CSE 352	**Artificial Intelligence**	**Handout A6**
Spring 2003	**Annie Liu**	**Apr. 16, 2003**
Stony Brook	**Assignment 6**	**No due date**

This assignment gives exercise problems for Exam 2. Solutions are provided and the assignment will not be graded. The exam will be open your own handwritten notes, and open textbook, but nothing else.

Problem 1. Planning for blocks world

Consider the blocks world as discussed in class. What is the STRIPS operator for "PutOn(x,y)"? What is a sequence of steps in solving the "Sussman anomaly" problem using partial-order planning?

See lecture slides.

Problem 2. Planning for register swapping

Consider the problem of swapping the contents of two registers, A and B. Suppose that there is available the single operator $Assign(x, v, lv, ov)$, which assigns value v, which is location lv, to location x, which previously contained value ov.

$Assign(x, v, lv, ov)$:
Precondition: $contains(lv, v)$, $contains(x, ov)$
Effect: $contains(x, v)$, $\neg contains(x, ov)$

Assume that there is an additional register C available. Solve this problem using partial-order planning, according to the following steps:

(a) For the initial plan.
(b) Pick a first open precondition to satisfy. What is an obvious action to achieve it? Show the result of adding this action to the plan.
(c) Pick a next open precondition to satisfy. What is a possible action and how to order it with respect to the action in (b)? Show the result of adding this action to the plan.
(d) Pick a next open precondition to satisfy. What is an obvious action to achieve it? Show the result of adding this action to the plan.

If you slow down your thinking, the following most naturally captures how you could have solved the problem the first time you had to do it yourself.

(a)

```
                    [Start]
contains(A,a)  contains(B,b)  contains(C,c)



       contains(A,b)  contains(B,a)
                    [Finish]
```

(b) Pick precondition $contains(A, b)$, say. An obvious action to add is $Assign(A, b, B, a)$, and call it S_1.

```
                    [Start]
contains(A,a)  contains(B,b)  contains(C,c)
         \        /
          \      /                                 assume all lines
        [Assign(A,b,B,a)]                          are with arrows
            |    \                                 pointing down.
            |    not contains(A,a)
            |
       contains(A,b)  contains(B,a)
                   [Finish]
```

(c) Pick precondition *contains*(B, a). An action to add could be $Assign(B, a, A, b)$, but it can not be before S_1, since it would destroy the precondition *contains*(B, b) for S_1; by putting it after S_1, it has to take value from a place other than A and B, since neither A or B contains a. So we use C and decide on the action $Assign(B, a, C, b)$.

```
                    [Start]
contains(A,a)    contains(B,b)   contains(C,c)
         \       /        \
          \     /          \                          assume all lines
       [Assign(A,b,B,a)]... \                         are with arrows
           |    \          . \                        pointing down.
           | ~contains(A,a) . \  contains(C,a)
           |                  . \/
           |          [Assign(B,a,C,b)]
           |                   |
       contains(A,b)     contains(B,a)
                   [Finish]
```

(d) Pick precondition *contains*(C, a). The action is $Assign(C, a, A, c)$, and it has to be before S_1.

```
                    [Start]
contains(A,a)    contains(B,b)   contains(C,c)
      \ \__________ /__\____________  |
       \           /    \           \ |
        \         /      \... [Assign(C,a,A,c)]
         \       /       .\           |
          \     /       .  \          |               assume all lines
       [Assign(A,b,B,a)]... \         |               are with arrows
           |    \          . \        |               pointing down.
           | ~contains(A,a) . \  contains(C,a)
           |                  . \/
           |          [Assign(B,a,C,b)]
           |                   |
       contains(A,b)     contains(B,a)
                   [Finish]
```

Problem 3. Plan execution monitoring

Consider the problem of getting milk, banana, and drill discussed in class. What are the preconditions to check at each step in the execution of the plan? What to do on failure of any such condition?

See lecture slides.

Problem 4. Basic concepts for probability and probabilistic inference

(a) What is conditional probability?
(b) What is chain rule?
(c) What is independence?
(d) What is conditional independence?
(e) What is Bayes' rule? (It is also called Bayes' theorem.)
(f) Why are Bayesian networks compact specifications of full joint distributions? (They are also called belief networks.)

