Review

- d-separation

When is \[ \begin{cases} P(Y \mid E, X) = P(Y \mid E) \\ P(X \mid E, Y) = P(X \mid E) \\ P(X, Y \mid E) = P(X \mid E)P(Y \mid E) \end{cases} \]?

True if all paths from X to Y are "blocked."

A path is blocked if it has a node Z that satisfies 1, 2, or 3:

1) \( Z \notin E \rightarrow Z \rightarrow \) intervening cause

2) \( Z \in E \leftarrow Z \rightarrow \) common cause

3) \( Z \notin E \rightarrow Y \leftarrow \text{desc}(Z) \notin E \) no observed common effect

- Markov blanket \( B_x \) of node X
Jconsists of parents, children, spouses of X.

Thm: \( P(X \mid B_x, Y) = P(X \mid B_x) \)

where \( Y \notin \{X, B_x\} \)

Proof:

For any \( Y \notin \{B_x, X\} \), the undirected path from Y to X must pass thru \( B_x \).

There are 5 cases of paths to consider:

1) from parent of parent \( P \) of node X
   blocked by \( P \) (I d-sep)

2) from child of parent \( P \)
   blocked by \( P \) (II)

3) from parent of spouse \( S \)
   blocked by \( S \) (III)

4) from child of spouse \( S \)
   blocked by \( S \) (IV)

5) from child of child \( C \)
   blocked by \( C \) (V)

All paths are blocked from Y to X:
\[ \Rightarrow P(X \mid Y \mid B_x) = P(X \mid B_x)P(Y \mid B_x) \]
Inference

• Problem
  \( E = \) set of evidence nodes
  \( Q = \) set of query nodes
  How to compute posterior probs \( P(Q|E) \)?

• Types of inference

  - diagnostic reasoning
    (from effects to causes)
    Ex: \( P(B=1|M=1) \)
  - causal reasoning
    (from causes to effects)
    Ex: \( P(M=1|B=1) \)
  - inter-causal reasoning
    (explaining away)
    Ex: \( P(B=1|A=1,E=1) \)

Q: When can we perform inference efficiently?
  (Polynomial time in size of DAG and CPTs)

A: Poly-trees

Def: poly-tree = singly connected networks at most one undirected path between any two nodes; no loops.

• Goal: compute \( P(X|E) \)

  boxes don't overlap:
  no loops in poly-trees!
Types of evidence

- $E^+_x$: evidence above $X$, connected thru parents
- $E^-_x$: evidence below $X$, connected thru children

$E = E^+_x \cup E^-_x$

Assume $X \notin E$; otherwise inference is trivial

"causal" reasoning from upstream evidence (only)

How to compute $P(X \mid E^+_x)$?

General strategy: recursion.

Upstream recursion

$$P(X \mid E^+_x) = \sum_{\tilde{u}} P(X, \tilde{u} = \tilde{u} \mid E^+_x) \text{ marginalization over parents}$$

$$= \sum_{\tilde{u}} P(X \mid \tilde{u} = \tilde{u}, E^+_x) P(\tilde{u} = \tilde{u} \mid E^+_x) \text{ conditionalized product rule}$$

$$= \sum_{\tilde{u}} P(X \mid \tilde{u} = \tilde{u}) P(\tilde{u} = \tilde{u} \mid E^+_x) \text{ d-separation case I or II}$$

$$= \sum_{\tilde{u}} P(X \mid \tilde{u} = \tilde{u}) \prod_{i=1}^{m} P(U_i = u_i \mid E^+_x) \text{ d-separation, case III (X is unobserved common effect)}$$

Let $E_{U_i \setminus X}$ = evidence connected to $U_i$ except via path through $X$

$E_{U_i \setminus X}$ (in $i^{th}$ parent's box)

$$P(X \mid E^+_x) = \sum_{\tilde{u}} P(X \mid \tilde{u} = \tilde{u}) \prod_{i=1}^{m} P(U_i = u_i \mid E_{U_i \setminus X}) \text{ d-separation case III (X is unobserved common effect)}$$

$P(X \mid E^+_x)$ solve by recursing on parents
Reasoning from downstream evidence

How to compute $P(E \perp X | X)$?

It is possible (but complicated) to derive a downstream recursion.

**General case**: reasoning from $E = E^+_x \cup E^+_x$

$$P(X | E) = P(X | E^+_x, E^+_x) = \frac{P(X, E^+_x | E^+_x)}{P(E^+_x | E^+_x)} \text{ conditionalized product rule}$$

$$= \frac{P(X, E^+_x | E^+_x)}{\sum_x P(X=x, E^-_x | E^+_x)} \quad \text{marginalization: denominator is same computation as numerator, summed over different values of } x.$$ 

**Focus on numerator**:

$$P(X, E^+_x | E^+_x) = P(E^+_x | X, E^+_x) P(X | E^+_x) \text{ product rule}$$

$$= P(E^+_x | X) P(X | E^+_x) \text{ d-sep (I) }$$

$$= \text{downstream recursion} \quad \text{upstream recursion}$$

**Termination conditions**

- root node (no parents)
- leaf node (no children)
- evidence node (trivial)

**Running time**

linear in # nodes

size of CPTs remember $\{ \sum_u P(X | U=u) \} \ldots$
Loopy BNs

Ex: medical diagnosis
    2-layer network

Ex: simpler example

How to do exact inference?

Turn a loopy BN into a poly-tree.

Ex: node clustering

Merge nodes to form poly-tree

Merge $s_1, s_2, s_3$ into mega-node $S$.

Merge CPTs $P(s_1 | D), P(s_2 | D), P(s_3 | D)$ into $P(S | D)$.

Apply poly-tree algorithm.

size of mega-node = $2^3$
size of mega-CPT = $2^4$

Poly-tree algorithm linear in CPT size,
but CPT size grows exponentially with clustering.

How to choose optimal clustering?
computationally hard problem.