Name: Andrew/CS ID:

15-780 Midterm Solutions, Fall 2006

November 15, 2006

- *Place your name and your andrew/cs email address on the front page.*
- *The exam is open-book, open-notes, no electronics other than calculators.*
- *The maximum possible score on this exam is 100. You have 90 minutes.*
- *Don't spend too much time on any one problem. If you get stuck on any of the problems, move on to another one and come back to that problem if you have time.*
- *Good luck!*

1. **Search [15 pts]**

   Answer true or false and provide a brief explanation of your answer. (**3 pts** each, **2 pts** for T/F; **1 pt** for discussion)

   (a) [**3 pts**] All of the search algorithms we discussed require space which is exponential in the solution depth (in the worst case).

   Answer: False. Depth first search uses linear space.

   (b) [**3 pts**] All of the search algorithms we discussed require time which is exponential in the solution depth (in the worst case).

   Answer: True. All the search algorithms we discussed will in the worst case have to expand at least the nodes at depth less than or equal to the solution depth, and there can be exponentially many such nodes.

   (c) [**3 pts**] Manhattan distance ($|x_1 - x_2| + |y_1 - y_2|$) is an admissible heuristic for any problem that involves finding a path through a maze.

   Answer: False. Manhattan distance is not admissible for maze problems where diagonal moves are valid, or when some moves cost less than 1. It is also inadmissible when searching for the longest path.

   (d) [**3 pts**] With an admissible heuristic, A* always finds an optimal goal while expanding fewer nodes than any of the other search algorithms we discussed.

   Answer: False. Depth-first search can get lucky and expand only nodes on the path to a goal. A* does provide a guarantee that it will expand only nodes with f value $\leq$ to the goal nodes, but this is a different guarantee.

   (e) [**3 pts**] Random restarts help when search times are sometimes short but the distribution has a heavy tail.

   Answer: True. In this case a restart may help prevent the search algorithm from getting "stuck" in a bad trial.

2. **Logic [7 pts]**

   (a) [**4 pts**] Write down the following sentences using first-order logic. Use predicates IsStudentOf, IsSmart, IsAdvisor and GetsJob.

   i. Anna is a student of Bill.
   ii. All students of a smart advisor are smart.
   iii. Bill is smart.

iv. Smart students get jobs.

Answer:

i. IsStudentOf(Anna, Bill)
ii. $\forall\ x, y$ IsStudentOf$(x, y) \wedge$ IsSmart$(y) \Rightarrow$ IsSmart(x)
iii. IsSmart(Bill)
iv. $\forall x\,(\exists y$ IsStudentOf$(x, y) \wedge$ IsSmart$(x)\ ) \Rightarrow$ GetsJob(x)

(b) [**3 pts**] Suppose we have the first-order sentence $\forall x$. happy(x) (call this sentence S). Also suppose that our universe consists of two people, Fred and Harry. Write a propositional sentence (that is, a sentence containing no quantifiers or variables) which is logically equivalent to S.

Answer: happy(Fred) $\wedge$ happy(Harry)

3. **Normal form games [13 pts]**

Consider the two-player general-sum game defined by the following payoff matrices: for the row player,

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

and for the column player,

$$\begin{pmatrix} e & f \\ g & h \end{pmatrix}$$

(a) [**3 pts**] Suppose the row player plays the strategy (.3, .7). (That is, 30% chance of playing the top row.) What is the column player's expected payoff for her first strategy? For her second strategy?

Answer: first strategy, $0.3e + 0.7g$; second strategy, $0.3f + 0.7h$

(b) [**3 pts**] Suppose the row player wants to pick a probability p of playing the first row so that the column player's two strategies have equal payoff. Write a linear equation which determines p as a function of $a, b, c, \ldots, h$. Don't worry about writing down the constraint $0 \leq p \leq 1$.

Answer: $pe + (1-p)g = pf + (1-p)h$ or $p = (h-g)/(e-g-f+h)$.

Now suppose we fill in the actual numerical values of the payoffs: for the row player,

$$\begin{pmatrix} 1 & 5 \\ 2 & 3 \end{pmatrix}$$

and for the column player,

$$\begin{pmatrix} 3 & 8 \\ 6 & 7 \end{pmatrix}$$

(c) [**3 pts**] Given the above payoffs, what numerical value of p solves the linear equation from (b) (again, ignore the inequality constraints)?

Answer: $p3 + (1 - p)6 = p8 + (1 - p)7$, so $p = -1/4$. The value outside of $[0, 1]$ means that the row player can't equalize payoffs for the column player's strategies. This happens because the column player's first strategy is dominated, and it means that there are no Nash equilibria in which the column player plays the first strategy.

