Assignment 9: Probabilistic Planning

CS 730/830, Fall 2015

Electronic submission due at **11:30pm on Mon, Nov 16**

Hardcopy submission due at **start of class on Tue, Nov 17**

Overview

You will write a planner that computes an optimal policy for a Markov decision process given on standard input using value iteration. Graduate students will also implement real-time dynamic programming. The MDP is represented using discrete atomic states and discrete actions. We will supply a test harness that will evaluate the performance of your policy by using it to control an agent over several trials in the MDP.

Input

The MDP format represents each state and action by an integer. The description starts by listing the number of states. Each state is then described, starting with state 0. The first line for each state lists the reward (a float), whether it is a terminal state (0 if non-terminal, 1 if terminal), and the number of actions available in that state. Then there is one line per action, starting with action 0, that starts with the number of successors and then lists each successor state along with its probability (a float). Comment lines start with #. For example:

```
# a tiny MDP that just happens to have two actions in
#   most states and two successors for every action
number of states: 4
start state: 0
# state 0
-1.0 0 2
# action 0 from state 0
2 2 0.9 1 0.1
# action 1 from state 0
2 1 0.9 2 0.1
# three more states
-10 0 2
2 3 0.9 0 0.1
2 0 0.9 3 0.1
-1 0 2
2 3 0.9 0 0.1
2 0 0.9 3 0.1
0 1 0
# EOF
```

Your program should accept three command line arguments:

algorithm the algorithm to run. For undergrads, this will also be `vi` but grads must also support `rtdp`. You should randomize your random number generator so that the simulation steps in `rtdp` give different results each time the program is run.

You may assume that when `rtdp` is selected, the MDP meets the necessary criteria (stochastic shortest path problem), all rewards are < 0, and $h = 0$ is an admissible initialization for the values of all states.

discount factor a float greater than 0 and ≤ 1.

termination criterion for `vi`, this will be a float representing the threshold for the maximum absolute change of any value during a Bellman backup before you can terminate. For `rtdp`, this will be an integer number of trials to run before you terminate.

Output

Your program should write a policy and the number of Bellman backups performed to standard output. The policy specifies the action to take in each state, one per line: the action for state 0 is listed on the first line, the action for state 1 on the second, and so on. For states with no actions available, the corresponding line can be left blank. For example:

```
0
0
0

345 backups performed.
```

Supplied Utilities

We supply:

sample-mdps.tar.gz some sample instances. The race track instances are named `track-`*width*-*height*-*width*-*max speed*`.mdp`.

`mdp-tester` your program's name is the first argument and your programs first two arguments are the next two arguments. More details about the remaining arguments will be forthcoming. Expects an MDP on standard input. Runs your program to derive a policy, then tests the policy by simulating its performance in the domain several times.

`mdp-reference` a sample solution.

Write-up

Submit a brief write-up with your hardcopy solution answering the following questions:

1. Describe any implementation choices you made that you felt were important. Mention anything else that we should know when evaluating your program.
2. What can you say about the the time and space complexity of your program?
3. What suggestions do you have for improving this assignment in the future?