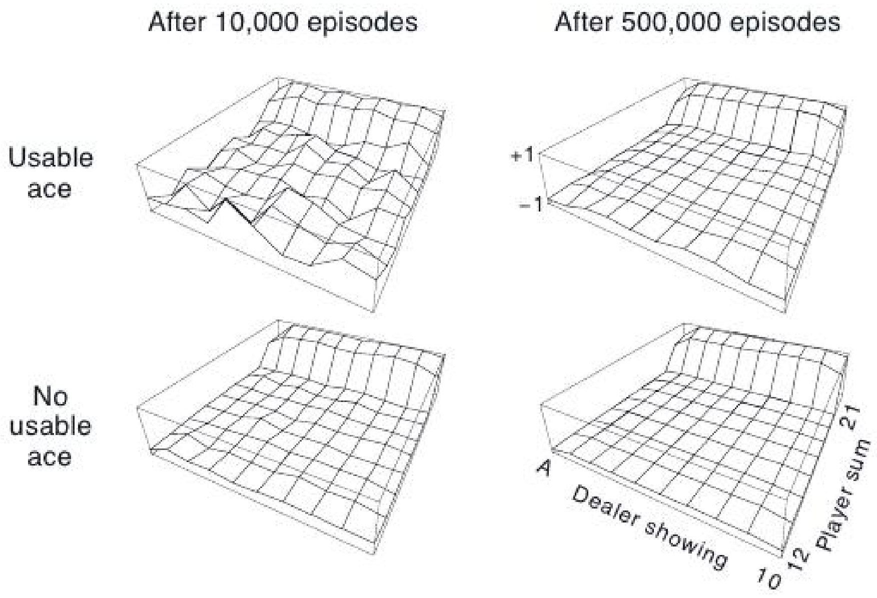
Reinforcement Learning: Tutorial 4

Problems and hints for solutions for the week from 23. 2. 2015

1. Is the reinforcement learning framework adequate to usefully represent all goal-directed learning tasks? Can you think of any clear exceptions?

   This problem is from Sutton & Barto's book There is an official set of solutions for these problems. I don't know whether my suggestions below are in agreement with these solutions. In principle, every problem with a goal is an RL problem (goal: *r=1*, non-goal: *r=0*), but some remarks are in order:
   (1) The goal must be a state of the problem (or perhaps a state-action pair or a state-state transition due to an action). If the goal is not part of the state space other methods are required. E.g. "leave the room" defines a clear goal, which would not be part of the state-space if the states are positions inside the room.
   (3) Goals can be on top of an existing RL problem, notable example is the goal "Explore properly" while doing RL. This could be represented as meta-RL problem where reward is given for exploratory actions in the primal RL task, but although such an approach is theoretically possible, it may not always be efficient. Note that there is the possibility to incorporate "exploration bonusses" in the reward function of the primal problem.
   (4) Goals can consist in changes of the representation (e.g. "to learn maths").
   (5) If the problem is relatively large and no further information is given, population-based methods might be better, i.e. genetic algorithms and related meta-heuristic search methods, in particular if the representation of the problem is also subject to search, these methods may help.
   (6) Is it useful? If an exact model exist, i.e. if the problem is deterministic and explicitly known, RL might be too general an approach.

2. The Figure shows the approximate state-value functions for the blackjack policy that sticks only on 20 or 21, computed by Monte Carlo policy evaluation. Why does the estimated value function jump up for the last two rows in the rear? Why does it drop off for the whole last row on the left? Why are the front-most values higher in the upper diagrams than in the lower?