(d) [**4 pts**] Use reasoning about the game's dominance structure to determine one of its Nash equilibria.

Answer: Row player always plays top row; column player always plays right column. We showed in (c) that the column player must play the right column, and given this fact, the row player gets 5 for playing the top row and only 3 for playing the bottom.

4. **Linear and integer programs [17 pts]**

Here is a linear program:

$$\begin{aligned} \text{maximize} \quad & x + y \\ \text{subject to} \quad & 2x + y \leq 4 \\ & 2x + 3y \leq 6 \\ & x, y \geq 0 \end{aligned}$$

(a) [**3 pts**] Sketch the feasible region of this LP.

Answer: See Figure 1.

(b) [**3 pts**] What is the optimal solution and its value?

Answer: $(x, y) = (3/2, 1)$, with value $5/2$. The active constraints are $2x+y = 4$ and $2x+3y = 6$. So, eliminating x, $3y-y = 6-4$, or $2y = 2$, or $y = 1$. Substituting in, $2x + 1 = 4$, so $x = 3/2$.

(c) [**3 pts**] Suppose we add the constraint that x and y must be integers. What is an optimal integer solution and its value?

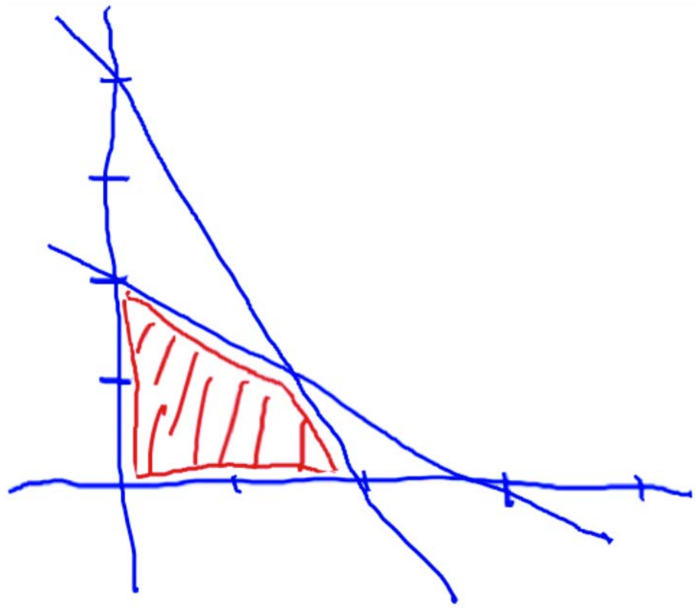

Figure 1: Answer for question 4(a)

Answer: $(2,0)$, $(1,1)$, or $(0,2)$, all with value 2.

(d) [**6 pts**] Write down the dual of this LP (without the integrality constraints mentioned in part (c)).

Answer: We will have one variable for each constraint in the original LP. Since the constraints are inequalities, the variables will be constrained to be positive; call them $a, b \geq 0$. Combining the two constraints, with weights a and b, yields

$$a(2x + y - 4) + b(2x + 3y - 6) \leq 0$$

or

$$(2a + 2b)x + (a + 3b)y \leq 4a + 6b$$

From this combined constraint, we want an upper bound on the objective $x + y$. If we had $a + 3b = 1$ and $2a + 2b = 1$, the combined constraint would become

$$x + y \leq 4a + 6b$$

which is a bound of the type desired. In fact, since y is positive, we can get the same bound as long as $a + 3b \geq 1$, since that will make $y \leq (a + 3b)y$. Similarly, since x is positive, we can get the same bound as long as $2a + 2b \geq 1$. So, the dual is:

$$\begin{aligned} \text{minimize} \quad & 4a + 6b \\ \text{subject to} \quad & a + 3b \geq 1 \\ & 2a + 2b \geq 1 \\ & a, b \geq 0 \end{aligned}$$

since we want to make the bound on the objective as tight as possible.

(e) [**2 pts**] What is the integrality gap of the integer program from (c), as compared to its linear program relaxation?

Answer: either $5/2 - 2 = 1/2$ or $(5/2)/2 = 5/4$ or $2/(5/2) = 4/5$.

5. **Complexity [10 pts]**

In this problem we will consider the integer and linear programming decision problems: given an integer or linear program determine whether or not its optimal value is $>= w$ for an integer w. Now suppose someone tells you that they have an algorithm which can take these inputs:

- an integer program A and an integer v with total description length (size) n

and in $O(n^2 log(n))$ time produce these outputs

- a linear program B and an integer v' with total description length (size) $n^2 + 3n/2$

The proposed algorithm guarantees that the optimal solution to B will have value at least v' if and only if the optimal solution to A has value at least v. If this algorithm and claims of its running time were correct which of the following would be true? (You should use both the putative existence of the above algorithm and your general knowledge about linear and integer programming when answering these questions.)

