Assignment A8: Policy Iteration

CS 4300
Fall 2017

Assigned: 17 October 2017

Due: 30 November 2017

For this problem, handin a lab report pdf (include name, date, assignment and class number in pdf) which examines the *value iteration* algorithm. The agent is to learn a policy for the following Wumpus world:

```
0 0 0 G
0 0 P 0
0 0 W 0
0 0 P 0
```

For this assignment, assume the actions available to the agent are

$$A = \{UP, LEFT, DOWN, RIGHT\}$$

where these are movements with probabilistic outcomes as described in the text (i.e., 0.8 probability of going the direction selected, 0.1 of going to either side). This requires development of a transition probability table for the 16 cells for the 4 actions. You are to implement the *modified policy iteration* algorithm given on p. 657 of the text and:

- comparable results to those given in Fig. 17.2 (a) and (b), p. 648 of R&N for the 16 utilities learned. In part (b) show results for several values of R so that some lead to the *gold* and some to *death*.
- what values you obtain comparable to those in Fig. 17.3, p. 651 in R&N.

Describe the algorithms in the method section, and verify their correctness. You should handin the report pdf as well as the Matlab source code used in the study. The code should conform to the style requested in the class materials (no matter what the language). In addition, please turn in a hardcopy of the report in class before the start of class on November 30, 2017.

Write a lab report in the format (please do not deviate from this format!) described in the course materials.

Here are the function descriptions of what you are to create:

```
function [policy,U,Ut] = CS4300_MDP_policy_iteration(S,A,P,R,k,gamma)
% CS4300_MDP_policy_iteration - policy iteration
%  Chapter 17 Russell and Norvig (Table p. 657)
% On input:
%     S (vector): states (1 to n)
%     A (vector): actions (1 to k)
%     P (nxk array): transition model
%     R (vector): state rewards
%     k (int): number of iterations
%     gamma (float): discount factor
% On output:
%     policy (nx1 vector): policy for problem
%     U (nx1 vector): final utilities found
%     Ut (num_iter by n array): trace of utilities (each is U at that ste
% Call:
%
%    Layout:              1
%                         ^
%     9 10 11 12          |
%     5  6  7  8       2 <- ->  4
%     1  2  3  4          |
%                         V
%                         3
%     [S,A,R,P,U,Ut] = CS4300_run_value_iteration(0.999999,1000);
%     [p,Up,Tpt] = CS4300_MDP_policy_iteration(S,A,P,R,10,0.999)
%     p'
%
% p =
%
```

```
%     1          corrresponds to:
%     2
%     2                 ->   ->  ->   X
%     2                 ^    X    ^   X
%     1                 ^    <-  <-  <-
%     1
%     1
%     1
%     4
%     4
%     4
%     1
%
% Author:
%     <Your name>
%     UU
%     Fall 2017
%
```