1	2	3	4	5	6	7	8	total
40	20	20	30	25	35	15	15	200

UMBC CMSC 471 01, Final Exam, 17 May 2019

Name:_____

Write all of your answers on this closed-book exam which has eight problems that add up to 210 points. You have the two hours to work on this exam. There are two blank pages at the end that you can use. Good luck.

1. True/False (40 points) Circle T or F for each statement

T F An optimal solution path for a search problem with only positive costs will never have repeated states. True

T F A simple breadth-first search always finds a shortest solution if one exists that is of finite length. True

T F Hill climbing search algorithms only work for search spaces that are two-dimensional or have solution-preserving projections onto two-dimensions. False

T F An inference procedure for logic that is not sound cannot be complete. False

T F There are eight models for the sentence (a ∨ b ∨ c) among all of the interpretations over the three Boolean variables? False

T F The Ockham's Razor heuristic prefers the simplest consistent explanation. True

T F Every model of (a ^ b) is also a model of (a ∨ b). True

T F A propositional sentence is valid, if and only if it is satisfied in all possible models. True

T F Solving constraint satisfaction problems with forward-checking and arc consistence algorithms means that backtracking search is not required. False

T F A partial order planner is one that always produces a shortest possible plan. False

T F When overfitting occurs, a machine learning model describes random error or noise instead of the underlying relationships. True

T F The ID3 decision tree induction algorithm uses information gain and is guaranteed to find the optimal decision tree consistent with a given training set. False

T F A SVM with a soft margin will allow some positive training examples to be on the negative side of the margin and also some negative training examples to be on the positive side of the margin. True

T F In a zero-sum, two player game there is necessarily always a winner and a loser. False

T F One drawback of the K-means clustering algorithm is that one needs to specify how many clusters the algorithms should find. True

T F The cells in a NumPy array can only hold numbers or strings. False

T F Overfitting occurs when a machine learning model over-generalizes from the its training data. False

T F The recall metric used in a machine learning binary classification task is defined as the ratio of the number of true positives to the sum of the number of true positives and false negatives. True

T F A learning curve in machine learning is a way to evaluate how a system's accuracy varies with the amount of training data. True

T F Information gain is used to determine the network structure in Recurrent Neural Network. False

2. Multiple choice (20 points: 5, 5, 5, 5)

Circle the letters of **all** of the correct answers for each question.

2.1 What are advantages of using the alpha-beta algorithm over simple mini-max for games?

(a) Alpha-beta can handle games with uncertainty, whereas mini-max cannot.

(b) Alpha-beta does not require a static evaluation function, whereas mini-max always does.

(c) Alpha-beta is more accurate than minimax.

(d) Alpha-beta has a lower worst-case time complexity.

(e) none of the above **True**

2.2 How many models does the propositional sentence $(A \wedge (B \Rightarrow C))$ have?

(a) one

(b) three **True**

(c) five

(d) seven

(e) eight

(f) none of the above

A	B	C
1	0	0
1	0	1
1	1	1

2.3 What are some advantages of a partial order planner (POP) over a state-space planner (SSP)?

(a) A POP can find plans that a SSP will not.

(b) A POP plan can represent sub-plans that can be done in any order **True**

(c) A POP plan is always optimal

(d) A POP plan can have loops where as a SSP cannot.

(e) none of the above

2.4 What are possible advantages using a random forest classifier over a simple decision tree one?

(a) It tends to require less training data

(b) It can handle a much larger number of features than a simple decision tree

(c) It can be more accurate **True**

(d) It is less prone to overfitting the training data **True**

(e) none of the above

3. Short Answers (20 points: 10, 10)

3.1 In a few sentences, describe the approach used in the classic STRIPS planning algorithm.

The STRIPS planner compared the current and goal states to find a list of facts that are true in the goal state but not the current one. It then loops over this list of missing facts and, for each, looks for an action that adds the missing fact as one of its effects. If one is found, the planner tries to execute the action using a recursive call to the planner in which the actions preconditions are a new goal state. If that process succeeds, the action is performed, and the loop continues. If it fails, then another action that adds the missing action as one of its effects is tried. When all of the condition in the goal state have been achieved, the planner stops with success.

3.2 Describe one problem that the simple STRIPS planning algorithm has that can cause it to produce a plan that is longer than necessary.

