UNIVERSITY OF BIRMINGHAM

School of Computer Science

First Year Undergraduate

06-34238

34238 LC Artificial Intelligence 1

Main Summer Examinations 2022

[Answer all questions]

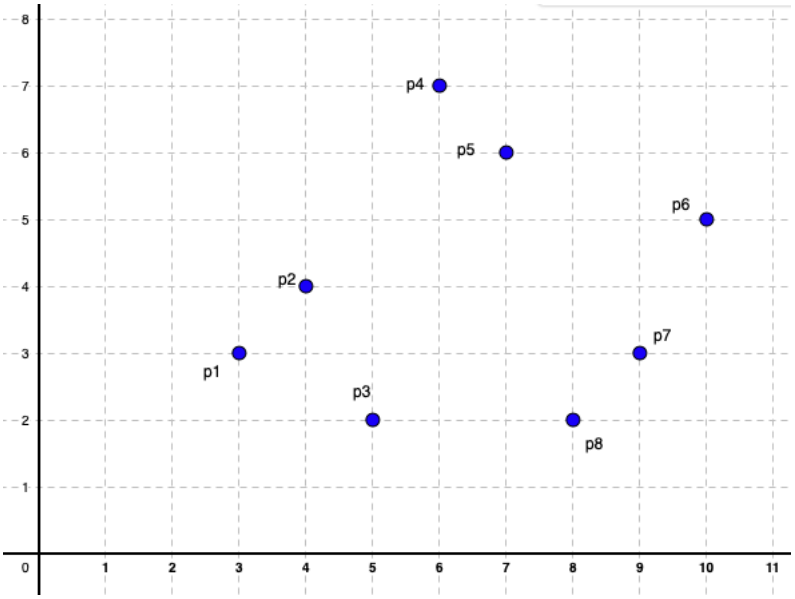
34238 LC Artificial Intelligence 1

Answer ALL questions. Each question will be marked out of 15. The paper will be marked out of 60, which will be rescaled to a mark out of 100.

Exam paper

Question 1 Clustering

Use DBSCAN based on Euclidean distance to cluster the data set shown in the picture below. This data set has eight 2-dimensional points, which are: p1=(3, 3), p2=(4, 4), p3=(5, 2), p4=(6, 7), p5=(7, 6), p6=(10, 5), p7=(9, 3), p8=(8, 2).



(a) The epsilon radius is set to 2, and the neighbourhood size (minPts) is set to 2. What cluster result would DBSCAN give? **Justify** your answer. **[8 marks]**

(b) If we would like to obtain three clusters produced by DBSCAN as: cluster1 (p1, p2, p3), cluster2 (p4, p5), cluster3 (p6, p7, p8), how would you set the epsilon radius and neighbourhood size? **Justify** your answer. **[7 marks]**

Question 2 Supervised Learning

(a) Give two advantages of k-Nearest Neighbours Regression over Linear Regression, and two advantages of Linear Regression over k-Nearest Neighbours Regression. Note that k-Nearest Neighbours algorithm can be used for both classification as well as regression. **[7 marks]**

Question 2 Supervised Learning

(a) Give two advantages of k-Nearest Neighbours Regression over Linear Regression, and two advantages of Linear Regression over k-Nearest Neighbours Regression. Note that k-Nearest Neighbours algorithm can be used for both classification as well as regression. **[7 marks]**

Linear Regression

Less computation required
(don't have to calculate distance
to every other point)

