Reinforcement Learning 2013

Tutorial 2: Hints and solutions

1. Discuss your solution of the 1D walker problem (see homework in lecture RL3, 21/1/2014). How do

   - initialisation
   - alternative reward definitions
   - exploration variants
   - 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. Discuss which variants that have not been tried yet, appear to be worth trying? What variant of the algorithm turns out (or is likely to be) most efficient? Is this variant likely to generalise to other problems?[1]

2. Discuss the solution of the 2D walker problem (see homework in lecture RL4, 24/1/2014). Same questions as above in problem 1.

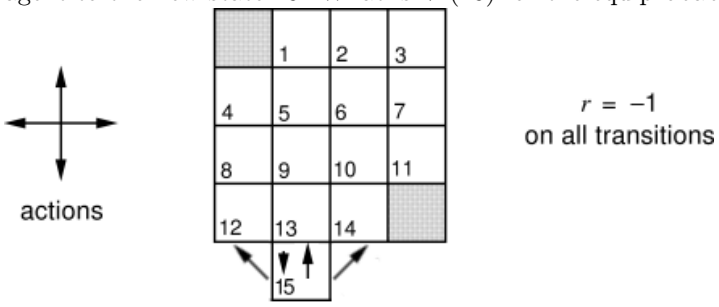
   Hints: Problems 1 and 2 are practical exercises, they are intended as a preparation for the first assignment. Although there is no shortcut to such experience, it is obviously not the idea to try out all possible combinations the parameters. To get an idea, it is practically reasonable to choose "typical" parameters and vary only one at a time (which, however, may lead to reconsideration of what is a typical parameter).
   In the lecture a number of hints were given such as
   – problem size: Usually given, here as small as possible, but trivial cases should be avoided. If the problem does not contain any further structure (e.g. a maze), then there is no reason to use more than 10 states in 1D and perhaps 25 in 2D. Make sure that a single run takes less than a minute.
   – parameters and parameter decay schemes: $\gamma = 0.8, \ldots 0.9$ (according to problem size), $\eta = 0.5$ (this is very large, but is fine for a non-stochastic problem were accuracy is not an issue). Decay: Gamma does not need to change, the learning rate η should not be too small when exploration becomes less.
   – exploration variants: ε-greedy, or by optimistic initialisation or ε-greedy exploration with linearly decaying ε (what are your experiences here?)
   – initialisation: Usually by small random values or optimistic.
   – alternative reward definitions: Either only at goal or cost (negative reward) per step or both.

3. 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 $\sum_{t=t_0}^{\infty} \gamma^{(t-t_0)} (r_t + K) = \sum_{t=t_0}^{\infty} \gamma^{(t-t_0)} (r_t + K) = \mathrm{V} + \frac{K}{1-\gamma}$

[1] Problems 1 and 2 are repeated from last week's sheet, where they appeared as 6 and 7.

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).

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?



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

$$V^\pi(s) = \sum_a \pi(s;a) \sum_{s'} P^a_{ss'}[R^a_{ss'} + \gamma V^\pi(s')]$$

must hold at each state for the value function V^π. Show, numerically in figure 1 (left), that this equation holds for the centre state, valued at +0.7, with respect to its four neighbouring states, valued at +2.3, +0.4, -0.4, and +0.7. No immediate reward and $\gamma = 0.9$
Note that $P^a_{s_0 s_1} = \delta(s_1, s(a|s_0))$
$\frac{1}{4}\left(\left(\sum_a R^a_{ss'}\right) + \gamma(2.3 + 0.4 - 0.4 + 0.7)\right) = \frac{1}{4}\gamma \cdot 3 = \frac{27}{40} \approx 0.7$
Correct within the given precision.

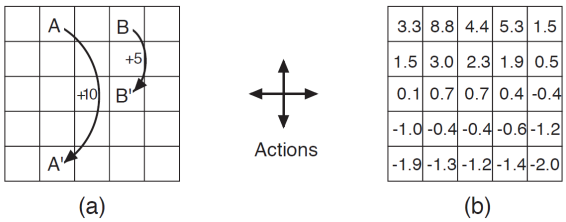


Figure from Sutton and Barto, 2nd ed.

6. Suppose that you are restricted to considering only policies that are ε-soft, meaning that the probability of selecting each action in each state, s, is at least $\frac{\varepsilon}{|A(s)|}$. Describe qualitatively the changes that would be required in each of the steps 3, 2, 1, in that order, of the policy iteration algorithm for V in figure 4.3 in the Sutton and Barto book.

1. Initialization
   $V(s) \in \Re$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Repeat
      $\Delta \leftarrow 0$
      For each $s \in \mathcal{S}$:
         $v \leftarrow V(s)$
         $V(s) \leftarrow \sum_{s'} \mathcal{P}_{ss'}^{\pi(s)} \left[\mathcal{R}_{ss'}^{\pi(s)} + \gamma V(s') \right]$
         $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
      $b \leftarrow \pi(s)$
      $\pi(s) \leftarrow \arg\max_a \sum_{s'} \mathcal{P}_{ss'}^{a} \left[\mathcal{R}_{ss'}^{a} + \gamma V(s') \right]$
      If $b \neq \pi(s)$, then *policy-stable* $\leftarrow$ $false$
   If *policy-stable*, then stop; else go to 2

- Policy Improvement: instead of an argmax use a soft-arg-max ε with the desired ε.
- Policy Evaluation: No changes (policy is fixed here), but for small ε the statistics might need to be improved in order to get samples for every action
- Initialisation: Initialise an ε-soft policy.