Reinforcement Learning 2014/2015

Tutorial 2 (week 4)

1. Discuss on-policy and off-policy RL for the 2D walker problem. Strengthen your argument by a simulation of a agent moving in an open space of $N \times N$ squares ($3 < N \lessapprox 10$) as well as in a maze (containing a fraction of inaccessible or strongly negatively rewarded squares) of the same dimensions. How do

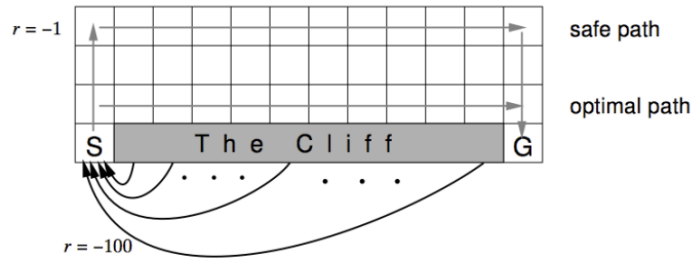
   - type of algorithm (i.e. whether on-policy or off-policy)
   - initialisation
   - exploration
   - parameters and parameter decay schemes
   - problem size

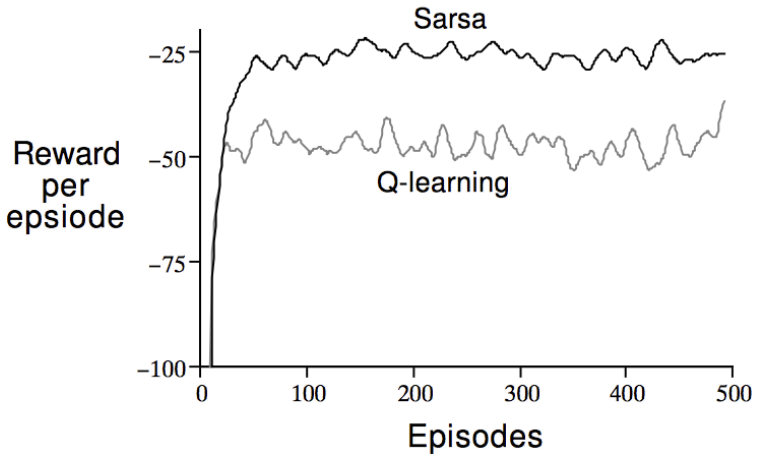
   influence the solution? Trying out several combinations of the setting of the algorithm should be a group effort. What variant of the algorithm turns out (or is likely) to be most efficient?

2. In the grid-world example, rewards are positive for goals, negative for running into the edge of the world, and zero otherwise. Are the signs of rewards important? Prove using the equation for expected discounted return (Eq. 3.2 in S+B) that adding a constant, C, to all the rewards adds a constant, K, to the values of all states. So, it does not affect the relative values of any states under any policies. What is K in terms of C and γ?
   Now, consider adding a constant C to all rewards in an *episodic* task, such as running a maze. Would this differ from the above case? Why or why not? Give an example to make your point.

   Hints: Only the relative sizes of the rewards are important.
   What is K in terms of C and γ? For $K = 0$ we have for the value of a state: $V = \sum_{t=t_0}^{\infty} \gamma^{(t-t_0)} r_t$, a different K gives $\sum_{t=t_0}^{\infty} \gamma^{(t-t_0)} (r_t + K) = \sum_{t=t_0}^{\infty} \gamma^{(t-t_0)} (r_t + K) = \mathrm{V} + \frac{K}{1-\gamma}$
   Would this differ from the above case? In a episodic task, the sum does not run to infinity, i.e. the result is a less simple in particular if the length of the episodes is not fixed.
   You may consider the MAB problem as a trivial episodic task, here K would not matter at all. If the episode length is determined by the actions (think of a board game), then for positive K a temporary tendency towards long games could arise, which could extend the learning time (assume a fast solution is optimal).

3. Discuss how on-policy and off-policy reinforcement learning algorithms solve the cliff-walking problem (see the figure below and lecture RL05).





ε-greedy with $\varepsilon = 0.1$

QL: $Q(s,a) = Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$ off-policy

SARSA: $Q(s,a) = Q(s,a) + \alpha[r + \gamma Q(s',a') - Q(s,a)]$ on-policy

In the cliff-walking task:
QL: learns optimal policy along edge
SARSA: learns a safe non-optimal policy away from edge

For $\varepsilon \to 0$ gradually both converge to the optimal path.
Note: The example has been been presented already in the lectures. It is important to really understand it. In the context of the cliff example also similar examples should be considered in the context of the on-policy vs. off-policy debate. E.g.
- obstacle avoidance (see Question 1. This quite similar to the cliff example but perhaps more realisitc)
- a mine field with known position of the mines
- learning default behaviours (i.e. continuing to follow a straight corridor, without taking other actions into consideration except at any intersections or doors etc.)
As a consequence these considerations you may find yourself facing the question, why $\mathcal{Q}$-learning (or off-policy in general) does actually make sense. An answer to this question that ε-greedy is not a good idea. If $\mathcal{Q}$-learning uses Boltzmann exploration, it may be as good as SARSA. Also SARSA becomes better (closer to the cliff) if Boltzmann exploration is used.

