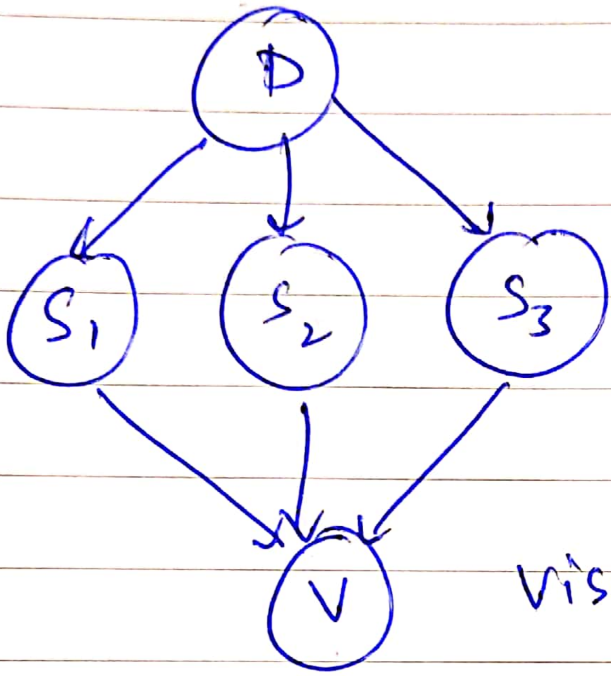
Review (Polytree)

- Singly connected, no loops, at most one path between two nodes
- Efficient algorithms for exact inference

Ex. Simpler (1-loop)

Disease $D \in \{0,1\}$.



Symptomps $S_i \in \{0,1\}$

Visit Doctor $V \in \{0,1\}$

How to compute $P(V=1)$?

$\boxed{\text{Exact inference in loopy BNs.}}$

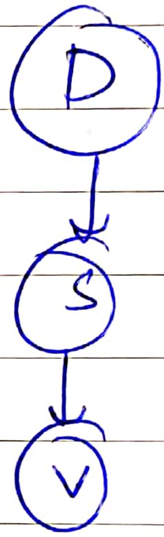
How to turn a loopy BN into a polytree?

1. Node Clustering
   - Merge Nodes to form polytree.
   ex. Cluster S_1, S_2, S_3 into mega-node.

- merge CPTs

ex. merge $P(S_1 \mid D)$, $P(S_2 \mid D)$, $P(S_3 \mid D)$ into $P(S \mid D)$

S_1	S_2	S_3	S
0	0	0	0
1	0	0	1
0	1	0	2
⋮	⋮	⋮	⋮
1	1	1	7



D	$P(S=0 \mid D)$	$P(S=1 \mid D)$	....	$P(S=7 \mid D)$
0				
1				

eg. $P(S=1 \mid D=1) = P(S_1=1, S_2=0, S_3=0 \mid D=1)$

$= P(S_1=1 \mid D=1)\, P(S_2=0 \mid D=1) \times P(S_3=0 \mid D=1)$

old CPT : $P(V=1 \mid S_1, S_2, S_3)$

S_1	S_2	S_3	S	$P(V=1 \mid S)$
0	0	0	0	
	0	0	1	
0	1	0	2	
			3	
			⋮	
1	1	1	7	

Apply polytree Algorithm to clustered BNs.
(No free Lunch)
* there's hidden complexity

- Tradeoff:
* Polytree Algorithm is linear in size of CPTs.
CPT size grows exponentially with clustering

Size of mega-node – 2^3
Size of CPT – 2^4

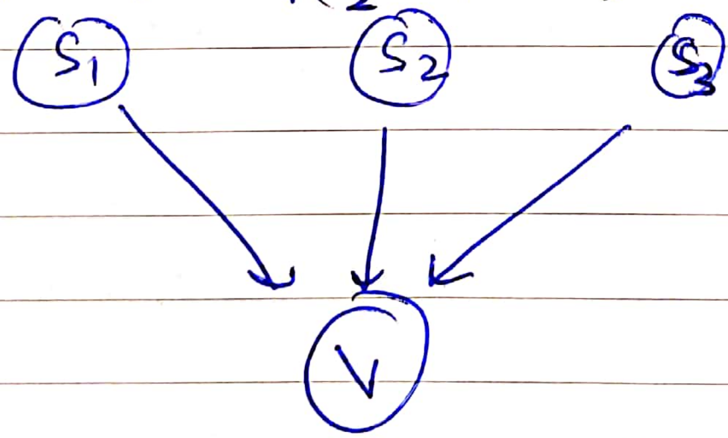
* How to choose optimal clustering of nodes (for maximally efficient inference)? — <u>HARD problem</u>

