Exercise Sheet 13

due: 09.02.2017

Reinforcement Learning

Exercise T13.1: Markov Decision Processes

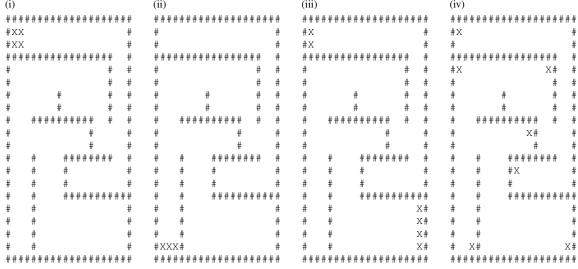
(tutorial)

- (a) What differentiates reinforcement learning from supervised and unsupervised learning?
- (b) Define a Markov decision process and a corresponding reinforcement learning agent.
- (c) How is the *value function* defined, and how can the *Bellman operator* be derived?
- (d) Describe two ways to compute the value function given models of the MDP and the agent.
- (e) How can the value function be *estimated* inductively?
- (f) Are the algorithm described in (d) and (e) contraction mappings? Are they also convergent?

Exercise H13.1: Value functions for mazes

(homework, 6 points)

In this exercise, you will construct a navigation maze from a text-array of m lines with n characters each. Each character represents either an unrewarded state (a space: ' '), a rewarded state (a capital X: 'X') or an impassable wall (a hash key: '#'). Note that walls are not states, as the agent cannot enter them. You are given the following 4 mazes:



All mazes can also be found in a text file on ISIS.

- (a) (1 point) Implement the above mazes and show them as an image-plot with some sensible color code (e.g. red walls, green rewards, blue unrewarded states).
- (b) (1 point) Implement a transition model $\mathbf{P} \in \mathbb{R}^{S \times S \times A}$ that moves an agent in one of the four adjacent states (e.g. 1: move right; 2: move down; 3: move left; 4: move up). Transitions that would end up in walls are blocked and no movement is performed. Plot $\sum_{j=1}^S P_{ijk}, \forall i \in \{1,\ldots,S\}, \forall k \in \{1,\ldots,A\}$, to verify that your model is indeed a probability distribution. Note that the walls are not states and need not adhere to this constraint.

- (c) (2 points) Compute the *analytic* value function for each of the mazes with the uniform policy $\pi(\underline{\mathbf{a}}_k|\underline{\mathbf{x}}_i) = \frac{1}{A}, \forall k \in \{1,\ldots,A\}, \forall i \in \{1,\ldots,S\}$. Every transition *from* a rewarded state (to any other) yields a reward of +1, otherwise the reward is 0. The discount factor shall be $\gamma = 0.9$. Plot the logarithm of the four value functions as image-plots and describe how you handled the walls in your computation.
- (d) (1 points) Show that *value iteration* with the Bellman operator \hat{B}^{π} converges to the analytical value calculated above, by initializing the value function with 0 everywhere and measuring the MSE to the analytical value function of all 4 mazes during the first 50 iterations.
- (e) (1 point) Show that *value iteration* with the Bellman operator \hat{B}^{π} is a contraction mapping by initializing 2 different value function from a normal distribution $\mathcal{N}(0,1)$ and show the MSE between them in all 4 mazes for the first 50 iterations.

Exercise H13.2: Find a good policy

(homework, 4 points)

This exercise extends the previous definition of navigation mazes by policies. Both locations that are indicated by a blank space () and rewarded states (marked with an X) have a uniform distribution among all actions, i.e., $\pi(\underline{\mathbf{a}}_k|\underline{\mathbf{x}}_i) = \frac{1}{A}$. However, locations that are marked with other symbols execute a deterministic policy: states marked with (>) always move right, states marked with (<) always move left, states marked with (\forall) always move down and states marked with (^) always move up. Locations marked by (#) are still walls and here no policy has to be defined. The only maze we will consider in this exercise is:

```
####################
#XX>>>>>>**
#XX<<<<<<<<<
#############*
#>>>>>>* #^v#
    v<< v#^v#
v#^ v#^v#
#^ v<<<<#^<<<<v#^v#
#^ V#########*^V#^V#
#^ >>>>>>**
#^ v<< v<<*#^<<<v#
#^ v#^ v#######**
#^ v#^ v#>>>>>
#^ v#^ v#^<<<<<#
#^ V#^ V##########
#^ v#^ >>>>>#
#^ v#^
         ₩
#^<<#^<<<<<
#####################
```

- (a) (1 points) Plot the analytical value function of this maze (with the indicated policy) as described in the previous exercise.
- (b) (1 point) Define an "optimal policy", that maximizes the value of all states. Define the text-array of that policy (except for the rewarded states). Plot the corresponding value as before.
- (c) (1 point) Given an example for another optimal policy with the corresponding value function.
- (d) (1 point) The value function of an optimal policy looks very similar to the value function of the uniform policy. Given an example (not necessarily a navigation maze) of an MDP in which this is not the case.

Total 10 points.