Q-Learning

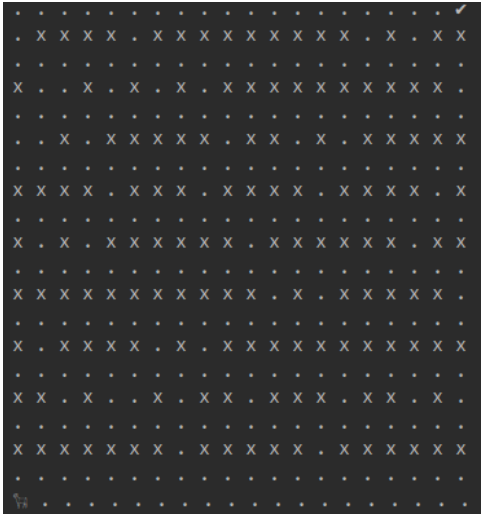
Pattern Recognition and Machine Learning - 2017

Andrea Palazzi, Davide Abati

December 13, 2017

University of Modena and Reggio Emilia

Problem setting



We have an agent stuck in a maze.

- state is (x,y) position
- reward is -1 for each time step
- when the exit of the labirinth is reached, the episode terminates
- allowed actions are N,S,W,E

Guide him out with reinforcement learning!

Q-learning algorithm for off-policy control

Initialize $Q(s, a), \forall s \in S, a \in \mathcal{A}(s)$, arbitrarily, and $Q(terminal - state, \dot{)} = 0$

for each episode **do**

  Intialise S

  for each step of episode **do**

    Choose A from S using policy derived from Q (e.g., ϵ-greedy)

    Take action A, observe R, S'

    $Q(S, A) \leftarrow Q(S, A) + \alpha(R + \gamma \max_{a'} Q(S', a') - Q(S, A))$

    $S \leftarrow S'$

  end for

end for

SARSA algorithm for on-policy control

Initialize $Q(s, a), \forall s \in S, a \in \mathcal{A}(s)$, arbitrarily, and $Q(terminal - state, \dot{}) = 0$

for each episode **do**

  Intialise S

  Choose A from S using policy derived from Q (e.g., ϵ-greedy)

  for each step of episode **do**

    Take action A, observe R, S'

    Choose A' from S' using policy derived from Q (e.g., ϵ-greedy)

    $Q(S, A) \leftarrow Q(S, A) + \alpha(R + \gamma Q(S', A') - Q(S, A))$

    $S \leftarrow S'; A \leftarrow A'$

  end for

end for

More fun with gym!

If you are curious about RL, try gym:

pip install gym