Homework 1

ECE 172A
Introduction to Intelligent Systems

Winter 2022

Make sure you follow these instructions carefully during submission. Not doing so may result in a significant penalty.

- All problems are to be solved using Python unless mentioned otherwise.
- Submit your homework electronically by following the steps listed below:
  1. Upload a pdf file with your write-up on Gradescope. This should include your answers to each question and relevant code snippets. We will only grade pages marked for each problem on Gradescope; problems without marked pages may receive a 0. Make sure the report mentions your full name and PID. Finally, carefully read and include the following sentences at the top of your report:
     Academic Integrity Policy: Integrity of scholarship is essential for an academic community. The University expects that both faculty and students will honor this principle and in so doing protect the validity of University intellectual work. For students, this means that all academic work will be done by the individual to whom it is assigned, without unauthorized aid of any kind. By including this in my report, I agree to abide by the Academic Integrity Policy.
  2. Upload a zip file with all your scripts and files on Gradescope. Name this file: ECE_172A_hw1_lastname_studentid.zip. This should include all files necessary to run your code out of the box.

1 Programming with Matrices (10 points)

(i) Using NumPy, set A = $\begin{bmatrix} 66 & -10 & 16 & 14 & 65 \\ -91 & 28 & 97 & -42 & 4 \\ 13 & 50 & -96 & 92 & -85 \\ 21 & -96 & 49 & 34 & 93 \\ -53 & 96 & 80 & 96 & -68 \end{bmatrix}$ and B = $\begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$

(ii) Find the row and col of all values in A that are less than -70.

(iii) Point-wise multiply A with B and set it to C.

(iv) Calculate the inner product of the 3rd column and 5th row of C.

(v) Find the maximum value in the 4th column of C. Then, find its corresponding row and column indices in matrix C.

(vi) Without using for loops, point-wise multiply the first row of C to all its rows (including the first row). Set the resulting matrix to D.

(vii) Repeat step (iv) for matrix D.

You may use library functions which find a particular value or maximum. In your report, include:

- The result for each sub-question above.
- Your code.

2 Robot Traversal (20 points)

Inspired by Shannon's mouse, you decide to simulate a robot to traverse a room without colliding with any objects. The goal of the simulation is to have the robot (a white pixel) move from the North side of the room to the South side while avoiding the objects (dark gray pixels). The simulated robot moves by adding a vector [v, h] to its current position, where v and h represent the vertical and horizontal directions respectively. The North-west pixel of the room has coordinates [0, 0], and the South-east pixel has coordinates [vSize - 1, hSize - 1], where vSize and hSize are provided in the code.

(i) The framework of the simulation is provided in HW1_2.py. Without changing the code, run the script and answer the following questions. Since there are pop-up images involved, it is recommended to run directly from the terminal instead of running embedded in a notebook. Note that you can press **q** (or close the image window) to advance to the next frame.

- What does *loc* keep track of?
- In what direction should the robot move when adding [1, 0]?
- Add a new object at [3, 6]. Explain what happens and why.
- If the robot were implemented based on the current state of the simulation, briefly explain why this robot would not be considered an intelligent system.

(ii) Uncomment the *if* statement in lines 74-75 and explain how this improved robot handles the object you placed at [3, 6]. Assuming that the function detectObject() can be practically implemented using sensors, briefly explain how this improved robot is a more intelligent system than the robot in part (i). Save the path of the simulated robot as an image once it reaches the goal and include it with your report.

(iii) Add another object at [4, 4]. Modify the code to solve this new case. The robot can only move one unit at a time (i.e. one move per iteration of the while loop), and it cannot move diagonally. Do not introduce any new variables or functions, and do not hard-code a solution. Only modify the logic in between the START and STOP comments. Save the path of the simulated robot as an image once it reaches the goal and include it in your report.

