CSE 150A/250A. Assignment 3

3.2 Variable Elimination Algorithm

(a) **Computation using variable elimination**

We start with the following factors:

- $f_0(T)$ for $P(T)$
- $f_1(F)$ for $P(F)$
- $f_2(T, F, A)$ for $P(A|T, F)$
- $f_3(F, S)$ for $P(S|F)$
- $f_4(A, L)$ for $P(L|A)$
- $f_5(L, R)$ for $P(R|L)$

Eliminating the evidence variables S and R yields factors $f_3'(F) = f_3(F, S{=}1)$ and $f_5'(L) = f_5(L, R{=}1)$:

F	$f_3'(F)$
0	0.01
1	0.9

L	$f_5'(L)$
0	0.01
1	0.75

To eliminate L, we can define $f_6(A) = \sum_L f_4(A, L) \cdot f_5'(L)$. This involves 4 multiplications and 2 additions:

A	$f_6(A)$
0	$f_4(A=0, L=0)f_5'(L=0) + f_4(A=0, L=1)f_5'(L=1) = \mathbf{0.01074}$
1	$f_4(A=1, L=0)f_5'(L=0) + f_4(A=1, L=1)f_5'(L=1) = \mathbf{0.6612}$

To eliminate A, we can define $f_7(T, F) = \sum_A f_6(A) \cdot f_2(T, F, A)$. This involves 8 multiplications and 4 additions:

T	F	$f_7(T, F)$
0	0	$f_6(A=0)f_2(T=0, F=0, A=0) + f_6(A=1)f_2(T=0, F=0, A=1) = \mathbf{0.01080505}$
0	1	$f_6(A=0)f_2(T=0, F=1, A=0) + f_6(A=1)f_2(T=0, F=1, A=1) = \mathbf{0.6546954}$
1	0	$f_6(A=0)f_2(T=1, F=0, A=0) + f_6(A=1)f_2(T=1, F=0, A=1) = \mathbf{0.563631}$
1	1	$f_6(A=0)f_2(T=1, F=1, A=0) + f_6(A=1)f_2(T=1, F=1, A=1) = \mathbf{0.33597}$

To eliminate F, we can define $f_8(T) = \sum_F f_7(T, F) \cdot f_1(F) \cdot f_3'(F)$. This involves 8 multiplications and 2 additions:

T	$f_8(T)$
0	$f_7(T=0, F=0)f_1(F=0)f_3'(F=0) + f_7(T=0, F=1)f_1(F=1)f_3'(F=1) = \mathbf{0.00599923}$
1	$f_7(T=1, F=0)f_1(F=0)f_3'(F=0) + f_7(T=1, F=1)f_1(F=1)f_3'(F=1) = \mathbf{0.00860368}$

An alternate calculation of $f_8(T)$ uses 6 multiplications and 2 additions by computing

- $f_8'(F) = f_1(F) \cdot f_3'(F)$, via 2 multiplications, followed by
- $f_8(T) = \sum_F f_8'(F) \cdot f_7(T, F)$, via 4 multiplications, and 2 additions.

To combine the remaining T factors, we can define $f_9(T) = f_0(T) \cdot f_8(T)$. This involves 2 multiplications:

T	$f_9(T)$
0	$f_0(T=0)f_8(T=0) = \mathbf{0.00587924}$
1	$f_0(T=1)f_8(T=1) = \mathbf{0.00017207}$

Lastly, we normalize the factor f_9 to compute a distribution over T:

$$P(T=0|R=1, S=1) = \frac{f_9(T{=}0)}{\sum_T f_9(T)} = \frac{0.00587924}{0.00587924 + 0.00017207} = \mathbf{0.97156429}$$

$$P(T=1|R=1, S=1) = \frac{f_9(T{=}1)}{\sum_T f_9(T)} = \frac{0.00017207}{0.00587924 + 0.00017207} = \mathbf{0.02843571}$$

