CSE 150A/250A. Assignment 3

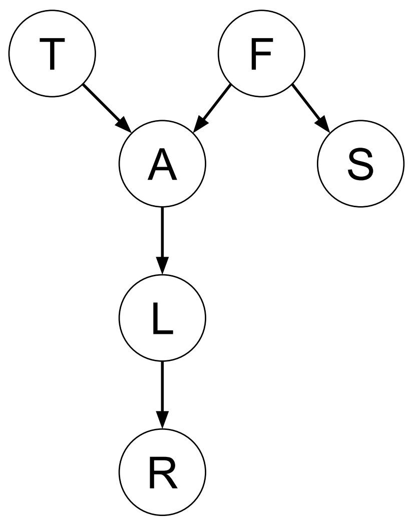
Due: *Mon April 28th* (by 11:59 PM, Pacific Time, via gradescope)
Grace period: 24 hours

3.1 Sign-up for PrairieLearn *(1pt)*

For your midterm exam, you will need to sign-up for PrairieLearn. Click on the course link provided on Canvas and finish Quiz 0. (The quiz will have one dummy question to ensure your sign-up.) Attach a screenshot showing your name and the quiz results.

3.2 Variable Elimination Algorithm

Consider the following belief network with binary random variables.



(Source: Adapted from Poole and Mackworth, *Artificial Intelligence 2E*, Section 8.3.2. For a description of the meaning of the variables in this belief network, see Example 8.15.)

Suppose we want to compute a probability distribution over T given the evidence $S = 1$ and $R = 1$. That is, we want to compute $P(T = 0|S = 1, R = 1)$ and $P(T = 1|S = 1, R = 1)$. In this question, we will evaluate the efficiency of two different methods of computing these probabilities.

The conditional probabilities of this belief network (expressed as factor tables) are as follows:

T	$f_0(T) = P(T)$
0	0.98
1	0.02

F	$f_1(F) = P(F)$
0	0.99
1	0.01

T	F	A	$f_2(T, F, A) = P(A\|T, F)$
0	0	0	0.9999
0	0	1	0.0001
0	1	0	0.01
0	1	1	0.99
1	0	0	0.15
1	0	1	0.85
1	1	0	0.5
1	1	1	0.5

F	S	$f_3(F, S) = P(S\|F)$
0	0	0.99
0	1	0.01
1	0	0.1
1	1	0.9

A	L	$f_4(A, L) = P(L\|A)$
0	0	0.999
0	1	0.001
1	0	0.12
1	1	0.88

L	R	$f_5(L, R) = P(R\|L)$
0	0	0.99
0	1	0.01
1	0	0.25
1	1	0.75

Note that the conditional probabilities are specified with some redundancy. For example, the tables store both $P(A = 0|T = 1, F = 1)$ and $P(A = 1|T = 1, F = 1)$, even though this information is technically redundant. Therefore, no addition or subtraction operations are required to compute $P(A = 0|T = 1, F = 1)$ based on $P(A = 1|T = 1, F = 1)$ or vice versa.

(a) **Computation using variable elimination**

Use the variable elimination algorithm (with elimination order S, R, L, A, F) to compute

$$P(T = 0|S = 1, R = 1) \quad \text{and} \quad P(T = 1|S = 1, R = 1).$$

You do not need to write code for this; you should apply the algorithm step-by-step and show your work, including any new factor tables that are computed along the way.

(b) **Counting calculations used by the variable elimination algorithm**

In the following table, fill in the number of multiplication, addition, and division operations needed for each phase of the VE algorithm. You should count operations using the same method used in class: for example, $a \times b + c \times d + e \times f$ involves 3 multiplications and 2 additions.

Phase of algorithm	**# multiplications**	**# additions**	**# divisions**
Eliminate S (evidence)	0	0	0
Eliminate R (evidence)	0	0	0
Eliminate L			0
Eliminate A			0
Eliminate F			0
Combine T factors			0
Normalize distribution over T	0	1	2
Total			2

(c) **Counting calculations used by the enumeration algorithm**

In the brute-force enumeration method, we can compute the desired probabilities via

$$P(T = 0|S = 1, R = 1) = \frac{P(T = 0, S = 1, R = 1)}{P(T = 0, S = 1, R = 1) + P(T = 1, S = 1, R = 1)}$$

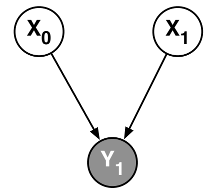
$$P(T = 1|S = 1, R = 1) = \frac{P(T = 1, S = 1, R = 1)}{P(T = 0, S = 1, R = 1) + P(T = 1, S = 1, R = 1)}$$

That is, we compute $P(T=0, S=1, R=1)$ and $P(T=1, S=1, R=1)$ and then normalize to obtain a probability distribution over T.

