Probabilistic Graphical Models

Lecture 9 – Undirected Models

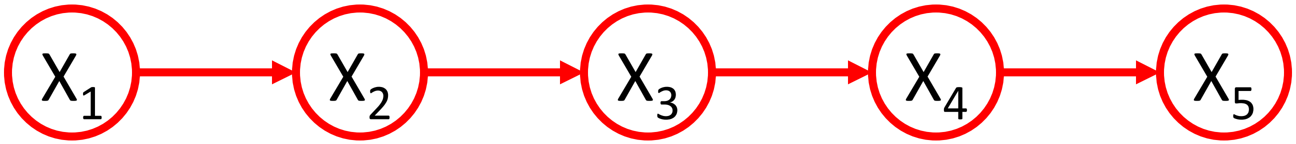
CS/CNS/EE 155
Andreas Krause

Announcements

- Homework 2 due next Wednesday (Nov 4) in class
  - Start early!!!
- Project milestones due Monday (Nov 9)
  - 4 pages of writeup, NIPS format
  - http://nips.cc/PaperInformation/StyleFiles

Best project award!!

Answering multiple queries



$$P(x_1, x_5) = P(x_1) \sum_{x_2} P(x_2 | x_1) \sum_{x_3} P(x_3 | x_2) \sum_{x_4} P(x_4 | x_3) P(x_5 | x_4)$$

Can compute in $O(n)$ operations $(+, *)$

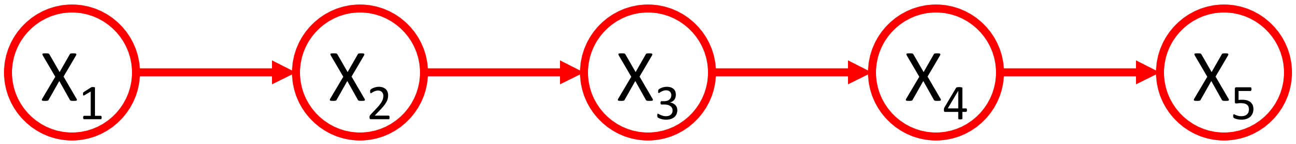
$P(x_2, x_5)$ costs $O(n)$ ops.

$P(x_i, x_5)$ — " $O(n)$ ops

If compute $P(x_i | x_j)$ $\forall i$ $\Rightarrow O(n^2)$

Can we reduce this to $O(n)$??

Reusing computation



$$P(x_1, x_5) = P(x_1) \sum_{x_2} P(x_2 | x_1) \underbrace{\sum_{x_3} P(x_3 | x_2) \underbrace{\sum_{x_4} P(x_4 | x_3) P(x_5 | x_4)}_{g_4(x_3, x_5)}}_{g_3(x_2, x_5)}$$

$$P(x_2, x_5) = \sum_{x_1} P(x_1) P(x_2 | x_1) \sum_{x_3} P(x_3 | x_2) \underbrace{\sum_{x_4} P(x_4 | x_3) P(x_5 | x_4)}_{g_4(x_3, x_5)}$$

already computed!

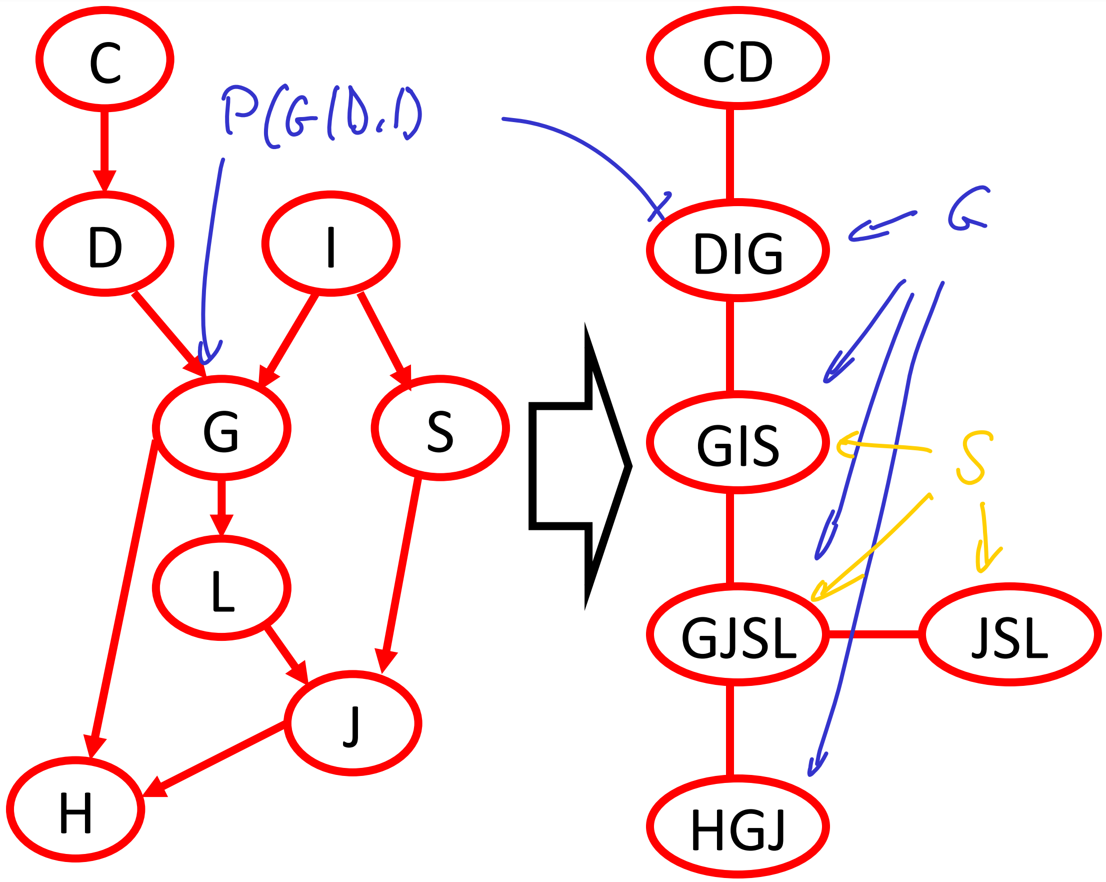
Want to "cache" our computations!