The planner is prone to the "Sussman anomaly", in which it achieves a necessary condition in an initial part of the plan only to undo the condition later in the plan in order to accomplish some other goal. Eventually, the planner will note that the first condition has become unsatisfied and achieve it again. But the result is a plan with needless steps.

4. Resolution Proof in Propositional Logic (30: 10, 10, 10)

Consider a propositional KB with three variables (P, Q, R) and just two sentences:

- $P \vee Q \Rightarrow R$
- $\sim P \Rightarrow Q$

A propositional sentence is in CNF if it is a set of one or more expressions where each is a disjunction of variables or negated variables

4.1 Construct a KB of propositional sentences using the three propositional variables (P, Q, R) and logical connectives ($\vee$, $\sim$). Encode each of sentences into *one or more* logic sentences in conjunctive normal form (CNF). (10 pts)

#	English statements	CNF clauses (one or more)
0	$P \vee Q \Rightarrow R$	$\sim P \vee R$ $\sim Q \vee R$
1	$\sim P \Rightarrow Q$	$P \vee Q$

4.2 How many models are there for this KB? Recall that a model is an assignment of true and false to each variable such that the KB is consistent. (10 pts)

3

The KB sentences requires that either P or Q is true and also that R is true. This is satisfied by three models.

P	Q	R
0	1	1
1	0	1
1	1	1

4.3 Show a resolution refutation proof that R is true given these two sentences. Start with the negation of what's to be proved, add statements from your KB and then use resolution to derive a contradiction (⊥). The table to the right shows an example resolution refutation proof (10 pts).

step	action	result
1	assume	~Q
2	KB	~P∨Q
3	KB	P
4	resolve 2,3	Q
5	resolve 1,4	⊥

Sample proof of **Q** given **P→Q** , **P**

step	action	result
1	**assume**	**~ R**
2	**KB**	**~ P ∨ R**
3	**KB**	**~ Q ∨ R**
4	**KB**	**P ∨ Q**
5	**Resolve 1, 2**	**~ P**
6	**Resolve 4, 5**	**Q**
7	**Resolve 1, 3**	**~ Q**
8	**Resolve 6, 7**	**⊥**
9		
10		
11		

5. English and logic (25 points, 5, 5, 5, 5, 5)

We want to represent data about polygons and some of their sub-types, including triangles, squares and simplePolygons, which includes triangles and squares. For each of the following English statements, write **two different** ways to express it in first order logic. (Hint: one can use ∀ and the other ∃) Use the unary predicates **triangle(x)**, **square(x)**, **simplePolygon(x)** and **polygon(x).** Use the standard notation for first order logic as shown in the table on the right and use parentheses as needed to ensure that the precedence intended is used.

∀	forall
∃	exists
∧	and
∨	or
⇒	implies
⇔	iff
~	not

5.1 All simplePolygons are polygons

- ∀x simplePolygon(x) ⇒ polygon(x)
- ~ (∃x simplePolygon(x) ∧ ~ polygon(x)

5.2 No triangle is a square

- ∀x triangle(x) ⇒ ~ square(x)
- ~ (∃x triangle(x) ∧ square(x))
-

5.3 All triangles and squares are simple polygons

- ∀x triangle(x) ∨ square(x) ⇒ simplePolygon(x)
- ~ (∃x (triangle(x) ∨ square(x)) ∧ ~ simplePolygon(x))
-

5.4 Every simple polygon is either a triangle or a square

- ∀x simplePolygon(x) ⇒ triangle(x) ∨ square(x)
- ~ (∃x simplePolygon(x) ∧ ~triangle(x) ∧ ~square(x))
-

5.5 Given these definitions, does it follow that every triangle is a polygon?

- Yes

6. Decision tree reasoning (35: 10, 15, 10)

Predictive features			*target*
credit	**income**	**debt**	**approve**
Good	Low	Low	Yes
Good	Medium	Low	Yes
OK	High	High	Yes
OK	High	High	Yes
OK	Medium	Low	Yes
OK	High	Low	Yes
Bad	High	Low	No
Bad	Medium	Low	No
OK	Low	High	No
OK	Low	High	No
OK	Low	Low	No
OK	Low	Low	No

The table on the right shows 12 examples of decisions about a credit application (approved or not) based on three variables: credit history, income level and current debt. You plan to use this data to train an ID3-based decision tree to predict if an application will be approved given the values of credit, income and debt.

