15381 Artificial Intelligence: Assignment 2

Due October 5 th, beginning of class

September 22th, 2010

Instructions: There are 4 questions on this assignment. Please send us emails at *15381-tas@lists.andrew.cmu.edu* when you have questions. Refer to the web page for policies regarding collaboration, due dates, and extensions. The TA names listed next to each problem show which TA wrote and is responsible for grading which problem. However, please feel free to speak to any TA or instructor regarding any problem.
Late Policy: Each student is allowed up to 2 late days for the whole semester, to be used with advance permission only. That means you must contact an instructor and request permission to use a late day for an assignment. Any late assignment without an instructor-permitted late day will not be graded.
Collaboration Policy: You can discuss questions with classmates, but the write-up should be of your own. If you discuss the problems with collaborators, you must list their names on your write-up.
Outside Sources: We expect you to solve these problems with the help of the textbook and the lecture slides (and the TAs and instructors) rather than searching for previous solutions and copying them. If you use any source other than the textbook or lecture slides, you must cite the source.
Code submission: The last problem involves coding. Do not attach your code to the writeup. Instead, add your code to the Digital Drop Box of the course Blackboard.

1 A* [25 pts] [Sam]

1. Recall that a heuristic function $h(s)$ is *admissible* if

$$h(s) \leq h^*(s) \text{ for all states } s,$$

where $h^*(s)$ denotes the true minimal cost to the goal from s.

For each of the following questions, please do the following. If the statement is true, prove it. If the statement is false, provide a counterexample by drawing a search graph and labeling the actual costs as well as $h(s)$ for the drawn states as done in the examples in lecture.

(a) [5 Points] A* using any admissible heuristic is optimal.

(b) [5 Points] A* using the heuristic $h(s) = h^*(s) + 1$ is optimal.

(c) [5 Points] A* using the heuristic $h(s) = 2h^*(s)$ is optimal.

(d) [5 Points] A* using any inadmissible heuristic is sound.

(e) [5 Points] A* using any inadmissible heuristic is complete.

2 STRIPS[10 pts] [Sam]

1. Congratulations, your class project has just been accepted at a prestigious conference to be held in Tokyo this summer! As a conscientious student of 15-381, you of course decide to model your problem of attending the conference as a planning problem. The Conference Attendance planning domain is defined as follows:

   - A student starts at Pittsburgh.
   - The student must travel to Tokyo, attend the conference, and return to Pittsburgh.

   Write the planning domain above using STRIPS. Recall that this involves specifying a set of state predicates and actions (an example is given in the lecture notes). Be sure to describe the pre-conditions and post-conditions of all of the actions you listed. If your representation contains more than 2–3 distinct operators, you are probably trying to put in more detail than is necessary for this problem. Also specify the initial state and goal predicate.

3 d-separation [21 pts] [Wooyoung]

Which of the following statements are true with respect to the following graphical model in Fig. 1?

1. [3 Points] $P(D, H) = P(D)P(H)$
2. [3 Points] $P(A, I) = P(A)P(I)$
3. [3 Points] $P(A, I|G) = P(A|G)P(I|G)$
4. [3 Points] $P(J, G|F) = P(J|F)P(G|F)$
5. [3 Points] $P(J, M|K, L) = P(J|K, L)P(M|K, L)$

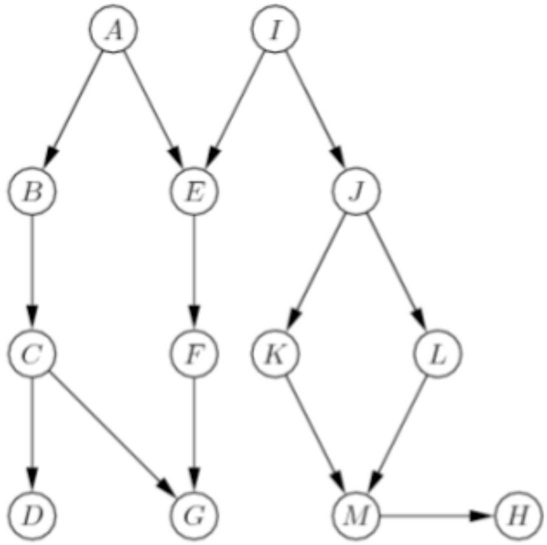


Figure 1: d-separation

6. [3 Points] $P(E, C|A, G) = P(E|A, G)P(C|A, G)$
7. [3 Points] $P(E, C|A) = P(E|A)P(C|A)$