Next

- Will learn about algorithm for efficiently computing all marginals $P(X_i \mid \mathbf{E}=\mathbf{e})$ given fixed evidence **E=e**
- Need appropriate data structure for storing the computation

  ➔ Junction trees

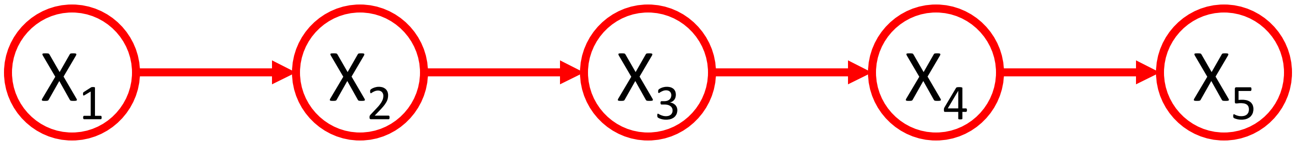
Junction trees



A junction tree for a collection of factors:

- A tree, where each node is a cluster of variables
- Every factor contained in some cluster C_i
- **Running intersection property**: If $X \in C_i$ and $X \in C_j$, and C_m is on the path between C_i and C_j, then $X \in C_m$

VE constructs a junction tree



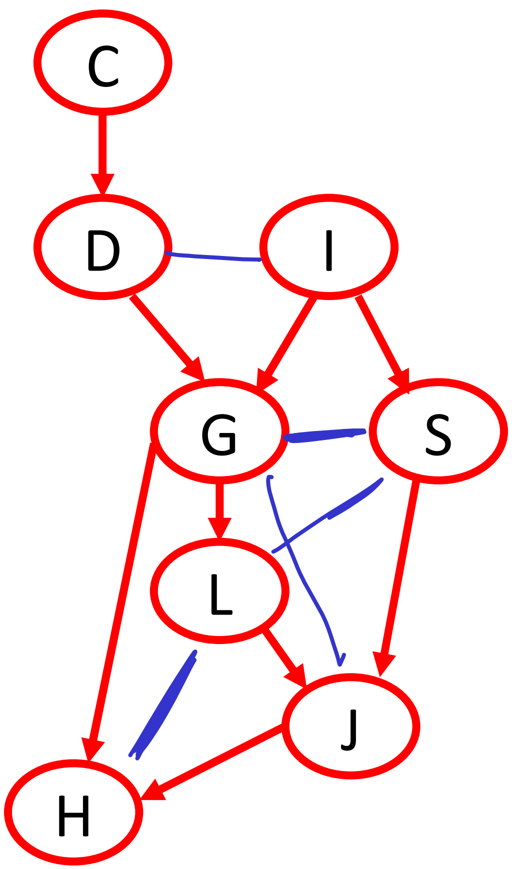
$P(X_1, X_5)$

Ordering 4, 3, 2

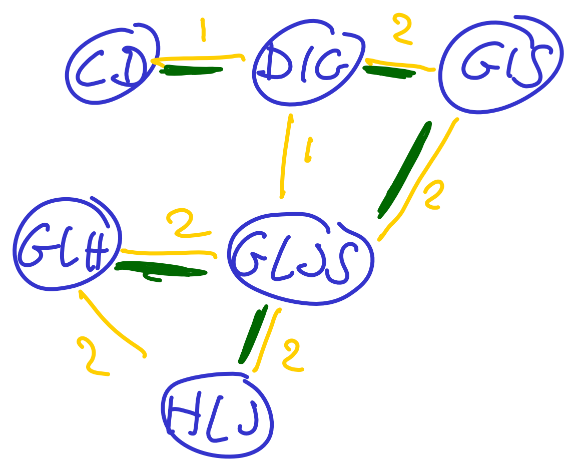
X_1, X_2 — $X_2\ X_3$ — $X_3\ X_4$ — X_4, X_5

- One clique C_i for each factor f_i created in VE
- C_i connected to C_j if f_i used to generate f_j
- Every factor used only once ➔ Tree
- **Theorem**: resulting tree satisfies RIP

Constructing JT from chordal graph

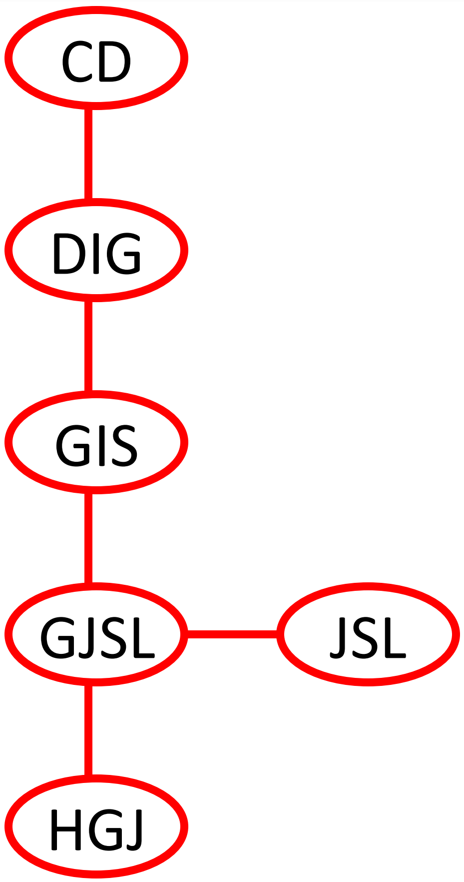


1. Moralize
2. Triangulate (make chordal)
3. Identify max. cliques
4. Connect cliques into undirected graph
   $w(C_i, C_j) = |C_i \cap C_j|$
5. Find Max ST



$\Rightarrow$ Results in valid junction tree

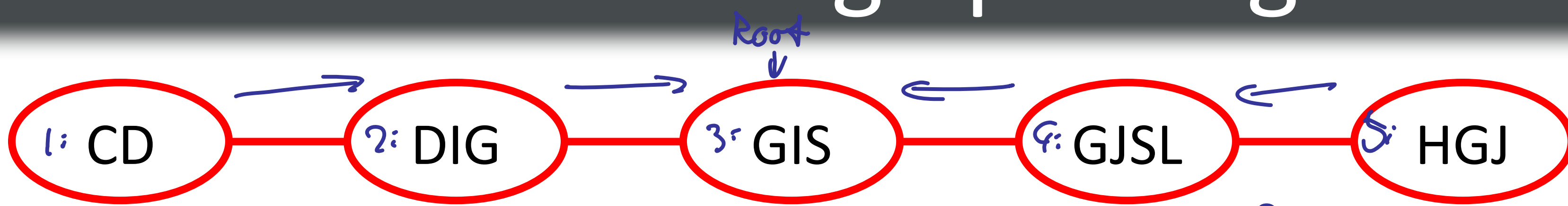
Junction trees and independence



Theorem:

- Suppose
  - T is a junction tree for graph G and factors F
  - Consider edge $\mathbf{C_i - C_j}$ with separator $\mathbf{S_{i,j}} = C_i \cap C_j$
  - Variables **X** and **Y** on opposite sites of separator
- Then $\mathbf{X} \perp \mathbf{Y} \mid \mathbf{S_{i,j}}$
- Furthermore, $I(T) \subseteq I(G)$

