RMIT University Vietnam

School of Science, Engineering & Technology

COSC2129 – Artificial Intelligence

Assignment 3: Reinforcement Learning & Bayesian Network

Due: **17:00 PM, January 16th, 2025, Week 12**

Introduction

This is the third assessment for the course COSC2129 Artificial Intelligence, Semester 3, 2024.

- **Deadline:** Thursday, January 16th, 2025 @ 17:00pm (Week 12)
- **Course Weight:** 45%
- **Assignment type:** Individual
- **Submission method:** Upload zip files (see below for instructions)

This assignment covers the topics of **Reinforcement Learning** (RL) and **Bayesian Networks** (BN) in Artificial Intelligence. You have three tasks to complete.

You must read fully and carefully the assignment specification and instructions detailed in this file. You are NOT to modify this file in any way, except if instructed by the teaching staff in writing.

Your tasks

PART 1: Implementation

Task 1: Reinforcement Learning

In this **task** you must complete 4 Questions, from Question 1 to Question 4, in the UCB Project 3: Reinforcement Learning. In this project, you will implement value iteration and Q-learning. You will test your agents first on the Gridworld, then apply them to a simulated robot controller (Crawler) and Pacman.

The code for this project is given in **task1.zip** file. As in previous projects, this project includes an autograder for you to grade your solutions on your machine.

Files to Edit and Submit: You will fill in portions of valueIterationAgents.py, qlearningAgents.py, and analysis.py during the assignment. Please do not change the other files in this distribution.

Task 2: Bayesian Networks

In this **task** you must complete 4 Questions, from Question 1 to Question 4, in the UCB Project 4: Ghostbusters. In this project, you will design Pacman agents that use sensors to locate and eat invisible ghosts. You will advance from locating single, stationary ghosts to hunting packs of multiple moving ghosts.

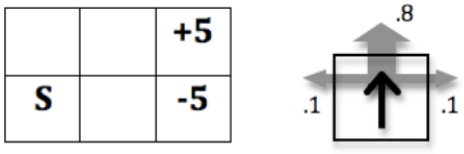
The code for this project is given in **task2.zip** file. As in previous projects, this project also includes an autograder for you to grade your solutions on your machine.

Files to Edit and Submit: You will fill in portions of bustersAgents.py, inference.py, and factorOperations.py during the assignment. Please do not change the other files in this distribution or submit any of original files other than these files.

PART 2: Writing presentation

Task 3

Question 1. (MDP and Reinforcement learning) Given a gridworld MDP of 2x3 grid as in Figure 1.a. The MDP operates like the one we had in lecture. The states are squares, specified by their (row, column) numbers. Agent S starts in state (1,1). Two terminal goal states are (2,3) with reward +5 and (1,3) with reward -5. State reward is used. Rewards are 0 in non-terminal states. The transition function is such that the intended agent movement (North, South, West, or East) happens with probability 0.8. With probability 0.1 each, the agent ends up in one of the states in directions as in Figure 1.b. If a collision with a wall happens, the agent stays in the same state.



(a) (b)

Figure 1: (a) Gridworld MDP. (b) Transition function.

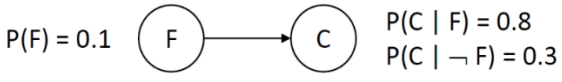
1.1. What is the optimal policy for this grid?
1.2. Suppose the agent knows the transition probabilities. Give the first two rounds of value iteration updates for each state, with a discount of 0.6. (Assume V_0 is 0 everywhere and compute V_i for times i = 1, 2).

Question 2. (Reinforcement learning)

2.1. For the following scenarios, briefly describe how we can model them as reinforcement learning problems (states, actions, type of rewards etc.).
i. Learning to play chess. Please provide a Q-Learning algorithm.
ii. You are about to graduate, and you decide to plan your finances for the next 30 years (until retirement). Consider what a reinforcement model might look like for financial planning. What can be challenges for the problem?

2.2. Consider chess. If we wanted to approximate the utility of the states using a sum of weighted features, discuss the type of features that might be useful for playing chess.

Question 3. Consider the following Bayesian network, where F = bad weather and C = traffic congestion.



