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

Reinforcement Learning Quiz (URL course - MAI)

Mario Martin

Last Revision Date: June 4, 2018
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, like you did for the quiz of the unsupervised learning part.
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 0.75 points (yes, I know... 0.5 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 before **June 12th at 23:59**.

Enter your name here: Josep Famadas Alsamora

Begin Quiz Answer each of the following.

1. Mark the true sentences:

(a) From only the V^π value function for each state for a policy π, we can compute the Q^π value function for each pair of state-action.

(b) 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.

(c) From state s_1, taking action a_1 always produces a reward of 2 and sends you to a state s_2 from which a long-term return of 10 is always received. The discount parameter γ is 0.9. In this case, we know for sure that $Q(s_1, a_1)$=11.

(d) If a policy π is greedy with respect to the value function for the equiprobable random policy, then it is an optimal policy.

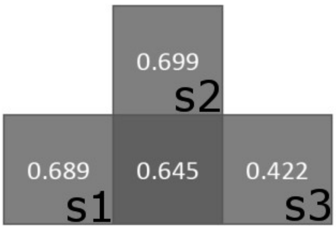

Figure 1: Figure for problem below

2. 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 policy for the green state given that there are three 4 possible actions: ←, →, ↑ and that probability transitions are the following:

$$\begin{aligned}
P(s_1|\text{green}, \leftarrow) &= 0.9, & P(s_1|\text{green}, \rightarrow) &= 0.0, & P(s_1|\text{green}, \uparrow) &= 0.05\\
P(s_2|\text{green}, \leftarrow) &= 0.05, & P(s_2|\text{green}, \rightarrow) &= 0.05, & P(s_2|\text{green}, \uparrow) &= 0.9\\
P(s_3|\text{green}, \leftarrow) &= 0.0, & P(s_3|\text{green}, \rightarrow) &= 0.9, & P(s_3|\text{green}, \uparrow) &= 0.05\\
P(\text{green}|\text{green}, \leftarrow) &= 0.05, & P(\text{green}|\text{green}, \rightarrow) &= 0.05, & P(\text{green}|\text{green}, \uparrow) &= 0.0
\end{aligned}$$

(a) $\pi(\text{green}) \; = \uparrow$

(b) $\pi(\text{green}) \; = \leftarrow$

(c) $\pi(\text{green}) \; = \rightarrow$

3. Assume you have to train with RL techniques a robot on an assembly line that has to remove defective elements from a continuous belt. How will you define the reward function?[1] (Select all correct choices)

(a) Finite horizon discounted long-term return with $\gamma = 0.9$ and reward +1 each time he removes a defective piece and -1 when he misses a defective piece or removes a faultless piece, 0 otherwise.

[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.

(b) Infinite horizon discounted long-term return with $\gamma = 0.9$ and reward +1 each time he removes a defective piece and -1 when he removes a faultless piece, 0 otherwise.

(c) Infinite horizon discounted long-term return with $\gamma = 0.9$ and reward +1 each time he removes a defective piece or allow to pass a faultless piece, 0 otherwise.

(d) Finite horizon discounted long-term return with $\gamma = 1$ and reward +1 each time he removes a defective piece and -1 when he removes a faultless piece, 0 otherwise.

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

(a) 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.

(b) Agent learning to play Go game against a *learning 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) Program learning to play poker with legal actions of the game and input the current hand of cards.

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

5. Select the true sentences:

(a) Monte Carlo is similar to Q-learning in the sense that both methods are on-policy.

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

(c) Q-learning obtains less biased estimations of $Q(s, a)$ than Monte Carlo.

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

6. Select the true sentences:

(a) *n-steps* policy learning is an on-policy learning method.

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

(c) Expected Sarsa can also be used with Boltzman exploration.

(d) Expected Sarsa shows less variance than Sarsa algorithm.

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

(a) Sarsa with fixed ϵ-greedy exploration cannot learn the optimal policy for an MDP.

(b) 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

(c) 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.

(d) On-policy learning is faster than off-policy learning

8. Select the true sentences:

(a) Exploration is done to make learning faster.

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

(c) In practical cases the value of α is equal for all states but it can decrease with learning experience.

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

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

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

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

10. Select the true sentences:

(a) A Good property of *linear* FA for Q-learning in policy evaluation (that makes it appealing) is that the gradient rule applied in this case is simple and exact.

(b) Linear function approximation cannot be used to represent problems with continuous variables.

(c) Batch methods solve the problem of the *moving target* by doing policy evaluation not only for the current state but for a larger number of samples.

(d) A Good property of *linear* FA for Monte Carlo in policy evaluation (that makes it appealing) is that it ensures convergence to the optimal policy because there is only a global optimal.

11. Select the true sentences:

(a) In practice, AC3 algorithm is faster than DQN because it more sample efficient (it takes more profit of each experience).

(b) In practice, AC3 algorithm is faster than DQN because (working several agents in parallel) it generates more samples in the same time.

(c) It has been proved in practice to use unmodified squared Bellman error as the Loss function train the DNN.

(d) In practice when an algorithm works well in one environment we don't need to run the algorithm on the environment again because we have shown that it works. We only have to repeat the experiment with another random seed when it is not able to learn to be sure of failure of the algorithm.

12. Select the true sentences:

(a) Advantage value $A(s, a)$ is an estimation of the advantage of action a over all other actions.

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

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

(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) 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.

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

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

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

14. Select the true sentences:

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

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

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

(d) Stochastic policies can represent policies for continuous action.

End Quiz