Reinforcement Learning 2013/2014: Tutorial 7

(Hints for solutions)

1. RL learning aims at related tasks of optimising the value function, the policy and the behaviour based on the reward signal. The rewards is used only locally (in simple RL algorithms), such that models are employed in order to enable the use of the experience of the agents in the environment more efficiently. Model learning faces analogous challenges: The model should be correct (w.r.t. the data available at a given time), it should not overfit but generalise beyond the data, and it should focus on relevant data (i.e. on regions of particularly high or low reward) although "relevance" might change during learning. Asymptotically, the model is (ideally) on the one hand perfect and on the other hand not needed as the information is available in the policy. Discuss the interaction of the aspects of the learning the action and the aspects of learning a model. It might make sense to remember here the Actor-Critic architecture. See also last tutorial

   This problem is to mainly to discuss the role of models in RL. It is obvious the a wrong model can be harmful if the agent does not explore beyond the model issues. As mentioned, the model cannot be correct with respect to the temporality of the agent (Niels Bohr: "Prediction is very difficult, especially about the future."), because the agent improves beyond its past experiences. The model can, in a minimal sense, be used to remember all previous examples and if this is done in an off-policy algorithm then there is no problem with actions that turn out to be "wrong". For continuous cases this might not be sufficient: Limited numbers of basis function need to be moved or local descriptions in terms of action-dependent state transitions may not be reliably integrable into trajectories (think of a pendulum near the upright position).

2. [from Sutton&Barto] The nonplanning method looks particularly poor in Figure 8.6 because it is a one-step method; a method using eligibility traces would do better. Do you think an eligibility trace method could do as well as the Dyna method? Explain why or why not.

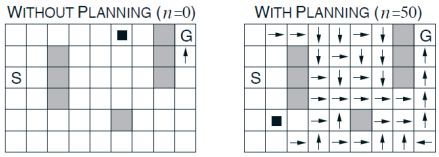

Figure 8.6: Policies found by planning and nonplanning Dyna-Q agents halfway through the second episode. The arrows indicate the greedy action in each state; no arrow is shown for a state if all of its action values are equal. The black square indicates the location of the agent.

   There was already a similar questions in the lecture. It can be useful to find simply a good λ value rather than adapting a full model, because sometimes also the simulation may have its cost. Obviously, even the best λ value may not apply everywhere in the environment in the same way. In the S&B book also the question of full vs. sample back-up is discussed. I have not mentioned this in the lecture, so here is an opportunity. It is not a very deep idea and can be mapped at the difference between eligibility traces (sample) and model (full), although obviously also in a model less than the full trajectory tree may be considered.

3. [from Sutton&Barto] Why did the Dyna agent with exploration bonus, Dyna-Q+, perform better in the first phase as well as in the second phase of the blocking and shortcut experiments? Careful inspection of Figure 8.8 reveals that the difference between Dyna-Q+ and Dyna-Q narrowed slightly over the first part of the experiment. What is the reason for this?

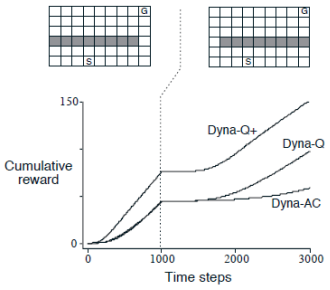

Figure 8.7: Average performance of Dyna agents on a blocking task. The left environment was used for the first 1000 steps, the right environment for the rest. Dyna-Q+ is Dyna-Q with an exploration bonus that encourages exploration. Dyna-AC is a Dyna agent that uses an actor–critic learning method instead of Q-learning.

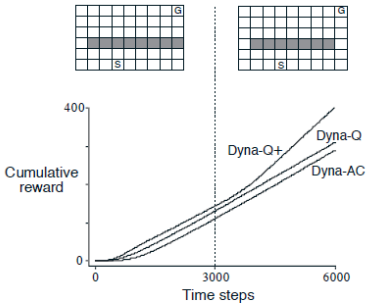

Figure 8.8: Average performance of Dyna agents on a shortcut task. The left environment was used for the first 3000 steps, the right environment for the rest.

You may remind the students first to the method of optimistic initialisation. Of course the answer depends on whether other exploration is used here, on any termination rules for the episode, and on the punishment per step. In most cases the "+" algorithm just explores better, while without "+" the algorithm tend to get stuck with earlier experiences. After the change the "+" version still explores, so it picks up what has changed.

