15-381 Artificial Intelligence: Representation and Problem Solving
Homework 2

Out: 09/18/08
Due: 10/02/08

Instructions

This assignment is due on Thursday, October 2, 2008. The written portion and societal question must be turned in at the beginning of class at noon on October 2. Type or write legibly; illegible submissions will not receive credit. Write your name and Andrew ID clearly at the top of the assignment. The programming portion of this assignment must be electronically submitted by noon on October 2. The submission instructions are described in the Programming section. You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas in the class in order to help each other answer homework questions. You are also welcome to give each other examples that are not in the assignment in order to demonstrate how to solve problems. But we require you to:

- not explicitly tell each other answers;
- not copy answers;
- not allow your answers to be copied.

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we ask that you specifically record on the assignment the names of the people you were in discussion with (or "none" if you did not talk to anyone else). This will help resolve the situation where a mistake in general discussion led to a replicated error among multiple solutions. This policy has been established in order to be fair to everyone in the class. We have a grading policy of watching for cheating and we will follow up if it is detected. Instructions for turning in programming component: The last question involves programming which should be done in MATLAB. Do *not* attach your code to the writeup. Make a tarball of your code and output, name it <userid>.tgz where your userid is your Andrew id, and copy it to /afs/andrew/course/15/381-f08/submit/hw2/<userid>/ . Refer to the web page for policies regarding collaboration, due dates, and extensions.

1 [10 pts] Probability miscellany

Calvin wants to choose between his two pet activities: playing with his pet tiger Hobbes in the garden, and tormenting his mom in the kitchen. He wants to choose uniformly at random between the two activities and decides to toss a coin to decide. Since he is a kid and is broke, he gets a coin from his mom to do this. However, he isn't sure if the coin is unbiased. Can you help him "simulate" a fair coin toss by using this coin? Explain how you would do this, and show that your procedure is indeed unbiased. While you don't know the bias of the coin, you may assume that the bias remains the same at all time. Some clarifications on terminology: a coin is considered "fair" or unbiased if p, the probability of heads equals 0.5. The question asks you to describe a procedure `toss_unbiased_coin()` that returns "Heads" or "Tails" with probability 0.5. The only source of randomness that the procedure has access to, is a procedure `toss_biased_coin()`

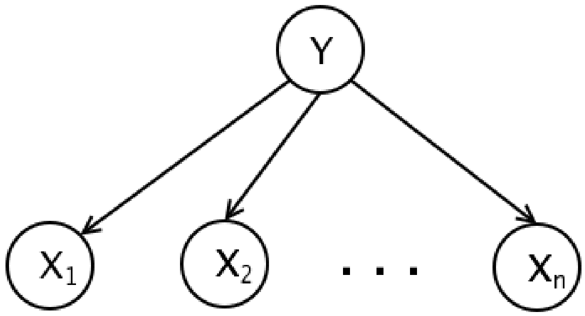

Figure 1: Bayesian network for Problem 3

that returns a "Heads" with some unknown probability p. Note: You aren't allowed to use any source of randomness other than the coin itself, so "call rand in matlab" is not a valid answer. :)

2 [10 pts] Representation

Suppose we have a boolean variable X. To complete describe the distribution $P(X)$, we need to specify one value: P(X=0)(since P(X=1) is simply 1-P(X=0)) Thus, we say, this distribution can be characterized with one parameter. Now, consider N+1 binary random variables $X_1 \dots X_N, Y$ that factorize according to Fig. 1

1. Suppose you wish to store the joint probability distribution of these N+1 variables as a single table. How many parameters will you need to represent this table?
2. Now, suppose you were to utilize the fact the joint distribution factorizes accoording to the Bayes Network. How many parameters will you need to completely describe the distribution if you use the Bayesian Network representation? In other words, how many parameters will you need to fully specify the values of all the conditional probability tables in this Bayesian Network.

3 [15 pts] Number of BNs

What is the maximum number of edges in a Bayesian network (BN) with n nodes? Prove that a valid BN containing this number of edges can be constructed (remember that the structure of a BN has to be a Directed *Acyclic* Graph)

4 [15 pts] Conditional Independencies in Bayes Nets

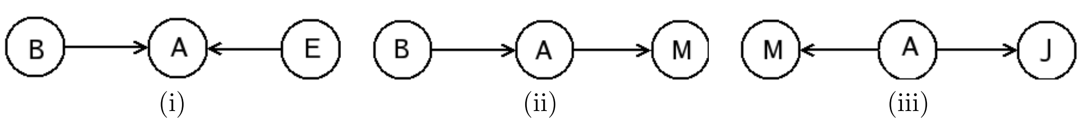

Figure 2: Parts of the alarm network

The Bayesian networks in Fig. 2 are all part of the alarm network introduced in class and in Russell and Norvig. We use the notation $X \perp Y$ to denote the variable X being independent of Y , and $X \perp Y | Z$ to denote X being independent of Y given Z.