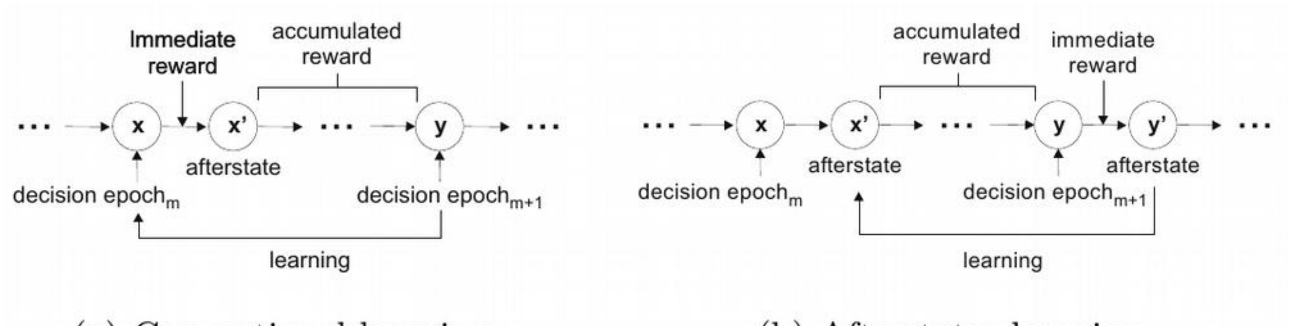


   If the policy sticks ("stands") already earlier (I'm not an expert on blackjack, there are many variants of the game, feel free to discuss), the transition along the player axis would be more gradual. Since the policy sticks at 20 or 21 one, there is no chance of "getting bust", i.e. getting immediate negative reward. If the total value is below 12, then no new card can lead to busting, therefore this is not considered here. An ace is good to have, as it can be 1 or 11 points, i.e. leaving an option, therefore the value is higher. However, it introduces also more uncertainty (which is later averaged out). Note that good players used to remember all cards in the deck and could thus profit of the slight non-Markovianity of the game (see also shuffle tracking).

3. Reinforcement learning systems do not need to be “taught” by knowledgeable “teachers”; they learn from their own experience. But teachers of various types can still be helpful. Describe three different ways in which a teacher might facilitate learning. For each, give a specific example scenario and explain what makes learning more efficient.

   (1) Known values: If the goal state is not a stop-state, on can initialise its value with r/(1-gamma)
   (2) Symmetries (or antisymmetries): when learning a particular state, the symmetric state can be learned as well. Approximative symmetries can be used at least in the early learning phase (e.g. keep a robot from falling forward is similar but not identical to the opposite of preventing it from falling backward)
   (3) Neighbourhood relations: Learn to a lesser extend the same for neighbouring states; value functions and policies can be smoothed.
   (4) in a hierarchical problem, the upper level can be a teacher for the lower level, i.e. the student has to complement general rules from the teacher by more detailed information from own experience.

4. Consider the game of Tic-Tac-Toe (or any board game with a finite number of moves). How do you define the state space and state value function for such a game? One could define an afterstate in terms of board positions after the agent has made its move. How does this impact on the information required to calculate a value function, and what might be the advantage of such a scheme - explain with a simple example scenario.



(a) Conventional learning. (b) Afterstates learning.

   Describe how the task of Jack’s Car Rental (see lecture or Example 4.2 in S&B) could be reformulated in terms of afterstates. Why, in terms of this specific task, would such a reformulation be likely to speed convergence?

   Afterstates can reduce the randomness of the problem, by taking the actual reaction of the opponent (or the customer) into account.

5. Suppose, a reinforcement learning algorithm is trained for game playing (e.g.

chess), but instead of playing against a random opponent, it played against itself. What do you think would happen in this case? Would it learn a different way of playing?

The algorithm might learn to produce (and defend against) some policies only and could thus become defenceless against certain types of attacks. Exploration is an important issue here. Discuss in this context: intra-species evolution (which brings about beautiful birds) and Nash equilibria.

6. Dynamic programming is said to be “model-based”. What does “model” mean in this context? Give an example of a “model-free” reinforcement learning method, explaining in what sense it is “model-free”.

   Discuss first, why models can help RL, even if DP approaches are not easily applicable. E.g. a model for a robot is never exact, but can help to move the algorithm to a region where little further exploration is needed.

7. Suppose you have the task of getting a mine-detection system to plot a safe path through a minefield. The mines are reasonably easily detectable and you have a few small, relatively cheap (and therefore, to some extent, disposable) robots that you can use to aid you. Discuss how you might use reinforcement learning to find a safe path.

   Explosion of a robot does not mean that all mines in that place are removed. Nevertheless, it can be assumed that there are less mines, e.g. a higher value for finding a safe path, such that it is a good idea to send more robot to this place. Costs are: Number of destroyed robots, length of path.

8. You find that in an RL problem discrete quantities is not accurate enough; instead, you use continuous quantities. How would you modify your learning system? What is a feature vector and how may one be used in the representation of a state? Why would one choose to use a feature-vector representation of a state?

   This is an outlook to the next lectures. Feature vectors can be defined continuously, and be adapted to a discrete algorithm e.g. by k-means clustering. Note that the clustering can be suboptimal after a bit of RL, such that several cycles may be needed.
   More generally, instead of a look-up table function approximation can be used. Learning (of a value function, e.g.) means now to induce a compatibility of the function values at different positions (and their neighbouring positions) to the reward by an appropriately adapted delta-rule. A simple approach would be linear interpolation between states (consider a 1D example).