CSE 150: Assignment 5
Value and Policy Iteration

Due : 2nd Dec, 2016

PROBLEM 1: POLICY IMPROVEMENT (5 POINTS)

(Individual written work: Please turn in 1 per team member and follow Gilligan's island rule)

Consider the Markov decision process (MDP) with two states s ∈ {0, 1}, two actions a ∈ {0, 1}, discount factor $\gamma = \frac{2}{3}$, and rewards and transition matrices as shown below:

s	R(s)
0	-2
1	4

s	s'	P(s' | s, a=0)
0	0	3/4
0	1	1/4
1	0	1/4
1	1	3/4

s	s'	P(s' | s, a=1)
0	0	1/2
0	1	1/2
1	0	1/2
1	1	1/2

a) (2.5 points) Consider the policy π that chooses the action a = 0 in each state. For this policy, solve the linear system of Bellman equations to compute the state-value function $V^{\pi}(s)$ for s ∈ {0, 1}. Your answers should complete the following table:

s	π(s)	Vπ(s)
0	0	
1	0	

b) (2.5 points) Compute the greedy policy $\pi'(s)$ with respect to the state-value function $V^{\pi}(s)$ from part (a). Your answers should complete the following table:

s	π(s)	π'(s)
0	0	
1	0	

Please show the calculations in your report. You can insert photos of your hand-written solution. For this question, each team member should attach his/her solution.

PROBLEM 2: VALUE AND POLICY ITERATION (15 POINTS)

In this problem, you will use value and policy iteration to find the optimal policy for the MDP described below. This MDP has |S| = 81 states, |A| = 4 actions, and discount factor γ = 0.99. The states are numbered 1..81 from top to bottom and left to right in the grid. Note that many of these states are unreachable!

Download the ASCII files from Piazza (assignment5_data.zip) that store the transition matrices and reward function for this MDP. The transition matrices are stored in a sparse format, listing only the row and column indices with non-zero values; if loaded correctly, the rows of these matrices should sum to one. There are 4 files (prob_*.txt) - each containing a transition matrix for one of the 4 actions - WEST, NORTH, EAST, SOUTH. The columns are in the order s, s', P(s' | s, a). "rewards.txt" contains 81 values corresponding to R(s).

(a) **(5 points)** Compute the optimal state value function V*(s) using the method of value iteration. Print a list of non-zero values of V*(s). Compare your answer to the numbered maze shown below. The correct value function will have positive values at all the numbered squares and negative values at the all squares with dragons.

(b) **(5 points)** Compute the optimal policy π*(s) from your answer in part (a). Interpret the four actions in this MDP as (probable) moves to the WEST, NORTH, EAST, and SOUTH. Your code "value_iteration.py" should print a list of (s, V*(s), π*(s)) tuples, for eg. (12, 1, "NORTH") - where 12 is the state number, 1 is optimal value and "NORTH" is optimal action.

Fill in the corresponding numbered squares of the maze with arrows that point in the directions prescribed by the optimal policy. Include this visualisation in your report and explain in short how you computed solutions to part (a) and part (b).

(c) **(5 points)** Compute the optimal policy π*(s) using the method of policy iteration. For the numbered squares in the maze, does it agree with your result from part (b)? Your code "policy_iteration.py" should print a list of (s, π*(s)) pairs. Explain how you did the computation for policy iteration in your report.

(d) Turn in your python files - value_iteration.py and policy_iteration.py along with the report. Also include a paragraph from each author stating his/her contribution.