4 Implementing Bayes Nets[24pts] [Wooyoung]

In this problem, we are going to implement the representation of Bayesian network and inference by enumeration. We strongly recommend using MATLAB for this problem. Download `bns_code.zip` from the Blackboard and use the templates provided for your implementation.

4.1 Representation

We represent a Bayesian Network with parents of each node and associated conditional probability tables with struct data type in MATLAB. You can see the examples in `gen_bn_burglary.m` and `gen_bn_student.m` which generate Bayesian Networks corresponding to the graphs in Fig. 3(a) and Fig. 3(b) each. *bn(i).pa* is a vector that represents the parents of i th variable and *bn(i).cpt* is a vector of length 2ˆ(the number of parents). All of our variables are binary in this problem(0=false,1=true). We assume that variables are ordered such that parents precede their children. We represent the evidence variables and their assigned values with vectors. Two vectors `vars` and `vals` are of the same length. The ordering of vars should match that of vals. For example, you want to represent $B = true, E = false$ in the burglary network, you can either set `vars` $= [1, 2]$, `vals` $= [1, 0]$ or `vars` $= [2, 1]$, `vals` $= [0, 1]$.

1. [7 Points] Implement a function that retrieves the distribution of a variable X given the values of its parents. Fill in the blank of `lookup_cpt.m`. As the variables are binary, you can represent their distributions with vectors of two entries. Let the first entry of your vector correspond to $P(X = 0|Pa_X = p_a)$ and the second entry to $P(X = 1|Pa_X = p_a)$. *Hints:You can use the binary values assigned to the parents to decide which entry of the conditional probability table vector you should look up. Make sure that your variables and corresponding values are ordered properly. Some helpful MATLAB functions are dec2bin.m, bin2dec.m, strcat.m,num2str.m.*

4.2 Inference by enumeration

In this section, you are going to implement inference by enumeration. Refer to Section14.4 of the textbook for more details of the algorithm.