In the following table, fill in the number of multiplication, addition, and division operations needed for each phase of this algorithm. Assume that the algorithm does a full brute-force calculation, even if some operations are redundant.

Phase of algorithm	**# multiplications**	**# additions**	**# divisions**
Compute $P(T=0, S=1, R=1)$			0
Compute $P(T=1, S=1, R=1)$			0
Normalize distribution over T	0	1	2
Total			2

3.3 Inference in a chain



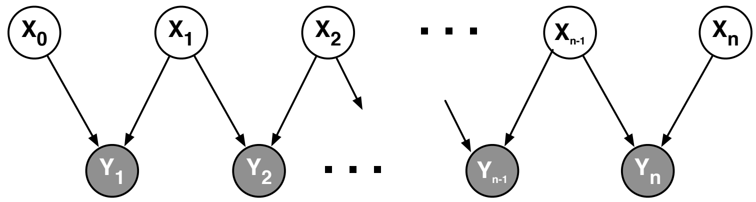
Consider the simple belief network shown to the right, with nodes X_0, X_1, and Y_1. To compute the posterior probability $P(X_1|Y_1)$, we can use Bayes rule:

$$P(X_1|Y_1) \;=\; \frac{P(Y_1|X_1)\,P(X_1)}{P(Y_1)}$$

(a) Show how to compute the conditional probability $P(Y_1|X_1)$ that appears in the numerator of Bayes rule from the CPTs of the belief network.

(b) Show how to compute the marginal probability $P(Y_1)$ that appears in the denominator of Bayes rule from the CPTs of the belief network.

Next you will show how to generalize these computations when the basic structure of this DAG is repeated to form a chain. Like the example we saw in class, this is another instance of efficient inference in polytrees.



Consider how to efficiently compute the posterior probability $P(X_n|Y_1, Y_2, \ldots, Y_n)$ in the above belief network. One approach is to derive a recursion from the *conditionalized* form of Bayes rule

$$P(X_n|Y_1, Y_2, \ldots, Y_n) \;=\; \frac{P(Y_n|X_n, Y_1, Y_2, \ldots, Y_{n-1})\,P(X_n|Y_1, Y_2, \ldots, Y_{n-1})}{P(Y_n|Y_1, \ldots, Y_{n-1})}$$

where the nodes $Y_1, Y_2, \ldots, Y_{n-1}$ are treated as background evidence. In this problem you will express the conditional probabilities on the right hand side of this equation in terms of the CPTs of the network and the probabilities $P(X_{n-1}\!=\!x|Y_1, Y_2, \ldots, Y_{n-1})$, *which you may assume have been computed at a previous step of the recursion*. Your answers to (a) and (b) should be helpful here.

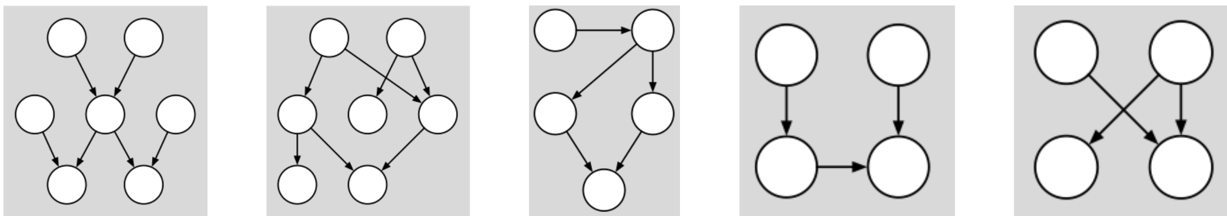
(c) Simplify the term $P(X_n|Y_1, Y_2, \ldots, Y_{n-1})$ that appears in the numerator of Bayes rule.

(d) Show how to compute the conditional probability $P(Y_n|X_n, Y_1, Y_2, \ldots, Y_{n-1})$ that appears in the numerator of Bayes rule. Express your answer in terms of the CPTs of the belief network and the probabilities $P(X_{n-1}\!=\!x|Y_1, Y_2, \ldots, Y_{n-1})$, which you may assume have already been computed.