VE as message passing



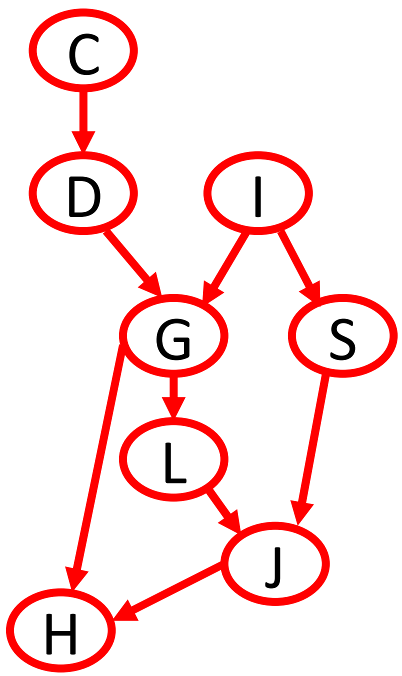
"Message": $\delta_{1\to2}(D) = \sum_{C} \pi(C,D)$

$\delta_{2\to3}(I,G) = \sum_{D} \pi(D,I,G)\cdot\delta_{1\to2}(D)$

$\delta_{4\to3}(JL) = \sum_{JL} \pi(GJSL)\,\delta_{5\to4}$

$\delta_{5\to4}(GJ) = \sum_{H} \pi(H,G,J)$

$$P(G) = \sum_{IS} \pi(GIS)\,\delta_{2\to3}(IG)\;\delta_{4\to3}(GS)$$



VE for computing X_i

- Pick root (any clique containing X_i)
- Don't eliminate, only send messages recursively from leaves to root
  - Multiply incoming messages with clique potential
  - Marginalize variables not in separator
- Root "ready" when received all messages

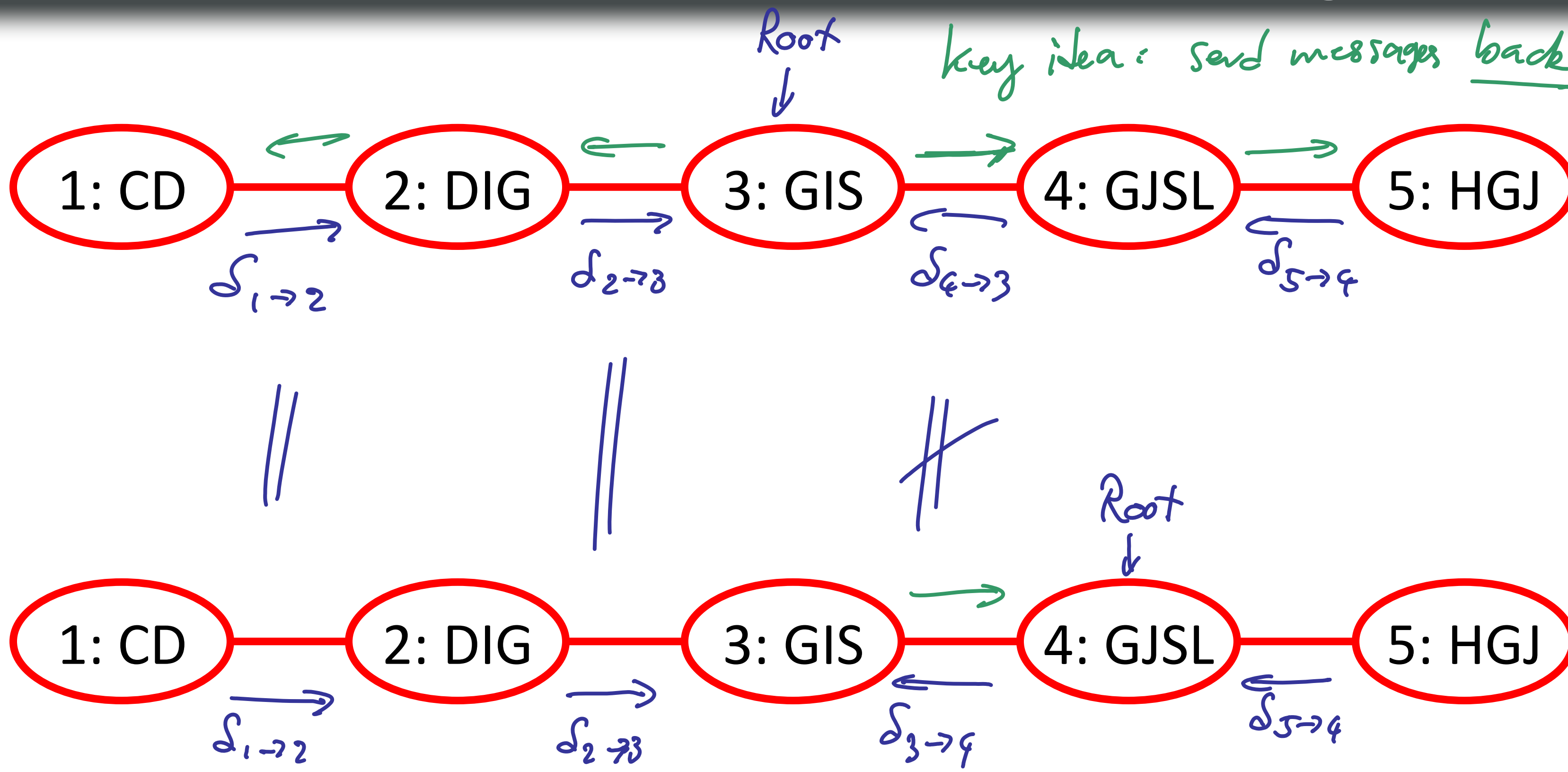
Correctness of message passing

- **Theorem**: When root ready (received all messages), all variables in root have correct potentials
  - Follows from correctness of VE

- So far, no gain in efficiency ☹

Does the choice of root affect messages?

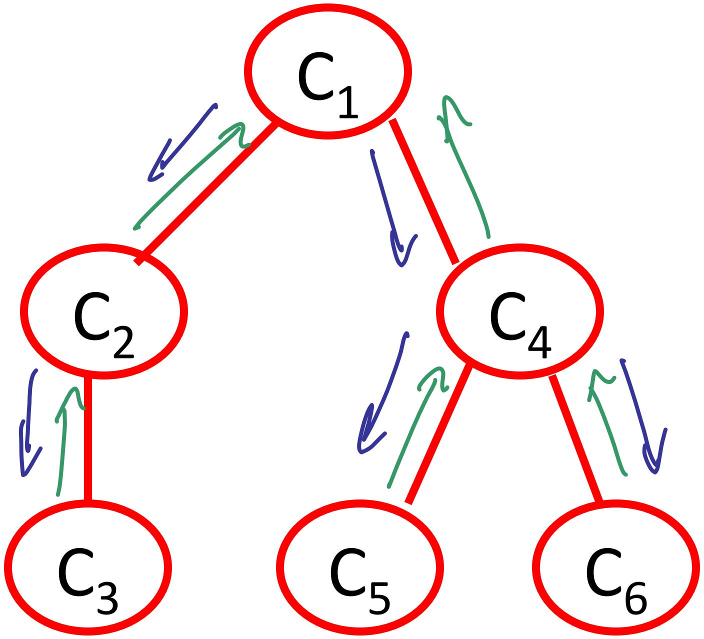
Key idea: send messages back



Instead of $O(n^2) \to O(n)$

1 message per edge per direction

Shenoy-Shafer algorithm



- Clique i ready if received messages from all neighbors but 1
  - Leaves always ready
- While there exists a message $\delta_{i \to j}$ ready to transmit send message

