

# CM20252 & CM50263 Artificial Intelligence

## Programming Coursework

Özgür Şimşek

**Date set:** 17 July 2018

**Date due:** 17 August 2018

**Total marks:** 100

**Where to submit:** CM20252 Moodle page

**What to submit:** Three completed Jupyter notebooks (.ipynb files)

**What you need to do:** This coursework has three parts, described below. You will be given three Jupyter notebooks to work with, each notebook corresponding to one part of the coursework.

You must follow the instructions on the Jupyter notebooks precisely.

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

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.

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>

### Part 1: Design and implement a Sudoku solver (40 marks)

8	6					9	3	1
	4	5			9	8		
			3			6	5	4
	5	7			6	1		3
	9	1				2	8	
4		3	5			7	6	
7	1	6			3			
		4	6			3	7	
9	3	8					1	6

You are familiar with the Sudoku puzzle from the lectures. You will work with the standard size of the puzzle, which is a  $9 \times 9$  grid. A sample puzzle is shown on the left. The objective is to fill each cell with a digit ranging from 1 to 9 so that each row, column and  $3 \times 3$  section (shown in the figure within darker boundary lines) contain every digit from 1 and 9. For example, each row of the grid has nine cells; to solve the puzzle correctly, every digit from 1 to 9 must be present in each row. Some of the cells are already filled in with digits; these digits cannot be altered.

Your algorithm will be tested on a set of 100 Sudoku puzzles. Your mark will be the number of these puzzles solved correctly within 1 minute of computation time on our test machine (iMac, 3.5 GHz). This is the total allocated time for all 100 puzzles. Very straightforward solutions require less than 5 seconds on our test machine on the 100 test problems. It is certainly possible to write much faster algorithms but that is not necessary for receiving full marks.

You will not be given the set of 100 test puzzles but you will be given a similar set of 100 puzzles on which you can test your code. The two sets are similar to each other (they are sampled from the same population of Sudoku puzzles). Please note that some of the puzzles may have no solution. How to handle such cases is explained in the Jupiter notebook.

### **Part 2 (30 marks)**

You will implement the interaction of a reinforcement learning agent with its environment. The environment is a small, simple gridworld explained in the Jupiter notebook. You will compare the performance of two agents: one that chooses actions randomly and one that uses Q-learning. You will need to produce a plot that shows the total reward obtained in each episode by the two agents. Your code needs to produce this plot within 10 minutes of execution time. In the Jupiter notebook, you will find detailed, precise instructions.

### **Part 3 (30 marks)**

You will build a spam filter using the naïve Bayes classifier. You will be given two data sets, one for training your classifier and one for testing your classifier. In the Jupiter notebook, you will find detailed, precise instructions that will guide you in developing your classifier.

### **Feedback**

You will be given feedback within two weeks of the submission deadline. You can get additional feedback on your submission from the unit leader via appointment.