Less space required
(don't have to store ever single point)

kNN Regression

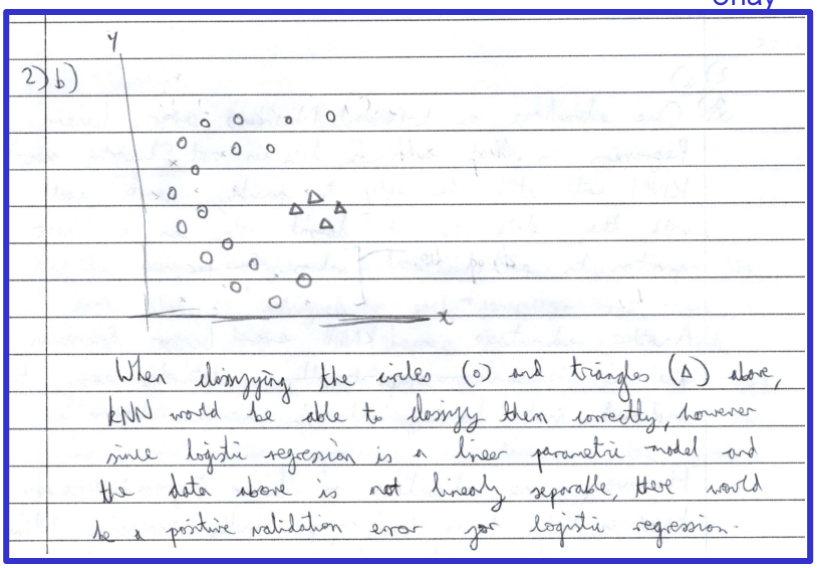
kNN can represent more
complex models

Can represent non-linear
relationships

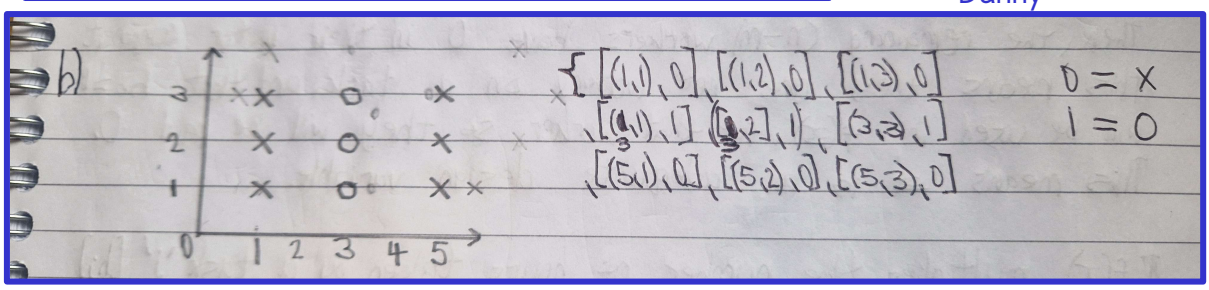
? Non-parametric = less
assumptions about data
distribution, so possibly
better fit

(b) Consider the following two classification methods: k-Nearest Neighbour with $k = 1$ employing the Euclidean distance (1NN), and Logistic Regression (LR). Create and draw a 2D labelled data set on which the leave-one-out validation error of 1NN is zero, but the leave-one-out validation error of LR is positive. **[8 marks]**

Shay



Danny



(b) Consider the following two classification methods: k-Nearest Neighbour with $k = 1$ employing the Euclidean distance (1NN), and Logistic Regression (LR). Create and draw a 2D labelled data set on which the leave-one-out validation error of 1NN is zero, but the leave-one-out validation error of LR is positive. **[8 marks]**

Question 3 Search Strategies

The Prisoner's Dilemma game is a game where two players interact. Each player has two possible actions: to cooperate or to defect. The payoffs associated with each combination of actions is shown in the table below, where the rows represent the actions chosen by player 1 and the columns represent the actions chosen by player 2.

Action	Cooperate (P2)	Defect (P2)
Cooperate (P1)	(3,3)	(0,5)
Defect (P1)	(5,0)	(1,1)

For example, if both players cooperate, they both get a payoff equal to 3. If player 1 chooses to cooperate and player 2 chooses to defect, the payoff for player 1 is 0, whereas player 2 gets a payoff equal to 5.