Complexity? $O(n\, 2^{\text{treewidth}})$

1 msg per edge per direction

"Only" exp. in treewidth

Theorem: At convergence, every clique has correct beliefs

Inference using VE

- Want to incorporate evidence E=e
- Multiply all cliques containing evidence variables with indicator potential 1_e

AB

$$1_{A=T}(a,b) = \begin{cases} 1 & \text{if } a=T \\ 0 & \text{if } a=F \end{cases}$$

- Perform variable elimination

Summary so far

- Junction trees represent distribution
  - Constructed using elimination order
  - Make complexity of inference explicitly visible
- Can implement variable elimination on junction trees to compute correct beliefs on all nodes

- Now:
  - **Belief propagation** – an important alternative to VE on junction trees.
  - Will later generalize to approximate inference!
  - Key difference: Messages obtained by division rather than multiplication

Message passing by factor division

- Variable elimination:
  - Message → Belief

$$\delta_{2\to3}(IG) = \sum_D \pi_2^{(0)}(DIG)\,\delta_{1\to2}(D)$$

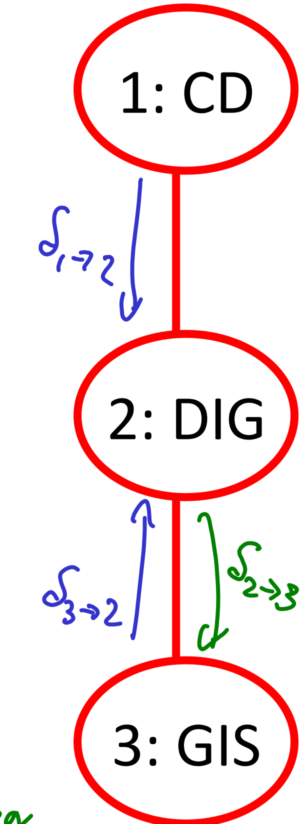
Belief at 3: $\pi_3^{(0)}(GIS)\cdot\delta_{2\to3}(IG)$

- Factor division:
  - Belief → Message

Belief at 2: $\pi_2^{(0)}(DIG)\cdot\delta_{1\to2}(D)\cdot\delta_{3\to2}(IG)$

Belief about Sep. IG: $\sigma_{2\to3}^{(t)}(IG) = \sum_D \pi_2^{(t)}(DIG)$ ← Send as msg to 3

Belief at 3: $\pi_3^{(t+1)}(GIS) = \pi_3^{(0)}(GIS)\cdot\dfrac{\sigma_{2\to3}^{(t)}(IG)}{\delta_{3\to2}(IG)} = \dfrac{\sum_D \pi_3^{(0)}(GIS)\,\pi_2^{(0)}\cdot\delta_{1\to2}\cancel{\delta_{3\to2}}}{\cancel{\delta_{3\to2}}}$



Factor division

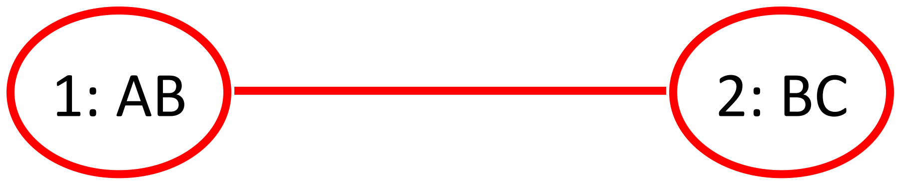
$f_1(A, B, C)$ $\qquad$ $f_2(B, C)$

$$f = \frac{f_1}{f_2} \qquad f(A, B, C) = \frac{f_1(A, B, C)}{f_2(B, C)}$$

with the convention that $\frac{0}{0} = 0$

Clique and separator potentials

$A \to B \to C$



$\Pi_1^{(0)}(AB) = P(A) \cdot P(B|A)$ $\qquad$ $\Pi_2^{(0)}(BC) = P(C|B)$

Belief about sep. B

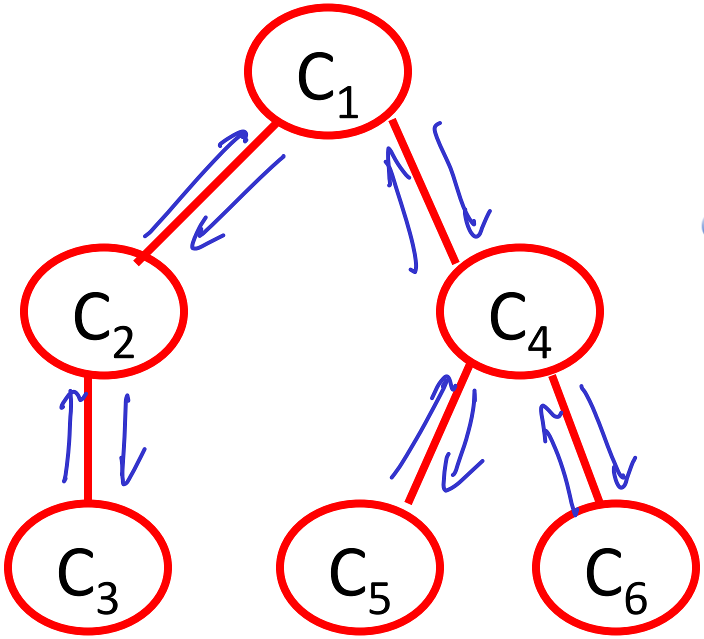
$$\sigma_{1\to 2}^{(0)}(B) = \sum_A \Pi_1^{(0)}(AB) \neq \sigma_{2\to 1}^{(0)}(B) = \sum_C \Pi_2^{(0)}(BC)$$

At convergence: $\Pi_1^{(*)}(AB) = P(AB)$ $\qquad$ $\Pi_2^{(*)} = P(BC)$

$$\sigma_{1\to 2}^{(*)}(B) = \sum_A \Pi_1^{(*)}(AB) = P(B) = \sigma_{2\to 1}^{(*)} = \mu_{12}(B)$$

Call such a JT with potentials $\Pi_1^{(*)} \dots \Pi_2^{(*)}$ that agree on separators "calibrated"

Lauritzen-Spiegelhalter algorithm (a.k.a. Belief Propagation)