(iv) Finally, add another object at [4, 6]. Modify the code once again to solve this new case. The same rules apply: the robot can only move one unit at a time (i.e. one move per iteration of the while loop), and it cannot move diagonally. Do not introduce any new variables or functions, and do not hard-code a solution. Only modify the logic between the START and STOP comments. Save the path of the simulated robot as an image once it reaches the goal and include it in your report.

Hints:

- You may use the function haveIBeenHereBefore() that we have provided.
- Ctrl-C can be used to stop a Python script from the terminal, should you find yourself caught in an infinite loop.

In your report, include:

- Answers to all questions above in no more than 2 sentences each.
- Images showing the path of the simulated robot once it reaches the goal for parts (ii), (iii), and (iv).
- Include the code block that implements your robots' logic (the part between START and STOP) for parts (ii), (iii), and (iv).

Figure 1: Scenario 1

3 Bug Algorithms (30 points)

In class, we will soon talk about different bug algorithms used for simple robotic motion planning. We will analyze each algorithm's performance on a few example scenarios in this problem.

3.1 Lower and Upper Bounds

Imagine that you are a robot wanting to go from a start point to an end point. Between the start and the end point, there are the following obstacles:

- A building with a perimeter of 600 meters
- 4 cars, each with a perimeter of 12 meters

The straight line distance between the start and end point is 200 meters. What are the upper and lower bounds for distance traveled using the Bug 1 and Bug 2 algorithms? Assume the goal line intersects the building and each car. Provide the following answers in your report:

(i) Upper and lower bounds for the Bug 1 algorithm

(ii) Upper and lower bounds for the Bug 2 algorithm

3.2 UCSD

Consider the map shown in Scenario 1, which has no obstacle on the road. You are at the start point indicated in red, and wish to reach the end point indicated in red. You can go to anywhere in the map except the highlighted buildings and obstacles. Provide answers to the following questions:

(i) If you were to use Bug 0, Bug 1, and Bug 2 beginning at the start point, which algorithm(s) will reach the end point?

(ii) Which algorithm would result in the least distance traveled? Explain your reasoning.

Figure 2: Scenario 2

3.3 UCSD with Obstacles

UCSD with Obstacles: Consider the map shown in Scenario 2, which has obstacles scattered all about.

Provide answers to the following questions:

(i) If you were to use Bug 0, Bug 1, and Bug 2 beginning at the start point, which algorithm(s) will reach the end point?

(ii) Which algorithm would result in the least distance traveled? Explain your reasoning.

3.4 One More Obstacle

Consider the map shown in Scenario 3, which has one more obstacle.

Provide answers to the following questions:

(i) If you were to use Bug 0, Bug 1, and Bug 2 beginning at the start point, which algorithm(s) will reach the end point?

(ii) Which algorithm would result in the least distance traveled? Explain your reasoning.

(iii) Draw the path that each algorithm will take. (Hand-drawn or computer-assisted is fine.)

In your report, include answers to the 8 questions above, plus your drawing.