Each state of the game is identified by the payoffs accumulated by each player so far after playing multiple times in a row. The goal states are identified by a payoff for player 1 greater than 10 and a payoff for player 2 smaller than 10, indicating that player 1 won the game. For example, a node identified by $(15, 0)$ would meet the requirement of the goal test.

This problem can be formulated as a search problem as follows:

- Initial state is $(0, 0)$ and goal state is given by $(x > 10, y < 10)$, where x is the payoff for player 1 and y is the payoff for player 2.
- The actions for this game are defined by the combinations of cooperation and defection in the above table, namely: (C,C), (C,D), (D,C), (D,D). When you expand the nodes, choose the next node corresponding to the action in the this order: (C,C), (C,D), (D,C), (D,D).
- Nodes are identified by the values of x and y.
- The cost of each action is equal to 1. Loopy paths are acceptable in this case as they correspond (in general) to different sequences of actions.

(a) Generate the breadth first tree until the goal node is found. Write down the steps to solve the problem, from the initial state to the goal state. When expanding the nodes, use the values of the payoffs to identify nodes. **[10 marks]**

(b) Based on the tree that you generated above, what is the solution for this problem? And what is the cost of this solution? **[5 marks]**

Question 3 Search Strategies

The Prisoner's Dilemma game is a game where two players interact. Each player has two possible actions: to cooperate or to defect. The payoffs associated with each combination of actions is shown in the table below, where the rows represent the actions chosen by player 1 and the columns represent the actions chosen by player 2.

Action	Cooperate (P2)	Defect (P2)
Cooperate (P1)	(3,3)	(0,5)
Defect (P1)	(5,0)	(1,1)

For example, if both players cooperate, they both get a payoff equal to 3. If player 1 chooses to cooperate and player 2 chooses to defect, the payoff for player 1 is 0, whereas player 2 gets a payoff equal to 5.

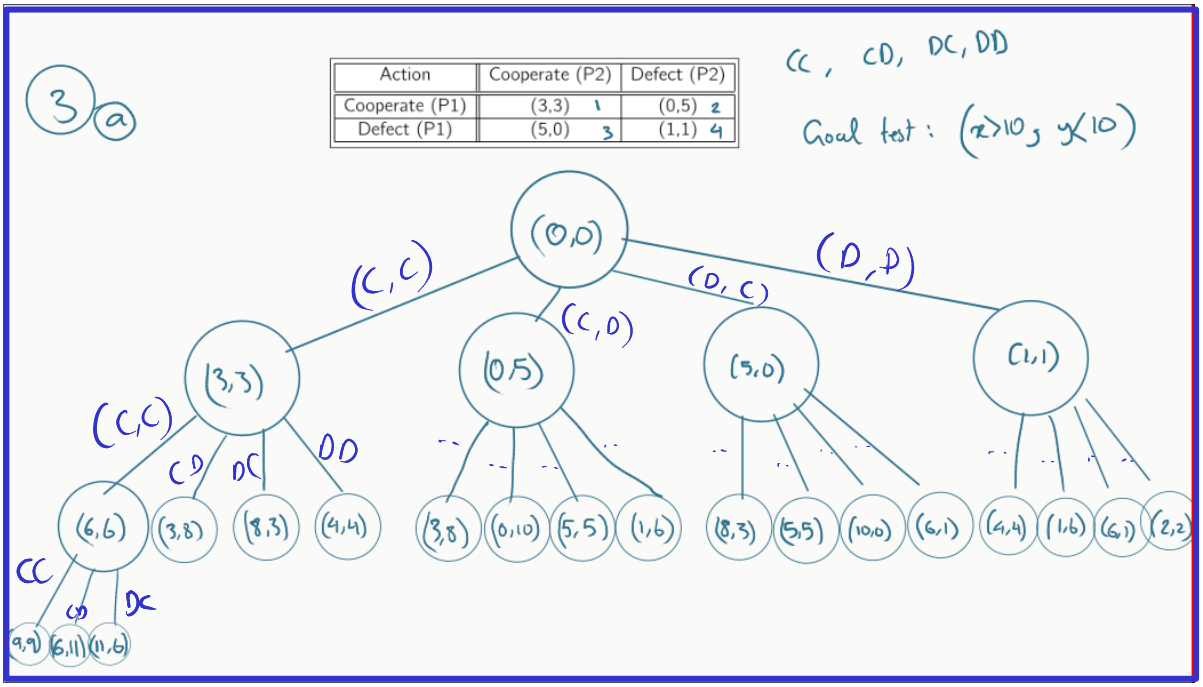
Each state of the game is identified by the payoffs accumulated by each player so far after playing multiple times in a row. The goal states are identified by a payoff for player 1 greater than 10 and a payoff for player 2 smaller than 10, indicating that player 1 won the game. For example, a node identified by $(15, 0)$ would meet the requirement of the goal test.

