Assignment 1: Optimizing Finite-Horizon Expected Total Reward

You can use any programming language for the project, including MATLAB. In addition to the write-up materials indicated below, you should also send all of your code to me when you submit the assignment. Submit the project directly to me via email at afern@eecs.oregonstate.edu. Make sure that you receive an acknowledgement that the instructor received your assignment.

1 Part I: Build a Planner

Implement an MDP planning algorithm for optimizing finite-horizon expected cumulative reward. In particular, you should implement finite-horizon value iteration. The input to your algorithm should be a description of an MDP and a time horizon $H$ (positive integer). The output should be an optimal non-stationary value function and non-stationary policy for the MDP and time horizon. As an example input format, you could specify the number of states ($n$), the number of actions ($m$), an $n \times n$ transition matrix for each of the $m$ actions, and an $n$-dimensional vector for the reward function. The output format could be two $n \times H$-dimensional matrices, one for the non-stationary value function (column $i$ gives the value function with $i$ steps-to-go) and one for the policy (column $i$ gives the policy for $i$ steps-to-go).

2 Part II: Test the Planner

Here you will test the algorithm on two simple MDPs of your choosing. Design simple MDPs where it is easy to see the optimal policy and where the optimal policy depends on the time horizon (i.e. the optimal action at some states will depend on the horizon). The only constraints on the MDPs are that they have at least 20 states and have some probabilistic transitions (i.e. not all actions can be deterministic). Ideally the MDPs will have a meaningful description, which will help in interpreting the policy. The MDPs can be variants of one another, but should have different optimal policies.

Run your planning algorithm on the MDPs for at least two different values of the horizon time. Note that for two different horizons $H_1$ and $H_2$ such that $H_1 < H_2$, the non-stationary value function for the $H_2$ case will simply be an extension of the $H_1$ case. That is, the value functions will be identical when the time to live is $H_1$ or less. This gives you one way to help check your code.

You should produce a concise write-up that describes the MDP and describes the results of running your algorithm of different MDPs. In particular, you should show the optimal value functions and policies, demonstrating that the algorithm is producing the correct results.

IMPORTANT NOTE: Your next assignment will be very similar, but will involve optimizing policies for infinite horizon discounted reward, as well as producing results on an MDP of my choosing. As you will see, the Bellman backup operation can be used in that setting as well. Thus, it will be best for you if you at least write the Bellman backup operation as a generic function, allowing you to reuse it later.