Northeastern University
CS 4180/5180 – Reinforcement Learning and Decision Making
Fall 2019, Chris Amato

Dynamic Programming Assignment

Name: _____

Problem	Points
V^*, Q^*, and π^*	/20
Working the numbers for one iteration of dynamic programming	/10
Programming: Value iteration	/25
Total	/55

Instructions

- Don't cheat, write clearly, and staple your sheets!
- Questions about this assignment should be posted on Piazza for all to see.
- Contribute to the Piazza discussions, without providing direct answers.
- Graded out of 55 points for graduate students and 50 for undergrads.

(20 pts.) V^*, Q^*, AND π^*

a. Give an equation for V^* in terms of Q^*. Break it down into a form with no expectations.

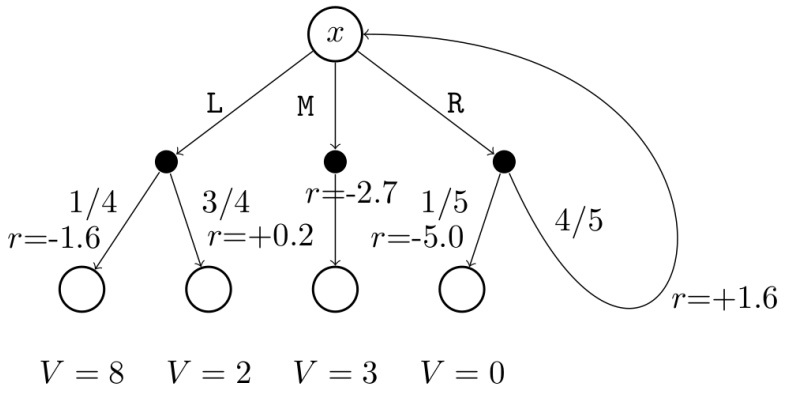
b. Give an equation for Q^* in terms of V^*. Break it down into a form with no expectations.

c. Give an equation for π^* in terms of Q^*, assuming that π^* is deterministic.

d. Give an equation for π^* in terms of V^*, assuming that π^* is deterministic.

(10 pts.) Working the numbers for one iteration of dynamic programming

Consider the following fragment of an MDP graph. The fractional numbers indicate the worlds transition probabilities and the decimal numbers indicate expected rewards. The three numbers at the bottom indicate what you can take to be the value of the corresponding states. The discount rate is 0.9.



$v_\pi(x)$ =

$v_*(x)$ =

a. Let π be the equiprobable random policy (all actions equally likely). What is the policy value $V^\pi(x)$ of the top node for this policy, assuming that the displayed values are themselves policy values V^π? Show your work.

b. What is the optimal value $V^*(x)$ of the top node, assuming that the displayed values are themselves optimal values V^*? Show your work.

(25 pts.) Programming: Value iteration

Complete the two programming exercises in the IPython/Jupyter notebook available on BlackBoard, named `notebook_dynamic_programming.ipynb`. Fill the coding sections, and submit your notebook on BlackBoard, naming it `notebook_dynamic_programming_firstname_lastname.ipynb`.

a. (20 pts) Make sure you are running python version 3.7 or higher. The programming homeworks will make use of the following libraries, which you should install and are encouraged to use yourself:

matplotlib : plotting library

numpy : numerical computing library

torch : numerical computing and automatic differentiation library

gym : popular interface for reinforcement learning environments

more_itertools : general-purpose recipes for iterables

ipdb : better python debugger

ipython : better command line interpreter

jupyter : runs interactive computing notebooks

tqdm : progress bar utility library

b. [**graduate**] (5 pts) The Gambler's problem has multiple optimal policies, and it is likely that your plot will not match the one shown in the book. Why is this the case? What do all the optimal policies have in common, and in what aspects can they differ?