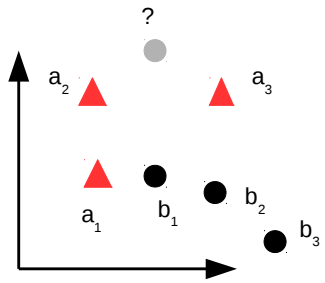
Nearest-neighbour methods

1. What is the difference between a parametric machine learning method and a non-parametric one? Is k-NN parametric?

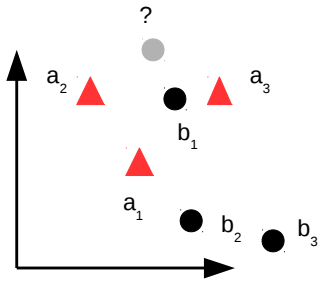
   Parametric methods have a pre-defined number of parameters, while non-parametric methods do not, and usually require memorising points from the dataset. K-NN is a non-parametric method.

2. Classify the data point with question mark with the 2-nearest neighbour method. Justify your answer.



   The two closest points are a_2 and a_3, and both are of class triangle, so the new point will also be classified as a triangle.

3. Classify the data point with question mark with the 4-nearest neighbour method. Justify your answer.



In this case the four closest points are a_1, a_2, a_3, b_1. The class that appears the most is triangle, so the new point will be classified as a triangle.

Reinforcement Learning

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.

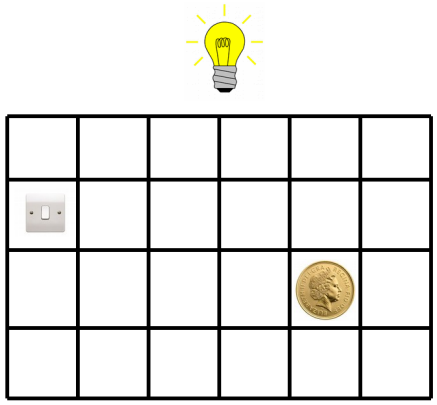
2. What makes a transition system Markovian?

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

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. Q-Learning boostraps 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

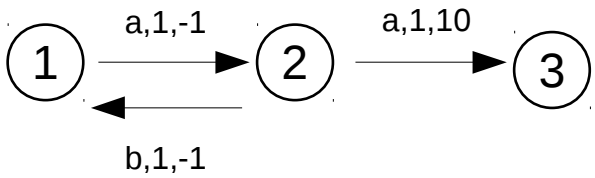


   agent are move `up`, `down`, `left`, `right`, `toggle switch` and `pick up`. The action `toggle switch` turns on and off the light in the room, and succeeds only if executed in the square with the switch, while it does not do anything anywhere else. The action `pick up` picks up the coin if executed 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. Compute the optimal action-value and value functions of the following MDP, with $\gamma = 0.5$:

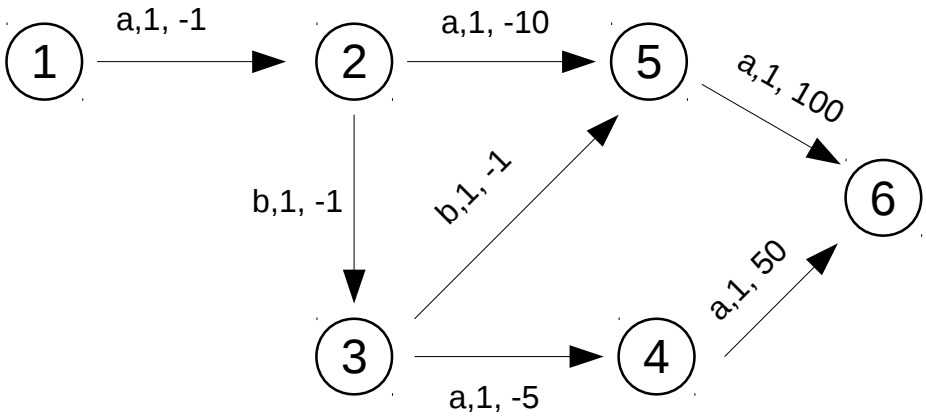


$q(2, a) = 10$
$q(1, a) = -1 + 0.5 \cdot q(2, a) = 4$
$q(2, b) = -1 + 0.5 \cdot q(1, a) = 1$

$v(1) = 4 \quad v(2) = 10 \quad v(3) = 0$

Note how if the agent chose action *b* in state 2 it would be locked in a loop in which it would only get negative reward, therefore the optimal action has to be *a* which has an immediate reward of 10.

6. 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.