4. Dyna-Q+ uses exploration bonuses, e.g. of the form $r + \kappa\sqrt{n}$. This is a kind of an intrinsic reward. Discuss advantages and problems in connection to intrinsic rewards. Consider also the aspect of evolutionary learning.

   Intrinsic rewards are a bias. If the problem are typically such that a bias is useful, then it makes sense to use it. A slow and steady evolutionary algorithm can find out what bias is useful.

5. Prioritised sweeping (see Sutton&Barto section 8.4 in 2nd ed.): While Dyna agents select state-action pairs uniformly at random from the previously experienced pairs, it might be more efficient. Why? Which state-action pairs should be preferred? Discuss the role of a goal states in this context.

   The idea is that "relevance" is not only about goals but about all changes (i.e. new information) in the reward/value structure. See also the questions 1 and 4 above.

6. Assume your are heading a large team of researches and technicians working on building a humanoid robot. Motivate your coworkers to use RL in various tasks related to the project. Consider hierarchical approaches, specify sub-tasks and required resources. Discuss alternatives. What options for hybrid algorithms might be interesting here?

   Try to compare to human development (which takes a couple of years), to include different aspects (biases, imitation learning, constrains, initialisation) as well as realisability (pretraining in a simulator, across environments, subtasks that are of realistic size)

7. Assume you are heading a large team of researches and technicians working in robot soccer. Motivate your coworkers to use RL in various tasks related to the project. Consider hierarchical approaches, specify sub-tasks and required resources. Discuss in particular team-level learning tasks and the role of models.

   Same as above, but now with the Multi-agent RL flavour. Discuss a mean field approach, stigmery (here: moving the ball) and a team-based solution (which is much higher dimension and requires communication). A team-based solution may not even be necessary, because its optimality is anyway compromised by the opponent team. In practice, hierarchical algorithm are used, where at the high-level the team-aspect is included, but at the low-level only agent-centered dimensions are explored.

8. Apply RL to the iterated prisoners dilemma. Discuss several scenarios: (i) Two prisoners, (ii) simultaneous plays between pairs randomly selected from many prisoners, (iii) prisoners are situated in a plane, plays with neighbours (iv) different state definitions in the group. Discuss also the effect of details of the RL algorithm.

   Existing papers on this (e.g. TW Sandholm, RH Crites, 1996) are not very conclusive. The general expectations is
   (i) Nash optimum is found easily
   (ii) same, but depends on the selection criteria, here subset could form play also other strategies.
   (iii) you would have domain boundaries, but eventually everywhere tit-for-tat wins
   (iv) It is interesting if the agents can determine themselves what memory size they can use. You may find here longer and longer states vectors (including the own and the opponents past actions, because this gives more information) but eventually this breaks down in favour of simple strategies (because the complex strategies cannot be explored everywhere), and then it starts again anew.
   Obviously everything depends on exploration, where on- or off-policy etc. Could be interesting to check whether a average-reward algorithm would go for something different than tit-for-tat.

9. Contraction Mapping [Vien Ngo & Marc Toussaint]: Define the backup operator H as

$$V_{t+1}(b) = HV_t = \max_a \left[\rho(b,a) + \gamma \sum_{o \in \mathcal{O}} p\left(o|b,a\right) V_t\left(\tau(b,a,o)\right) \right]$$

   Prove that H is a contractive mapping which means $|HV_t - HU_t|_\infty \leq \gamma \, |V_t - U_t|_\infty$, if U, V are two value functions.

   This is for belief states, but we did something similar in the lecture (RL16). The problem is that ρ might not be the same for both realisations, such that it is advisable to assume a deterministic algorithm. Alternatively, one could wait for ρ to have averaged out for longer and longer intervals to arrive at a probability 1 result.

10. Lipschitz Condition [Vien Ngo & Marc Toussaint]: If b_1 and b_2 are two certain belief points such that $|b_1 - b_2|_1 \leq \delta$, prove that the optimal value function satisfies the Lipschitz condition: $|V^*(b_1) - V^*(b_2)| \leq \frac{R_{\max}}{1-\gamma}\delta$.
    Hint: Assuming that the optimal value function is piece-wise linear convex using a set of α-function Γ^*:

$$V^*(b) = \max_{\alpha \in \Gamma^*} b \cdot \alpha$$

    This was also part of the proof for the fully observable case, but was not referred to as a Lipschitz condition.

The last few problems are not meant as "typical exam questions" but to encourage you to get interested in the theoretical/general quesitons of RL.