(g) What are the two kinds of variables used in Markov processes? What is Markov assumption? What is first-order Markov process? What is sensor Markov assumption? What are the two kinds of probabilities given in hidden Markov models? How to use them in general?

See lecture slides for (a) to (e).

(f) Suppose there are n random variables and they are Boolean valued, then a Bayesian network where each variable has no more than k parents requires no more than $O(n \cdot 2^k)$ numbers, but full joint distribution requires $O(2^n)$ numbers.

(g) Two kinds of variables are state variables and evidence variables; state variables are unobservable, and evidence variables are observable.

Two kinds of probabilities given in hidden Markov models: those for transition of state from each time step to the next, $P(X_t|X_{t-1})$, and those for sensing the evidence given the state at each time step, $P(E_t|X_t)$; they are easily given as matrices, called transition matrix and sensor matrix, respectively.

As typical probabilistic inference, hidden Markov models are used to infer conditional probabilities of state conditioned on evidence based on conditional probabilities of evidence conditioned on state, i.e., based on $P(E_t|X_t)$; but it is further used to accumulate knowledge over a sequence of evidence and states by combining $P(E_t|X_t)$ with $P(X_t|X_{t-1})$ for a sequence of t's.

See lecture slides for answers to other questions.

Problem 5. Independence

Exercise 13.7 of textbook.

By definition of conditional probability $P(a|b) = \frac{P(a \wedge b)}{P(b)}$, we have $P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$.

If $P(a|b) = p(a)$, then $P(b|a)P(a) = P(a|b)P(b) = P(a)P(b)$ and thus $P(b|a) = P(b)$.

If $P(b|a) = P(b)$, then $P(a \wedge b) = P(b|a)P(a) = P(b)P(a) = P(a)P(b)$, i.e., $P(a \wedge b) = P(a)P(b)$.

If $P(a \wedge b) = P(a)P(b)$, then $P(a|b)P(b) = P(a \wedge b) = P(a)P(b)$ and thus $P(a|b) = P(a)$.

Problem 6. Normalization

Exercise 13.12 of textbook. Note that the letters m and s should be in lower case. That is, this question asks you to compute $P(m|s)$ and $P(\neg m|s)$ using $P(s|\neg m)$ and normalization for the meningitis example on page 480 of textbook.

Say $P(s|\neg m) = 1/30$. (Probably a smaller value would have made better sense, but you won't get points off for such things that are not your expertise and are not common knowledge.) Then

$$\begin{aligned} \langle P(m|s), P(\neg m|s)\rangle &= \alpha\langle P(s|m)P(m), P(s|\neg m)P(\neg m)\rangle \\ &= \alpha\langle 0.5 \times 1/5000, 1/30 \times 49999/50000\rangle = \langle 0.003, 0.997\rangle \end{aligned}$$

Problem 7. Belief network design

Consider the following random variables, pertaining to driving home after a New Year's Eve party in Lake Tahoe:

BrakeFailure — whether you brakes fail
Drunk — whether you are actually over the limit
AccidentSeverity — values *None*, *FenderBender*, *Severe*
IcyWeather — whether the weather is icy
Arrested — whether you get arrested
Injured — whether you are injured
Jailed – whether you go to jail

(a) Pick a reasonable ordering for the variables and use it to construct a network topology. Try to minimize the amount of information required for he conditional probability tables, while respecting the obvious causal influences in the domain.
(b) Label each node with the number of independent probabilities that must be supplied for the associated conditional probability table.
(c) Give a reasonable conditional probability table associated with the *Jailed* node.
(d) Is your network singly-connected?

