Calculators may be used in this examination provided they are not capable of being used to store alphabetical information other than hexadecimal numbers

UNIVERSITY OF BIRMINGHAM

School of Computer Science

Artificial Intelligence 1

Mock Exam 2023

Time allowed: 2 hours

[Answer all questions]

Question 2 Supervised Learning

(a) The following pseudo-code represents one iteration through the training set for gradient descent applied to univariate second order polynomial regression.

```
cost = 0;
w0 = 0;
w1 = 0;
w2 = 0;
for j in size(trainingSet)
      f = w0 + w1*x(j) + w2*x(j)^2
      cost = cost + (y(j) - f)^2
      w0 = w0 - a*(f - y(j))
      w1 = w1 - a*(f - y(j))*x(j)
      w2 = w2 - a*(f - y(j))*x(j)^2
endfor
```

Assume that the value of the learning rate, a is 1.

Give the numerical values of 'w0', 'w1', 'w2' and 'cost' at the end of this pseudo-code for the following training set: $\{(1,1),(2,5)\}$. Show all your working. **[10 marks]**

Shay

2) a) $w_0 = 0$ $w_1 = 0$ $w_2 = 0$ $cost = 0$

(1,1)

$f = 0 + 0(1) + 0(1)^2 = 0$

$cost = 0 + (1-0)^2 = 1$

$w_0 = 0 - 1(0-1) = 1$

$w_1 = 0 - 1(0-1)(1) = 1$

$w_2 = 0 - 1(0-1)(1)^2 = 1$

$w_0 = 1$ $w_1 = 1$ $w_2 = 1$ $cost = 1$

(2,5)

$f = 1 + 1(2) + 1(2)^2 = 7$

$cost = 1 + (5-7)^2 = 1 + 4 = 5$

$w_0 = 1 - 1(7-5) = 1 - 1(2) = -1$

$w_1 = 1 - 1(7-5)(2) = 1 - 1(4) = -3$

$w_2 = 1 - 1(7-5)(2)^2 = 1 - 1(8) = -7$

$\underline{w_0 = -1}$ $\underline{w_1 = -3}$ $\underline{w_2 = -7}$ $\underline{cost = 5}$

(b) Consider a multivariate data set with 2 classes that are not linearly separable. Is it true that the classes will still be not linearly separable

- (i) if you remove one point from this data?
- (ii) if you remove one feature from this data?

In both cases, justify your answers in the following way: if your answer is yes, then explain why; if your answer is no then give a counter-example. **[10 marks]**

Shay

b) If we remove a point from the data set, it may now become linearly separable. For example,

if we remove this point then this data will become linearly separable.

Shay

2) b) If we remove a feature from this data, it will remain not linearly separable. Removing a feature can be likened to multiplying the original value in a linearly ~~separable~~ linear equation by 0, as if it was not there in the first place. If we can't linearly separate the data in n+1 dimensions then we definitely won't be able to with n (one less) dimensions.

continued

Question 3 Optimisation

(a) Explain what are constraints in optimisation problems, and how they are usually mathematically depicted in a problem formulation. **[10 marks]**

Ashprit

```
Constraints are mathematical conditions that need to be satisfied in order for a can
didate solution to be accepted in an
optimisation problem formulation. They consist of the design variables and ensuring
that some mathematical equation or function
applied to these variables are subject to conditions ensuring feasibility. There can
be 0 or more of these constraints. They can
only ensure feasibility but also discourage infeasible solutions through constraint
handling like death penalties or levels of infeasibility in which we can apply a con
straint and change our objective function such that it is subject to a constraint th
at will discourage highly infeasible solutions more than lowly infeasible solutions
which prevent a local optima from being reached and converging, hence improving the
optimisation search formulation since we look for global optima which are better tha
n local optima in some cases.

F(w) is the cost function for the linear regression model. This allows us to calcula
te the cost between the predicted value from the hypothesis function and the output
value from the dataset labels. this is known as the mse because we square that diffe
rence sum all the differences and average it over all of the distances. And we are t
rying to minimise the cost in this optimisation formulation which is what we do duri
ng gradient descent / training a linear regression model.
```

Shay

3) a) Constraints in an optimisation problem define whether or not a candidate solution is feasible or not. We usually mathematically depict them under the objective function, denoted by "such that". For example,

max $f(x)$] objective function

such that $g(x): x^2 \leq 0$
$h(x): 2x - 23 > 0$

] constraints, typically depicted as some function being $=, >, <, \geq,$ or $\leq$ zero.

(b) Consider a regression task represented by a (potentially noisy) training set as follows:

$$D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})\},$$

where, for any $i \in \{1, 2, \ldots, n\}$, $x^{(i)}$ and $y^{(i)}$ are real values.

Consider a linear regression model for this regression task. The weights of this model are stored in a 2-dimensional vector $\mathbf{w}$ of real values, and the output of the model for an input x is given by $h(x; \mathbf{w})$. Assume that one proposes to formulate the machine learning problem of learning the weights $\mathbf{w}$ to be adopted for the regression task as an optimisation problem as follows:

$$\text{minimize} \quad f(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - h(x^{(i)}; \mathbf{w}))^2$$

(i) Explain your understanding of what the function $f(\mathbf{w})$ is calculating.

Shay

b) i) The function $f(w)$ is calculating the mean squared error (L2) loss function by calculating the difference between each labelled output and it's respective model output, squaring it, adding them up and dividing by the number of values to calculate a mean average.

(ii) Consider that one wishes to solve the problem given by the proposed formulation using simulated annealing with the design below:

Representation: direct representation of the design variable. In other words, a 2-dimensional vector $\mathbf{w}$ of real values.

