Reinforcement Learning for Robotics
Comparative Report
Manu Parashar

1. E-SARSA vs Q-Learning

E-SARSA or Expected SARSA is an on policy reinforcement learning algorithm. It is very similar to the SARSA algorithm covered in the course.

The only difference being that E-SARSA uses the expected value of the action values under the current policy as the target instead of choosing a random action under the policy in the case of SARSA.

Expected SARSA is much more computationally expensive than SARSA due to the need of calculating the expected value at each step, but E-SARSA is also a more stable algorithm than SARSA since it always updates in the direction of the expected value and therefore eliminating any variance from selecting a random action.

The policy learnt from E-SARSA is ε-greedy.

Q-Learning on the other hand is an off policy control algorithm in reinforcement learning.

Q-Learning always chooses the max action value (Q - Value) as the target in TD-Error. This is what makes it an off policy algorithm, since regardless of the behaviour policy the policy being learnt chooses the action which has the maximum action value.

The learnt policy in Q-Learning is greedy.

PERFORMANCE

1) Number of Episodes

```
EPSILON     = .1
NUM_EPISODES = 100
GAMMA       = .9
ALPHA       = .1
```

E-SARSA:
Unable to Learn the correct policy

Q-Learning:
Correct Policy Learnt
('maze_transitions Q-Learning ::', [2, 2, 0, 0, 0, 2])
('drone motion Q-Learning Q-Learning ::', [-1, 0, 1, 0, 0, -1])

```
EPSILON     = .1
NUM_EPISODES = 1000
GAMMA       = .9
ALPHA       = .1
```

E-SARSA
Correct Policy Learnt
('maze_transitions E-SARSA ::', [0, 0, 0, 2, 2, 2])
('drone motion E-SARSA ::', [0, 0, 0, -1, 0, 0])

Q-Learning:
Correct Policy Learnt
('maze_transitions Q-Learning ::', [2, 2, 0, 0, 0, 2])
('drone motion Q-Learning Q-Learning ::', [-1, 0, 1, 0, 0, -1])

CONCLUSION
Q-Learning is able to learn the optimal policy with much less number of episodes than SARSA

2) Epsilon

```
EPSILON     = .01
NUM_EPISODES = 1000
GAMMA       = .9
ALPHA       = .1
```

E-SARSA:
Unable to Learn the correct policy

Q-Learning:
Correct Policy Learnt
('maze_transitions Q-Learning ::', [0, 0, 0, 2, 2, 2])
('drone motion Q-Learning ::', [0, 0, 0, -1, 0, 0])

```
EPSILON     = .1
NUM_EPISODES = 1000
GAMMA       = .9
ALPHA       = .1
```

E-SARSA
Correct Policy Learnt
('maze_transitions E-SARSA ::', [0, 0, 0, 2, 2, 2])
('drone motion E-SARSA ::', [0, 0, 0, -1, 0, 0])

Q-Learning:
Correct Policy Learnt
('maze_transitions Q-Learning ::', [2, 2, 0, 0, 0, 2])
('drone motion Q-Learning Q-Learning ::', [-1, 0, 1, 0, 0, -1])

CONCLUSION:
Q-Learning, unlike E-SARSA is able to learn an optimal policy even with very little exploration.

3) Alpha - Learning Rate/Step Size

```
EPSILON     = .1
NUM_EPISODES = 1000
GAMMA       = .9
ALPHA       = .001
```

E-SARSA:
Unable to Learn the correct policy

Q-Learning:
Correct Policy Learnt
('maze_transitions Q-Learning ::', [0, 0, 0, 2, 2, 2])
('drone motion Q-Learning ::', [0, 0, 0, -1, 0, 0])

```
EPSILON     = .1
NUM_EPISODES = 1000
GAMMA       = .9
ALPHA       = .01
```

E-SARSA
Correct Policy Learnt
('maze_transitions E-SARSA ::', [0, 0, 0, 2, 2, 2])
('drone motion E-SARSA ::', [0, 0, 0, -1, 0, 0])

Q-Learning:
Correct Policy Learnt
('maze_transitions Q-Learning ::', [2, 2, 0, 0, 0, 2])
('drone motion Q-Learning Q-Learning ::', [-1, 0, 1, 0, 0, -1])

```
EPSILON     = .1
NUM_EPISODES = 1000
GAMMA       = .9
ALPHA       = .1
```

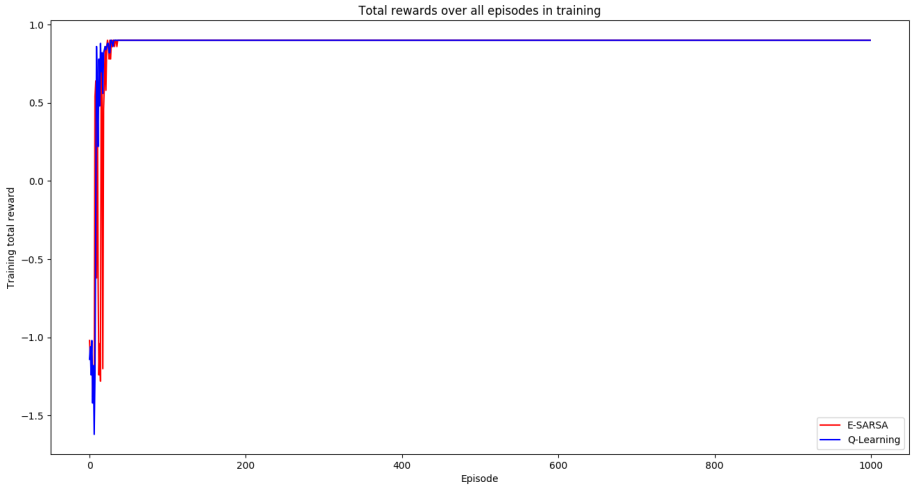
E-SARSA
Correct Policy Learnt
('maze_transitions E-SARSA ::', [0, 0, 0, 2, 2, 2])
('drone motion E-SARSA ::', [0, 0, 0, -1, 0, 0])

Q-Learning:
Correct Policy Learnt
('maze_transitions Q-Learning ::', [2, 2, 0, 0, 0, 2])
('drone motion Q-Learning Q-Learning ::', [-1, 0, 1, 0, 0, -1])

CONCLUSION:
Q-Learning, unlike E-SARSA is able to learn an optimal policy even with a very small learning rate. Which means Q-Learning converges faster than E-SARSA.

4) GRAPH

I put in a decay for epsilon, so that after a certain number of episodes the algorithms stop exploring.



It can be seen that even though both the algorithms converge to the optimal policy Q-Learning performs better in early episodes.

2. Suggestion

One suggestion to improve the course would be to go into some detail into Deep Reinforcement Learning. Deep RL combines RL and deep neural networks. It is one of those popular buzzwords in the field of computer science these days. It would be interesting to learn a little about it.