Name: ________________________________ Andrew Login: __________________

15-381: Artificial Intelligence
Final Exam, Spring 2000

Instructions

This is an open-book examination. You may use any passive information sources: notes, books or calculators, but you may *not* consult any active sources such as other people, live internet/email connections, etc. You have up to 3 hours to complete all of the problems, although it should only take you only 2 hours or less to finish. Please write your answers directly on these sheets, and continue on the back if necessary.

Be sure to write your name and Andrew login on **every** page.

	Problem	Score	Possible
1	Short Answer		20
2	NLP		10
3	Probability		15
4	Machine Learning		20
5	Neural Networks		5
6	MDPs		15
7	RL		15
	Total		100

Name: ________________________________ Andrew Login: __________________

Problem 1: Short Answer [20 points]

This is the short-answer section spanning the entire course. Please respond *briefly*, and if you do not know the answer go on to the next item.

a) In bi-directional breadth-first search, if the distance between the start state and the nearest solution is d, and the average forward branching factor is b and the average backward branching factor is c, how many states will be visited? Express your answer as precisely as you can in terms of b, c and d. Partial credit for big-O approximation.

b) Assuming x and y are variables, A and B constants, are the following two expressions unifiable? If so, what is the unifier? If not, why not?

f(x, g(y, A, h(y, B))), and
f(h(A, B), g(A, y, x))

Name: __________________________________ Andrew Login: ___________________

Problem 1 Continued

c) Recall the vector-space model for information retrieval. If $w = |Dict|$ is the size of the dictionary after stop-word removal, $d = |Documents|$ is the number of documents in the collection, and $q =$ number of words in the query, what is the big-O complexity for finding the cosine similarity between the query and all the documents in the collection, in terms of w, d and q? Assume that document vectors contain only term-frequency counts, and that no precomputations have been done (no indexing) except for having precalculated and stored the vector-length of each document.

d) In speech recognition we calculate

$$\mathbf{arg} max_{word_i \in Dict} P(word_i | \text{acoustic signal})$$

to find the maximally probable word. Apply Bayes theorem to the above and conclude with the simplest expression that we need to maximize in speech recognition.

Name: _________________________________ Andrew Login: __________________

Problem 2: NLP [10 points]

In natural language processing, you need only a "bag-of-words" model for relevance assessments in information retrieval, yet you need full analysis of the language structure to parse text for tasks such as question answering and machine translation.

a) What is a potentially useful language-relevant task that requires a level of analysis beyond bag-of-words but less than full parsing? State the task and the level of analysis required.

b) Suppose we wanted much higher level of precision for retrieval from long queries and we wanted to index phrases as well as words (e.g., "air force", "stock market", "gross national product"). However if the dictionary contains 100,000 words, the number of trigram phrases is 10^{15} and therefore beyond reason. How would you get around this problem? Be creative.

Name: ________________________________ Andrew Login: __________________

Problem 3: Probability [15 points]

You have three binary variables:

- DAY (what day it is: tues/thurs)
- $LECT$ (who's lecturing today: Carbonell/Cohn)
- $LATE$ (is the lecturer late: true/false).

You are given $p(DAY)$, $p(LECT|DAY)$, and $p(LATE|LECT)$.

a) What other piece of information do you need in order to compute $p(LATE)$? Note: there is more than one correct answer!

Extra credit

We noted above that there was more than one correct answer. Can you give another one?

b) You startle awake at 1:35 and notice that you're in class, and the lecturer hasn't arrived ($LATE$=true). Using the (minimal) additional information you assumed in part 1, write a formula for the probability that it's Tuesday.

Name: ________________________________ Andrew Login: __________________

Problem 4: Machine Learning [20 points]

Consider an inductive learning situation where a prospecting robot is sent looking for gold into the modern equivalent of "them thar hills." Unfortunately, ye olde prospectors of the bygone gold-rush era are no longer among us to teach the robot how to recognize gold ore. That is, we can't simply do our two-phase protocol analysis and code up the rules. However, the geological museum has many labeled examples of gold ore and many other examples of non-gold-bearing rocks and minerals. For simplicity, assume that the robot has a reliable vision system that will convert an image of a geological specimen into an feature-value description language.

a) What learning method would you use if the gold and non-gold examples are linearly separable in feature space (i.e., a single hyperplane in feature space divides positive and negative instances)? What method might you use if they are not? Justify your choices.

b) Suppose the museum had 500 positive and 1000 negative samples of gold ore. Given that you would like the learning method to be likely to learn a hypothesis that does well on new (unseen) data, how would you set up training and testing to estimate well the accuracy of the learning method?