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

Phase of algorithm	**# multiplications**	**# additions**	**# divisions**
Eliminate S (evidence)	0	0	0
Eliminate R (evidence)	0	0	0
Eliminate L	**4**	**2**	0
Eliminate A	**8**	**4**	0
Eliminate F	**6** or **8**	**2**	0
Combine T factors	**2**	**0**	0
Normalize distribution over T	0	1	2
Total	**20** or **22**	**9**	2

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

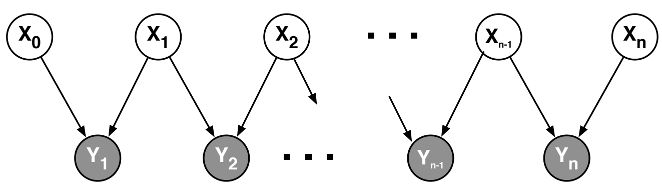
After marginalizing over F, A, and L, applying the product rule, and using the network's conditional independence relationships, we have

$$P(T=0, S=1, R=1) = \sum_f \sum_a \sum_l P(T=0)P(F=f)P(A=a|T=0, F=f)$$
$$\cdot\, P(S=1|F=f)P(L=l|A=a)P(R=1|L=l)$$

Here, there are $2^3 = 8$ terms, each with 6 numbers multiplied together. So each term requires 5 multiplications, and they are summed together using 7 additions. The same process is repeated for $T = 1$, and we end by normalizing to get a distribution over T.

Phase of algorithm	**# multiplications**	**# additions**	**# divisions**
Compute $P(T=0, S=1, R=1)$	$5 \times 2^3 = \mathbf{40}$	$2^3 - 1 = \mathbf{7}$	0
Compute $P(T=1, S=1, R=1)$	$5 \times 2^3 = \mathbf{40}$	$2^3 - 1 = \mathbf{7}$	0
Normalize distribution over T	0	1	2
Total	**80**	**15**	2

3.3 More inference in a chain *(10 pts)*



(a) **First term in numerator** *(2 pts)*

$$\begin{aligned}
P(Y_1|X_1) &= \sum_x P(Y_1, X_0{=}x|X_1) \quad \textbf{(marginalization)} \\
&= \sum_x P(X_0{=}x|X_1)\, P(Y_1|X_0{=}x, X_1) \quad \textbf{(product rule)} \\
&= \sum_x P(X_0{=}x)\, P(Y_1|X_0{=}x, X_1) \quad \textbf{(conditional independence)}
\end{aligned}$$

(b) **Denominator** *(2 pts)*

$$
\begin{aligned}
P(Y_1) &= \sum_{x'} P(Y_1, X_1 = x') \quad \textbf{(marginalization)} \\
&= \sum_{x'} P(X_1 = x')\, P(Y_1 | X_1 = x') \quad \textbf{(product rule)} \\
&= \sum_{x,x'} P(X_0 = x)\, P(X_1 = x')\, P(Y_1 | X_0 = x, X_1 = x') \quad \textbf{(substitution from (a))}
\end{aligned}
$$

(c) **Second term in numerator** *(1 pt)*

$$
P(X_n | Y_1, Y_2, \ldots, Y_{n-1}) = P(X_n) \quad \textbf{(marginal independence)}
$$

