Homework Assignment # 4

Due: April 6, 2017, 11:59 p.m.

Total marks: 100

Question 1. [10 MARKS]

Why is Dyna-Q considered a planning method? A good way to answer this is to say in words what makes something a planning method, and then say, in words, why Dyna-Q is such a something. (And do not say that something is a planning method because it plans.)

Question 2. [90 MARKS]

Programming question

You are to conduct are careful experiment with Dyna-Q with Prioritized Sweeping in the gridworld depicted in Figure 8.3 and described in Example 8.1 (page 173). The idea is to get you comfortable with sweeping parameters in a systematic way and summarizing the results in a compact and informative way. The Dyna-Q algorithm with prioritized sweeping is described in section 8.4.

[**Part one: worth 40**] Implement Dyna-Q algorithm with prioritized sweeping. Plot the number of steps per-episode over the first 50 episodes, averaged over 30 runs, with the following parameter settings ($\epsilon = 0.1$, $\alpha = 0.5$, γ=0.95, 5 steps of planning). Investigate different values of θ (the priority threshold). You should be able to make Dyna-Q significantly outperform Q-learning on this domain. The graph should have a similar form to Figure 8.3 in the text—but not likely to be exactly the same.

[**Part two: worth 40**] The second step is to test the sensitivity of Dyna-Q with prioritized sweeping with respect to α. To do this you will need to test Dyna-Q with prioritized sweeping with 9 different values of $\alpha \in \{0.025,0.05,0.1,0.2,0.4, 0.5, 0.8\}$, while holding $\epsilon = 0.1$, $\gamma = 0.95$, 5 steps of planning, and the value of θ you found to work well in part one.You will have to run 50 episodes and 30 runs for each of the 7 parameter settings. To summarize your performance you will plot the total steps to goal summed over 50 episodes (averaged over 30 runs) on the y-axis, verses α value on the x-axis. You may want to cut-off long episodes by using RL_episode(maxSteps)—this prevents episodes longer than maxSteps. Try maxSteps=2000. There should be 7 points on your graph; make a line plot to connect the points.

[**Part three: worth 10**] The third part of this question involves a careful scientific summary of your results. What are your conclusions from this experiment with regard to α in this domain?

Please read the page titled "Conducting a scientific experiment" for an explanation of how to discuss and summarized the results of a reinforcement learning experiment.

This question will require you to implement several things:

1. a simulation of the gridworld problem described in example 8.1 (an Environment program). Part 1
2. a Dyna-Q prioritized sweeping agent (Agent program). Part 1
3. code to run the experiment for 50 episodes, averaging over 30 runs, testing 9 different values of α (Experiment program). Part 2

Please submit your **two** plots and ALL your code (including any scripts and data processing).

[**Bonus (not required): worth 10**] Compare Dyna-Q with prioritized sweeping (using the best value of α found in part three) with n-step Sarsa (described on page 157, and implemented in question 7 in assignment # 3). Can you get n-step Sarsa to outperform Dyna-Q with prioritized sweeping in the gridworld from part 1? Use the following parameter settings, number of planning steps for Dyna-Q prioritized sweeping equal to 5; best α and θ for Dyna-Q found in the previous questions. You will have to experiment with the n and α parameters of n-step Sarsa to find good performance. This Bonus question requires one graph with the best performance of Dyna-Q with prioritized sweeping (line one), and the best performance of n-step Sarsa (line two)—or two plots with one line each. As before make a learning curve: number of episodes of the x-axis (1-50), and steps per episode on the y-axis. Results averaged over 30 runs.

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!