Reinforcement Learning

Friday, 14 September 2018 11:41 AM

Reinforcement learning: what if we do not know transitions *P* and reward function *r* of an MDP?

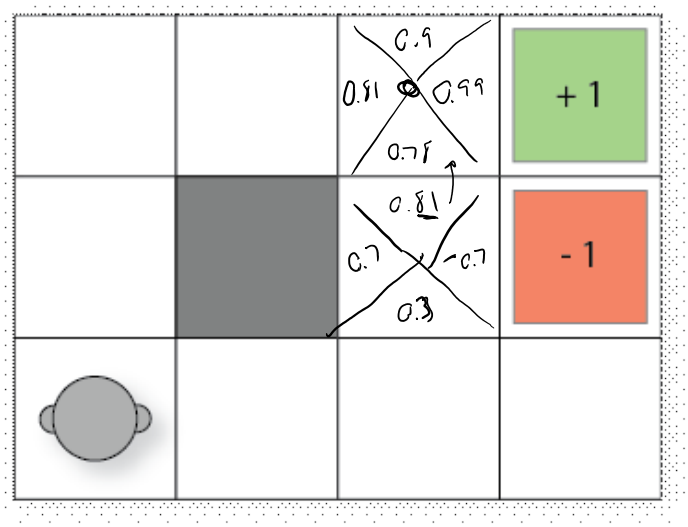
The Mystery Game:
https://programmingheroes.blogspot.com/2016/02/udacity-reinforcement-learning-mystery-game.html

Q-learning

1. Initialise *Q(s,a)* arbitrarily
2. For each episode:
   a. Initialise *s* (go to the initial state)
   b. Repeat for each step in the episode
      i. Select the next action *a* to apply from *s* (using e.g. epsilon greedy, UCT) use *Q(s,a)*
      ii. Execute action *a* and observe the reward *r* and new state *s'*
      iii. $Q(s,a) := Q(s,a) + \alpha[r + \gamma \max a' Q(s',a') - Q(s,a)]$
      iv. $s := s'$
   c. *Until s is terminal*

Q-Tables

State	Action: North	South	East	West
(0,0)	0.53	0.36	0.36	0.21
(0,1)	0.61	0.27	0.23	0.23
. . .				
(3,3)	0.90	0.78	0.99	0.81



Learning rate $\alpha = 0.1$
Discount reward factor γ 0.9

Q-learning:
Q((2,3), North) = 0.81 + 0.1*(0 + 0.9*0.99 - 0.81) = 0.8181

SARSA, with assumption that a' is West
Q((2,3), North) = 0.81 + 0.1*(0 + 0.9*0.81 - 0.81) = 0.8019

SARSA, with assumption that a' is East is just the same as for Q-learning

SARSA: On-policy learning

1. Initialise *Q(s,a)* arbitrarily
2. For each episode:
   a. Initialise *s* (go to the initial state)
   b. Select the next action *a* to apply from *s* (using e.g. epsilon greedy, UCT)
   c. Repeat for each step in the episode
      i. Execute action *a* and observe the reward *r* and new state *s'*
      ii. Select the next action *a'* to apply from *s'* (using e.g. epsilon greedy, UCT)
      iii. $Q(s,a) := Q(s,a) + \alpha\,[r + \gamma\, Q(s',a') - Q(s,a)]$
      iv. *s := s'; a := a';*
   d. Until *s* is terminal

off-policy

Q-learning:

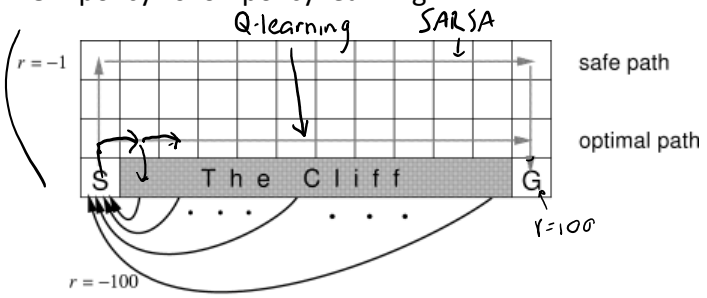
Initialize $Q(s,a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    Choose a from s using policy derived from Q (e.g., ε-greedy)
    Take action a, observe r, s'
    $Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$
    $s \leftarrow s'$;
  until s is terminal

