Intelligent Robotics Systems

Exercise 5

Lecturer: Armin Biess, Ph.D.

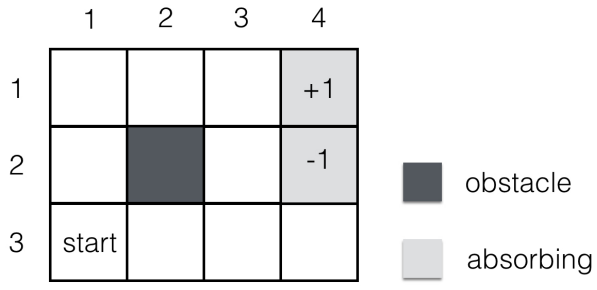
Due date: 2.7.2018

1. Define and ***explain*** the following terms [10 pts]

1. Markov Decision Process (MDP). Which problems can be solved with an MDP? Provide an example.
2. State-value function
3. Action-value function
4. Policy
5. Dynamic programming
6. Value iteration
7. Policy iteration
8. Reinforcement learning (RL). Which problems can be solved with RL? What is the difference to an MDP? Provide an example.

2. Markov decision process (MDP) in a robot gridworld [90 pts]

Given is a robot in a start state of a 3×4 planar gridworld with one obstacle and two absorbing (terminal) states. In the terminal states the robot receives rewards of $+1$ (goal state) and -1, respectively, whereas for all other states a reward of $r = -0.04$ is collected. In this gridworld the robot can perform four actions $= \{N, E, S, W\}$. The probability for the commanded direction to occur is 0.8, but with probability 0.2 the robot moves at right angles to the intended direction. It is assumed that the boundaries are reflective, i.e., if the robot hits a wall or obstacle it remains in the same grid cell. Find the optimal policy for the robot in this gridworld by solving the corresponding MDP.



For the implementation of the MDP a Matlab `cWorld` class is provided. The `cWorld` class includes three functions:

- `plot`: plots the robot gridworld
- `plot_value(v)`: takes a column vector of values $v(s)$ as input and plots the values into the gridworld
- `plot_policy(π)`: takes a column vector of actions $a = \pi(s)$ (deterministic policies) or a matrix of actions $\pi(a|s)$ (stochastic policies) as inputs and plots the policy into the gridworld

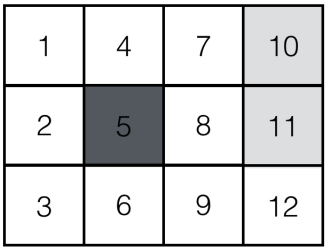
Experiment with the `cWorld` class by typing at the Matlab prompt:

```
>> myworld = cWorld();
>> myworld.plot;
>> myworld.plot_value(rand(myworld.nStates,1));
>> myworld.plot;
>> myworld.plot_policy(randi(myworld.nActions,myworld.nStates,1));
```

a.)[30 pts] Construct the transition model $p(s'|s,a)$ and reward function $r(s)$ for this MDP in Matlab.
b.)[30 pts] Solve the MPD using *value iteration* with $\gamma = 1$ and a termination threshold of $\theta = 10^{-4}$. Plot the optimal values into the gridworld by using the `plot_value` of the `cWorld` class. Plot the optimal policy in a different figure by using the `plot_policy` function of the `cWorld` class. Now, repeat your value iteration algorithm with (i) $\gamma = 0.9$ and $r = -0.04$ and (ii) $\gamma = 1$ and reward $r = -0.02$. Explain how these changes in parameters affect the optimal policy. For each run generate two plots showing the optimal value function and optimal policy.
c.)[30 pts] Solve the MPD using *policy iteration* with $\gamma = 0.9$ and $r = -0.04$. Initialize your policy iteration algorithm with a uniform random policy. Plot the value function and policy after each iteration step into two different figures of the gridworld by using the `plot_value` and `plot_policy` function of the `cWorld` class, respectively. Compare your results with the results obtained using value iteration.

Remarks:

1. To facilitate the checking of the exercise label your states in the gridworld as



   Label the actions as follows: $\{N, E, S, W\} = \{1, 2, 3, 4\}$.

2. Note that for $p(s'|s,a)$ to be a proper probability distribution function over successor states s' it is $\sum_{s' \in \mathcal{S}} p(s'|s,a) = 1$ for all s and a. Use this property to check your transition model.