(e) Show how to compute the conditional probability $P(Y_n|Y_1, Y_2, \ldots, Y_{n-1})$ that appears in the denominator of Bayes rule. Express your answer in terms of the CPTs of the belief network and the probabilities $P(X_{n-1}\!=\!x|Y_1, Y_2, \ldots, Y_{n-1})$, which you may assume have already been computed.

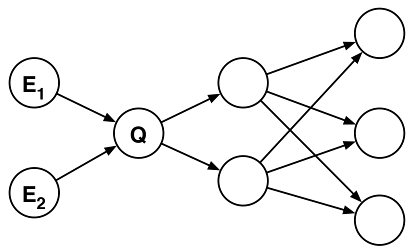
3.4 Node clustering and polytrees

In the figure below, *circle* the DAGs that are polytrees. In the other DAGs, shade **two** nodes that could be *clustered* so that the resulting DAG is a polytree.



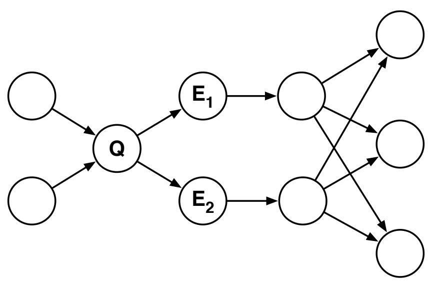
3.5 Cutsets and polytrees

Clearly not all problems of inference are intractable in loopy belief networks. As a trivial example, consider the query $P(Q|E_1, E_2)$ in the network shown below:

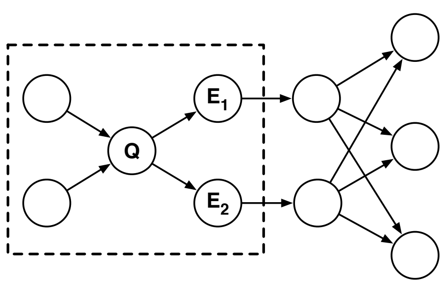


In this case, because E_1 and E_2 are the parents of Q, the query $P(Q|E_1, E_2)$ can be answered directly from the conditional probability table at node Q.

As a less trivial example, consider how to compute the posterior probability $P(Q|E_1, E_2)$ in the belief network shown below:



In this belief network, the posterior probability $P(Q|E_1, E_2)$ can be correctly computed by running the polytree algorithm on the subgraph of nodes that are enclosed by the dotted rectangle:



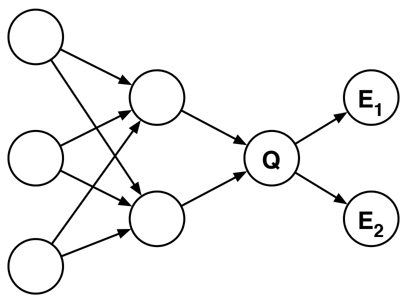
In this example, the evidence nodes d-separate the query node from the loopy parts of the network. Thus for this inference the polytree algorithm would terminate before encountering any loops.

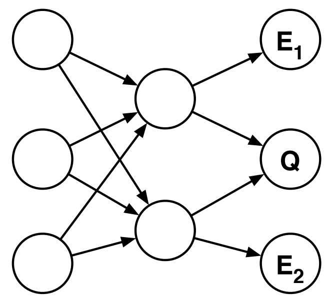
(continued)

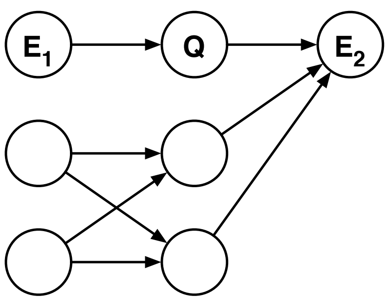
For each of the five loopy belief networks shown below, consider how to compute the posterior probability $P(Q|E_1, E_2)$.

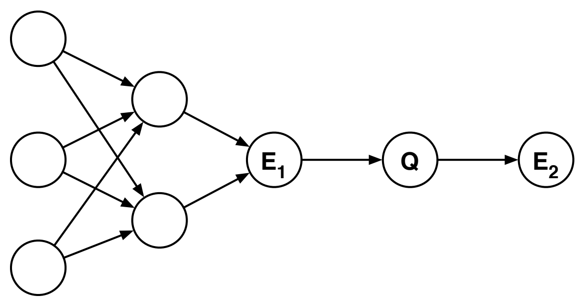
If the inference can be performed by running the polytree algorithm on a subgraph, enclose this subgraph by a dotted line as shown on the previous page. (The subgraph should be a polytree.)