2. Cutset Conditioning

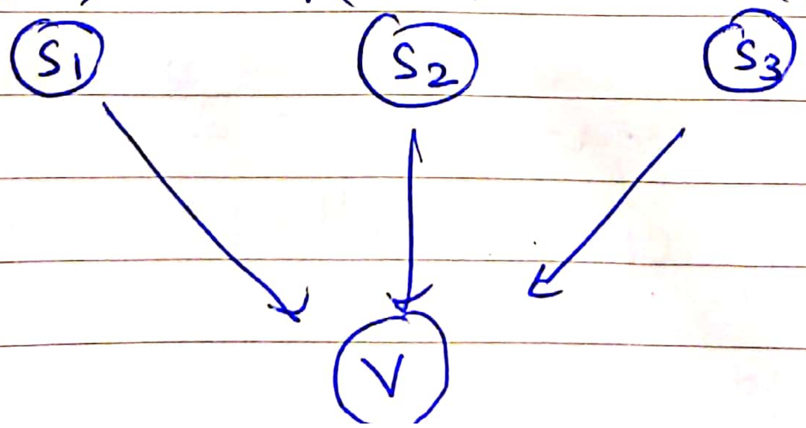
— Instantiate nodes.
So that the remaining nodes form a polytree

ex. instantiate $D=0$ or $D=1$

$P(S_1=1 \mid D=0)$ $P(S_2=1 \mid D=0)$ $P(S_3=1 \mid D=0)$



$P(S_1=1 / D=1)$ $P(S_2=1 \mid D=1)$ $P(S_3=1 \mid D=1)$



- Apply polytree algorithm to compute $\begin{cases} P(V=1 \mid D=0) \text{ in left BN} \\ P(V=1 \mid D=1) \text{ in right BN} \end{cases}$

- Then combine

$$P(V=1) = \sum_{d} P(D=d, V=1)$$

(marginalization)

$$= \sum_{d} P(D=d)\, P(V=1 \mid D=d)$$

(prod. rule)

$P(D=d) \rightarrow$ original BN CPT

$P(V=1 \mid D=d) \rightarrow$ results of polytree algorithm

- Set of instantiated nodes is "<u>cut-set</u>".

- <u>tradeoff</u>: # of terms to combine grows exponentially with size of cutset.

How to choose optimal cutset for maximally efficient inference

Approximate inference in loopy BNs.

Exact inference is NP-hard

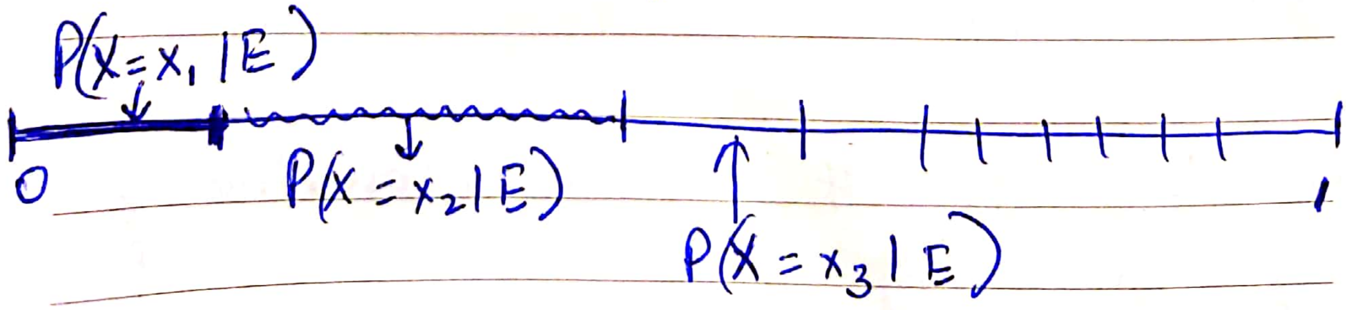
Approx methods often best choice in large loopy BNs.

Basics of stochastic sampling

Given discrete random variable X, the probs. $P(X = x_i | E)$ for all values of x_i, and access to uniform random # generator $r \in [0, 1]$,

$\Rightarrow$ how to sample $X \sim P(X|E)$?

Intuitively, $P(X = x_i | E)$ defines a portion of unity



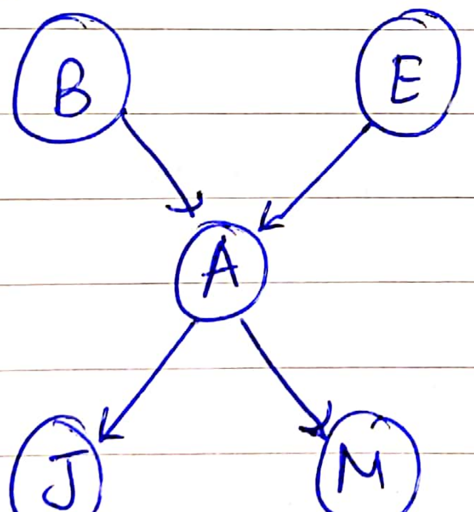
Partition maps random number $r \in [0,1]$ into discrete value of X.