Algorithm 1: Initialisation Procedure.

Output: Candidate solution $\mathbf{w}$.

$\mathbf{w}$ = new vector of size 2;
w_1 = real value picked uniformly at random between 0 (inclusive) and 1 (inclusive);
w_2 = real value picked uniformly at random between 0 (inclusive) and 1 (inclusive);
return $\mathbf{w}$

Algorithm 2: Neighbourhood Operator.

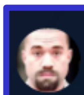
Input: Current solution $\mathbf{w}$.
Output: Neighbour $\mathbf{w}'$.

$\mathbf{w}'$ = copy of $\mathbf{w}$;
i = value picked uniformly at random from the set $\{1, 2\}$;
j = value picked uniformly at random from the set $\{-1, +1\}$;
$w'_i = w_i + j$;
return $\mathbf{w}'$

Discuss a key potential weakness of this simulated annealing design in the context of solving the problem given by the proposed formulation.

Note: assume that the proposed formulation truly represents the problem to be solved.

[10 marks]



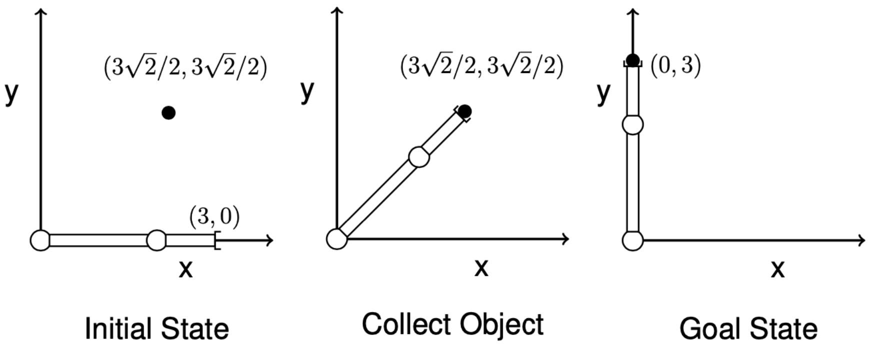
sid Today at 16:22
One potential weakenss of the design is that the neighbourhood operator only allows for local changes in range ± 1. This restricts the algorithm from being able to explore more potential solutions which may be located in distant regions of the solution space, due to this small range. Without sufficient ability to explore the soln space, the algo may converge slowly or potentially get stuck in local optima, hindering the overall ability to potentially find a global optimum or a satisfactory soln.

Shay

ii) In the neighbourhood operator we're picking a random weight and then either increasing it by +1 or -1. The step size here might be too big, so we aren't able to make more granular changes to weights e.g. ±0.1 or ±0.01. Therefore we might skip past potentially good solutions.

Question 4 Search

A planar robot with two degrees of freedom consists of two links that can rotate around the two rotational joints. The planar robot is placed at the origin as shown in the image below (initial state). The first link has length 2, while the second link has length 1 so that the end effector (i.e., the end of the robotic arm that is used to manipulate objects) is placed at coordinates $(3, 0)$ in the initial state.



The goal of the robot is to collect an object placed at coordinates $(3\sqrt{2}/2, 3\sqrt{2}/2)$ and move this object to the position identified by coordinates $(0, 3)$, as shown in the Collect Object and Goal State figure above, respectively. This problem can be formulated as a search problem as follows:

Question 1 continued over the page

Turn Over

- Initial and goal states as shown in the figure above.
- Actions: you can rotate one of the links by 45° or −45°, and you can collect the object only if the end effector is placed above it.
- Nodes are identified by the coordinates of the end effector and by the information if the robot is holding the object. To calculate the coordinates, use the following equations (forward kinematics):

$$x = 2\cos(\theta_1) + \cos(\theta_1 + \theta_2), \quad y = 2\sin(\theta_1) + \sin(\theta_1 + \theta_2),$$

  where θ_1 and θ_2 are the angles of rotation of the first and second joint, respectively, and cos and sin are the cosine and sine functions.
- The cost of each action is equal to 1. Always avoid loopy paths.

To calculate the cosine and sine of a given angle, please refer to the table below.

angle	cosine	sine
0	1	0
45°	$\sqrt{2}/2$	$\sqrt{2}/2$
90°	0	1

(a) Consider the state space of the above problem. In a real-life context, the robotic arm can rotate its links by any specified value, even non-integer values, e.g., rotate by 22.5° or 1.6666°. What impact would this have on the execution of a breadth first search on this problem and, more generally, how would this relate to the formulation of a search problem? **Justify** your answer. **[10 marks]**

(b) Generate the breadth first tree until the goal node is found. **[10 marks]**

When choosing which node to expand in the frontier and all nodes are at the same depth, always expand the node corresponding to the action in the following order: collect object (only if above the object), rotate link 1 by 45°, rotate link 1 by −45°, rotate link 2 by 45° and rotate link 2 by −45°. **Important: we only consider rotations if both coordinates of the position of the end effector are positive.**

Write down the following:

- Search tree produced by breadth first search, indicating which nodes are in the frontier when the algorithm terminates.
- The solution retrieved by breadth first search and its cost.
- The order in which the nodes are visited by breadth first search.

Question 4 Search

(a) Consider the state space of the above problem. In a real-life context, the robotic arm can rotate its links by any specified value, even non-integer values, e.g., rotate by 22.5° or 1.6666°. What impact would this have on the execution of a breadth first search on this problem and, more generally, how would this relate to the formulation of a search problem? **Justify** your answer. **[10 marks]**

BFS isn't complete if we're given an infinite search space - we're not guaranteed to find a solution.

(completeness = the ability for an algorithm to find a solution provided a solution exists)