Both:

$q(4,a)=50 \quad q(5,a)=100$

$q(3,a)=-5+\gamma\, q(4,a)=20 \quad q(3,b)=-1+\gamma\, q(5,a)=49$

$q(2,a)=-10+\gamma\, q(5,a)=40$

Q-learning:

$q(2,b)=-1+\gamma \max(q(3,a),q(3,b))=23.5$

$q(1,a)=-1+\gamma \max(q(2,a),q(2,b))=19$

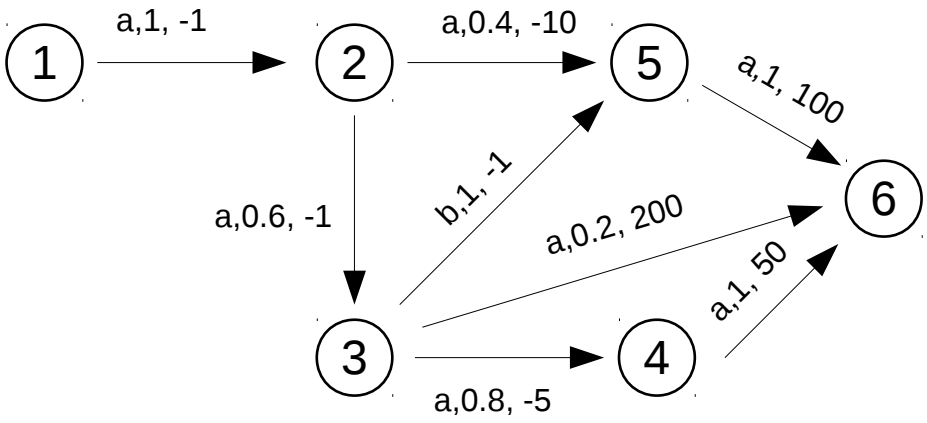
Sarsa:

$q(2,b)=-1+\gamma((1-\epsilon+\frac{\epsilon}{2})q(3,b)+\frac{\epsilon}{2}\cdot q(3,a))=-1+0.5((1-0.1+0.05)49+0.05\cdot 20)=22.775$

$q(1,a)=-1+\gamma((1-\epsilon+\frac{\epsilon}{2})q(2,a)+\frac{\epsilon}{2}\cdot q(2,b))=18.569$

Note that the value computed by Sarsa is slightly lower, since it takes exploration (that is, taking suboptimal actions with low probability) into account.

7. Calculate the action-value function that Q-learning and Sarsa would compute on the following MDP, with γ = 0.9 and $\epsilon=0.1$.



Both:

$q(4,a)=50 \quad q(5,a)=100$

$q(3,a)=0.8(-5+\gamma\, q(4,a))+0.2\cdot 200=56 \quad q(3,b)=-1+\gamma\, q(5,a)=49$

Q-learning:

$q(2,a)=0.4(-10+\gamma q(5,a))+0.6(-1+\gamma\, q(3,a))=32.2$

$q(1,a)=-1+\gamma\, q(2,a)=15.1$

Sarsa:

$q(2,a)=0.4(-10+\gamma q(5,a))+0.6(-1+\gamma\,(1-\frac{\epsilon}{2})q(3,a)+\frac{\epsilon}{2}\cdot q(3,b)))=32.095$

$q(1,a)=-1+\gamma\, q(2,a)=15.048$

K-means

1. Compute the new position of the cluster centres after 1 step of k-means. Data: <1, 1>, <0, 2>, <-1, 2>, <5, 6>, <7, 5>. Cluster centres: <-1, -1>, <4,6>.

   Points assigned to cluster <-1, -1>: <1, 1>, <0, 2>, <-1, 2>
   Points assigned to cluster <4,6>: <5, 6>, <7, 5>

   New cluster centres: $\langle (1+0-1)/3, (1+2+2)/3 \rangle = \langle 0, 1.66 \rangle$ and
   $\langle (5+7)/2, (6+5)/2 \rangle = \langle 6, 5.5 \rangle$

2. Same as (1), with the single cluster centre: <1, -1>

   New cluster centre: $\langle 2.4, 3.2 \rangle$.

3. Same as (1), with data: <1, 3>, <2, 2>, <3, -1>, <4,2>, <5,-3>, <5,4>, <4,5>, <3,-6>, <2,5>; and centres: <0,1>, <0, -1>.

   Points assigned to cluster <0,1> = <1,3>, <2,2>, <4,2>, <5,4>, <4,5>, <2,5>
   Points assigned to cluster <0,-1> = <3, -1>, <5,-3>, <3,-6>

   New cluster centres: <3,3.5>, <3.66, -3.33>