- consider also the lunar lander task (a historic computer game, http://en.wikipedia.org/wiki/Lunar_Lander_(1979_video_game), actions: thruster on or off, reward when reaching the surface with a small velocity): The closer the spaceship is to the moon's surface, the higher the gravitational force becomes and the more critical are single movements. Q-learning has a chance to learning the final approach, while SARSA may tend to avoid it.
"In real-life terms, a SARSA person deciding which party to go to would choose the one he expects to be the most fun, while a Q-Learning person would choose the one that could be the most fun, if he made all the right moves." (from an answer at stackoverflow)
Conclusion: If a reasonable good initial policy is known, try SARSA. If not but, exploration is safe, Q-learning might be faster. If nothing is known and exploration is unsafe, you have a problem.

4. Consider the grid world example in Chapter 4 of the Sutton and Barto book (Example 4.1). Suppose a new state 15 is added just below state 13, and its actions, *left, up, right, down*, take the agent to states 12, 13, 14 and 15 respectively. Assume that the transitions from the original states are unchanged. What then is $V^\pi(15)$ for the equiprobable random policy?
Now suppose the dynamics of state 13 are also changed, such that action *down* from state 13 takes the agent to the new state 15. What is $V(15)$ for the equiprobable random policy in this case?

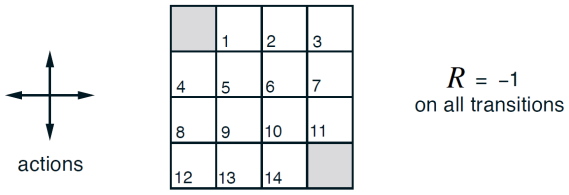
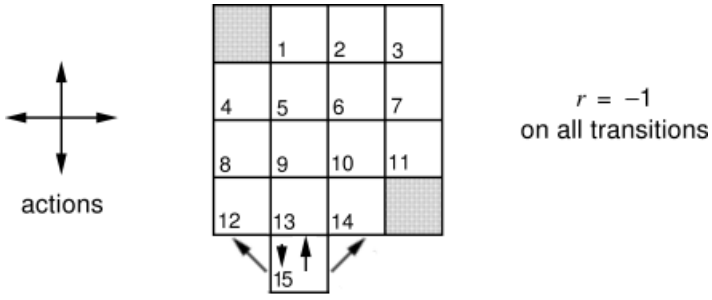


Figure from Sutton and Barto, 2nd edition

Hints:
$V(12) = -22$, $V(13) = -20$, $V(14) = -14$
$V(15) = -1 + \frac{1}{4}(-22 - 20 - 14) + \frac{1}{4}V(15)$
$V(15) = \frac{4}{3}\left(-1 - \frac{56}{4}\right) = -20$



Now with recalculation of state 13. Assuming the values of all other states remain unchanged this can be solved by simultaneously solving the equations for $V(13)$ and $V(15)$. But if state 15 can be traversed then all average route lengths and thus the values change, which requires in principle a complete recalculation.

5. Recall the set-up of an RL algorithm. Considering algorithmic details, general feasibility and computational complexity as well as the options implied by the previous problem, discuss, in as much details as reasonably possible, applications of RL to problems such as (but not excluding further examples)
   (a) Finding your way in a maze
   (b) Finding your way inside the Forum
   (c) Balancing a pole
   (d) Playing Tetris
   (e) Playing Kendama
   (f) Playing football
   (g) Part-painting problem
   (h) Routing network traffic
   (i) Running a shop
   (j) Running a big company
   (k) Guiding a robot arm to a target
   (l) Coaching a team of soccer robots
   (m) Driving a car
   (n) Driving assistance at intersections
   (o) HCI (Spoken Dialogue Systems)
   (p) Equity trading
   (q) Organising your day
   (r) Biological modelling

Hint: Select a few examples and ask in each case for states, actions (continuous or discrete?), rewards, expectations for parameter values, suitable algorithms, practical feasability, existence of better solutions (outside RL) and possibility of an application to a sub-problem. Try to find other eamples where RL appears as a promising approach.