1. For each of these three networks, write the factored joint distribution implied, in the form of $p(X, Y) = p(X)p(Y|X)$

2. Using the joint distribution you wrote down for Fig. 2(i), write down a formula for $P(B, E)$.
3. Now prove that $B \perp E$.
4. Similarly, prove that $B \perp M | A$ in the Bayesian network of Fig. 2(ii), and $M \perp J | A$ in the Bayesian network of Fig. 2(iii).

5 [15 pts] Elimination

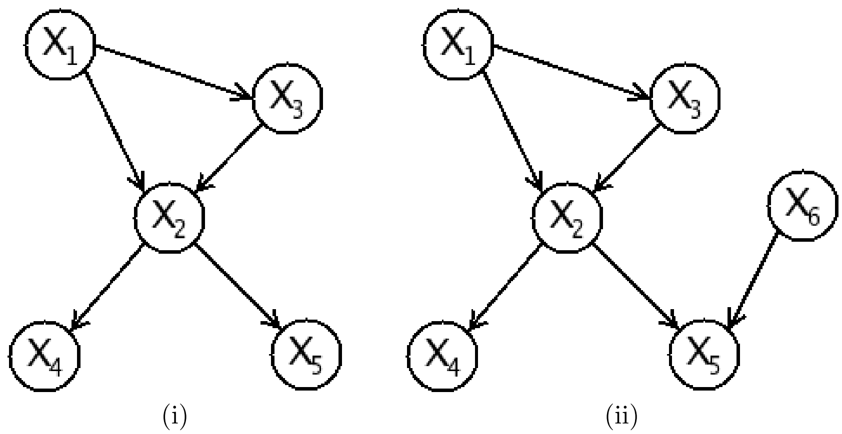

Figure 3: Bayesian networks for Problem 6

Consider the Bayesian Network given in Fig. 3(i). Assume that each of the variables are boolean valued. For each of the following, state the total number of operations(multiplication and addition) the variable elimination algorithm will take to compute the answer. For example, if A and B are binary variables, $\sum_a P(B)P(a|B)$ will take two multiplications($P(B) * P(A|B)$ and $P(B) * P(\neg A|B)$) and one addition(adding those two terms). Assume that the algorithm avoids unnecessary computation, so any summations that are irrelevant to the query will be avoided(see the textbook for an example of an *irrelevant* summation).

1. $P(X_3|X_4)$ i.e., the probability that X_3 is true given that X_4 is true. Assume that the variables are eliminated in the order: X_5, X_1, X_2.
2. $P(X_3|X_4)$ but this time, with the elimination ordering X_2, X_1, X_5
3. Now suppose that you had to answer the last two questions, this time with the Bayes Net given in Fig. 3(ii). Would your answers change?(You don't have to state the number of operations; just if they will be different from previously with a brief explanation)

6 [35 pts] Inference by enumeration

1. Implement exact inference by enumeration. Write a function that calculates the conditional probability distribution of one query variable given a set of evidence variables in a Bayesian network. See Algorithm 1 for the pseudo-code and Russell and Norvig chapter14.4(pp. 504) for reference. For simplicity, all variables are binary, so all distributions mentioned in the pseudocode and in Russell and Norvig, including the returned value of the function, can be represented by the probability of the variable being true. Two sample Bayesian network will be given to you in the support archive, named alarm and pedigree, details below. The following Matlab functions are provided, to access the Bayesian network data structure:

```
Enumeration_Ask(X,e,bn) begin
    Data: X, the query variable; e, the observed values for variables E; bn, a Bayes net with variables
          X ⋃ E ⋃ Y
    Result: distribution over X
    Q(X) ← a distribution over X, initially empty;
    foreach value x_i of X do
        extend e with value x_i for X;
        Q(X_i) ← Enumerate_All(Vars[bn], e);
    end
    return Normalize(Q(X))
end
Enumerate_All(vars, e) begin
    if Empty?(vars) then
        return 1.0
    end
    Y ← First(vars);
    if Y has value y in e then
        return P(y|Parents(Y)) × Enumerate_All(REST(vars, e))
    else
        return Σ_y P(y|Parents(Y)) × Enumerate_All(Rest(vars), e_y) where e_y is e extended with Y = y
    end
end
```

Algorithm 1: The enumeration algorithm

- *create_alarm_bn*: creates the alarm Bayesian network.
- *create_pedigree_bn*: creates the pedigree Bayesian network.
- *bn_vars*: returns the variables in a Bayesian network, partially ordered from parents to children.
- *bn_parents*: returns the parents of a variable.
- *bn_cpt*: returns the conditional distribution of a variable, given the specified values of its parents. This corresponds to one row of the conditional probability table(CPT).