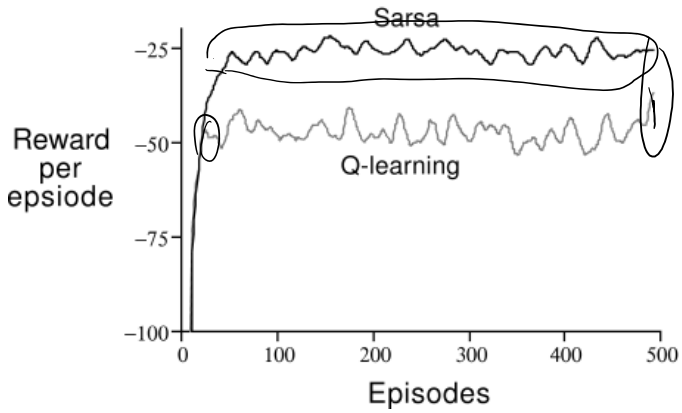
on policy

SARSA:

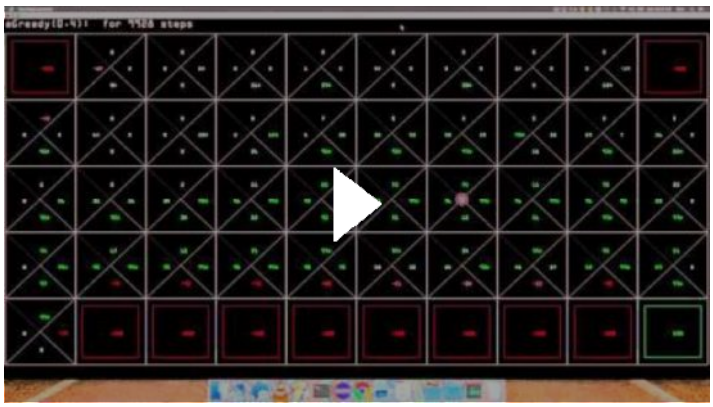
Initialize $Q(s,a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Choose a from s using policy derived from Q (e.g., ε-greedy)
  Repeat (for each step of episode):
    Take action a, observe r, s'
    Choose a' from s' using policy derived from Q (e.g., ε-greedy)
    $Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma Q(s',a') - Q(s,a)\right]$
    $s \leftarrow s'$; $a \leftarrow a'$;
  until s is terminal

Off-policy vs. on policy learning

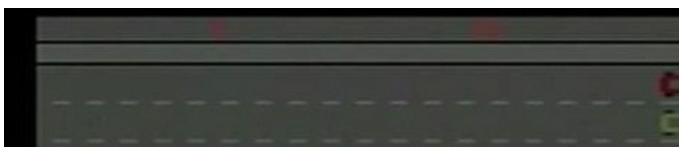


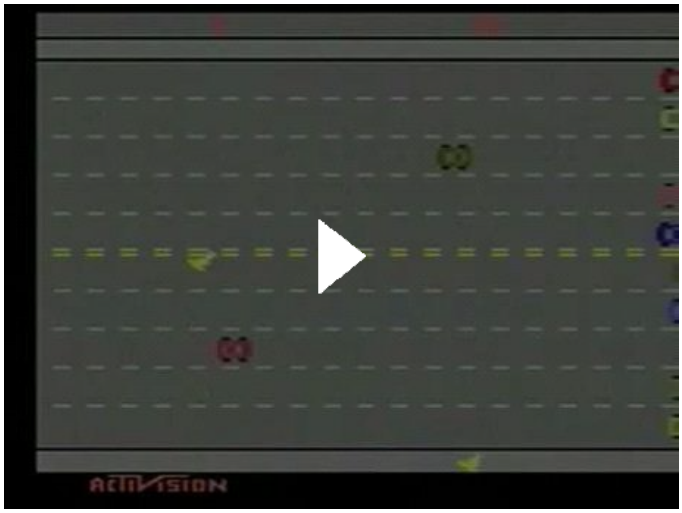


Gridworld Q-Learning - Example 3 - The Cliff



Learning to Play Freeway, using Reinforcement Learning





Learning Hand-Eye Coordination for Robotic Grasping