(a) [**2 pts**] (T F) This algorithm is a reduction from the integer programming decision problem to the linear programming decision problem.

Answer: True. The algorithm specified as above constitutes a reduction between the two problems.

(b) [**2 pts**] (T F) This algorithm is a reduction from the linear programming decision problem to the integer programming decision problem.

Answer: False. A reduction from the LP decision problem to the IP decision problem would take an LP and a target value as input and produce an IP and a target value.

(c) [**2 pts**] (T F) There is an efficient (polynomial time) algorithm for testing whether a linear program has an optimal value of at least w for an integer w.

Answer: True. Linear programming is in P, so we don't even need to use the algorithm mentioned above.

(d) [**2 pts**] (T F) There is an efficient (polynomial time) algorithm for testing whether an integer program has an optimal value of at least w for an integer w.

Answer: True. We can use the algorithm from above to turn our integer program into a linear program, and then solve the LP decision problem.

(e) [**2 pts**] (T F) We can prove $P = NP$.

Answer: True. We've just given an algorithm to solve an NP-complete problem (the IP decision problem) in poly time. We could also solve SAT in poly time, if desired, by turning a SAT instance into an IP decision problem instance (we went over this reduction in the first review session) and from there into an LP decision problem instance.

6. **Bayesian Networks [10 pts]**

This question focuses on the independency assumptions that are encoded by the network in Figure 2. Define $I(X, Y|Z)$ to mean that the variables X and Y are independent given the variables in Z. Z can be an empty set, a set containing one variable or a set containing multiple variables. For example, the statement $I(D, E|C)$ is true for the network in Figure 2.

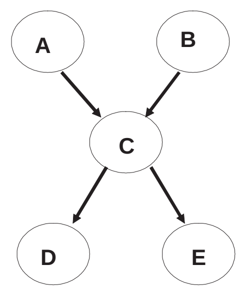

Figure 2: Bayesian Network for question 6

Specify TRUE or FALSE for each of the statements below (no need to explain, each is worth 2 points).

(a) [**2 pts**] $I(D, E|C, A)$ TRUE FALSE

Answer: TRUE, A variable is independent of all variables above it in the network when conditioned on its parents. Since C is the parent of D and E the addition of A does not change the fact that they are independent given C as mentioned in the question.

(b) [**2 pts**] $I(D, E|\{\})$ TRUE FALSE

Answer: FALSE. Consider for example the case where $P(D = 1|C = 1) = P(E = 1|C = 1) = 1$ and $P(D = 0|C = 0) = P(E = 0|C = 0) = 1$. Also, assume that $1 > P(C = 1) > 0$. Then $P(D = 1) > 0$ and also $P(E = 0) > 0$ but $P(D = 1, E = 0) = 0$.

(c) [**2 pts**] $I(A, B|\{\})$ TRUE FALSE

Answer: TRUE. Both A and B have no incoming connections and so the lack of edge between them indicates that they are independent when no other information is provided about their descendants.

(d) [**2 pts**] $I(A, B|C)$ TRUE FALSE

Answer: FALSE. Again, an example would help. Consider the case of $P(A = 1) = P(B = 1) = 1/2$ and $P(C = 1) = 1$ if $A = 1$ and $B = 1$ and $P(C = 1) = 0$ if either A or B or both are 0. Then, $P(A = 1, B = 1|C = 0) \neq P(A = 1|C = 0)P(B = 1|C = 0)$.

(e) [**2 pts**] $I(A, E|C)$ TRUE FALSE

Answer: TRUE. As we mentioned in the solution to A above, E is independent of A given C.

7. **HMM [16 pts]**

Assume a Hidden Markov Model (HMM) with three states. Each state outputs the result of a coin flip (H for heads or T for tails). The output distribution is deterministic, that is, state A either outputs H or T but it cannot output both. The same applies to states B and C. The HMMs always starts in state A. For each of the questions below circle the correct sign that should replace '?'.