3.1. Write down the joint probability table specified by the Bayesian network.

3.2. Are C and F independent in the given Bayesian network?

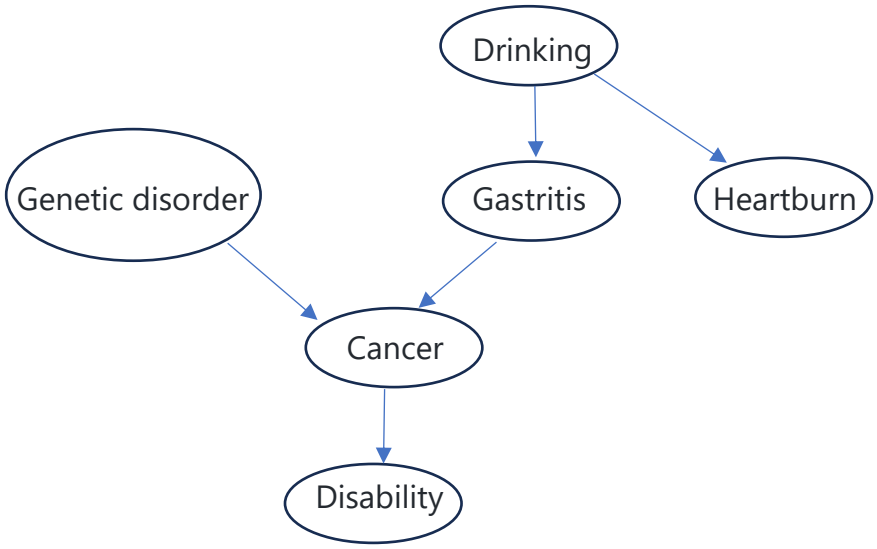
Question 4. Bayesian Networks

Consider the network for a disease (cancer) diagnosis shown in the figure below.

4.1. Extend the network with the Boolean variables *Hardworking* and *Stress*. State your assumptions and reason how components might affect each other.

4.2. Give reasonable conditional probability tables for all the nodes.

4.3. Assuming that no conditional independence relations are known to hold among the Boolean nodes, how many independent values are contained in the joint probability distribution?



Marking criteria

Task 1	Points
Question 1	4
Question 2	3
Question 3	4
Question 4	4

Task 2	Points
Question 1	4
Question 2	4
Question 3	4
Question 4	3

Task 3	Points
1.1	1
1.2	2
2.1	2
2.2	1.5
3.1	1
3.2	2
4.1	1.5
4.2	2
4.3	2

Evaluation for Task 1 and Task 2:

- Your code will be autograded for technical correctness. Please do not change the names of any provided functions or classes within the code, or you will wreak havoc on the autograder. However, the correctness of your implementation – not the autograder's judgements – will be the final judgement of your score. The teaching staff reserves the right to run more tests, inspect your code manually, and **arrange a face-to-face meeting** for discussion and demo of your solution to ensure that you receive due credit for your work. Failure to follow instructions can result in loss of points.
- Your code must run error-free on Python 3.10.6+. Staff will not debug/fix any code. Using a different version will risk your program not running with the Pacman infrastructure and may risk losing (all) marks.
- Your code **must not contain any personal information**, like your student number or your name. If you use an IDE that inserts your name, student number, or username, you should disable that.
- You must follow good SE practice during your development; please refer to Marking criteria below.
- You are free to add additional testing scenarios under the `test_cases/` folder.

Submissions not compatible with the instructions above will attract zero marks and do not warrant a re-submission. We will not debug or fix errors in your submission. Read carefully and ask for help (in forum or lab) if needed.

Important information

You must ALWAYS keep your work **private** and **never share it** with anybody in or outside the course, except your teammates (if it is a teamwork project), *even after the course is completed*. You are not allowed to make another repository copy outside the provided Classroom without the written permission of the teaching staff. Please respect the authors request.

Please do not distribute or post solutions to any of the projects.

Corrections: From time to time, students or staff find errors (e.g., typos, unclear instructions, etc.) in the assignment specification. In that case, a corrected version of this file will be produced, announced, and distributed to you. Because of that, you are NOT to modify this file in any way to avoid conflicts.

Late submissions & extensions: Late submission is not allowed. Extensions will only be permitted in *exceptional* circumstances under the University's rules.

Academic Dishonesty: This is an advanced course, so we expect full professionalism and ethical conduct. Plagiarism is a serious offense. Please **do not let us down and risk our trust**. Sophisticated *plagiarism detection* software via Codequiry may be used in this edition to check submitted code against other submissions in the class as well as resources available on the web. We will pursue the strongest consequences available according to the **University Academic Integrity policy**. In a nutshell, **never look at solutions done by others**, either in (e.g., classmate) or outside (e.g., web) the course.

Silent Policy: A silent policy will take effect **24 hours** before this assignment is due. This means that no question about this assignment will be answered, whether it is asked on the newsgroup, by email, or in person.

Code of Honour

We expect every RMIT student taking this course to adhere to the **Code of Honour** under which every learner-student should:

- Submit their own original work.
- Do not share answers with others.
- Report suspected violations.
- Not engage in any other activities that will dishonestly improve their results or dishonestly improve or damage the results of others.

Submission

Zip your Python files for Task 1 in **Task1.zip.** Zip your Python files for Task 2 in **Task2.zip**. **Task 3** should be presented in **one PDF file**. Then zip the three files and submit your Assignment 3 in **one final zip file**.
File name of the final zip file is your student number, for example 1234567.zip and 1234567.pdf, if your student ID is s1234567. Do not submit unnecessary files.

Acknowledgements

Task 1 and Task 2 are UCB Project 3 and Project 4 from the set of UC Pacman Projects. We are grateful to UC Berkeley CS188 for developing and sharing their system with us for teaching and learning purposes.