(d) **First term in numerator** *(3 pts)*

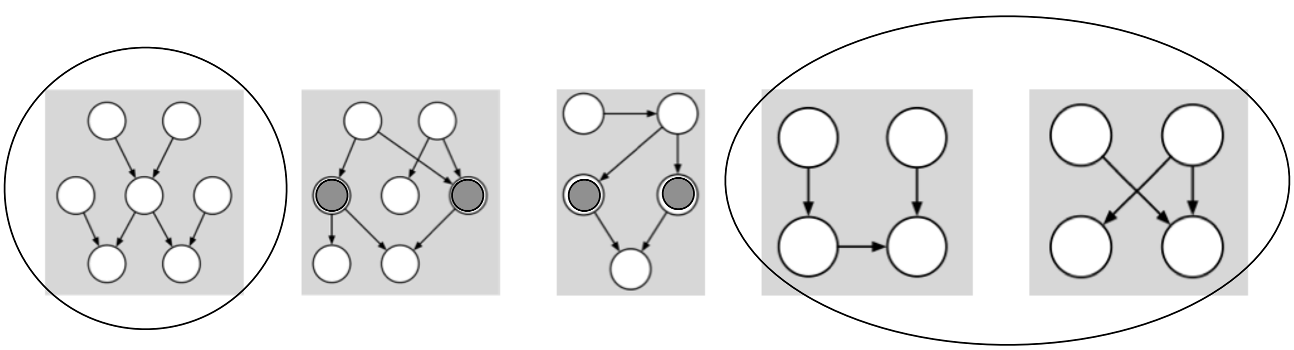
$$
\begin{aligned}
&P(Y_n | X_n, Y_1, Y_2, \ldots, Y_{n-1}) \\
&\quad = \sum_{x} P(Y_n, X_{n-1} = x | X_n, Y_1, Y_2, \ldots, Y_{n-1}) \quad \textbf{(marginalization)} \\
&\quad = \sum_{x} P(X_{n-1} = x | X_n, Y_1, Y_2, \ldots, Y_{n-1})\, P(Y_n | X_{n-1} = x, X_n, Y_1, Y_2, \ldots, Y_{n-1}) \quad \textbf{(product rule)} \\
&\quad = \sum_{x} P(X_{n-1} = x | Y_1, Y_2, \ldots, Y_{n-1})\, P(Y_n | X_{n-1} = x, X_n) \quad \textbf{(conditional independence)}
\end{aligned}
$$

(e) **Denominator** *(2 pts)*

$$
\begin{aligned}
&P(Y_n | Y_1, Y_2, \ldots, Y_{n-1}) \\
&\quad = \sum_{x'} P(Y_n, X_n = x' | Y_1, Y_2, \ldots, Y_{n-1}) \quad \textbf{(marginalization)} \\
&\quad = \sum_{x'} P(Y_n | X_n = x', Y_1, Y_2, \ldots, Y_{n-1})\, P(X_n = x' | Y_1, Y_2, \ldots, Y_{n-1}) \quad \textbf{(product rule)} \\
&\quad = \sum_{x,x'} P(X_{n-1} = x | Y_1, Y_2, \ldots, Y_{n-1})\, P(Y_n | X_{n-1} = x, X_n = x')\, P(X_n = x') \quad \textbf{(substitution from c\&d)}
\end{aligned}
$$

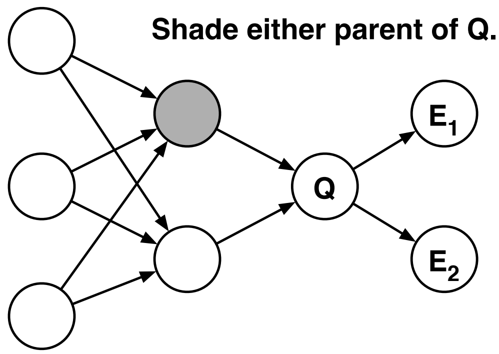
3.4 Node clustering and polytrees *(5 pts)*

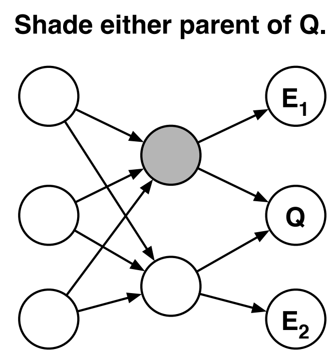
In the figure below, *circle* the DAGs that are polytrees. In the other DAGs, shade **two** nodes that could be *clustered* so that the resulting DAG is a polytree.

