Planning, Learning and Decision Making

Group 27

78375 - João Pirralha

84758 - Rafael Ribeiro

Homework 5. Reinforcement learning

a)

In [1]:

```
import numpy as np

X = ("(4, b̄, r)", "(5, b̄, r)")
S4bR = 0; S5bR = 1
UP = 0; DOWN = 1; LEFT = 2; RIGHT = 3
GAMMA = 0.99
STEP_SIZE = 0.1
Q_t = np.array([[3.23, 3.34, 3.25, 3.22], [3.08, 3.25, 3.57, 3.22]])

# Slide 15 of lec21.pdf
def Q_learning_update(x_t, a_t, c_t, x_t1):
    result = np.copy(Q_t)
    result[x_t, a_t] = Q_t[x_t, a_t] + STEP_SIZE * (c_t + GAMMA * np.min
(Q_t[x_t1]) - Q_t[x_t, a_t])
    return result

print("Q-values after a Q-learning update resulting from the transition
at time step t:")
Q_t1_Q_learning = Q_learning_update(S5bR, UP, 1.0, S5bR)
for state, Q in zip(X, Q_t1_Q_learning):
    print(state + ":", Q)
```

```
Q-values after a Q-learning update resulting from the transition at time
step t:
(4, b̄, r): [3.23 3.34 3.25 3.22]
(5, b̄, r): [3.17692 3.25    3.57    3.22   ]
```

b)

In [2]:
```
# Slide 32 of lec21.pdf
def SARSA_update(x_t, a_t, c_t, x_t1, a_t1):
    result = np.copy(Q_t)
    result[x_t, a_t] = Q_t[x_t, a_t] + STEP_SIZE * (c_t + GAMMA * Q_t[x_
t1, a_t1] - Q_t[x_t, a_t])
    return result

print("Q-values after a SARSA update resulting from the transition at ti
me step t:")
Q_t1_SARSA = SARSA_update(S5bR, UP, 1.0, S5bR, RIGHT)
for state, Q in zip(X, Q_t1_SARSA):
    print(state + ":", Q)
```

```
Q-values after a SARSA update resulting from the transition at time step
t:
(4, b̄, r): [3.23 3.34 3.25 3.22]
(5, b̄, r): [3.19078 3.25    3.57    3.22   ]
```

c)

On-policy learning improves the same policy that is used to make exploration decisions, while off-policy learning improves a different policy than the one used to make exploration decisions. This results in on-policy conservatively learning a near-optimal policy instead of the optimal policy, unless policy improvement (e.g. ε-greedy with decaying ε) is used, while off-policy learns the optimal policy at the expense of more mistakes (whose cost may be significant).

Q-learning (1.a) is off-policy as the Q-values are updated based on the optimal (in this case minimizing) action for the next state. On the other hand, SARSA (1.b) is on-policy as the Q-values are updated using the performed action for the next state, which was chosen by the same policy being improved.