National University of Singapore
School of Computing

Semester 2, AY2021-22

CS4246/CS5446 AI Planning and Decision Making

Issued: 21 Feb 2022 **Due: 21 Mar 2022 at 23:59**

Assignment 2

Information

- This assignment is worth 10 marks out of 100. The score you obtain will be scaled to 10.
- This assignment should be submitted individually.

On collaboration & information source

- You are allowed to discuss solution ideas on the forums. However, you *must write up the solutions independently*.
- It is considered as plagiarism if the solution write-up is highly similar to other students write-ups or to other sources.
- If you obtained the solution through research, e.g. through the web, state your source in the answer rationale in the quiz.

Submission

- The written part of the assignment should be submitted via the corresponding LumiNUS quiz.
- Things to note:
  - You will have only one attempt at the quiz on LumiNUS.
  - There is no time limit for the attempt. However, the quiz closes on the designated deadline.
  - Use the *Rationale* field in the LumiNUS quiz to show your working, or paste the well-formatted answer, or both!
- For the programming part, please **upload the zip to the aiVLE evaluation server**.
- Additionally, the zip file containing your solution to the programming assignment should be submitted on LumiNUS
- ***Late submission:***
  - Deadline for the **written part** is fixed, no late submissions are allowed.
  - For the **programming component**, you will incur a late penalty of 50% of your score for late submissions.
  - Hard deadline for programming component: 25 March 2022 @ 23:59. No submission will be accepted after that.

Written questions (14 marks)

1. **[6 marks] Value of perfect information**

   DecisionMakers, a consultancy company is planning its advertisement budget for the next year. There are two modes of advertisement: via Mass media (MM) and via social media (SM). The company can choose to use either MM or SM or Both for airing advertisements. The costs involved are given in the matrix below (in thousands of dollars).

Economy → Advertisement mode ↓	Good	Bad
MM	$9	$11
SM	$11	$7
Both	$15	$12

   The company is trying to reduce the advertisement costs; hence will choose the most economical mode of advertisement. The economy is reviving from COVID-19. It has a 80% chance of being good and 20% chance of being bad.

   a) (2 marks) With the current information above, the company would choose:
      i. Social media (SM)
      ii. Mass media (MM)
      iii. Both channels

      Choose the most appropriate answer. In the rationale, please show your calculations to support your answer. (Answers without rationale will receive only half the marks).

   b) (2 marks) If there was an oracle that accurately tells the state of the economy (as being good or bad), what is the expected minimum cost DecisionMakers will incur? Fill in the blank correct to one decimal place. In the rationale, please show your calculations to support your answer. (Answers without rationale will receive only half the marks).

   c) (2 marks) DecisionMakers should pay less than $_____ (in thousands) to the oracle to obtain the information on the economy conditions. (Assume that the company will choose the most economical mode of advertisement)? In the rationale, please show your calculations to support your answer. (Answers without rationale will receive only half the marks).

2. **[6 marks] Markov Decision Process — Modeling**

   In this problem, we will try to model the COVID infection cycle in humans as a Markov Decision Process [1]. Given the current wave of infections, we can assume everyone is susceptible. When an individual comes in contact (action: *contact*) with someone carrying

[1] This is a very simplified model to be of any use in the real-world. For instance, we are not considering the effect of time or the viral load in the modeling. We are only concerned about the state transitions. Nonetheless, it gives you a chance to think about modeling real-world problems using MDP.

the COVID-19 virus, the individual is said to be exposed. For simplicity, we will assume a 50% chance of exposure. Every 1 in 10 people who is exposed gets infected (action: $progress\_infection$), whereas the others return to be susceptible. Once infected (next action: $progress\_recover$), there is a very low chance of mortality (2 in 1000 infected people). The others recover. However, recovery is not uniform. Half of the recovered individuals will have long-term effects and the other half recover to be fully well. Nevertheless, as long as the people are not dead, they will become susceptible again (action: $regress$)!

a) (4 marks) Identify the states and the transition probabilities that will help in modeling the above. If writing it in words is painful, you can draw the state transition diagram clearly indicating state and action nodes and label the nodes properly. Please write the transition probabilities on the arcs. Paste the image in the answer field.
Note: Assume generic actions mentioned above. This will allow you to define abstract actions that subsume a many real actions.
   - Individuals could go to a party or participate in mask-off activities in the presence of an infected (but asymptomatic) peer in the office, OR may maintain poor hygiene. But you can subsume them as $contact$ action.
   - Individuals can take medication or practice proning [2] to improve their breathing. But you can subsume them is an abstract action $progress\_recover$.

b) (2 marks) Comment on the nature of the transitions in your model above (Markovian or non-Markovian). Substantiate your choice/assumptions; any reasonable argument will receive marks.