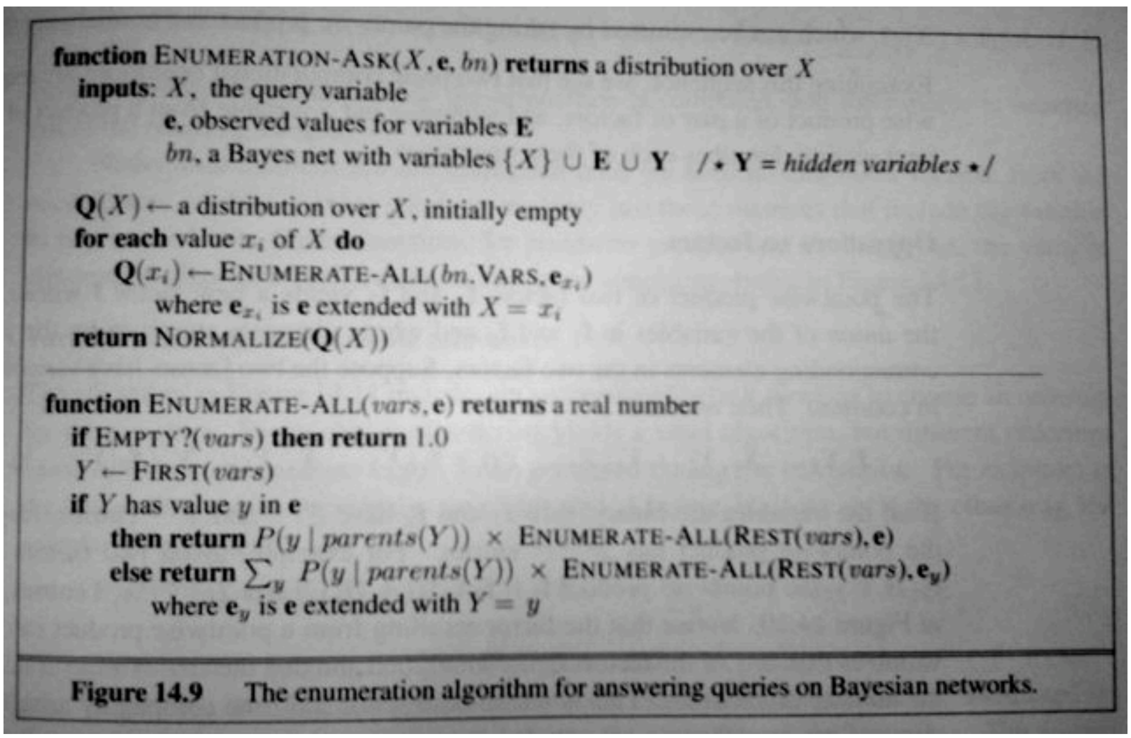
1. [7 Points] Implement `enumeration_all.m`(Psedocode in Algorithm Fig. 2). Fill in the blank of the template `enumeration_all.m`. You can call `lookup_cpt.m` you implemented in the previous problem if you need to.
2. [4 Points] Implement `enumeration_ask.m`. You need to call `enumeration_all.m` you implemented in the previous problem. Fill in the blank of the template `enumeration_ask.m`.
3. Using your implementations, answer the following queries in the burglary network,
   (a) [2 Points] If John calls you but Mary doesn't, what is the probability that a burglar has entered your house? Compare the conditional probability to the marginal probability of burglary.
   (b) [2 Points] This time, Mary calls you but no call from John, what is the probability of burglary? Compare the conditional probability to the one calculated in the previous problem.
   (c) [2 Points] If neither of Mary or John calls you, what is the probability of earthquake? Compare this value to the marginal probability of earthquake.
4. [8 Points Extra Credit] You are interviewing students to hire for your company. Of course, you want intelligent students. You have access to their SAT scores, but SAT($S = true$ if SAT score is high) scores are not absolutely trustworthy measure of intelligence, so you read Yaser's recommendation letters. Yaser writes a good($L = true$) or bad($L = false$) letter based only on whether the students passed($G = true$) or failed 15381. You also want to know how hard 15381 was because sometimes even highly intelligent students fail courses when they are hard($D = true$) and some lucky but unintelligent students pass the course when it is easy($D = false$). Answer the following queries using your implementation.

(a) Student A has high SAT score but has a bad recommendation letter. What is the conditional probability of the difficulty of the course? Compare this with the marginal distribution of the difficulty of the course.

(b) Student B has low SAT score and a good recommendation letter. What is the conditional probability of the difficulty of the course? Compare this with the marginal distribution of the difficulty of the course.

(c) Student C has high SAT score and also has a good recommendation from Yaser. What is the probability that the student is intelligent?

