Homework Assignment 1

This assignment asks you to implement versions of value iteration (VI) and policy iteration (PI)
for a particular MDP and evaluate their performance.
The assignment is due by Wednesday September 24, 1:30pm - in hardcopy (in class) and electron-
ically (via provide).

1 Comparing VI and PI

Consider a generalized version of the example MDP from class. We have a $n \times n$ grid with $n = 15$,
with a single reward of 10 at the top right state, and where the episode ends when this state is
reached. Intended moves: left and down are deterministic. Intended moves: up and right do not
always perform as intended and instead have the following transitions. Action: up goes up with
probability 0.5, left with probability 0.25, and right with probability 0.25. Action: right goes right
with probability 0.9, left with probability 0.05, and down with probability 0.05. In the above, if
the implied move hits the boundary the agent stays in the same position. We assume a discount
factor of 0.9 in this problem.

Your task is to implement the VI and PI algorithms where in PI we perform iterative policy
evaluation. Fix the stopping criterion of the $L_\infty$ norm error to be 0.001 in VI and in iterative
policy evaluation. Your implementation should be such that you can “empirically evaluate” the
greedy policy with respect to the current value function after every iteration. In “empirically
evaluate” a policy, we run it for 100 episodes where each episode is at most 100 steps long. Each
run starts at the bottom left state and the quality of the policy is evaluated as the average of the
total discounted reward received in these runs.

You should implement the algorithms and measure the quality of policies after every iteration.
Then plot the quality of the policies as a function of the “effort” of the algorithm. You should
produce two plots comparing VI and PI. In the first, quality is plotted as a function of the number
of iterations. In the second the quality is plotted as a function of the number of “single action
backups”. By the latter we mean the number of times you calculate $r(s, a) + \gamma \sum_{s'} Pr[s'|s, a] V(s')$
during the algorithm. Assuming we have $S$ states and $A$ actions in each state, a single VI iteration
costs $SA$ “single action backups” since we have $A$ such backups per state. A single PI iteration
costs $SA + SI$ such backups where $I$ is the number of internal iterations in policy evaluation. Note
that this means that in your code you must also store the number of such backups associated with
each iteration so that you can plot by this quantity as well.
2 Submitting Your Assignment

- You should submit the following items both electronically and in hardcopy:
  1. All your code for the assignment (please write clear code and document it as needed),
  2. Results of the simulations which should include the plots explained above and any observations on them.

- Please submit a hardcopy in class.

- Please submit electronically using provide: Put all the files from the previous item into a zip or tar archive, for example call it myfile.zip. Then submit using provide comp150aml a1 myfile.zip.

3 For Extra Fun and Credit

Those wishing to explore further may want to explore some variations of the above. Here are 3 ideas; you can try others as well. (1) How do the results change as a function of the error parameter chosen? (2) Change your PI code to MPI so that the number of internal iterations is manually controlled to be small. How does performance vary with this parameter? (3) Modify the problem so that several “walls” block the path in the grid and the effective paths to the goal get longer. How do the two algorithms fare in this case?