answers you mark are visible.

   If all else fails, you can send me a file containing for each number of question a list of the options checked.
4. This quiz is about the reinforcement learning part of the course. Each question has a value of 0.65 points (yes, I know... 0.4 over 10 bonus ... you're welcome). Each question has an unknown number of correct answers and each **incorrect answer will discount** accordingly to the probability of choosing it randomly so the expectation of a randomized answer is 0. Choose carefully.
5. You have to upload the printed file with your answers to the *Racó* in the entry corresponding to the questionnaire in the *Practicals* section before **April 19th at 23:59**.
6. The grade of this evaluation act has a weight of 20% on the final mark of the course. You can consult your notes, slides, etc. to answer it. The goal is to help to ask yourself if you have the knowledge you thought on the theory of RL and find the answers by yourself using the materials of the course (of even with the help of Google). Some of the questions are tricky because that reason.

Enter your name here: Alberto Becerra Tomé

Begin Quiz Answer each of the following.

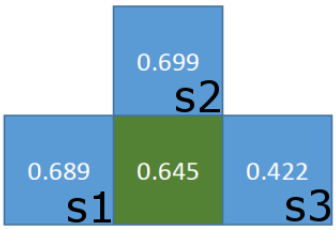

Figure 1: Figure for problem below

1. Figure 1 represents 3 states of a larger grid. You have learned the Value of each state, in particular, you have learned the values around the green state. Which will be the greedy policy for the green state given that there are three 4 possible actions: $\leftarrow, \rightarrow, \uparrow$ and that probability transitions are the following:

$$\begin{array}{lll} P(s_1|\text{green},\leftarrow) = 0.8, & P(s_1|\text{green},\rightarrow) = 0.0, & P(s_1|\text{green},\uparrow) = 0.1 \\ P(s_2|\text{green},\leftarrow) = 0.1, & P(s_2|\text{green},\rightarrow) = 0.1, & P(s_2|\text{green},\uparrow) = 0.8 \\ P(s_3|\text{green},\leftarrow) = 0.0, & P(s_3|\text{green},\rightarrow) = 0.8, & P(s_3|\text{green},\uparrow) = 0.1 \\ P(\text{green}|\text{green},\leftarrow) = 0.1, & P(\text{green}|\text{green},\rightarrow) = 0.1, & P(\text{green}|\text{green},\uparrow) = 0.0 \end{array}$$

(a) $\pi(\text{green}) = \rightarrow$
(b) $\pi(\text{green}) = \uparrow$
(✔) $\pi(\text{green}) = \leftarrow$

2. Assume you have to train with RL techniques a robot that in a tennis court collects balls as soon as possible during the rest time between two sets of a match. How would you define the reward function?[1] (Select all correct choices)

(a) Infinite horizon undiscounted long-term return with +1 reward each time he takes a ball, 0 otherwise.
(✔) Finite horizon discounted long-term return with $\gamma = 1$ and reward +1 each time he takes a ball, and -1 at each time step.
(c) Finite horizon discounted long-term return with $\gamma = 0.9$ and reward +1 each time he takes a ball, and -1 at each time step.
(d) Infinite horizon discounted long-term return with $\gamma = 0.9$ and reward +1 each time he takes a ball, 0 otherwise.

3. Mark the true sentences:

(✔) From only the Q value function we can compute the greedy policy for Q.
(✔) Consider a reward function R and the optimal policy π for that function. If we define a new reward function R' built adding a constant $c \in \mathbb{R}$ to R for each state, the optimal policy will be also π.

[1] Use long-term definitions of reward that make sense. Some of the combinations could work but are overcomplicated with unnecessary assumptions for the problem at hand. Don't mark them.

(c) From only the V value function we can compute the greedy policy for V.

(d) ✔ Consider a reward function R and the optimal policy π for that function. If we define a new reward function R' built multiplying R by constant $c > 0$ for each state, the optimal policy will be also π.

4. Mark problems that can be represented as a standard MDP (Select all correct choices)

(a) Agent learning to control the voltage of a DC motor where the goal is to achieve a given constant number of revolutions per minute as soon as possible starting from the rest state. Assume the motor has the necessary sensors.

(b) ✔ Agent learning to play Othello game against a fixed program opponent that only considers current state of the board, with actions the possible legal moves on the game and input the current state of the board.

(c) Solving the Atari game of Pong with actions being the 4 possible buttons to play in that game and input only the current frame displayed on the screen.

(d) Program learning to play poker with legal actions of the game and input the current hand of cards.

5. Select the true sentences:

(a) ✔ TD(λ) is an on-policy learning method.

(b) ✔ When $\epsilon = 0$, Q-learning and Sarsa are the same algorithm.

(c) Off-policy learning is better than on-policy when there is a lot of stochasticity in the environment transitions between states.

(d) Expected Sarsa can only be used with ϵ greedy exploration.

6. Considering learning in the tabular case (without function approximation), select the true sentences:

(a) Q-values learned using Q-learning and Monte-Carlo using ϵ-greedy exploration ($\epsilon = 0.01$), converges to the same limit when we give enough experiences for learning.

(b) On-policy learning is faster than off-policy learning.

(c) ✔ Given a large enough set of tuples s, a, r, s' obtained choosing always random actions, we can compute the $Q^\pi(s, a)$ for a given deterministic policy π

(d) ✔ Sarsa with fixed ϵ-greedy exploration ($\epsilon > 0$) cannot learn the optimal policy for an MDP.