(a) A good ordering (root causes to final symptoms) might be *BrakeFailure*, *IcyWeather*, *Drunk*, *Accident-Severity Arrested*, *Injured*, *Jailed*. The topology would look like the following:

```
BrakeFailure     IcyWeather         Drunk
    |_____________   |   ____________| | |
                 |   |  |              | |
                 V   V  V              | |
                 Accident              | |
                  | | |______    ______| |
                  | |        |  |        |
                  | |        V  V        |
                  | |      Arrested      |
                  | |_______  |  _______|
                  |         | | |
                  V         V V V
                Injured     Jailed
```

(b) In order of (a), we have 1, 1, 1, 16 ($= 2^3 + 2^3$, two entries of each combinations of predecessor are needed since *AccidentSeverity* is three-valued), 6 ($= 3 \times 2$), 3, 12 ($= 3 \times 2 \times 2$).

(c) The main things are: *Jailed* is only possible if *Arrested* is true. It is more likely if *Drunk*, and if not *Drunk* then unless *AccidentSeverity* is high the probability of *Jailed* is low or zero. It should increase with *AccidentSeverity* for drunks.

(d) No, because of multiple paths from *Drunk* to *Jailed*.

Problem 8. Probabilistic inference

Consider the belief network below with Boolean variables H=`Honest`, S=`Slick`, P=`Popular`, E=`Elected`. Assume all lines are with arrows pointing down.

```
   P(H)   H
   .1     |  \
          |   \     H P(S)
          |     S   T .3
H S P(P)  |   /     F .9
T T .9    |  /
T F .5    P
F T .8     \
F F .2      \    P P(E)
             E   T .6
                 F .1
```

(a) Which, if any, of the following are asserted by the network structure (ignoring the CPTs for now)? Note that any subset of these maybe be correct.
(i) $\mathbf{P}(H, S) = \mathbf{P}(H)\mathbf{P}(S)$
(ii) $\mathbf{P}(E|P, H) = \mathbf{P}(E|P)$
(iii) $\mathbf{P}(E) \neq \mathbf{P}(E|H)$

(b) Calculate the value of $P(h, s, \neg p, \neg e)$.
(c) Calculate the probability that someone is elected given that they are honest.
(d) Suppose we want to add the variable $L =$ *LotsOfCampainFunds* to the network; describe, with justification, all the changes you would make to the network.
(e) If we sample from an empty network using the network, what is the probability that the sample $h, s, \neg p, \neg e$ is generated?
(f) If we want to do sampling given certain evidence, how can results of sampling from an empty network be used?
(g) What kind of conditional probabilities can be used for sampling using Markov Chain Monte Carlo?

(a) (ii) is asserted, by the local semantics of belief networks: a node is conditionally independent of its nondescendants given its parents. (i) is not asserted since H and S are linked by an arc. (iii) is not asserted by the structure alone, because arcs do not deny independence. (The CPTs can deny it, however.)

(b) $P(h, s, \neg p, \neg e) = P(h)P(s|h)P(\neg p|h, s)P(\neg e|\neg p) = 0.1 \times 0.3 \times 0.9 \times 0.1 = 0.00027$.

(c) Probably the simplest way to do this is to construct the part of the full joint for H true (8 rows) and then add up. The following uses the enumeration algorithm:

$$\begin{aligned} P(E|h) &= \alpha P(h)\Sigma s P(s|h)\Sigma p P(p|h, s)P(E|p) \\ &= \alpha \times 0.1 \times [\ 0.3 \times (0.9 \times \langle 0.6, 0.4\rangle)+ \\ &\qquad\qquad\quad 0.7 \times (0.1 \times \langle 0.1, 0.9\rangle)+ \\ &\qquad\qquad\quad 0.5 \times (0.9 \times \langle 0.6, 0.4\rangle)+ \\ &\qquad\qquad\quad 0.5 \times (0.1 \times \langle 0.1, 0.9\rangle)\ ] \\ &= \langle 0.41, 0.59\rangle \end{aligned}$$

(d) Let's assume honesty doesn't influence fund-raising ability, but slickness does. Funds support advertising which increases popularity, but do not directly affect electability otherwise. So L should be a child of S and parent of P. We would need a CPT for $P(L|S)$ and an augmented CPT $P(P|H, S, L)$. Any CPTs reflecting the above-mentioned influences will do.

(e) Same as (b).