This problem can be formulated as a search problem as follows:

- Initial state is $(0, 0)$ and goal state is given by $(x > 10, y < 10)$, where x is the payoff for player 1 and y is the payoff for player 2.
- The actions for this game are defined by the combinations of cooperation and defection in the above table, namely: (C,C), (C,D), (D,C), (D,D). When you expand the nodes, choose the next node corresponding to the action in the this order: (C,C), (C,D), (D,C), (D,D).
- Nodes are identified by the values of x and y.
- The cost of each action is equal to 1. Loopy paths are acceptable in this case as they correspond (in general) to different sequences of actions.

(a) Generate the breadth first tree until the goal node is found. Write down the steps to solve the problem, from the initial state to the goal state. When expanding the nodes, use the values of the payoffs to identify nodes. **[10 marks]**

Sid



(b) Based on the tree that you generated above, what is the solution for this problem? And what is the cost of this solution? **[5 marks]**

Solution: (C,C), (C,C), (D,C)

Cost: 3

"visiting"

|

expanding neighbours

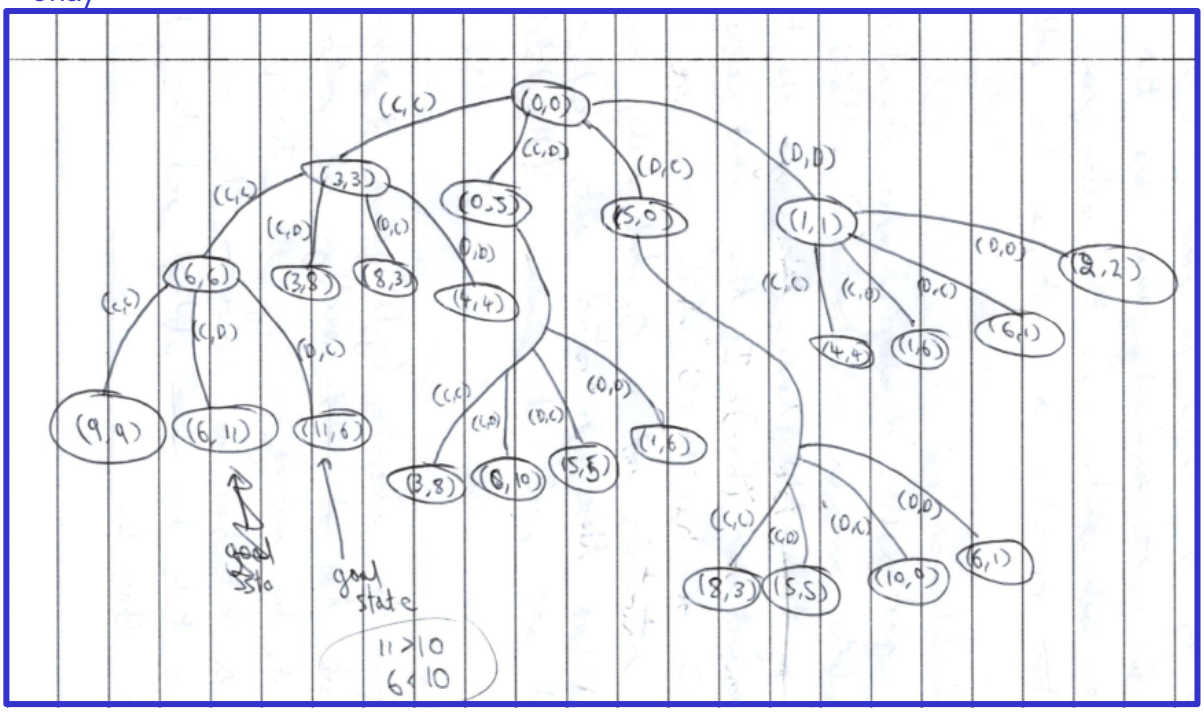
≠

"adding to frontier"

|

adding to queue

Shay



Shay

3) b) Solution:

((0,0), (3,3), (0,5), (5,0), (1,1), (6,6), (11,6), (8,10), (5,5), (10,9))

Cost = 3

Question 4 Optimisation Problem Formulation

Consider an illustrative problem where you wish to allocate n contractors to m tasks in a project, where $n > 0$, $m > 0$ and $n > m$. Each contractor i, $0 < i \leq n$, has a cost c_i corresponding to the amount of money they charge per hour. Each task j, $0 < j \leq m$, requires h_j hours of work. Each contractor must be allocated to at most one task, whereas each task must be allocated to one and only one contractor. We would like to find an allocation of contractors to tasks that minimises the total cost of the project, in terms of monies paid to contractors.

Consider that someone tried to formulate this optimisation problem as follows:

$$\begin{aligned} &\text{minimize} && f(\mathbf{x}) = \sum_{i=1}^{n} h_{x_i} \times c_i \\ &\text{subject to} && g_j(\mathbf{x}) = \sum_{i=1}^{n} I(x_i = j) = 1, \quad \forall j, 0 < i \leq m, \end{aligned}$$

where $\mathbf{x}$ is a vector of size n. Each position x_i, $0 < i \leq n$, of this vector contains an integer number j, $0 \leq j \leq m$, corresponding to a task j (or zero for the absence of a task). $I(x_i = j)$ is 1 if $x_i = j$ and 0 otherwise. The value of h_0 is 0.

Is this problem formulation adequate for the problem described above? **Justify** your answer by explaining the following in detail:

(a) explain the design variable of this problem formulation and why it is adequate / inadequate; **[5 marks]**

(b) explain the objective function $f(\mathbf{x})$ and why it is adequate / inadequate; **[5 marks]**

(c) explain the constraints $g_i(\mathbf{x})$ and why they are adequate / inadequate. **[5 marks]**

PS: if you believe you need to make any assumptions when answering the question, please list these assumptions in your answer.

Shay

4) a) The design variable x is suitable since it stores the task that is assigned to each contractor, and this is what we're trying to change to minimise the cost. It's adequate since we can only assign one task per contractor, since each position in the vector can only have one integer representing a task, as the problem states.

b) The objective function $f(x)$ is adequate since it calculates the hourly rate for a contractor (c_i) multiplied by the ~~number of tasks~~ total hours taken for the task they are assigned to (hx_i). If they aren't assigned a task, $x_i = 0$, so $hx_i = h_0 = 0$ which means we will consider that contractor's cost is 0, which makes sense. We then add up these contractors' costs (Σ), and this is what we want to minimise (the total cost).

c) The constraint is also adequate ($g_i(x)$) since it ensures that every task has ~~at most~~ exactly one contractor assigned to it, by making use of the $I(x_i = j)$ notation. If a task has no contractors, this will be 0, and if a task has ~~more~~ multiple contractors, this will be more than 1.