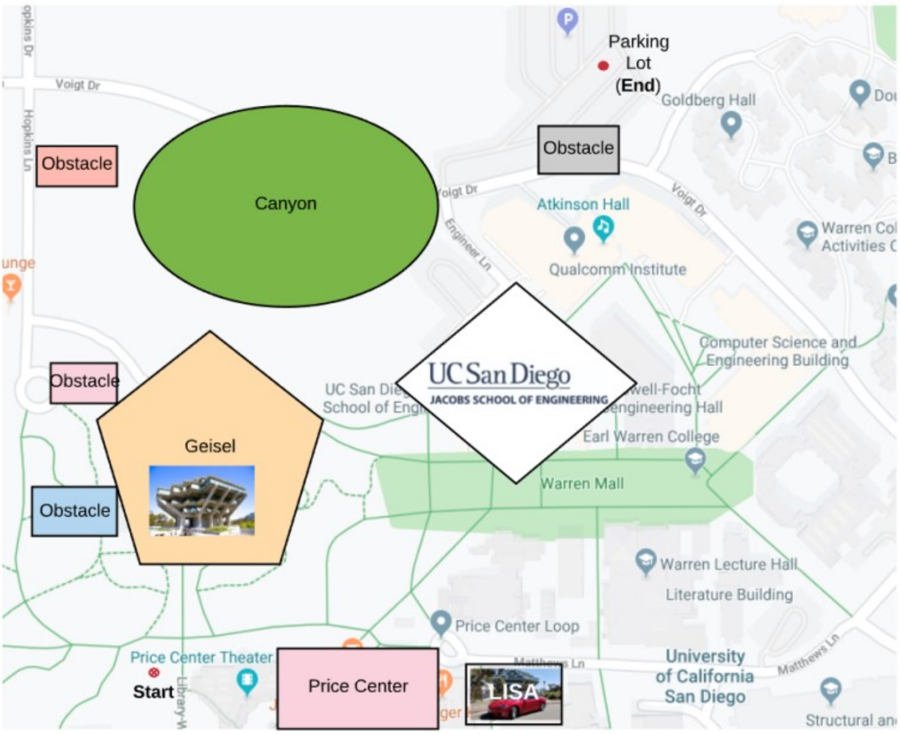

Figure 3: Scenario 3

4 BONGO, An Example of the Turing Test (30 points)

CAPTCHA, an acronym for Completely Automated Public Turing Test to Tell Computers and Humans Apart, is used to distinguish between computers and humans every day on the internet. One of the interesting CAPTCHAs is BONGO, which asks users to solve a visual pattern recognition problem. (Optional reading: Why is it called BONGO?) In this problem, we are going to build some simple visual recognition algorithms to hack a similar system like BONGO.

Consider four different BONGO generators. Each generator will create a set of binary images. Each set includes a pair of randomly generated BONGO blocks, which are labeled 0 and 1, and a test image. For each of the four generators, your task is to write a function to classify which BONGO block the test image belongs to. The input of your classify function will be three images (a pair of binary BONGO images, and a binary test image). Your output should be 0 (meaning the test image belongs to BONGO block 0) or 1 (meaning the test image belongs to BONGO block 1).

Your algorithm for each question will be evaluated on 100 sets of images. Feel free to use built-in functions in your approach.

Begin by running HW1_4_visualize.py (again, recommended to run from terminal, not notebook) to observe some BONGO examples. Then, answer the following questions:

1. If a system guesses the BONGO block at random, what is the expected value for the number of correct predictions out of 100 trials?
2. Describe the pattern observed in each of the four BONGO puzzles.
3. For each of the four BONGO puzzles, write a function to solve the puzzle, completing the code blocks in HW1_4_evaluate.py. You may then run the script to evaluate your algorithms' performance. The threshold for successful performance is 95%. Explain your approach in your report. If you are having trouble reaching the performance benchmark, you can still earn partial credit by explaining (and providing code for) the approaches you attempted, and discussing why the approaches may have failed.
4. Would you consider your approach to be an example of an intelligent system? Please explain, using the criteria introduced during lecture. Include in your discussion your thoughts on whether any of your approaches could be applied at once to all four of the puzzle patterns presented in the assignment, and whether your approaches would generalize well to other possible BONGO puzzles.

In your report, include:

- Answers for each question.
- Your classification function for each puzzle.
- Accuracy of your classification functions for each puzzle.

5 Literature Survey (10 points)

Prepare a one page report on a specific, recent intelligent robotics related story (not older than 2015) that you find interesting. Consider the following points in your report:

1. Identify the type of robot, types of sensors (both external and internal), types of planning and controllers that may be used, application domain, and your assessment of the significance of such robotic systems.
2. Make sure that one of the sensory modality used is vision.
3. What challenges may be encountered in the development of such a vision subsystem?
4. Discuss any technical issues which need resolution in development of this robotic system.
5. Discuss any ethical issues which should be considered in making such robots.