3. **[2 marks] Markov Decision Process — Solution algorithm**

In an MCTS simulation, a state has been visited 32 times and has two child nodes, A and B. Node A has a won 4 out 8 times whereas node B has won 10 out of 24 times. Consider $c = 1$ and natural log in the UCT bound.

a) The value of U_{UCB} for node A is ___

b) The value of U_{UCB} for node B is ___

[2] https://en.wikipedia.org/wiki/Proning

Programming Assignment (6 marks)

Setup

OpenAI Gym (version ≥ 0.20) has introduced has introduced some breaking changes, due to which the environment built using the first assignment will not work for the MCTS task. To mitigate the task Gym Grid Driving environment has pegged the gym version to be 0.19 (by updating the `setup.py`).

If you have setup a stand alone version of the environment please take note of the above.

If you are using the Docker image, please re-pull the image to fetch the fix mentioned above.

Programming problem 1: Value Iteration

In the crossing the road task in Assignment 1, we assumed all the cars move with a constant speed of -1. Most of the times, this assumption does not hold. There is inherent noise in the car speeds due to reasons like bumpy roads, uneven driving, etc. This non-deterministic behaviour is not captured in the PDDL format.

For problems that are not very large, planning algorithms like Value Iteration can be used to handle non-deterministic behaviours. In this task, we model the problem as a Markov Decision Process and use Value Iteration algorithm to find the optimal policy.

We have provided the script which models the problem as an MDP. Every state in this MDP is represented as a tuple of features: $\{\text{Agent}_x, \text{Agent}_y, \text{Car}_x^1, \text{Car}_y^1, \cdots, \text{Car}_x^N, \text{Car}_y^N, \text{isTerminal}\}$, where:

- $\text{Agent}_x, \text{Agent}_y$ denote the x and y coordinates agent.
- $\text{Car}_x^N, \text{Car}_y^N$ denote the x and y coordinates of Car^N.
- isTerminal is a boolean variable denoting whether the state is terminal.

Your task is to fill up the function that implements the Value Iteration algorithm which is marked with "FILL ME".

You can test your code with the test configurations given in the script by running `python __init__.py N`, where N is an int value between 0 to 5, on the docker (using the `docker run ...` command).

HINT : You can use Matrix multiplication to optimise the code, but refrain from using the function `np.matmul()`/`np.multiply()`/`np.dot()` as these functions interfere with the evaluation script. Instead you can use function `matmul()` defined in the same script.

Programming problem 2: Monte Carlo Tree Search

Unfortunately, we moved to a bigger parking lot and Value Iteration is not feasible as it does not scale well with large state spaces. However, we can use Monte Carlo Tree Search (MCTS) to handle such cases.

We have provided a separate script which solves the problem with MCTS, Your task is to complete the script by filling up 2 code snippets inside the functions marked with "FILL ME".

The functions required to be filled up are:

a) backpropagate() : This function should implement the backpropation step of MCTS.

b) chooseBestActionNode() : Populate the list bestNodes with all children having maximum value. The value of the i^{th} child node can be calculated with the formula:

$$\frac{v_i}{n_i} + c\sqrt{\frac{\log N}{n_i}}$$

where,

- v_i : sum of returns from the i^{th} child node
- n_i : Number of visits of the i^{th} child node
- N : Number of visits of the current node
- c : The exploration constant. We set it to be 1.

For more details on MCTS, it might be helpful to look here. You are also encouraged to look through the rest of the code to see how the entire MCTS algorithm is implemented.

You can test your code with the test configurations given in the script by running `python __init__.py N`, where N is an int value between 0 to 5, on the docker (using the `docker run ...` command).