6.1 What is the initial entropy of the target variable, i.e., E(approve), which is defined as the sum of the probability of each value times the negative of the log base 2 of that probability. Recall that $\log_2$ of ½ is -1.

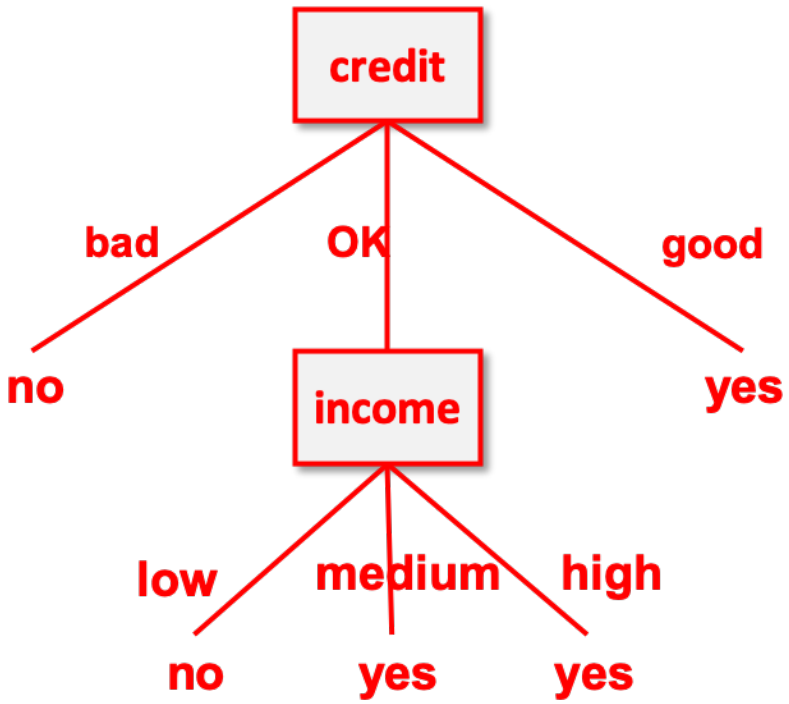
1.0

6.2 The ID3 algorithm selects the variable at each level that maximizes the information gained. Which attribute would be chosen as the root of the decision tree? (5 pts)

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

credit

6.3 Show the entire decision tree that would be constructed by ID3, i.e., by recursively applying information gain to select the roots of sub-trees after the initial decision. If the training data does not determine the predicted value for a case, use a "?" for the decision.(10 pts)



6.3 If we add an additional column to the predictive features which is an applicant's unique application number and then use that in addition to the other predictive features, what impact will it have on the decision tree that is learned by the ID3 algorithm?

The tree will only have one node which asks about the application number. This is an extreme case of overfitting! Worse yet, for any data with a new application number, the decision tree will not be able to predict the target

7. Perceptron vs. feed forward neural network (15)

Explain in a few sentences the difference between a simple perceptron and a feed forward neural network

A perceptron is a single-layer neural network with a single output that uses a step activation function to produce either a 0 or a 1. Learning is done by adjusting the weights on the input nodes.

A feed forward neural network is a collection of perceptrons organized in layers, so that the output of one layer is the input to the next. The perceptrons are modified so that the output is a continuous function of the inputs, rather than being a step. The learning algorithm is similar in principle to the perceptron learning algorithm, but involves a system of message passing from the output layer backwards to the input layer to adjust the weights on the links.

8. Batches, Epochs and backpropagation (15)

Two important parameters when training a neural network are **batch size** and **epochs**. Explain what each means and how they affect the backpropagation algorithm when a neural network is trained.

When training a neural network with a given collection of training data, we pass all of training data through the network many times. Each of these times is called an **epoch**. The number of epochs to use is typically determined experimentally by observing the error after each epoch and stopping when the desired accuracy is obtained, or improvements in the error approach an asymptote.

Systems that support **batches** divide the training data in to subsets where the batch size is the number of training examples in a subset. Training for one epoch is done in a loop where in each iteration, we run one of the training batches through the network, and note the error(s) for that batch. Depending on the algorithm, backpropagation is done at the end of each batch or the error information is collected for each batch and backpropagation done using all of the batch errors at the end of the loop.

For example, suppose our training data has 1000 instances and our batch size is 10. During one epoch, the system will process 100 batches. For each batch, 10 training examples will be sent through the network and at the end, the resulting error noted.