CM50270 Reinforcement Learning
Coursework 3
Özgür Şimşek

Date set: 9 April 2018
Date due: 19 April 2018, 8 pm
Total marks: 100 (This coursework will determine 30% of your mark for the unit.)
Where to submit: CM50270 Moodle page
What to submit: Completed Jupyter notebook (.ipynb file). Optionally, an additional pdf file.
You will be given a Jupyter notebook to work with. You must follow the instructions on this notebook and submit this particular notebook. Otherwise you will receive 0 marks.
Late submissions: We will follow the university policy on late submissions.
Coursework submitted after the deadline will receive a maximum mark of 40 (out of 100).
Coursework submitted after five working days will receive a mark of zero.
Feedback: Within two weeks of the submission deadline, your Jupyter file will be returned to you (via Moodle), showing the marks you received from each part of the coursework. You can get additional feedback from the unit leader or from the unit tutor on request.

You are required to work individually. You are welcome to discuss ideas with others but you must design your own implementation and write your own code.

This coursework will be marked anonymously. Please do not include any identifying information on the files you submit.

Do not plagiarise. Plagiarism is a serious academic offence. For details on what it is and how to avoid it, please visit the following webpage:
http://www.bath.ac.uk/library/help/infoguides/plagiarism.html

Two weeks of lab sessions are allocated for you to work on this coursework. Our tutor will be available during these lab sessions to answer your questions and to help you in any way possible. You can also consult the unit leader in her office hours, during the lectures, or by appointment.

What you need to do

Earlier in this unit, you have completed assignments where you were asked to think about the entirety of the reinforcement learning problem, creating the complete interaction of the agent with its environment. In this coursework, you will focus on implementing more advanced algorithms for the agent. Specifically, you will work on value function approximation in *Mountain Car,* a test environment well-known in the field.

You will be provided with the code that implements this environment. It will use the specification described in the textbook (Sutton & Barto, Reinforcement learning, First edition: Example 8.2 on pages 214–215), also available in the online edition of the book: http://www.incompleteideas.net/book/ebook/node89.html

Part 1 (50 marks)

Apply linear, gradient-descent Sarsa(λ) with tile coding and epsilon-greedy policy. Use ten 9×9 tilings, each offset by a random fraction of a tile width. This is the implementation in the textbook, in Example 8.2. The choice of parameters is up to you, including λ, α, type of eligibility trace (replacing or accumulating), and ε. Choose settings that allow efficient learning in the environment. Note that a more *efficient* method needs fewer observations to reach a certain level of performance. You can use the results in the textbook as a guide for choosing your parameter settings.

Marking guideline: If you can demonstrate some amount of learning, you will receive 40 marks or above, depending on how well your agent performs on average. If you can demonstrate accurate implementation of the algorithm and efficient learning, you will receive additional marks. Your code is expected to be well organised and readable. Otherwise, your marks may be lower than indicated in this guideline. Additional details are provided in the Jupyter file, including instructions on how you can demonstrate accurate, efficient learning.

Part 2 (50 marks)

Develop an agent that learns more efficiently, a better policy, or both, compared to the agent you developed in Part 1. You are free to explore any approach you find suitable. For example, you may decide to add more tiles, or tiles of different shapes, to your algorithm in Part 1. Or you may choose a different algorithm to implement. Explain the rationale for your approach, describe your results, and discuss your findings. Did your approach work better or worse than you expected? What did you learn from your experimentation? If you had more time, what other approaches would you try?

Please follow the instructions provided in the Jupyter notebook. You can optionally submit an <u>additional</u> pdf document of at most two pages describing your approach, your results, and conclusions.

Your work will be marked on the performance of your agent, how well you can justify your algorithmic choices, and how well you describe your results. Even if your approach does not work out well, you can still get high marks by explaining why you expected the approach to work well and what you learned from your experiments. Feel free to try multiple approaches and report your findings. Once again, your code is expected to be well organised and readable. Otherwise, you may lose marks.

<u>**You must follow the instructions in the Jupyter notebook precisely.**</u>