

CS 747: Programming Assignment 4

This assignment is meant to give you the experience of developing both agent and environment. Consequently it is more open-ended than your previous assignments. As a part of this assignment, you will implement the *Windy Gridworld* task given as Example 6.5 by Sutton and Barto (2018). You will program some agent-environment interactions, record your results, and present your interpretations. You can use any programming language of your choice for this assignment.

Tasks

1. Implement Windy Gridworld as an episodic MDP. The core of your code will have to be a function (or functions) to obtain next state and reward for a given state and action. You can use your own function names and conventions.
2. Implement a Sarsa(0) agent as described in the example, and obtain a baseline plot similar to the one accompanying the example (episodes against time steps). You can set learning and exploration rates as you see fit (just be sure to describe them in your report).
3. Get another plot when King's moves are permitted (that is, 8 actions in total), as described in Exercise 6.9.
4. Add stochasticity to the task as described in Exercise 6.10, and again plot the resulting performance of the Sarsa agent. Make sure you note down your convention for modeling corner cases.

In all your experiments, generate at least ten independent runs by varying the random seed. Plot the average statistic in the graphs.

Submission

Create a directory called `submission` and place the following material in it.

1. Your code for implementing the task and its variants;
2. Code for your Sarsa agent;
3. A script to run your simulations and gather data;
4. Plots of your agent's performance;
5. A README file describing how to run your code and obtain the plots; and
6. A report presenting your observations from these experiments (as a pdf file named `report.pdf`).
Place the plots in the report and provide accompanying commentary, rather than keeping the plots and text separate.

Compress the directory into `submission.tar.gz` and upload on Moodle under Programming Assignment 4.

Convince yourself that the results obtained match your expectations. Feel free to be creative and use the simulation environment to test related hypotheses you might find interesting. Your observations (under 6) must explain the variations observed across the three task settings, and report any particular issues you encountered while experimenting with this task. Don't hesitate to include additional numbers or graphs.

Evaluation

Your marks will be divided roughly equally among the three tasks you have to implement, in each case determined by the plot and the accompanying observations.

The TAs and instructor may look at your source code and notes to corroborate the results obtained by your program, and may also call you to a face-to-face session to explain your code.

Deadline and Rules

Your submission is due by 11.55 p.m., Friday, November 8. Finish working on your submission well in advance, keeping enough time to validate your code and to upload your submission to Moodle.

Your submission will not be evaluated (and will be given a score of zero) if it is not uploaded to Moodle by the deadline. Do not send your code or report to the instructor or TAs through any other channel. Requests to evaluate late submissions will not be entertained.

Your submission will receive a score of zero if your code does not execute on the sl2 machines. To make sure you have uploaded the right version, download it and check after submitting (but before the deadline, so you can handle any contingencies before the deadline lapses). If your code needs any special libraries to run on the sl2 machines, it is **your** responsibility to get them installed. You can do so by [filing a bug](#) with the CSE system administrators.

You are expected to comply with the rules laid out in the "Academic Honesty" section on the course web page, failing which you are liable to be reported for academic malpractice.