CSE 250A-Assignment 3

3.1) <u>Base case</u>: $t=1$

a) We compute $P(X_{t+1}=j \mid X_t=i) = P(X_2=j \mid X_1=i)$

$$[A^t]_{ij} = [A^1]_{ij} = [A]_{ij} \text{ and } P(X_2=j \mid X_1=i) = A^1_{ij}$$

Thus, $P(X_{t+1}=j \mid X_1=i) = [A^t]_{ij}$ for the base case of $t=1$.

<u>Inductive Step:</u>

Assume that $P(X_{t+1}=j \mid X_1=i) = [A^t]_{ij}$ for the case of t.

We want to prove that this holds for the case of $t+1$. That is, it holds for $P(X_{t+2}=j \mid X_1=i) = [A^{t+1}]_{ij}$.

$$P(X_{t+2}=j \mid X_1=i) =$$
$$\sum_a P(X_{t+2}=j, X_{t+1}=a \mid X_1=i) =$$
$$\frac{\sum_a P(X_{t+2}=j, X_{t+1}=a, X_1=i)}{P(X_1=i)} =$$
$$\frac{\sum_a P(X_{t+2}=j, X_{t+1}=a, X_1=i)}{P(X_1=i)} \cdot \frac{P(X_{t+1}=a)}{P(X_{t+1}=a)} =$$
$$\frac{\sum_a P(X_{t+2}=j, X_{t+1}=a, X_1=i)}{P(X_{t+1}=a)} \cdot \frac{P(X_{t+1}=a)}{P(X_1=i)} =$$
$$\sum_a P(X_{t+2}=j, X_1=i \mid X_{t+1}=a) \cdot P(X_{t+1}=a \mid X_1=i) =$$
$$\sum_a P(X_{t+2}=j \mid X_{t+1}=a) \cdot \underbrace{P(X_{t+1}=a \mid X_1=i)}_{\text{By induction}} =$$
$$\sum_a A_{aj} \cdot [A^t]_{ia} =$$
$$[A^{t+1}]_{ij}$$

b) Recall from part a that, to compute $[A^{t+1}]_{ij}$, we needed to compute the product of $\sum_a A_{aj} \cdot [A^t]_{ia}$.

The algorithm is essentially multiplying the vector formed by $\sum_a A_{aj}$ with the row i in $[A^t]$. Doing this for all rows in matrix $[A^t]$ is computing vector-matrix multiplication, which given that the vector is size m and the matrix is size $m \times m$, this would result in a computation time of $O(m^2)$. Given that we have t matrices, we thus have a runtime of $O(m^2 \cdot t)$.

c) Note that the matrix A^t can be computed as $A^t = A^{\frac{t}{2}} \cdot A^{\frac{t}{2}}$ when t is even and $A^t = A^{\lfloor \frac{t}{2} \rfloor} \cdot A^{\lfloor \frac{t}{2} \rfloor} \cdot A$ if t is odd. Given that we can keep recursively dividing these matrices into half of t until we hit A^1 as the base case, we can solve this with divide and conquer. Each matrix multiplication takes $O(m^3)$ time. Thus, given that we split the matrix $O(\log(t))$ times, we get a runtime of $O(m^3 \cdot \log(t))$ for computing A^t. To get the inference result, however, we run the usual vector-matrix multiplication again, which takes $O(m^2)$ time. Overall, this would take $O(m^3 \cdot \log(t))$ time.

d) Given that $s \ll m$ such that there are at most s non-zero elements each row, this implies that the number of multiplications with a vector of size m would be $O(s)$. Thus, we reduce vector-matrix multiplication from $O(m^2)$ to $O(s \cdot m)$ - that is, we have m iterations of vector-matrix multiplication with at most s multiplications. Once again, since we have t matrices, this gives us an overall run time of $O(s \cdot m \cdot t)$.

e) $$P(X_1 = i \mid X_{T+1} = j) = \frac{P(X_1 = i, X_{T+1} = j)}{P(X_{T+1} = j)} = \text{Baye's rule}$$

$$\frac{P(X_1 = i, X_{T+1} = j)}{\sum_a P(X_{T+1} = j, X_1 = a)} = \text{Baye's rule}$$

$$\frac{P(X_{T+1} = j \mid X_1 = i) \cdot P(X_1 = i)}{\sum_a P(X_{T+1} = j \mid X_1 = a) \cdot P(X_1 = a)} = \text{Baye's rule}$$

$$\frac{[A^T]_{ij} \cdot P(X_1 = i)}{\sum_a [A^T]_{aj} \cdot P(X_1 = a)}$$

3.2) We'd need to consider the other parent

a) node X_0

$P(Y_1 | X_1) = \boxed{\sum_a P(Y_1 | X_1, X_0 = a) \cdot P(X_0 = a)}$ X_1 and X_0 are independent of each other.