Problem 4 Continued

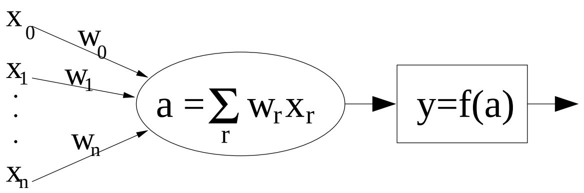
c) If you were building a decision tree and training it with 300 positive and 600 negative instances, what would be the information gain (IG) be for binary feature f that was present in 250 of the positive examples and only 50 of the negative ones? Would this be preferable as the root node of a decision tree compared with feature g that was present in 40 of the positive and none of the negative? Why or why not?

d) Suppose the robot's camera was less than reliable and failed to pick up the values of some features in the training and testing samples. (In 3-valued logic, these failures would be "?") How would you handle this situation for decision trees?

Name: _________________________________ Andrew Login: __________________

Problem 5: Neural networks [5 points]

You are constructing a feedforward neural network from units like the one in the accompanying diagram. Recall that the activation function f(a) does not necessarily need to be a sigmoid; in fact, there's no requirement that all units use the same type of activation function.



Given that you're planning on training the network using the error backpropagation algorithm, which of the following are legal activation functions?

1. $f(a) = \frac{1}{1+e^{-a}}$
2. $f(a) = sin(a)$
3. $f(a) = sgn(a) = \begin{cases} 1 & if\ a>0 \\ -1 & otherwise \end{cases}$
4. $f(a) = a^2$
5. $f(a) = 2a$
6. $f(a) = a$
7. $f(a) = |a|$

Extra credit

Are there some of these functions that you would use in the output layer of a network but not a hidden layer, or vice versa? Why or why not?

Name: ________________________________ Andrew Login: __________________

Problem 6: MDPs [15 points]

After graduating from CMU, you have been employed by the UpDown Elevator Company. Your first assignment is to write an elevator schedule for SleepWell Hotel. They want to install 3 elevators to service their 15 floors. Their top priority is to minimize the longest time a guest must wait for an elevator. Coming prepared with your knowledge from 15-381, you decide to try to model this as a Markov decision process.

a) Describe a reasonable state space, set of possible actions, and a reward function.

b) Does the Markov property hold for your system? Why or why not?

c) What type of learning algorithm would you apply to this problem? Why do you think this algorithm would work well and what possible difficulties could you face?

Name: __________________________________ Andrew Login: ___________________

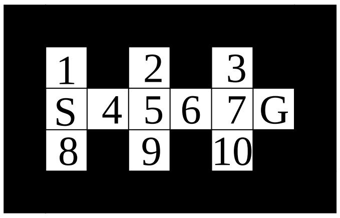
Problem 7: RL [15 points]

Having survived the dreaded alien cheese maze several times, you manage to convince your captors that, as a student of artificial intelligence, you can create for them a device that will learn to run the maze optimally.

You must design an algorithm that will run the maze below, starting at state S and running until it reaches state G.

Consider an algorithm that conducts a depth-first search to find a solution plan, then executes it open loop. The shortest such path plan is the policy "always go right," if we're willing to assume that nothing can go wrongcangowrongcan$\wedge C \wedge C$.

But the aliens are running their robot on an advance copy of Windows2157, so the device is slightly faulty: on any time step, there is an 85% chance that it will go in the intended direction, and a 5% chance that it will try to go in one of the other directions. If it attempts to go in a direction where there is a wall, it will remain in the current cell.



a) Will the robot always eventually reach the goal?

Problem 7 Continued

b) You decide to use dynamic programming to compute the expected number of steps to reach the goal when following the policy of "always go right." Now, a state's value is its "cost to go," defined to be the number of steps needed to reach the goal state. Bellman's equation still applies, except that our immediate reward is now an immediate cost of 1 for each step.

You decide to initialize all state values (except the goal state) to 1, assuming that you're always one step from the goal. You also decide to back up the state values in the order listed below. Next to each state, fill in the state value (expected cost to go) after that backup.

state	value after one backup
S	
4	
5	
6	
7	
6	
5	
4	
S	
7	
3	
3	

Name: ______________________________ Andrew Login: ________________

Problem 7 Continued

c) Consider a more clever algorithm that observes its state, and takes a greedy step towards the "best" next state. That is, this algorithm's policy for each state is always to move to the "best" next state according to the current value function.

Again, you decide to initialize all state values (except the goal state) to 1. You also decide to back up the state values in the order listed below. Next to each state, fill in the state value (expected cost to go) after that backup.

state	value after one backup
S	
4	
5	
6	
7	
6	
5	
4	
S	
7	
3	
3	