See README and type help func name in Matlab (or read corresponding scripts) for documentations and examples. Note that the recursive enumeration must be performed from parents to children, i.e. the list of variables must be partially ordered such that parents are always before their children. The function bn vars will provide the ordered variables list(actually the variable index itself in the Bayesian networks given below are already ordered this way, so the ordered list is just 1,2,. . .,N). Please write your code by modifying the provided file enumeration ask.m, which contains suggested API with documentation. Matlab is STRONGLY recommended, particularly because writing necessary support code in other languages is time consuming.

2. Run your exact inference implementation on the following two Bayesian network inference problems. It should be straightforward for you to convert the question below into a function call to your code (by using appropriate variable indexes labeled on the graph below); type "help enumeration_ask" for an example. Both questions ask the conditional probability of one query variable being true given a set of evidence variables. Report run time and results.

    (a) The first Bayesian network, created by create_alarm_bn, is the alarm network in Russell and Norvig. As shown in Fig. 4, variable are indexed from 1 to 5 (denoted as X1 to X5), and CPT are the same as in Russell and Norvig figure 14.2. Evidence variables are shaded, while query variables are shaded and circled by a thick line. Calculate the conditional probability p(X1|X4,¬X5). Hint: verify with some queries that you know the answer first, e.g. p(X1|X4,X5) should be about 0.284.

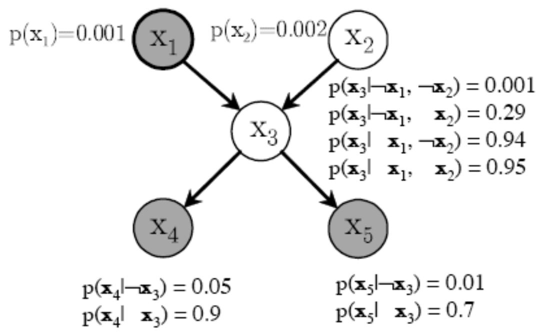

Figure 4: Bayesian network for Problem 3(a)

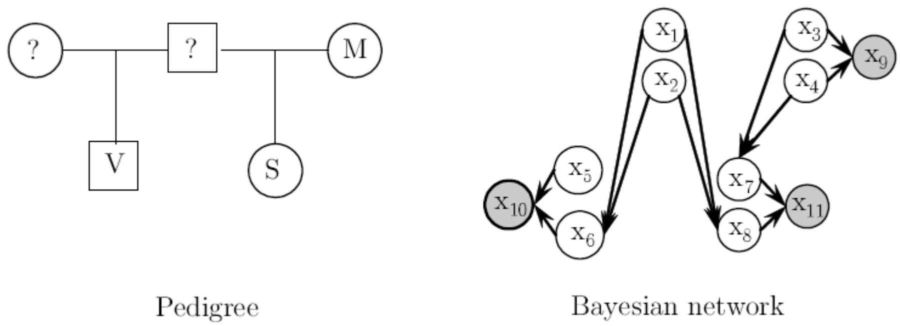

Figure 5: Bayesian network for Problem 3(b)

(b) The second Bayesian network, created by create_pedigree_bn, is about genetic inference. Consider a victim V in a plane crash, whose only family members are his half-sister S and the sister's mother M (not V's mother). Their pedigree is shown below. You need to determine whether certain remains belong to V based on genetic fingerprints of S and M. This can be solved by a Bayesian network shown in Fig. 5, indexed from 1 to 11. Evidence and query variables are shaded, while normal circles are hidden variables. You do not need to worry about the CPT if you are using Matlab; otherwise the CPT is explained in the documentation of create_pedigree_bn.m The variables (X1,X2,...,X8) correspond to unobserved genetic information in the so-called Mendelian inheritance: humans have two copies of each chromosome, one from the father and one from the mother. During reproduction, one copy (chosen randomly) will be passed to the next generation. Assume you cannot determine which copy is from which parent, but can only obtain partial information in the observed variables (X9,X10,X11). However, you do not have to understand Mendelian inheritance to solve this problem. Now using the structure and CPT provided in the archive, calculate the conditional probability p(X10| $\neg$ X9,X11).