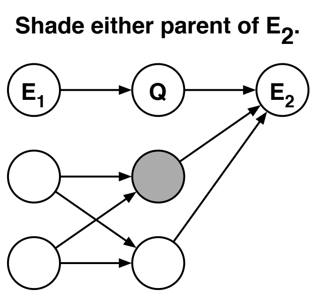


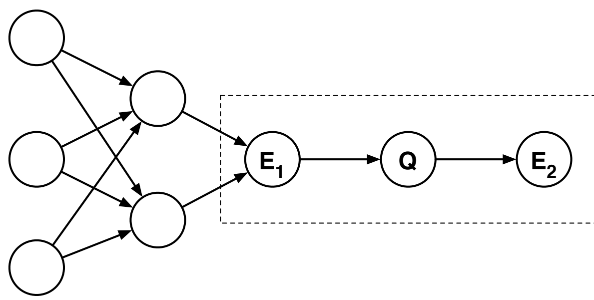
3.5 Cutsets and polytrees *(5 pts)*

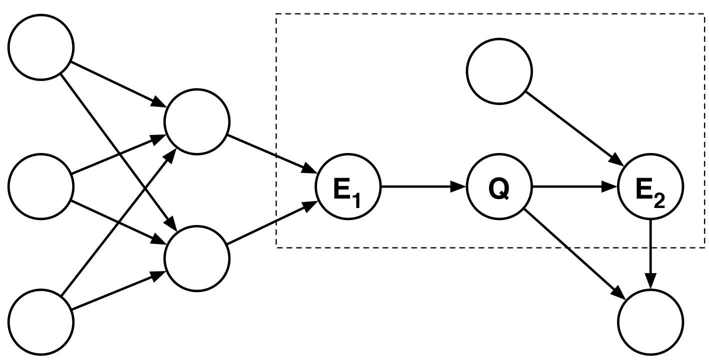
For each of the five loopy belief networks shown below, consider how to compute the posterior probability $P(Q|E_1, E_2)$. If the inference can be performed by running the polytree algorithm on a subgraph, enclose this subgraph by a dotted line as shown on the previous page. On the other hand, if the 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.





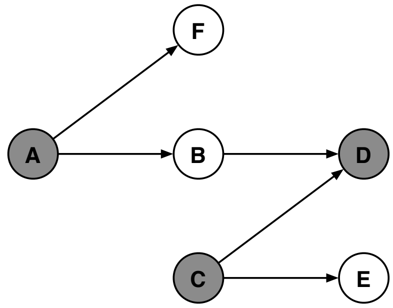






For the last belief network, note that $P(Q|E_1, E_2)$ can be computed in terms of the CPTs for $P(Q|E_1)$, $P(\mathrm{pa}(\mathrm{E}_2))$, and $P(E_2|Q, \mathrm{pa}(E_2))$.

3.6 Even more inference *(5 pts)*



(a) **Markov blanket** *(3 pts)*

$$\begin{aligned}
P(B|A,C,D) &= \frac{P(D|B,A,C)\,P(B|A,C)}{P(D|A,C)} \quad \textbf{(conditionalized Bayes rule)} \\
&= \frac{P(D|B,C)\,P(B|A)}{P(D|A,C)} \quad \textbf{(conditional independence)} \\
&= \frac{P(D|B,C)\,P(B|A)}{\sum_b P(B\!=\!b, D|A,C)} \quad \textbf{(marginalization)} \\
&= \frac{P(D|B,C)\,P(B|A)}{\sum_b P(B\!=\!b|A,C)\,P(D|A,B\!=\!b,C)} \quad \textbf{(product rule)} \\
&= \frac{P(B|A)\,P(D|B,C)}{\sum_b P(B\!=\!b|A)\,P(D|B\!=\!b,C)} \quad \textbf{(conditional independence)}
\end{aligned}$$

(b) **Conditional independence** *(1 pt)*

The nodes A, C, and D form the Markov blanket of B.
Hence $P(B|A,C,D,E,F) = P(B|A,C,D)$.
The answer is the same as part (a).