(f) Two ways. One is to throw out samples that do not match evidence. The other is to weight each sample by the likelihood it accords to the evidence.

(g) Conditional probabilities conditioned on Markov Blanket.

Problem 9. Value of information using Bayes' rule

Exercise 16.11 b-e of textbook, on used-car buyer testing car.

This question is a simple exercise in sequential decision making. It also emphasizes the point that the value of information is computed by examining the conditional plan formed by determining the best action for each possible outcome of the test. Incidentally, the question assumes that utility and monetary value coincide, and ignores the transaction costs involved in buying and selling.

(b) The expected net gain in dollars is

$$P(q^+)(2000 - 1500) + P(q^-)(2000 - 2200) = 0.7 \times 500 + 0.3 \times -200 = 290.$$

(c) The question could probably have been stated better: Bayes' theorem is used to compute $P(q^+|Pass)$, $P(q^-|Pass)$, $P(q^+|\neg Pass)$, and $P(q^-|\neg Pass)$, whereas using conditional probabilities alone is sufficient to compute $P(Pass)$ and $P(\neg Pass)$. Alternatively, normalization can be used to avoid computing $P(Pass)$ and $P(\neg Pass)$. Below we use the first method, not using normalization:

$$P(Pass) = P(Pass|q^+)P(q^+) + P(Pass|q^-)P(q^-) = 0.8 \times 0.7 + 0.35 \times 0.3 = 0.665.$$

Using Bayes' theorem:

$$\begin{array}{lcl} P(q^+|Pass) & = & \frac{P(Pass|q^+)P(q^+)}{P(Pass)} = \frac{0.8\times 0.7}{0.665} \approx 0.8421 \\ P(q^-|Pass) & \approx & 1 - 0.8421 = 0.1579 \\ P(q^+|\neg Pass) & = & \frac{P(\neg Pass|q^+)P(q^+)}{P(\neg Pass)} = \frac{0.2\times 0.7}{1-0.665} \approx 0.4179 \\ P(q^-|\neg Pass) & \approx & 1 - 0.4179 = 0.5821 \end{array}$$

(d) If the car passes the test, the expected value of buying is

$$P(q^+|Pass)(2000-1500) + P(q^-|Pass)(2000-2200) = 0.8421 \times 500 + 0.1579 \times -200 = 378.92.$$

Thus, buying is the best decision given a pass. If the car fails the test, the expected value of buying is

$$P(q^+|\neg Pass)(2000-1500) + P(q^-|\neg Pass)(2000-2200) = 0.41279 \times 500 + 0.5821 \times -200 = 92.53.$$

Buying is again the best choice.

(e) Since the action is the same for both outcomes of the test, the test itself is worthless (if it is the only possible test) and the optimal plan is simply to buy the car without the test. (This is a trivial conditional plan.) For the test to be worthwhile, it would need to be more discriminating in order to reduce the probability $P(q^+|\neg Pass)$. The test would also be worthwhile if the market value of the car were less, or if the cost of repairs were more.

Problem 10. Speech recognition

(a) How does Bayes' rule help decompose the speech recognition problem into acoustic modeling and language modeling?

(b) Give three components where hidden Markov models are used in speech recognition. How are they combined?

(a) $P(Words|signal) = \alpha P(signal|Words)P(Words)$

(b) Recognizing a phone from phone states (onset, mid, end); recognizing a word from a sequence of phones; recognizing a sentence from a sequence of words.

States in a combined hidden Markov model are labeled with the word we are in + the phone in that word + the phone state in that phone.

Problem 11. Decision tree

Exercise 18.3 of textbook.