Stochastic Sampling in BNs | RN 14.5

* "Belief Network" defines "generative model"

$$P(X_1, X_2, \ldots X_n) = \prod_i P(X_i \mid Pa(X_i))$$

* Easy to draw samples from joint distribution.



Repeat N times:

$b_i \sim P(B)$

$e_i \sim P(E)$

$a_i \sim P(A \mid B=b_i, E=e_i)$

$(i = 1, 2, \ldots N)$ $\quad j_i \sim P(J \mid A=a_i)$

$m_i \sim P(M \mid A=a_i)$

$\Rightarrow$ Sample $\{(b_i, e_i, a_i, j_i, m_i)\}_{i=1}^{N}$

* Statistics of large sample

$$\frac{\text{Count}(B=b, E=e, A=a, J=j, M=m)}{N}$$

Converges as $N \to \infty$ to $P(B=b, E=e A=a, J=j, M=m)$

* But hard (in general) to draw samples from. posterior distributions (eg. $P(B=1 \mid M=1)$)

* How to estimate (or approximate) $P(Q=q \mid E=e)$?

E = evidence node(s)
Q = query node(s)

1) Rejection Sampling in BNs:
To estimate $P(Q=q \mid E=e)$
Generate N samples from joint distribution.

So we have N tuples of all variables in the BN.

- Generate N samples from joint distbtⁿ
- Count the # samples $N(q,e)$

$$N(q,e) = \text{count}(Q=q, E=e)$$

where $Q=q$ and $E=e$

- Count the # samples $N(e)$

$$N(e) = \text{count}(E=e)$$

where $E=e$

Estimate $P(Q=q \mid E=e) \approx \dfrac{N(q,e)}{N(e)}$

with $N(q,e) \leq N(e) \leq N$

Converges as $N \to \infty$ to correct answer

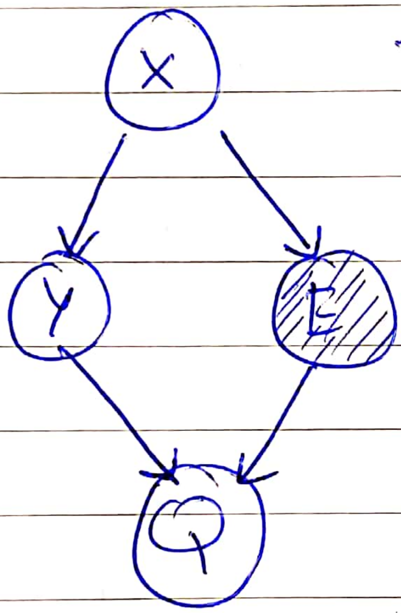
Inefficient!

- Discards all samples without $E=e$
- Takes many samples to converge when evidence is rare

2) Likelihood Weighting

- Instantiate evidence nodes instead of sampling them.
- Weight each sample using CPTs at evidence nodes.

ex.



To estimate $P(Q=q \mid E=e)$, draw samples $\{x_i, y_i, q_i\}$ $i = 1$ to N

Do N times:

Sample $x_i \sim P(X)$

Sample $y_i \sim P(Y \mid X = x_i)$

fix $E = e$

sample $q_i \sim P(Q \mid Y = y_i, E = e)$

Define indicator function

$$I(q, q') = \begin{cases} 1 & \text{if } q = q' \\ 0 & \text{if } q \neq q' \end{cases}$$

$$\overbrace{}^{\text{Likelihood weight.}}$$

Estimate

$$P(Q=q \mid E=e) = \frac{\sum_{i=1}^{N} I(q, q_i)\,P(E=e \mid X=x_i)}{\sum_{i=1}^{N} P(E=e_i \mid X=x_i)}$$

Converges.

Compare to rejection sampling

Do N times:

$$x_i \sim P(x)$$

$$y_i \sim P(Y \mid X=x_i)$$

$$e_i \sim P(E \mid X=x_i)$$

$$q_i \sim P(Q \mid X=x_i, E=e_i)$$

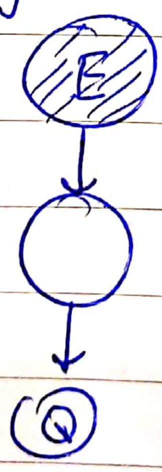
estimate. $P(Q=q \mid E=e) = \dfrac{N(q,e)}{N(e)}$

$$= \frac{\sum_{i=1}^{N} I(q, q_i)\,\underbrace{I(e, e_i)}}{\sum_{i=1}^{N} \underbrace{I(e, e_i)}}$$

* Note: In larger BNs, with multiple evidence nodes, the likelihood weights multiply in both numerator and denominator of ratio.

* LW much faster than rejection sampling.
  - Uses all samples to improve estimates.
  - But still slow for rare evidence (especially when evidence is descended from query nodes)

Good case LW



Bad case LW