(c) **More conditional independence** *(1 pts)*

$$\begin{aligned}
P(B,E,F|A,C,D) &= P(B|A,C,D)\,P(E|A,B,C,D)\,P(F|A,B,C,D,E) \quad \textbf{(product rule)} \\
&= P(B|A,C,D)\,P(E|C)\,P(F|A) \quad \textbf{(conditional independence)}
\end{aligned}$$

The first term on the right side is given by the answer in part (a).

3.7 Inference in a chain *(11 pts)* **(250A only)**

(a) **Powers of transition matrix** *(3 pts)*

We will prove the claim by iiduction. First we note that the claim is trivial at node $t = 1$. Next, assume that the claim is true at node $t-1$. Then at node t:

$$\begin{aligned}
P(X_{t+1}=j|X_1=i) &= \sum_k P(X_{t+1}=j, X_t=k|X_1=i) \quad \boxed{\textbf{marginalization}} \\
&= \sum_k P(X_{t+1}=j|X_t=k, X_1=i)\, P(X_t=k|X_1=i) \quad \boxed{\textbf{product rule}} \\
&= \sum_k P(X_{t+1}=j|X_t=k)\, P(X_t=k|X_1=i) \quad \boxed{\textbf{conditional independence}} \\
&= \sum_k P(X_t=k|X_1=i)\, P(X_{t+1}=j|X_t=k) \\
&= \sum_k [A^{t-1}]_{ik} A_{kj} \quad \boxed{\textbf{induction}} \\
&= [A^t]_{ij}.
\end{aligned}$$

(b) **Linear-time complexity** *(2 pts)*

Note that computing any entry in the jth column of A^t requires only the jth column of A^{t-1}:

$$[A^t]_{ij} = \sum_k A_{ik}[A^{t-1}]_{kj}.$$

Inductively, we need only compute the jth column of each power of A. Each column computation takes $O(m^2)$ time; thus the total time to compute $[A^t]_{ij}$ is $O(m^2 t)$.

(c) **Log-time complexity** *(2 pts)*

Alternatively, we can compute $A, A^2, A^4, \ldots, A^k$, where $k = 2^{\lfloor \log_2 t \rfloor}$ is the largest power of 2 less than t, by repeated squaring in time $O(m^3 \log_2 t)$. A^t can then be computed by multiplying the matrices corresponding to ones in the binary expansion of t. (For example, $A^{11} = A^8 A^2 A$.) This process takes time $O(m^3 \log_2 t)$, since matrix multiplication involves $O(m^3)$ computations. Hence the total computation is $O(m^3 \log_2 t)$.

(d) **Sparse transition matrices** *(1 pt)*

If the transition matrix A_{ij} is sparse, then the sum in (b) can be restricted to the $s \ll m$ non-zero elements of each row. This reduces the computation to $O(smt)$.

(e) **Posterior probability** *(3 pts)*

$$
\begin{aligned}
P(X_1 = \imath | X_{T+1} = j) &= \frac{P(X_{T+1} = j | X_1 = i) P(X_1 = i)}{P(X_{T+1} = j)} \quad \boxed{\textbf{Bayes rule}} \\
&= \frac{P(X_{T+1} = j | X_1 = i) P(X_1 = i)}{\sum_k P(X_1 = k, X_{T+1} = j)} \quad \boxed{\textbf{marginalization}} \\
&= \frac{P(X_{T+1} = j | X_1 = i) P(X_1 = i)}{\sum_k P(X_1 = k)\, P(X_{T+1} = j | X_1 = k)} \quad \boxed{\textbf{product rule}} \\
&= \frac{[A^T]_{ij} P(X_1 = i)}{\sum_k [A^T]_{kj} P(X_1 = k)} \quad \boxed{\textbf{substitution}}
\end{aligned}
$$
