MDPs and RL

Useful Formulas

- Q-learning update: $Q_{k+1}(s,a)=Q_k(s,a)+\alpha\left(R_{t+1}+max_{a'}\gamma\, Q_k(s',a')-Q_k(s,a)\right)$.
- Sarsa update: $Q_{k+1}(s,a)=Q_k(s,a)+\alpha\left(R_{t+1}+\gamma Q(s',a')-Q_k(s,a)\right)$.

Questions

1. What is an MDP? What are the elements that define an MDP?

   A Markov Decision Process is a controllable stochastic process in which the next state and reward depend solely on the current state. It is a tuple $\langle S, A, T, r \rangle$ where S is a set of states, A is a set of actions, T is the transition function, and r is the reward function.

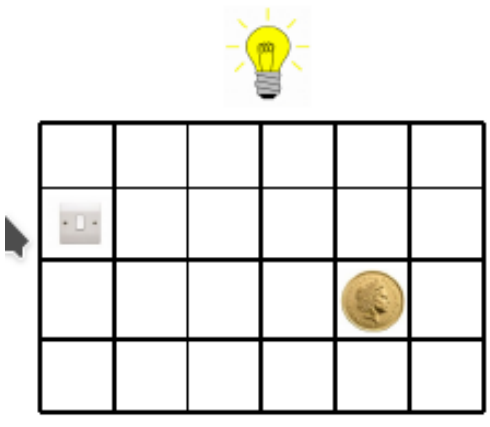
2. What makes a transition system Markovian?

   The distribution of the next state and the reward depend only on the current state, that is:
   $p(s_{t+1}=s, r_{t+1}=r|s_0, a_0, s_1, a_1, \ldots, s_t, a_t)=p(s_{t+1}=s, r_{t+1}=r|s_t, a_t)$

3. What does it mean that an RL method bootstraps? Provide an example of an RL algorithm that bootstraps and one that does not.

   An RL method bootstraps when it computes a prediction based on another prediction. TD methods bootstrap, for instance Q-Learning and Sara, while Monte Carlo does not.

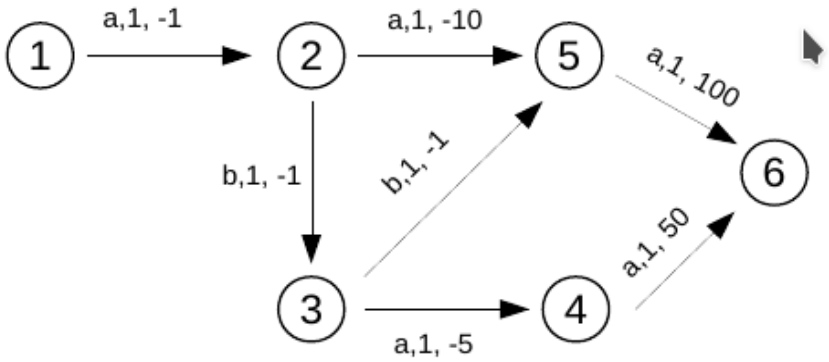
4. An agent has to find the coin in the MDP below, and pick it up. The actions available to the agent are move up, down, left, right, toggle switch and pick up. The action toggle switch turns on and off the light in the room, and succeeds only if executed in the square with the switch, while it does not do anything anywhere else. The action pick up picks up the coin if executed in the square with the coin and if the light is on, while does nothing anywhere else, or with the light off. How would you model this domain so that the representation is Markovian?



A state, in order for the representation to be Markovian, must have at least the following components: position, LightOn, HoldingCoin. The position can be represented in different ways, as long as there is one coordinate per square. LightOn and HoldingCoin are two boolean variables, that are true when the light is on, and when the agent is holding the coin respectively.

Note on notation: in the following MDPs, each state is labeled with an id. Each transition is labeled with the name of the corresponding action, the probability of landing in the next state, and the reward for that transition. If a state has no outgoing edges, it is an absorbing state.

5. Calculate the action-value function that Sarsa and Q-learning would compute on the following MDP, while acting with an ε-greedy policy with and ε = 0.1 and γ = 0.5.