- Initialize separator potentials μ_{ij}
  - One per edge, initialized to 1

    stores last msg across edge $i - j$

- Messages $i \to j$

$$\delta_{i \to j}(C_j \cap C_i) = \sum_{C_i \setminus C_j} \pi_i^{(t)}$$

$$\pi_j^{(t+1)} = \pi_j^{(t)} \cdot \frac{\delta_{i \to j}}{\mu_{ij}}$$

$$\mu_{ij} = \delta_{i \to j}$$

Correctness of Belief propagation

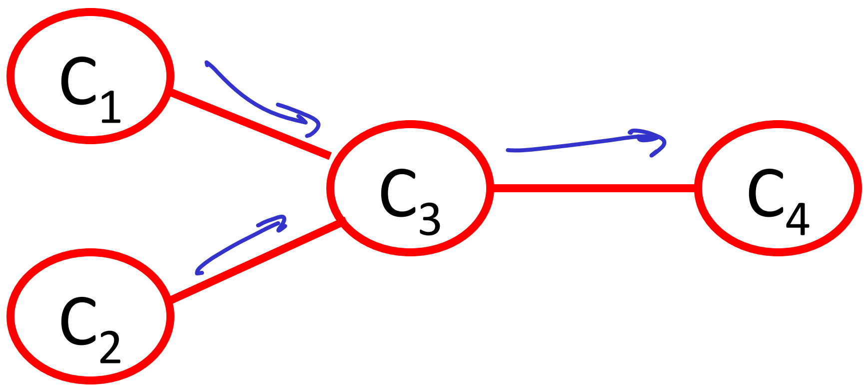
- Complexity linear in #cliques

- **Theorem**:
  At convergence, every clique has correct beliefs
  (when using correct message order, i.e., leaves to root and back)

  ➔ **Corollary**:
  Junction tree is calibrated (cliques agree on separator)

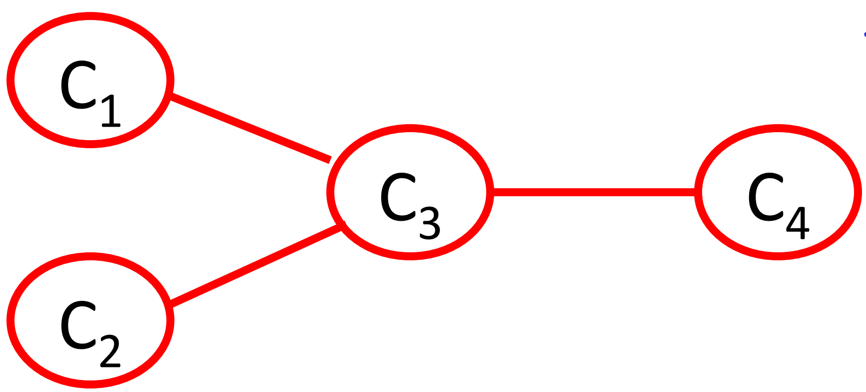
Comparison of VE and BP messages

- Variable elimination



$$\delta_{3\to4} = \sum_{C_3 \setminus C_4} \Pi_3^{(0)} \delta_{1\to3}\, \delta_{2\to3}$$

- Belief propagation



$$\sigma_{3\to4}^{(t)} = \sum_{C_3 \setminus C_4} \Pi_3^{(t)}$$

$$\Pi_4^{(t+1)} = \Pi_4^{(t)} \cdot \frac{\sigma_{3\to4}}{\mu_{34}}$$

$$\mu_{34}^{(t+1)} = \sigma_{3\to4}^{(t)}$$

Understanding BP

- Junction tree potential

$$\pi_T(X) = \frac{\prod_i \pi_{C_i}(C_i)}{\prod_{ij} \mu_{ij}(C_i \cap C_j)}$$

- Junction tree invariant

$$\pi_T(X) = P(X)$$

- Theorem: BP maintains Junction tree invariant

➔ BP reparametrizes clique and separator potentials

Advantages and disadvantages of junction tree inference

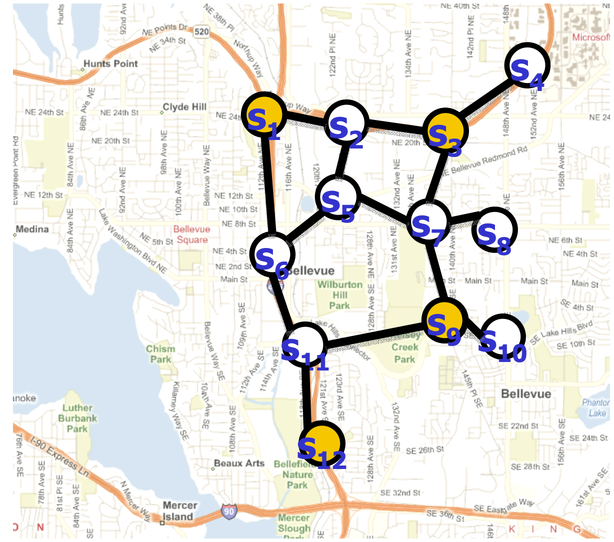
- Advantages
  - Can answer multiple queries (for same evidence) efficiently
  - Can perform incremental updates
- Disadvantages
  - No factors are “deleted”
  - Can’t prune away unnecessary variables
  - Slower for a single query

Summary so far

- Bayesian Networks
  - Representation
  - Learning (MLE / Bayesian) with fully observed data
  - Exact Inference
- Next
  - Undirected models
  - Approximate inference
  - Hidden variables

Representing the world using BNs

represent



True distribution P'
with cond. ind. I(P')

Bayes net (G,P)
with I(P)

- Want to make sure that $I(P) \subseteq I(P')$
- Ideally: $I(P) = I(P')$
- Want BN that **exactly** captures independencies in P'!

Perfect maps

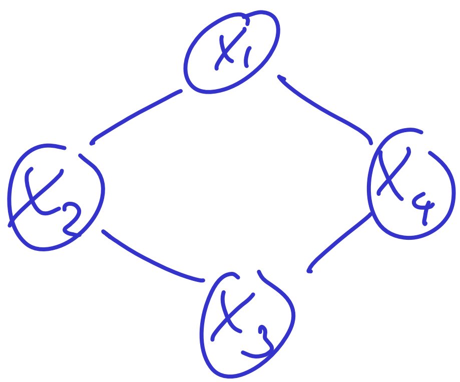
- Minimal I-maps are easy to find, but can contain many unnecessary dependencies.

- A BN structure G is called **P-map** (perfect map) for distribution P if **I(G) = I(P)**
- Does every distribution P have a P-map?