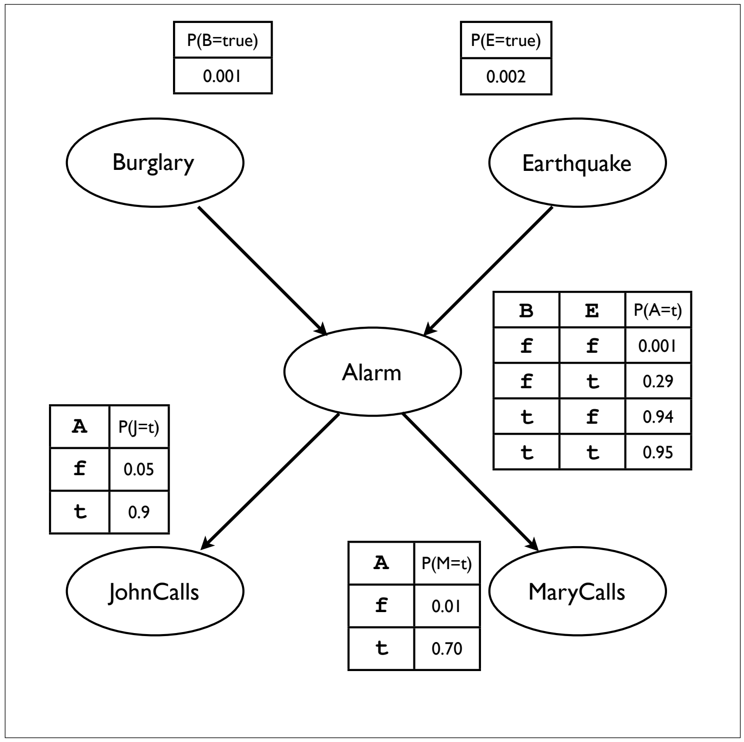
(d) Student D has low SAT score and but has a good recommendation from Yaser. What is the probability that the student is intelligent? Compare it to the marginal probability of a student being intelligent.



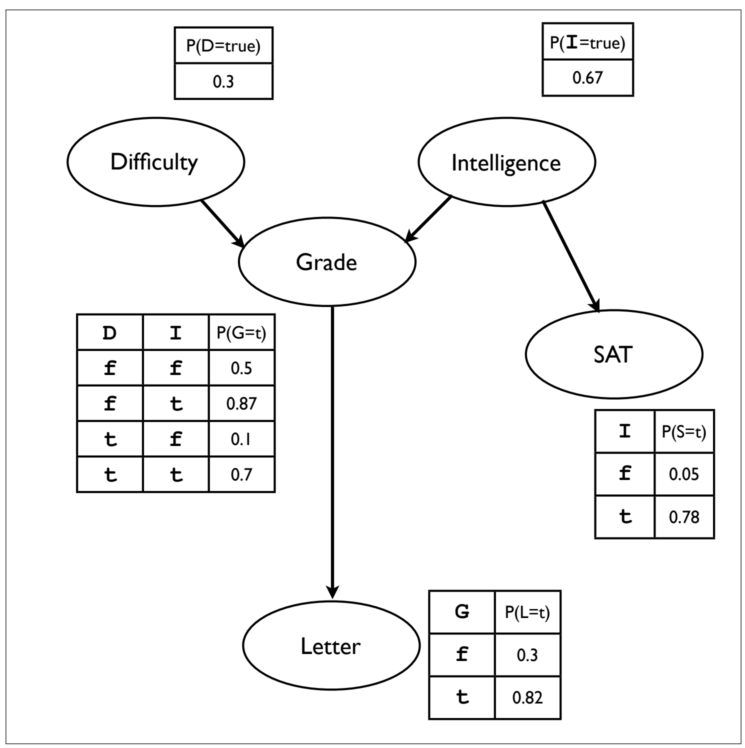
```
function ENUMERATION-ASK(X, e, bn) returns a distribution over X
  inputs: X, the query variable
          e, observed values for variables E
          bn, a Bayes net with variables {X} ∪ E ∪ Y   /* Y = hidden variables */

  Q(X) ← a distribution over X, initially empty
  for each value x_i of X do
      Q(x_i) ← ENUMERATE-ALL(bn.VARS, e_{x_i})
          where e_{x_i} is e extended with X = x_i
  return NORMALIZE(Q(X))

function ENUMERATE-ALL(vars, e) returns a real number
  if EMPTY?(vars) then return 1.0
  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
```

Figure 14.9 The enumeration algorithm for answering queries on Bayesian networks.

Figure 2: Inference by enumeration



(a) Burglary network



(b) Student network

Figure 3: Bayesian Networks