b) Similar to above,

$P(Y_1) = \sum_a P(Y_1 | X_1 = a) \cdot P(X_1 = a)$

$\boxed{\sum_b \sum_a P(Y_1 | X_1 = a, X_0 = b) \cdot P(X_1 = a) \cdot P(X_0 = b)}$

c) Note: In the probability of $P(X_n | Y_1, Y_2, \ldots, Y_{n-1})$, there is no Y_n, so it isn't in our evidence set. In the belief network for problem 3.2, notice how node Y_n blocks any path from node X_n to every other node because of the unobserved common ~~efect~~ effect condition in d-separation. Thus,

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

d) We use the same process that I did in part a.

$P(Y_n | X_n, Y_1, \ldots, Y_{n-1}) =$ X_{n-1} and X_n are independent

$\sum_a P(Y_n | X_{n-1} = a, X_n, Y_1, \ldots, Y_{n-1}) \cdot P(X_{n-1} = a | X_n, Y_1, \ldots, Y_n) =$

$\sum_a P(Y_n | \underbrace{X_{n-1} = a, X_n, Y_1, \ldots, Y_{n-1}}_{\text{common cause condition}}) \cdot P(X_{n-1} = a | Y_1, \ldots, Y_n) =$

$\boxed{\sum_a P(Y_n | X_{n-1} = a, X_n) \cdot P(X_{n-1} = a | Y_1, \ldots, Y_n)}$

e) Note: X_{n-1} and X_n are parents of Y_n, so we need to bring both in.

$P(Y_n \mid Y_1, \ldots, Y_{n-1}) =$

$\sum_a P(Y_n \mid X_n = a, Y_1, \ldots, Y_{n-1}) \cdot P(X_n = a \mid Y_1, \ldots, Y_{n-1}) =$

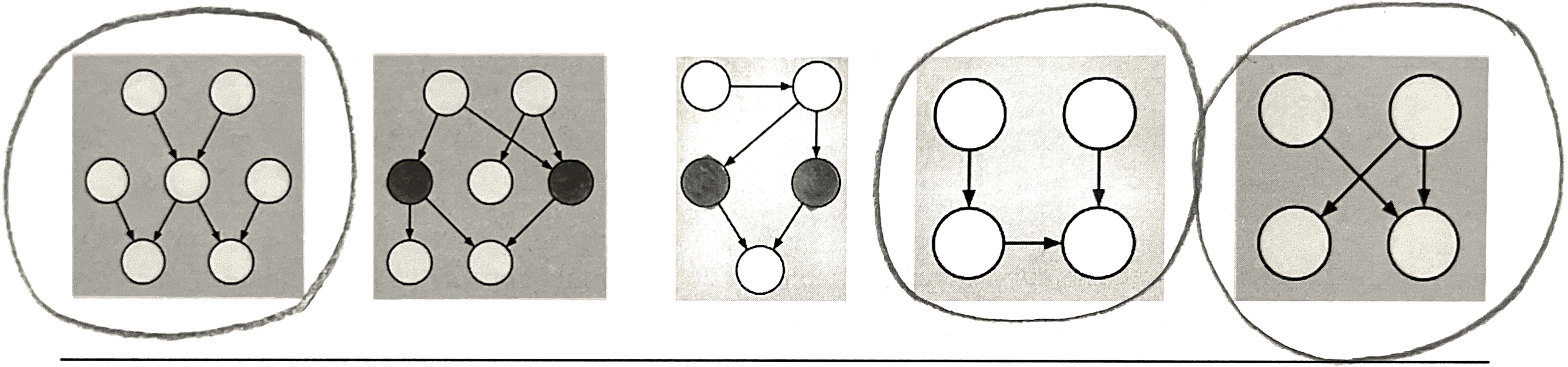
$\sum_b \sum_a P(Y_n \mid X_n = a, X_{n-1} = b, Y_1, \ldots, Y_{n-1}) \cdot \underbrace{P(X_n = a \mid Y_1, \ldots, Y_{n-1})}_{\text{unobserved common effect at } Y_n} \cdot P(X_{n-1} = b \mid Y_1, \ldots, Y_{n-1}) =$

$\sum_b \sum_a P(Y_n \mid X_n = a, \underbrace{X_{n-1}}_{\text{common cause condition}} = b, Y_1, \ldots, Y_{n-1}) \cdot P(X_n = a) \cdot P(X_{n-1} = b \mid Y_1, \ldots, Y_{n-1}) =$

$\boxed{\sum_b \sum_a P(Y_n \mid X_n = a, X_{n-1} = b) \cdot P(X_n = a) \cdot P(X_{n-1} = b \mid Y_1, \ldots, Y_{n-1})}$

3.3 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.



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.