Existence of perfect maps

$X_1, \ldots, X_4$ $\quad$ $X_1 \perp X_3 \mid X_2 X_4$

$X_2 \perp X_4 \mid X_1, X_3$

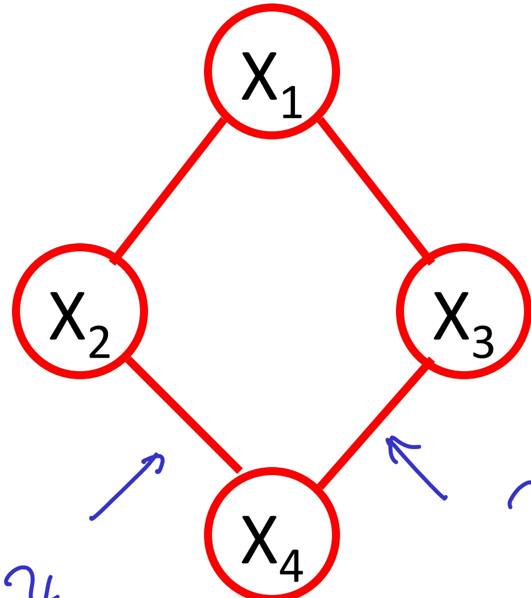


undirected GM is a P-map

but NO BN is P-map

- Will have undirected model as P-map

Undirected Parameterization



Specify factors over cliques in undirected graph

$\psi_{34}(x_3, x_4) \geq 0$

ψ_{24}

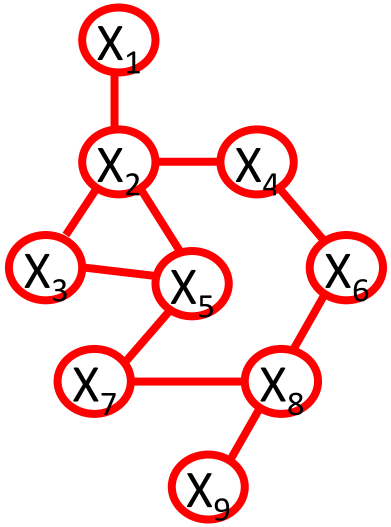
x_2	x_4	$\psi_{24}(x_2, x_4)$
0	0	3
0	1	17
1	0	.1
1	1	3.1415

$$P(x_1, \ldots x_4) = \frac{1}{Z} \psi_{12}(x_1, x_2)\, \psi_{13}(x_1, x_3)\, \psi_{24}\, \psi_{34}$$

$$Z = \sum_{x_1 \ldots x_4} \prod_{ij} \psi_{ij}(x_i, x_j)$$

Markov Networks
(a.k.a., Markov Random Field, Gibbs Distribution, …)

- A Markov Network consists of
  - An undirected graph, where each node represents a RV
  - A collection of factors defined over cliques in the graph
- Joint probability



$$P(X) = \frac{1}{Z} \prod_i \psi_i(C_i)$$

P

- A distribution factorizes over undirected graph G if

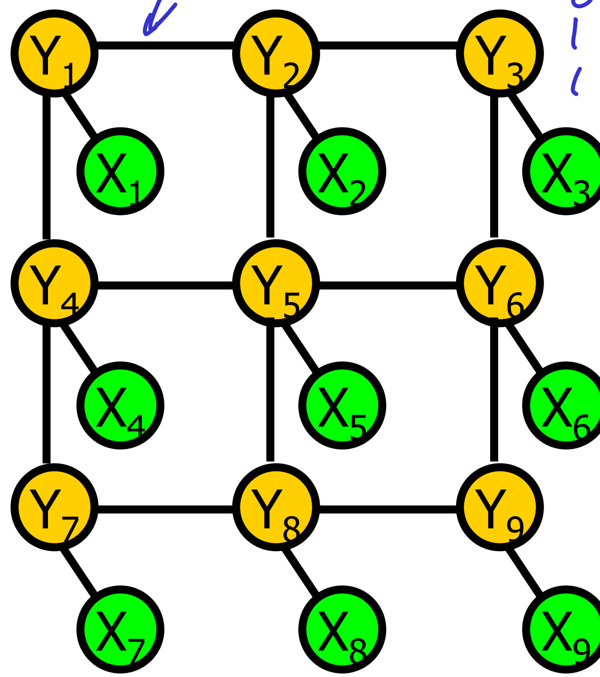
$\exists$ factors $\psi_1 \cdots \psi_k$ over cliques of G s.t.

$$P(X) = \frac{1}{Z} \prod_i \psi_i(C_i)$$

Example MN: Image denoising

Markov Network

$\psi_{12}(Y_1, Y_2)$



Y_1	Y_2	$\psi_{12}(Y_1, Y_2)$
0	0	5
0	1	1
1	0	1
1	1	5

X_i: noisy pixels

Y_i: "true" pixels

30

Computing Joint Probabilities

- Computing joint probabilities in BNs

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

$P(X_1 \mid X_n)$

actually comp. $P(X_1, X_n)$

- Computing joint probabilities in Markov Nets

Can do VE

$$P(X_1 \ldots X_n) = \frac{1}{Z} \prod_i \psi_i(C_i)$$

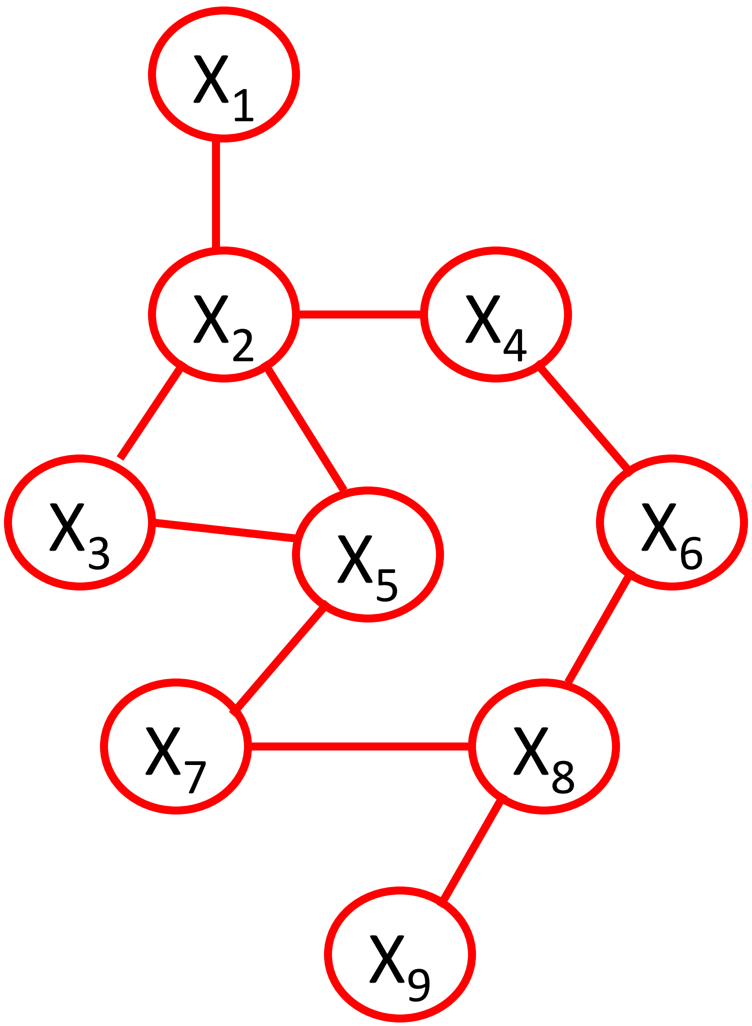
Need to know partition "function" Z

Can compute $$\frac{P(x_1 \ldots x_n)}{P(x_1' \ldots x_n')} = \frac{\prod_i \psi_i(C_i)}{\prod_i \psi_i(C_i')}$$

Independences in Markov Nets?