7. Select the true sentences:

(a) Exceptionally, exploration probability of one action a in one state s can be set to zero when $Q(s, a) < 0$.

(b) ✔ Exploration is necessary to guarantee convergence of Monte Carlo.

(c) ✔ Boltzman exploration is different from ϵ-greedy because the former automatically adapts to learnt experience.

(d) In ϵ-greedy exploration, it is mandatory that ϵ should be constant.

8. Select the true sentences:

(a) ✔ Q-learning has less variance in estimations of $Q(s, a)$ than Monte Carlo.

(b) Monte Carlo, estimation of $Q(s, a)$ is done applying Bellman equations.

(c) Monte Carlo is different form Q-learning in the sense that Monte Carlo does not need to apply exploration to find the optimal policy while Q-learning needs exploration.

(d) Monte Carlo policy evaluation method can be done in continuous learning if we apply the adequate discounting long-term return.

9. Select the true sentences:

(a) ✔ Batch methods for reinforcement learning can be applied to on-policy methods like Sarsa.

(b) ✔ Tile coding in linear function approximation represent states using large sparse vectors with dimension a lot larger than the original space of variables.

(c) ✔ Tile coding allows generalization by moving space of input features to a larger dimensional space.

(d) ✔ Generalization in RL is needed mainly when the problem has a very large number of states that cannot be stored using a table.

10. In the linear function approximation framework, select the true sentences:

(a) Given a large as needed set of tuples s, a, r, s' obtained using random actions and using an on-line off-policy method (like Q-learning), we can learn best parameters θ that minimize the MSE error in linear function approximation.

(b) Only gradient descent methods can be applied to linear function approximation.

(c) ✔ Expected Sarsa algorithm converges when applying gradient descent with linear function approximation.

(d) Considering the properties of the *n-steps* algorithm, it should converge when applying gradient descent with linear function approximation.

11. Select the true sentences:

(a) ✔ A difference between *DQN* and *Neural Fitted Q-learning* is that the later cannot be used for on-line learning tasks.

(b) DQN does not need the property of markovian transitions to converge because it generates a non-linear approximation.

(c) DQN uses two copies of the DNN with different weights, one of them frozen during several experiences to solve the problem of *correlated data*.

(d) Replay buffer has a limited capacity to remove old samples s, a, r, s' because they were generated by old policies and rewards are not right.

12. Select the true sentences:

(a) ✔ An improvement over DQN is to use in the update of the weights two set of weights, one to compute the policy and another to compute the value of the policy.

(b) *Double Q-learning* is, in general, faster than DQN because it does not suffer from the *moving target*.

(c) ✔ *Dueling Network Architectures* work because learning for one action a in one state is propagated to other actions in the same state.

(d) ✔ *Prioritized Experience Replay* is faster because it chooses those examples in the Replay Buffer with higher error in the estimation of return.

13. Select the true sentences:

(a) ✔ Clipping is done to avoid large changes in the weights.

(b) ✔ DQN is an off-policy method.

(c) ✔ Looking at the results of *Rainbow*, it seems more useful the use of *n-step backups* than the *Double* technique.

(d) ✔ AC3 algorithm does not need a Replay Experience Buffer because using several workers we train the network with uncorrelated data.

14. Select the true sentences:

(a) α parameter in Policy gradient algorithms is critical because direction of the gradient is misleading in some cases.

(b) Actor-Critic methods do not need the assumption of markovian transitions.

(c) ✔ Pure *policy gradient* techniques do not need the assumption of markovian transitions.

(d) In order to evaluate the goodness of a policy π, we always need to estimate Q^π or V^π value functions.

15. In the algorithm PPO we use the following loss:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right]$$

What is r and why we need it?

Answer:

Answers to Q15 and Q16

Q15: In PPO, the term r is the ratio that compares the probability of selecting action a in state s under the current policy parameterized by theta with the probability of the same action under the previous policy. This ratio is key because it quantifies how much the current policy differs from the older one, helping to ensure that policy updates are not overly drastic, thus promoting stable and gradual learning. The Clipped Surrogate Objective function employs this ratio with a clipping range defined by eps. This approach has a dual purpose: it allows for controlled exploration by encouraging changes in the policy that yield positive outcomes, and it restricts extreme policy adjustments by constraining r to the range [1 - eps, 1 + eps], thus ensuring steady learning without excessive volatility.

16. In the Soft-Actor critic algorithm the loss contains also the *entropy* of the policy, so we want to find a policy that maximize the combination of expected long term reward and entropy. Why and which are the advantages of maximizing the entropy?

Answer:

Answers to Q15 and Q16

Q15: In PPO, the term r is the ratio that compares the probability of selecting action a in state s under the current policy parameterized by theta with the probability of the same action under the previous policy. This ratio is key because it quantifies how much the current policy differs from the older one, helping to ensure that policy updates are not overly drastic, thus promoting stable and gradual learning. The Clipped Surrogate Objective function employs this ratio with a clipping range defined by eps. This approach has a dual purpose: it allows for controlled exploration by encouraging changes in the policy that yield positive outcomes, and it restricts extreme policy adjustments by constraining r to the range [1 - eps, 1 + eps], thus ensuring steady learning without excessive volatility.

End Quiz