Homework Assignment # 2

Due: Tuesday, February 21, 2017, 11:59 p.m.
Total marks: 100

Question 1. [2 MARKS]

In Example 4.1 (from the SB textbook), if π is the equiprobable random policy, what is $q_\pi(4, \text{up})$? What is $q_\pi(5, \text{left})$? Please explain how you arrived at your answer.

Question 2. [6 MARKS]

Exercise 4.2 in Sutton and Barto 2nd Ed., 2016. Answer both parts and explain your answer.

Question 3. [6 MARKS]

Exercise 4.6 in Sutton and Barto 2nd Ed., 2016. Your algorithm should only use state-action values (Q), not state-value functions (v).

Question 4. [4 MARKS]

Exercise 4.10 in Sutton and Barto 2nd Ed., 2016. (value iteration equation for action values). Explain your answer. Your equations should only use state-action values (Q).

Question 5. [25 MARKS]

Programming Question

Exercise 4.9 in Sutton and Barto 2nd Ed., 2016.

24 marks: Implement value iteration (algorithm box page 90) for the Gambler's problem. Create a plot of $v_\star$ and $\pi_\star$ similar to the figure from the book (figure 4.3). Experiment with two case: (1) where bet can be zero, and (2) where bet must be greater than or equal to one. Please pay attention to how you break ties! For this simple setting we do not have to use RL-glue, but you will be required to code in c or c++.

Please submitted your plots and ALL your code (including any scripts and data processing). I am aware that there are implementations of this problem available on the internet. **It is not acceptable to find the code online, modify it and submit it as your own. I expect you to implement this yourself, from scratch.**

Discuss why your plot sof $v_\star$ differs from figure 4.3 in the textbook.

Question 6. [3 MARKS]

Exercise 5.1 in Sutton and Barto 2nd Ed., 2016. [there are three subquestions] (discuss blackjack value function)

Question 7. [3 MARKS]

Exercise 5.6 in Sutton and Barto 2nd Ed., 2016.

Question 8. [6 MARKS]

Exercise 5.7 in Sutton and Barto 2nd Ed., 2016.

Question 9. [35 MARKS]

Programming: **Part 1, worth 30 marks**: Implement *Every-visit Monte Carlo Control with Exploring Starts* for state-action-values (Q(s,a)) on the *Gambler's problem* from Chapter 4. We will make two changes from the problem specification in the book. As in Question 4 above, let $\mathbf{pr}(heads) = 0.25$. In addition we will **not allow the zero bet action**; the agent must always make a bet of one or greater. Plot the $V_\pi(s)$ ($V_\pi(s) = \max_a Q_\star(s,a)$) for each state (on the x-axis). Plot $V_t(s)$ after 100 episodes,10000 episodes, and 10000000 episodes, producing a graph similar to the top plot in Figure 4.6. $V_t(s)$ should be produced by averaging over 30 runs. That is, your plot should contain 3 lines: $V_{t=100}$ averaged over 30 runs, $V_{t=10000}$ averaged over 30 runs, and $V_{t=10000000}$ averaged over 30 runs. You use may $t > 10000000$ if you like. My implementation takes approximately 15 minutes to run. It is running on a 2 yr old macbook pro, but could probably be further speedup.

You will use RL-Glue. Note you cannot send a query to the agent program from the experiment program; that is, the experiment program cannot ask the agent to send back V_t. However, you need to average V_t over runs. You could do this by writing V_t to a file from inside your agent on episodes 100,10000, and 10000000, on each run. Afterwards you could load the files into a data processing program like excel and compute the average and plot. Alternatively, you could make V_t a global variable. There is yet another solution that involves RL_agent_message and pointers.

This will require you to implement three things:

1. a simulation of the gamblers problem, that allows exploring starts (an Environment program)
2. the Every-visit Monte Carlo Control Algorithm with Exploring Starts (an Agent program)
3. code to run the experiment (Environment program)

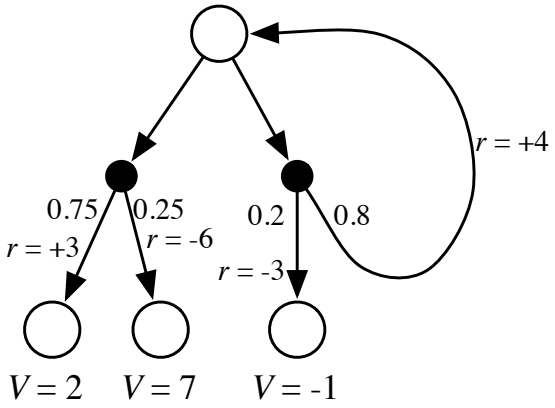
Please submit your plot and ALL your code (including any scripts and data processing).

Part 2, worth 5 marks: Discuss and compare how the plot produced by Monte Carlo Control with Exploring Starts is similar and different from the plot produced by value iteration (in question 5). Discuss why Dynamic Programming is suitable for the Gambler's problem and why the Monte Carlo method with exploring starts is less suitable for the Gambler's problem. Experiment with $\mathbf{pr}(heads)$ different than 0.25, to see what happens.

Question 10. [10 MARKS]

Consider the following fragment of an MDP graph. The fractional numbers indicate the world's transition probabilities and the whole numbers indicate the 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. What is the value of the top node for the equiprobable random policy (all actions equally likely) and for the optimal policy? Show your work.

$v_\pi =$ $v_* =$



Homework policies:

Your assignment will be submitted as a single pdf document and a zip file with code, on canvas. The questions must be typed; for example, in Latex, Microsoft Word, Lyx, etc. or must be written legibly and scanned. Images may be scanned and inserted into the document if it is too complicated to draw them properly. All code (if applicable) should be turned in when you submit your assignment. Use the RL-glue framework available on the course webpage (your code will be in c/c++), and any language of choice for plotting the results (learning curves).

Policy for late submission assignments: Unless there are legitimate circumstances, late assignments will be accepted up to 5 days after the due date and graded using the following rule:

on time: your score 1

1 day late: your score 0.9

2 days late: your score 0.7

3 days late: your score 0.5

4 days late: your score 0.3

5 days late: your score 0.1

For example, this means that if you submit 3 days late and get 80 points for your answers, your total number of points will be $80 \times 0.5 = 40$ points.

All assignments are individual work, no exceptions. All the sources used for problem solution must be acknowledged, e.g. web sites, books, research papers, personal communication with people, etc. You are expected to solve problems from scratch. That means, if you are asked to code something, do not find code online and modify it. Write it from scratch yourself. Academic honesty is taken seriously; for detailed information see Indiana University Code of Student Rights, Responsibilities, and Conduct.

Good luck!