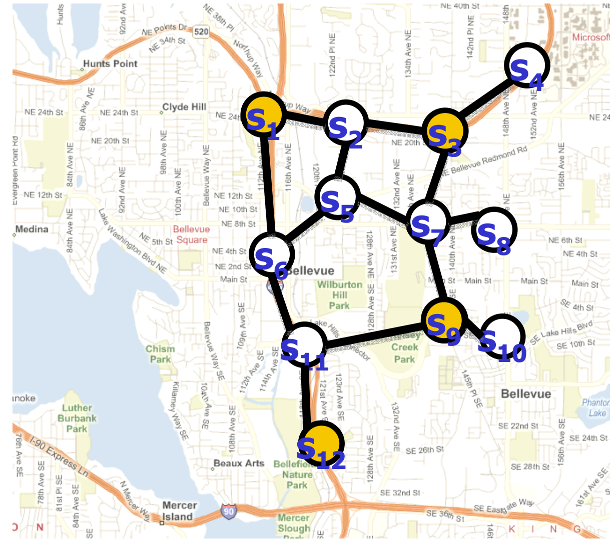
- In Bayes Nets (G,P)
  - Local Markov Assumption: $X \perp$ **NonDesc(X)** | $\mathbf{Pa_X}$
  - G is I-map for distribution P if Local Markov Assumption holds
  - Factorization Thm: P factorizes over G ⇔ G is an I-map
  - Global independences: d-separation
  - Completeness and soundness of d-separation
- How about Markov Nets?
  - What's the analog of the Local Markov Assumption?
  - Is there a factorization theorem for Markov Nets?
  - What replaces d-separation?

Local Markov Assumption for MN



- The **Markov Blanket MB(X)** of a node X is the set of neighbors of X
- Local Markov Assumption: X $\perp$ EverythingElse | MB(X)
- $I_{loc}(G)$ = set of all local independences
- G is called an I-map of distribution P if $I_{loc}(G) \subseteq I(P)$

Factorization Theorem for Markov Nets "→"



True distribution P
can be represented exactly as
a Markov net (G,P)

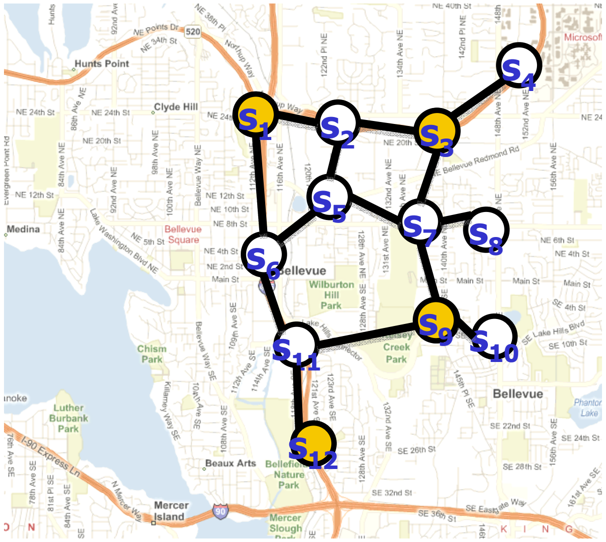
$$P(X_1, ..., X_n) = \frac{1}{Z} \prod_i \phi_i(\mathbf{C}_i)$$

⇒

$I_{loc}(G) \subseteq I(P)$

G is an **I-map** of P
(independence map)

Factorization Theorem for Markov Nets "←"



True distribution P can be represented exactly as

$$P(X_1, \ldots, X_n) = \prod_i P(X_i \mid \mathbf{Pa}_{X_i})$$

$I_{loc}(G) \subseteq I(P)$ $\not\Rightarrow$

Not true in general

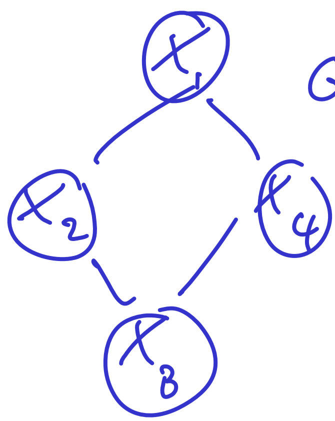
G is an **I-map** of P (independence map)

i.e., P can be represented as a Markov net (G,P)

Counterexample

- G an I-map for P does not imply that P factorizes
- Binary variables $X_1,\dots,X_4$.
- Only positive states
  (0,0,0,0), (1,0,0,0), (1,1,0,0), (1,1,1,0)
  (0,0,0,1), (0,0,1,1), (0,1,1,1), (1,1,1,1)