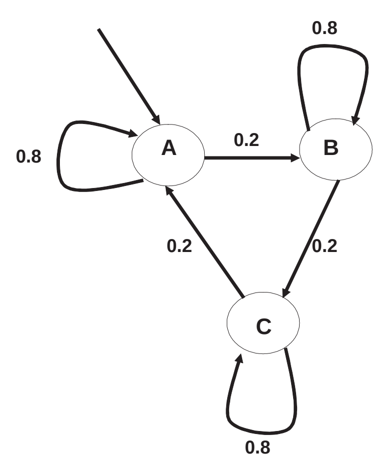

Figure 3: HMM for question 7(a)

(a) The following questions refer to the HMM in Figure 3. Assume we observe the following output sequence: $HTTTH$. We would like to use this to infer the likelihood of states at a specific time point. Use 'cannot be determined' in cases where there is not enough data to determine the relationship.

a.1 [**4 pts**] $P(q_4 = B)$? $P(q_4 = C)$ $>$ $<$ $=$ cannot be determined

Answer: $>$.
We start at state A and transition to B after the first emission. The only uncertainty is in the last three states and there are only three possible models for these states:
BBC, BCA, CCA
In the first case, $q_4 = B$ whereas in the other two, $q_4 = C$. To determine which one is more probable we only need to compute the probability of each and sum up the two cases where $q_4 = C$. $P(B, B, C) = 0.8 \times 0.8 \times 0.2$, $P(B, C, A) = 0.8 \times 0.2 \times 0.2$, $P(C, C, A) = 0.2 \times 0.8 \times 0.2$
So $P(q_4 = B) = 0.8 \times 0.8 \times 0.2 > P(q_4 = C) = 2 \times 0.8 \times 0.2 \times 0.2$

a.2 [**4 pts**] $P(q_2 = B)$? $P(q_3 = B)$ $>$ $<$ $=$ cannot be determined

Answer: $>$.
q_2 must be B since we start at A and output H and so the next output (T) must come from a different state. In contrast, q_3 can be C if C also outputs T. So $P(q_2 = B) = 1$ whereas $P(q_3 = b) < 1$.

(b) The following questions refer to the HMM in Figure 4. For this HMM we know the deterministic output probabilities (specified in each state outputs) but not the transition probabilities. Assume we observe the following output sequence: $HHTTTTHH$. We would like to use this to infer the transition probabilities. The questions below refer to the MLE estimates of these probabilities.

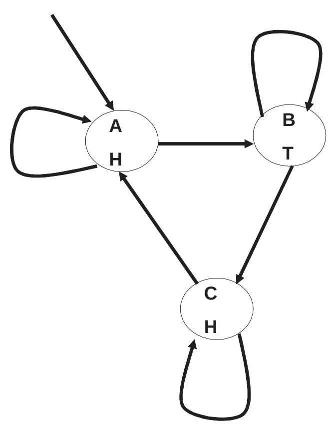

Figure 4: HMM for question 7(b)

b.1 [**4 pts**] $P(B|A)$? $P(C|B)$ $>$ $<$ $=$ cannot be determined
Answer: $>$.

b.2 [**4 pts**] $P(C|B)$? $P(B|B)$ $>$ $<$ $=$ cannot be determined
Answer: $<$.

For this part we know that we start in A and spend the first two states there. We then transition to B prior to the third emission ($q_3 = B$) and stay in B until q_6 and thus $q_7 = C$. q_8 may either be C or A, fortunately, it does not influence the computations we need to do for this question. If we know the sates then computing the MLE of the transition probabilities reduces to simple counting. Thus $P(A|A) = P(B|A) = 1/2$, $P(B|B) = 3/4$, $P(C|B) = 1/4$.

8. **Decision Trees, Neural Nets and MDP [12 pts]**

   Answer true or false and provide a brief explanation of your answer. (**3 pts** each, **2 pts** for T/F; **1 pt** for discussion)

   (a) [**3 pts**] For any set of attributes, and any training set generated by a deterministic function f of those attributes, there is a decision tree which is consistent with that training set.

   Answer: True.

   At worst, there is 1 leaf for each element of the training set.

   (b) [**3 pts**] It is important to use accuracy on held out data rather than accuracy on training data in order to decide when to stop pruning a decision tree.

   Answer: True. Pruning using the training data would typically not have any effect because the recursive algorithm generates a tree that classifies the training data well. That is, deciding to stop when training accuracy decreased would cause an immediate stop. When pruning yields a decline in held-out data accuracy, on the other hand, the overfitting subtrees have been pruned away but the classifier is still expressive enough to model trends in the data.

   (c) [**3 pts**] Multi-layer feed forward neural networks can ONLY solve linearly separable problems.

   Answer: False. We know that XOR isn't linearly separable, but it can be solved by multi-layer feed forward neural networks.

   (d) [**3 pts**] Q-learning is only guaranteed to converge to the optimal q-values if the underlying MDP has deterministic state transitions.

   Answer: False. Q-learning is guaranteed to converge to optimal Q if the world is really an MDP.