CS 199: Homework 4

Due by 10:00 a.m. on Monday, April 30th

1. **Concept learning**

   (a) Read sections 10.4 and 10.5 in the handout that was passed out, and understand in particular the learning method of 10.5.1 which keeps, as it goes, the most specific hypothesis that is consistent with positive training data seen so far.

   In this problem we consider a slight modification to the learning method of 10.5.1. Notice that when the second (positive) training example is processed, the attribute for "distance from car in front" changes in the hypothesis from "30 ft" to "?". (In class we used "*" to indicate that any value will match the rule.) This is because the two training examples had different values (the first had 30 feet, the second 20 feet).

   However, another possibility would be to represent a range of acceptable values - the tightest range possible that would still include the positive examples seen. Thus, instead of the second hypothesis being (slow, ?, ?, 0, evening, ?), it would be (slow, ?, [20-30], 0, evening, ?).

   Keeping track of the smallest (most specific) range for a numeric attribute is often done instead of generalizing immediately to "?" when two positive training examples are seen with different values for a numeric attribute. This allows greater expressivity in the learned rule.

   We consider a learning problem faced by an auto insurance company trying to determine which drivers are likely to be good insurance risks. They have a database containing those with good driving records (positive training data) and bad driving records (negative training data), along with various attributes of each customer. The data is displayed in the table below.

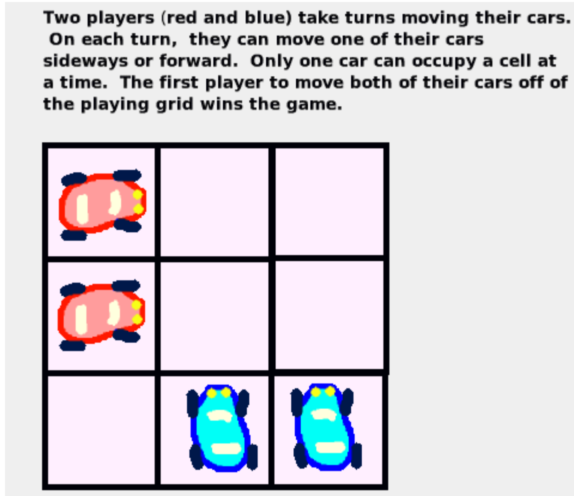
No:	Marital Status	Salary	Gender	Own/Rent	Car Type	Age	Driving Record
1	married	35,000	F	rent	sports	25	good
2	married	80,000	F	rent	sedan	35	good
3	married	50,000	F	rent	sports	20	bad
4	single	100,000	M	own	sedan	35	bad
5	married	30,000	F	rent	wagon	40	good
6	married	25,000	M	own	sports	20	bad
7	single	40,000	F	rent	sports	45	good
8	single	25,000	F	rent	sedan	40	bad
9	married	45,000	M	own	wagon	20	bad
10	married	45,000	F	own	sports	30	good

   (b) Using the modification of the technique described above, list the five hypotheses that the algorithm outputs after processing, in order, the five positive training examples given by customers 1, 2, 5, 7, and 10.

   (c) For each of the negative training examples, explain the reason(s) that the example is correctly classified (i.e., does not match the final hypothesis).

   (d) Suppose customer number 6 actually had a good driving record. What would the final hypothesis look like? How would your answer to the previous question change?

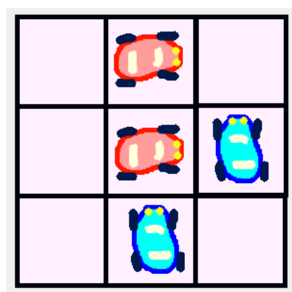
2. **A Game Tree Problem:** A game between two players is played on a 3 by 3 grid, as shown in the picture below. The players take turns trying to move their cars off of the grid in the forward direction; the red player goes first. Since only one car can occupy a cell at any time, blocking the other player is an important consideration during play. Cars may only move sideways or forwards, and a player must move on her turn. The player who first removes both cars from the grid wins the game.