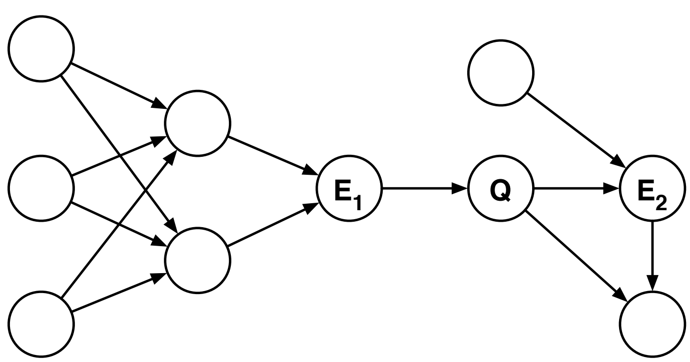
On the other hand, if the inference cannot be performed in this way, shade **one** node in the belief network that can be instantiated to induce a polytree by the method of cutset conditioning.









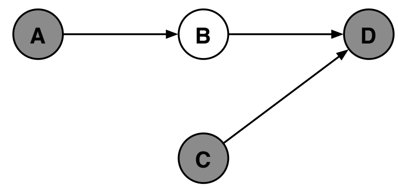


3.6 Even more inference

Show how to perform the desired inference in each of the belief networks shown below. Justify briefly each step in your calculations.

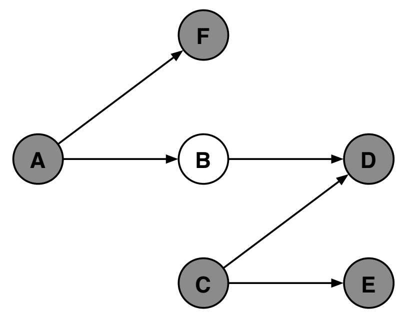
(a) **Markov blanket**

Show how to compute the posterior probability $P(B|A, C, D)$ in terms of the CPTs of this belief network—namely, $P(A)$, $P(B|A)$, $P(C)$, and $P(D|B, C)$.



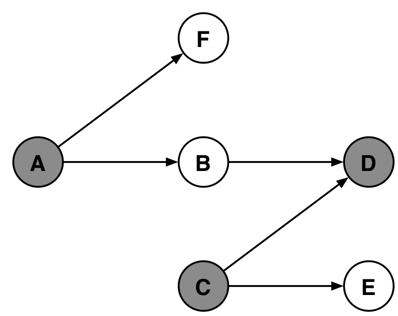
(b) **Conditional independence**

This belief network has conditional probability tables for $P(F|A)$ and $P(E|C)$ in addition to those of the previous problem. Assuming that all these tables are given, show how to compute the posterior probability $P(B|A, C, D, E, F)$ in terms of these additional CPTs and your answer to part (a).

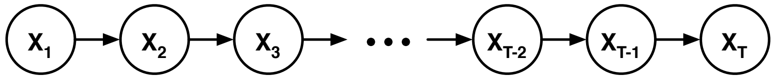


(c) **More conditional independence**

Assuming that all the conditional probability tables in this belief network are given, show how to compute the posterior probability $P(B, E, F|A, C, D)$. Express your answer in terms of the CPTs of the network, as well as your earlier answers for parts (a) and (b).



3.7 Inference in a chain (250A only)



Consider the belief network shown above with random variables $X_t \in \{1, 2, \ldots, m\}$. Suppose that the CPT at each non-root node is given by the same $m \times m$ matrix; that is, for all $t \geq 1$, we have:

$$A_{ij} \;=\; P(X_{t+1} = j | X_t = i).$$

(a) Prove that $P(X_{t+1} = j | X_1 = i) = [A^t]_{ij}$, where A^t is the t^{th} power of the matrix A.
Hint: use induction.

(b) Consider the computational complexity of this inference. Devise a simple algorithm, based on matrix-vector multiplication, that scales as $O(m^2 t)$.

(c) Show alternatively that the inference can also be done in $O(m^3 \log_2 t)$.

(d) Suppose that the transition matrix A_{ij} is sparse, with at most $s \ll m$ non-zero elements per row. Show that in this case the inference can be done in $O(smt)$.

(e) Show how to compute the posterior probability $P(X_1 = i | X_{T+1} = j)$ in terms of the matrix A and the prior probability $P(X_1 = i)$. *Hint:* use Bayes rule and your answer from part (a).