Assignment 11

Machine Learning, Summer term 2013, Norman Hendrich

Solutions due by July 1

Exercise 11.1 (Example reinforcement learning tasks, 3 points) Devise and describe three example tasks of your own that might be solved by the reinforcement learning approach. For each of your three examples, identify the states and actions, and sketch a suitable rewards-function. Also estimate the size of the state-space: how many dimensions do you need to represent the (s, a) state-action space, and how many states are there in total? Try to make your examples as different from each other as possible, using different contexts and application areas.

Exercise 11.2 (RL approach, 2-points) Is the reinforcement learning framework adequate to represent *all* goal-directed learning-tasks? Try to find a clear exception, where RL will obviously fail. Otherwise, explain why RL should in principle be applicable for any problem.

Exercise 11.3 (Grid-world V-function, 1+2 points) The maximum value of V-function for the grid-world example (slides 60/66) is given as 24.4, to one decimal place.

- What is the optimal policy for this grid-world?
- For this optimal policy, derive the expression for the expected reward of the best state symbolically, then compute it to three decimal places (using $\gamma = 0.9$).

Exercise 11.4 (Bellman Equation for the Q-function, 3 points) Derive the Bellman equation for the action-value function $Q^\pi(s, a)$. The equation must give the action value $Q^\pi(s, a)$ in terms of the action values $Q^\pi(s', a')$ of possible successors to the state-action pair (s, a).

Exercise 11.5 (Maze, 4 points) Imagine that you are designing a robot to run a maze. You decide it to give it a reward of +1 for escaping from the maze, and a reward of zero at all other times. This is an episodic task (the successive runs through the maze), so you decide that the goal is to maximize expected total reward. After running the learning agent for a while, you find that it is showing no improvement from escaping from the maze. What is going wrong? How you could you change the reward-function to improve the behaviour of your agent?