This is a deceptively simple question, designed to point out the plethora of "exceptions" in real-world situations and the way in which decision trees capture a hierarchy of exceptions. One possible tree is shown below. One can, of course, imagine many more exceptions.

```
                FrontOfQueue?
                  /      \
               no/        \yes
                /           \
     CarAheadMoving?    IntersectionBlocked?
         /  \                   /    \
      no/    \yes            no/      \yes
       /      \                /        \
     NO       YES     CrossTraffic?   NO
                          /  \
```

```
                    no/    \yes
                     /      \
              Pedestriants?  NO
                    /  \
               no/    \yes
                /      \
             Turning?    NO
              /  |  \
         left/   |no \right
           /     |    \
OncomingTraffic?  YES  Cyclist?
     /  \                / \
  no/    \yes        no/   \yes
   /      \           /     \
  YES     NO         YES    NO
```

Problem 12. Decision tree learning

(a) Exercise 18.4 of textbook.
(b) Consider learning a classification with 4 possible outcomes. If we have 12 examples, and there are 3 in each category, how many bits are needed for this information?

(a) In standard decision trees, attribute tests divide examples according to the attribute value. Therefore any example reaching the second test already has a know value for the attribute and the second test is redundant.

In some decision tree systems, however, all tests are Boolean even if the attributes are multivalued or continuous. In this case, additional tests of the attribute can be used to check different values or subdivide the range further (e.g., first check if $X > 0$, and then if it is, check if $X > 10$)

(b) The probability for each of the categories is 1/4. So

$$H(\langle 1/4, 1/4, 1/4/, 1/4\rangle) = \Sigma_{i=1}^{4} - 1/4 \log_2(1/4) = 4 \times (-1/4 \times (-2)) = 2.$$

Problem 13. Neural network construction

Exercise 20.11 of textbook.

XOR (in fact any Boolean function) is easiest to construct using step-function units. Because XOR is not linearly separable, we will need a hidden layer. It turns out that just one hidden node suffices. To design the network, we can think of the XOR function as OR with the AND case (both inputs on) ruled out. Thus the hidden layer computes AND, while the output layer computes OR but weights the output of the hidden node negatively. The network below achieves this, where t is the threshold.

```
    [] _________________
     \                  \ w=0.3                         
w=0.3 \                  \                           assume all lines
       \          w=-0.6  \                          are with arrows
        (t=0.5) ---------- (t=0.2) ----              pointing right
       /                  /
w=0.3 /                  /
     / _________________/ w=0.3
    []
```

Problem 14. Linear activation functions

Exercise 20.17 of textbook.

This exercise reinforces one's understanding of neural networks as mathematical functions that can be analyzed at a level of abstraction above their implementation as a network of computing elements. For simplicity, we assume that the activation function is the same linear function at each node: $g(x) = cx + d$. The argument is the same (only messier) if we allow different c_i and d_i for each node.

(a) The outputs of the hidden layer are

$$H_j = g(\Sigma_k W_{k,j} I_k) = c\Sigma_k W_{k,j} I_k + d$$

The final outputs are

$$O_i = g(\Sigma_j W_{j,i} H_j) = c(\Sigma_j W_{j,i}(c\Sigma_k W_{k,j} I_k + d)) + d$$

Now we just have to see that this is linear in the inputs:

$$O_i = c^2 \Sigma_k I_k \Sigma_j W_{k,j} W_{j,i} + d(1 + c\Sigma W_{j,i})$$

Thus we can compute the same function as the two-layer network using just a one layer perceptron that has weights $W_{k,i} = \Sigma_j W_{k,j} W_{j,i}$ and an activation function $g(x) = c^2 x + d(1 + c\Sigma W_{j,i})$.

(b) The above reduction can be used straightforwardly to reduce an n-layer network to an $(n-1)$-layer network. By induction, the n-layer network can be reduced to a single-layer network. Thus, linear activation function restrict neural networks to represent only linear functions.

Problem 15. Minimizing error

Exercise 20.19 of textbook.

Intuitively, the data suggest that a probabilistic prediction $P(Output = 1) = 0.8$ is appropriate. The network will adjust its weights to minimize the error function. The error is

$$E = \frac{1}{2}\Sigma_i (T_i - O_i)^2 = \frac{1}{2}[80(1 - O_1)^2 + 20(0 - O_1)^2] = 50O_1^2 - 80O_1 + 50$$

We use the hint (for minimizing the error), computing the derivative with respect to O_1:

$$100O_1 - 80$$

and setting the derivative to zero, we find that $O_1 = 0.8$.