(a) Read sections 6.2 and 6.3 from the handout passed out in class.

(b) Play this game with your friends a dozen times or more, to get a feel for the game. A good way to play is at a restaurant while waiting for your food, and using the pink and blue sugar-substitute packets. (A grid can be imagined, or laid out with silverware.)

(c) Draw a game tree for this game, with root node the starting position in the picture. Draw the tree ONLY THREE LEVELS DEEP. If you do this properly, you should end up with between twenty and thirty leaves (the nodes at the bottom of your tree.) There should be three edges coming out of the root node, corresponding to the three possible first moves for the red player. The second picture below is one that should appear as a leaf node in your game tree, because it can be reached in three moves - first red moves the top car forward, then blue moves the rightmost car forward, then red moves his lower car forward.



(d) Construct an evaluation function that you think would be useful for a computer to play this game. Considerations include how many cars you have removed from the board so far, how far along your cars are, whether your car(s) are blocking those of your opponent, and similar information about your opponent's cars and positions.

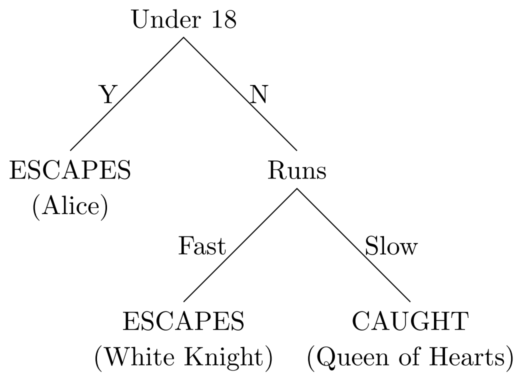
(e) Use the MIN/MAX method described in section 6.3, with your evaluation function, to compute the value at the root node, and what the best move is, assuming that we look ahead only to the third level of the tree, as you have drawn.

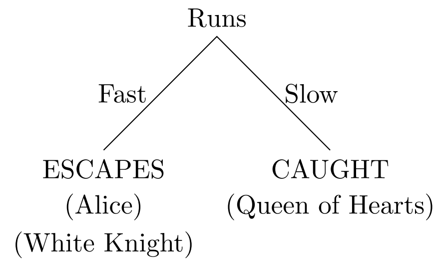
3. **Decision Trees:** In a decision tree, each node in the tree that is not a leaf corresponds to asking a question about one particular attribute. Depending on the answer to the question, you go to one of the children of that node. By the time we reach a leaf, we should know what the "decision" is. Here's a small example: The table below shows which, of several people, escaped the Jabberwock when it chased them.[1]

Name	Escapes?	Under 18? (Y/N)	Runs
Alice	Yes	Yes	Fast
White Knight	Yes	No	Fast
Queen of Hearts	No	No	Slow

[1] The Jabberwock is a fearsome monster. If you're familiar with the 'Alice' stories, forgive us for taking liberties with the characters and events.

Here are two trees that ask questions about the attributes “Under 18” and “Runs”, and decides whether or not the characters escape:





The second tree is smaller (it has 3 nodes, instead of 5), and probably a better predictor. How old you are probably doesn’t affect whether or not you can escape the Jabberwock, but it does matter whether or not you run quickly.

The table below lists the outcomes of several basketball games, along with lots of other information about those games. Try to build as small a decision tree as possible that correctly predicts when the team will win a game. Draw your decision tree, labeling each non-leaf node with the attribute corresponding to the question asked at that node. Label each edge from this node to a child with the appropriate answer to that question (we call this the “value” of that “attribute”). Finally, label each leaf “WIN” or “LOSE”, depending on the outcome of the relevant games.