The difference between the two algorithms is that Q-learning is off-policy, while Sarsa is on policy. Therefore, Sarsa converges to:

$$q_\pi(s,a)=\sum_{s',r} p(s',r|s,a)(r+\gamma\sum_{a'}\pi(a'|s)q_\pi(s',a'))\quad,$$

while Q-learning converges to:

$$q^*(s,a)=\sum_{s',r} p(s',r|s,a)(r+\gamma max_{a'}q^*(s',a'))\quad,$$

where the policy has been substituted by the max operator.

States where there is no choice are shared by the two algorithms. Applying the two equations above, we obtain:

for both Sarsa and Q-learning:

$$q(5,a)=100\quad,\quad q(4,a)=50\quad,\quad q(3,a)=-5+\gamma q(4,a)=-5+0.5\cdot 50=20\quad,$$
$$q(3,b)=-1+\gamma q(5,a)=-1+0.5*100=49\quad,$$
$$q(2,a)=-10+\gamma q(5,a)=-10+0.5\cdot 100=40\quad.$$

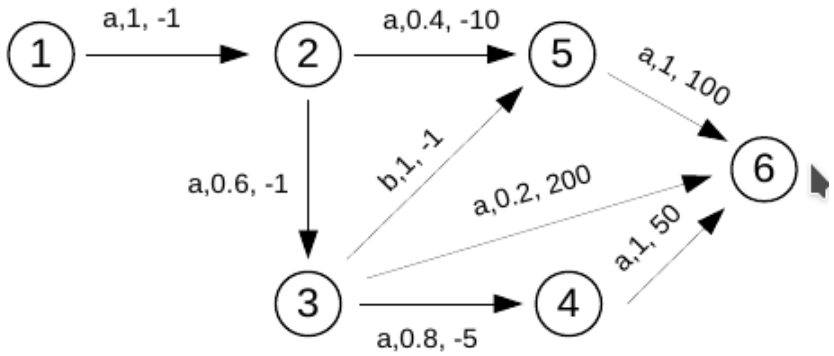
For Sarsa:

$$q(2,b)=-1+\gamma((1-\epsilon)q(3,b)+\epsilon q(3,a))=-1+0.5(0.9\cdot 49+0.1\cdot 20)=22.05\quad,$$
$$q(1,a)=-1+\gamma((1-\epsilon)q(2,a)+\epsilon q(2,b))=-1+0.5(0.9\cdot 40+0.1\cdot 22.05)=18.103\quad.$$

For Q-learning:

$q(2,b)=-1+\gamma q(3,b)=-1+0.5\cdot 49=23.5$, $q(1,a)=-1+\gamma q(2,a)=-1+0.5\cdot 40=19$. Note how Q-learning does not take the policy into account, and the values it computes are higher because they are the values under the optimal policy.

6. Calculate the action-value function that Q-learning and Sarsa would compute on the following MDP, with γ = 0.5 and ε=0.1.



For both Q-learning and Sarsa:
$q(5,a)=100$, $q(4,a)=50$,
$q(3,a)=0.8(-5+\gamma q(4,a))+0.2(200)=0.8(-5+0.5*50)+0.2\cdot 200=56$,
$q(3,b)=-1+\gamma q(5,a)=-1+0.5*100=49$.

For Sarsa:

$q(2,a)=0.4(-10+\gamma q(5,a))+0.6(-1+\gamma((1-\epsilon)q(3,a)+\epsilon q(3,b)))=$
$=0.4(-10+0.5\cdot 100)+0.6(-1+0.5(0.9\cdot 56+0.1\cdot 49))=31.99$,
$q(1,a)=-1+\gamma q(2,a)=-1+0.5\cdot 31.99=14.995$.

For Q-learning:
$q(2,a)=0.4(-10+\gamma q(5,a))+0.6(-1+\gamma q(3,a))=0.4(-10+0.5\cdot 100)+0.6(-1+0.5\cdot 56)=32.2$, $q(1,a)=-1+\gamma q(2,a)=-1+0.5\cdot 32.2=15.1$.