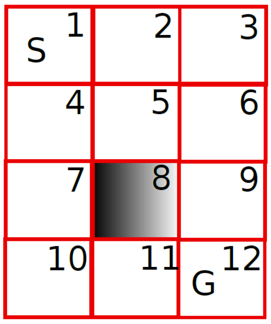
Reinforcement Learning – Exercise 08

Hung Ngo & Vien Ngo
Machine Learning & Robotics lab, University of Stuttgart
Universittsstrae 38, 70569 Stuttgart, Germany

June 2, 2016

1 RMAX

Consider a 4×3 maze as in the figure. The agent always starts at state 1, and finds an optimal path to the goal state (12). State 8 is a trap. The agent is equipped with four movement actions. If the agent falls into 8 (and receive reward -10) or 12 (and receive reward +10), the episode terminates. The environment dynamics is deterministic, except state 6. If taking actions in state 6, the agent moves to the intended direction with probability of 0.8, and to two perpendicular directions with each 0.1. The movement cost is -1.0 each step.



The sample code (download here) provides you with i) a simulator of the 4×3 maze, and ii) the RMAX's code structure.

a) Fill in the code of the Value Iteration algorithm to solve for an optimal policy of the approximate MDP model.

b) Implement the RMAX algorithm using $m = 20$. Report the average performance (total reward of each episode) w.r.t #episodes (averaging over 10 runs).

c) (Optional) Choose to implement some other learning-and-exploration strategies of interest for this domain. For fair comparison, fine tune each strategy (for RMAX, tuning the threshold m) for their best performance.

2 Reward Shaping

As we have seen in the lecture, E3 and RMAX and many other exploration strategies use the intuition of optimism in the face of uncertainty. This is done by introducing bonuses to the reward function, hence one could call this technique *reward shaping*.

Assuming our original MDP is $\mathcal{M}$ with a reward function $R(s, a, s')$. We construct another MDP $\mathcal{M}'$ by transforming the reward function R of $\mathcal{M}$ to $R'(s, a, s') = R(s, a, s') + F(s, a, s')$ of $\mathcal{M}'$, where $F(s, a, s') = \gamma\psi(s') - \psi(s)$ is called a potential-based shaping function, with $\psi : \mathcal{S} \rightarrow \mathcal{R}$ a real-valued function mapping from states to rewards.

Prove that every optimal policy in $\mathcal{M}'$ is optimal in $\mathcal{M}$ (and vice versa).

Hint: i) Given an optimal policy derived from $Q^*(s, a)$, prove that the optimal policy does not change under the translation $Q^*(s, a) + g(s)$, where $g(s)$ is an arbitrary function. ii) Use Bellman equations for optimal policies.

3 (Optional)

a) Prove that a reinforcement learner with initial Q-values based on the shaping algorithm's potential function makes the *same updates* throughout learning as a learner receiving potential-based shaping rewards!

b) Implement and compare the performance of linear Dyna-2 using TD-search with linear Dyna using prioritized sweeping (HW7.2) for the mountain car domain with RBF basis function of HW6.

4 Other Notes

HW7 questions (and the slides) have been revised! We will go through HW7 again in the coming tutorial session.