Outome	Where	When	Fred Starts	JoeOffence	JoeDefense	OppCenter
Won	Home	7 p.m.	Yes	Center	Forward	Tall
Won	Home	7 p.m.	Yes	Forward	Center	Short
Won	Away	7 p.m.	Yes	Forward	Forward	Tall
Lost	Home	5 p.m.	No	Forward	Center	Tall
Lost	Away	9 p.m.	Yes	Forward	Forward	Short
Won	Away	7 p.m.	No	Center	Forward	Tall
Lost	Home	7 p.m.	No	Forward	Center	Tall
Won	Home	7 p.m.	Yes	Center	Center	Tall
Won	Away	7 p.m.	Yes	Center	Center	Short
Lost	Home	9 p.m.	No	Forward	Center	Short
Lost	Away	7 p.m.	No	Forward	Forward	Short
Won	Away	5 p.m.	No	Center	Forward	Tall
Lost	Home	7 p.m.	No	Center	Center	Tall
Lost	Home	9 p.m.	No	Forward	Forward	Short
Lost	Away	9 p.m.	Yes	Center	Forward	Short
Won	Home	7 p.m.	Yes	Center	Center	Short
Won	Home	7 p.m.	Yes	Center	Forward	Short
Lost	Home	5 p.m.	No	Forward	Center	Short
Won	Home	7 p.m.	Yes	Center	Forward	Tall
Lost	Away	5 p.m.	No	Center	Center	Tall

4. **Inference:** An *inference engine* is a system that tries to conclude new facts from existing facts, and valid rules of inference. In class, we saw an inference engine that tried to find out who had committed a crime.

   Some rules of inference are listed below; for each row in the table, IF we know the statements in the left column are true, we can conclude that THEN, the statements in the right column are true. (We use commas to separate two different statements in one table cell; also, recall that ¬ means NOT, ∨ means OR, ∧ means AND, and → means IMPLIES.[2])

[2] That is, $a \rightarrow b$ means a IMPLIES b, which is another way of saying IF a THEN b.

IF		THEN
p		$p \vee q$ (for any q)
p,	q	$p \wedge q$
p,	$p \rightarrow q$	q
$\neg q$,	$p \rightarrow q$	$\neg p$
$p \vee q$,	$\neg q$	p

Pause for a few minutes to convince yourselves that these are *valid* rules of inference. That is, try to see why each of these rules makes sense. For instance, the first rule says that for any statement p, if I know that p is true, then p OR q is true (regardless of what q is). But certainly at least one of p or q is true, because I already know that p is true. Therefore, this is a reasonable rule to use.

Here is an example of using these rules to conclude something about Alice and the Jabberwock. Suppose we know the following facts:

(a) Alice runs fast.

(b) Alice does not carry a sword.

(c) If Alice runs fast or carries a sword, the Jabberwock will not attack her.

Let r stand for the statement "Alice runs fast", and s for the statement "Alice carries a sword". Let j mean "The Jabberwock will attack Alice". Then, our three facts are equivalent to r, $\neg s$, $(r \vee s) \rightarrow \neg j$. (Take a minute to see that these symbols represent the three facts we listed.) Our inference engine could derive the following facts.

- r.
- Therefore, $(r \vee s)$ (From the first rule in the table).
- $(r \vee s)$ is true, and $(r \vee s) \rightarrow \neg j$ is true. Therefore, from rule 3 of the table, $\neg j$ is true.

Therefore, the inference engine concludes $\neg j$, that is, the Jabberwock will *not* attack Alice. You may have already realized this as soon as you read the list of facts, but computers aren't quite that intelligent. They have to run through inference procedures like this; for all we know, the human brain does some version of this as well, but it happens sub-consciously.

Suppose we had an inference engine that used the rules in the table, and after investigation of a homicide, the police knew the set of facts below:

(a) Victor was killed in the library.

(b) If Victor is dead, his murderer is Arthur, Bertram, or Carleton.

(c) If Arthur was the murderer, then he must have entered the library to kill Victor.

(d) If Carleton was the murderer, then he must have touched a statue just outside the library

(e) Arthur has never been to the library

(f) Carleton has never touched the statue.

Who is the murderer? Use only the inference rules listed in the table. You may need to come up with symbols such as *al* to denote the statement "Arthur has been in the library". List the set of facts the inference engine derives, as we did for the example with Alice and the Jabberwock. You should also explain which rule from the table you are using at each step.