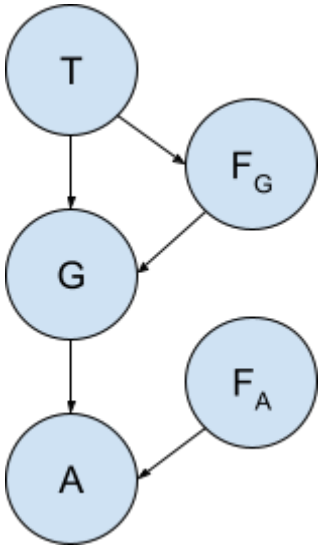
Fall 2022 - CSC545/645 Artificial Intelligence - Assignment 9

Due date: Thursday, November 17, 2022, 2:00pm. Please create a folder called assignment9 in your local working copy of the repository and place all files and folders necessary for the assignment in this folder. Once done with the assignment, add the files and folders to the repo with svn add *files,folders* and then commmit with svn ci -m "SOME USEFUL MESSAGE" *files,folders* .

Exercise 9.1 [20 points]
Read chapter ”Probabilistic Reasoning” of the textbook.

1. **There is an alarm in your local nuclear power station, that senses when a temperature gauge exceeds a given threshold. The gauge measures the core temperature. Consider the Boolean variables *A* (alarm sounds), F_A (alarm is faulty), and F_G (gauge is faulty), and the multi-valued nodes *G* (gauge reading) and *T* (actual core temperature).**

   (a) Draw a Bayesian network for this domain, given that the gauge is more likely to fail when the core temperature gets too high.



(b) Is your network a polytree?

No it is not. Since temp affects the gague and the faultiness of the gague, and the faltiness also affects the gague, there are multiple ways to get to gague from temp - therefore this is not a polytree.

(c) Suppose there are just two possible actual and measured temperatures, Normal and High; the probability that the gauge gives the correct temperature is *x* when it is working, but *y* when it is faulty. Give the conditional probability table associated with *G*.

	F_G = False		F_G = True	
	Temp = Norm	Temp = High	Temp = Norm	Temp = High
G = Norm	x	1-x	y	1-y
G = High	1-x	x	1-y	y

(d) Suppose the alarm works correctly unless it is faulty, in which case it never sounds. Give the conditional probability table associated with *A*.

	F_A = False		F_A = True	
	G = Norm	G = High	G = Norm	G = High
A	0	1	0	0

[10 points]

2. This exercise is concerned with the variable elimination algorithm given in figure 14.11 below.

(a) Section 13.4/14.4 (3rd/4th edition) in the textbook applies variable elimination to the query

P(*Burglary*|*JohnCalls* = *true, maryCalls* = *true*).

Perform the calculations indicated and check that the answer is correct.

P(B | j, m) = $\alpha * \Sigma\Sigma P(b)P(e)P(a \mid b, e)P(j \mid a)P(m \mid a)$

= $\alpha * P(b) \sum_e P(e) \sum_a P(a \mid b, e)P(j \mid a)P(m \mid a)$

= $\alpha * P(b) \sum_e P(e) \left(\left[\frac{.95 \quad .29}{.94 \quad .001}\right] *.9 *.7 + \left[\frac{.05 \quad .71}{.06 \quad .999}\right] *.05 *.01 \right)$

$= \alpha * P(b) \sum_e P(e) \left(\left[\frac{.5985 \quad .1827}{.5922 \quad .00063} \right] + \left[\frac{.000025 \quad .000355}{.00003 \quad .0004996} \right] \right)$

$= \alpha * P(b) \left(.998 \left[\frac{.598525}{.59223} \right] + .002 \left[\frac{.183055}{.0011295} \right] \right)$

$= \alpha * [.001] \left[\frac{.592242}{.001492} \right]$

$= \alpha * \quad < .000592242, .001490508 >$

Just as in the textbook!

(b) Count the number of arithmetic operations performed and compare this with the number performed by the enumeration algorithm.

Multiplications: 2 for the beginning decimals + 2*4 for each a matrix + 2*2 for e matrix + 2 for P(B): **16 multiplications**

Additions: 4 for adding the 4 values in Σa matrixes + 2 for adding the 2 values in the Σe matrixes: **6 additions**

Enumeration would have needed to recompute P(j | a) and P(m | a) for both values of e, which would have cost an **additional 2 multiplications** to have them explicitly solved this way.

(c) Suppose a network has the form of a *chain* – a sequence of Boolean variables $X_1, \ldots, X_n$ where *Parents*$(X_i) = X_{i-1}$ for $i = 2, \ldots, n$. What is the complexity of computing P($X_1 | X_n = true$) using enumeration? Using variable elimination?

Enumeration considers the entire expression trees depth-first and therefore always runs in $O(2^n)$ complexity.

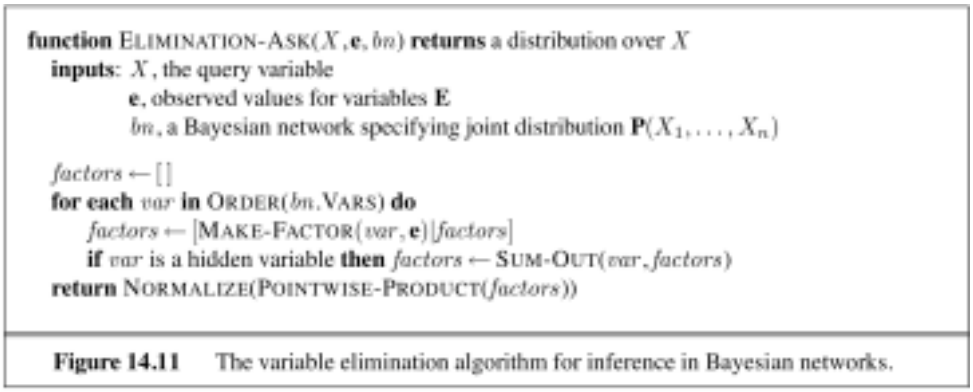
Variable Elimination works by finding out that "every variable that is not an ancestor of a query variable or evidence variable is irrelevant to the query" and therefore only needs to operate on the n relevant known probabilities between X_1 and X_n so its complexity would be O(n).

(d) Prove that the complexity of running variable elimination on a polytree network is linear in the size of the tree for any variable ordering consistent with the network structure.

Running variable elimination of a polytree of size n will always run in linear time O(n). This is because, using variable elimination to create "new nodes" or functions that represent the effect of a node, is calculated bottom-up and only uses information from a nodes immediate parents. If the network is a polytree, this is possible due to conditional independence. Therefore, there are only so many calculations done

to calculate one node. In the best case a node only has one parent which the number of calcuations is linear depending on the size of the current node, or in worse case a node has all n-1 parents, but then after those calculations the network is then reduced to only indepenent nodes (because its a polytree). Therefore regardless of the size of the network the comlexity will be linear. Even more explicitly, we can see from the pseudocode below that we only consider each variable once, create factors or sum through factors those factors, then forget the variable. This is representative of code that runs in O(n) time.

[10 points]



function ELIMINATION-ASK(X, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $factors \leftarrow [\,]$
  for each var **in** ORDER(bn.VARS) **do**
    $factors \leftarrow [\text{MAKE-FACTOR}(var, \mathbf{e}) | factors]$
    if var is a hidden variable **then** $factors \leftarrow \text{SUM-OUT}(var, factors)$
  return NORMALIZE(POINTWISE-PRODUCT($factors$))

Figure 14.11 The variable elimination algorithm for inference in Bayesian networks.