each happens with prob $\frac{1}{8}$



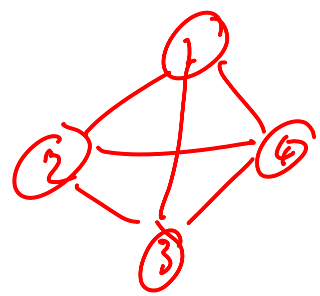
G

G is I-map for P

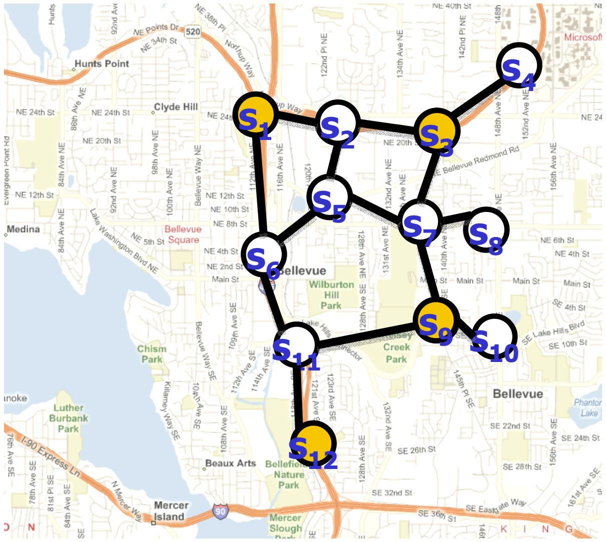
$X_1 \perp X_3 \mid X_2, X_4$

Eg.: $X_2 = 1, X_4 = 1$

But to represent P, need fully connected graph



Factorization Theorem for Markov Nets "$\Leftarrow$" Hammersley-Clifford Theorem



True distribution P
can be represented exactly as

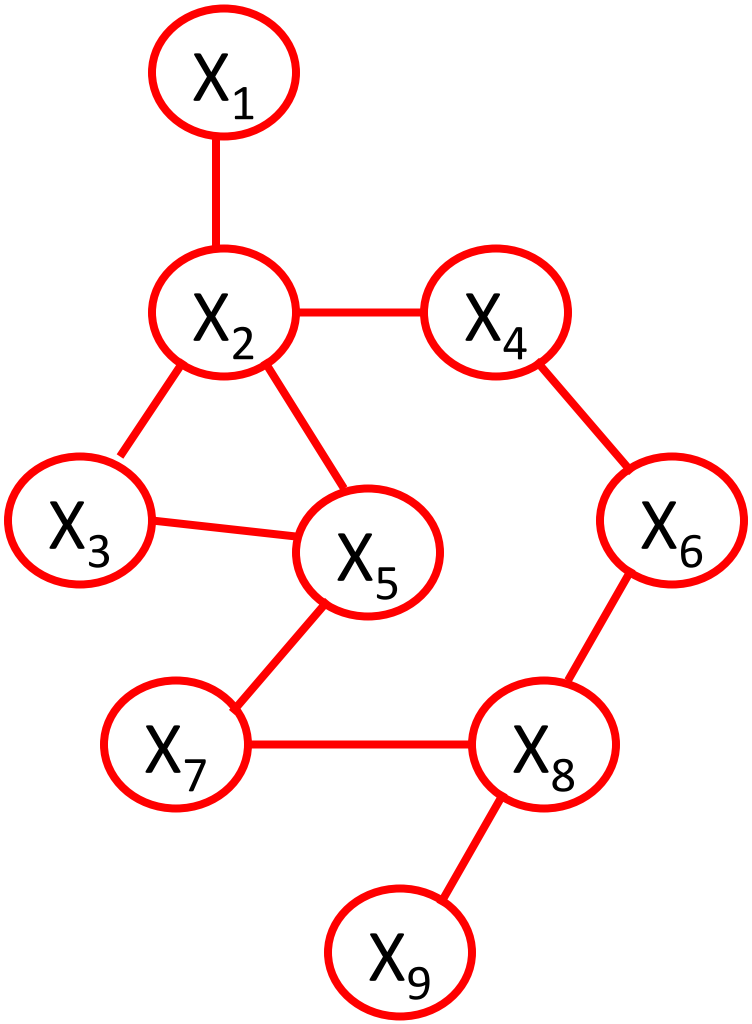
$I_{loc}(G) \subseteq I(P)$ $\Rightarrow$

$$P(X_1, ..., X_n) = \prod_i P(X_i \mid \mathbf{Pa}_{X_i})$$

G is an **I-map** of P
(independence map)
and P>0

i.e., P can be represented as
a Markov net (G,P)

Global independencies



- A trail $X—X_1—\ldots—X_m—Y$ is called active for evidence $\mathbf{E}$, if none of $X_1,\ldots,X_m \in \mathbf{E}$
- Variables X and Y are called **separated** by **E** if there is no active trail for **E** connecting X, Y
  Write sep(X,Y | **E**)

- $I(G) = \{X \perp Y \mid \mathbf{E}: \text{sep}(X,Y \mid \mathbf{E})\}$

Soundness of separation

- Know: For positive distributions P>0

  $I_{loc}(G) \subseteq I(P) \Leftrightarrow$ P factorizes over G

- **Theorem**: Soundness of separation

  For positive distributions P>0

  $I_{loc}(G) \subseteq I(P) \Leftrightarrow I(G) \subseteq I(P)$

- Hence, separation captures only true independences

- How about I(G) = I(P)?

Completeness of separation

Theorem: Completeness of separation

$$I(G) = I(P)$$

for "almost all" distributions P that factorize over G

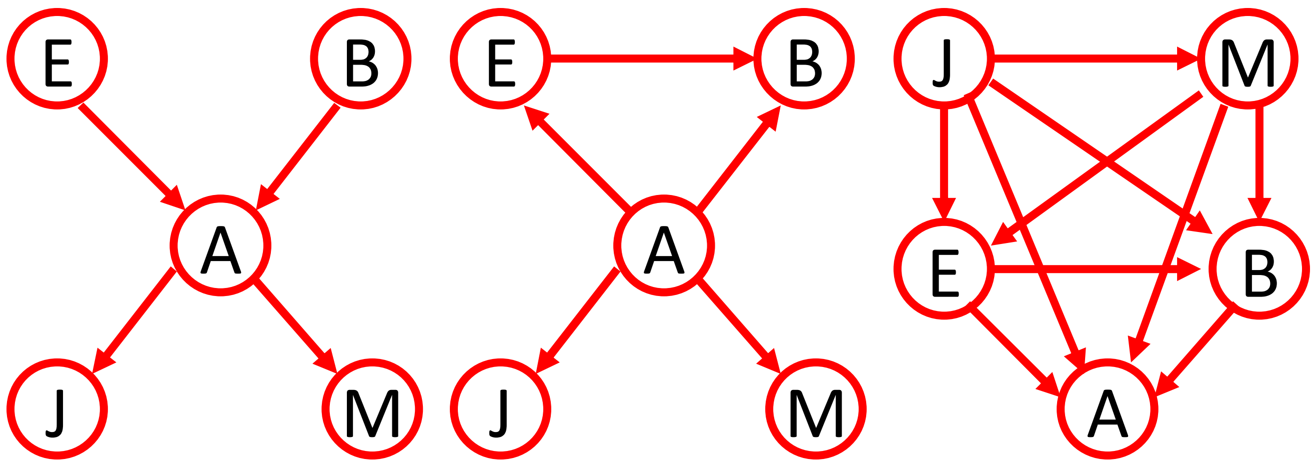
"almost all": Except for of potential parameterizations of measure 0 (assuming no finite set have positive measure)

Independences in Markov Nets?

- In Bayes Nets (G,P)
  - Local Markov Assumption: X $\perp$ **NonDesc(X)** | $\mathbf{Pa_X}$
  - G is I-map for distribution P if Local Markov Assumption holds
  - Factorization Thm: P factorizes over G $\Leftrightarrow$ G is an I-map
  - Global independences: d-separation
  - Completeness and soundness of d-separation
- How about Markov Nets?
  - Local Markov Assumption: X $\perp$ EverythingElse | MB(X)
  - Factorization Thm: For positive P, P factorizes $\Leftrightarrow$ G is an I-map
  - Global independences: separation
  - For positive P: separation is complete and sound
- How about minimal I-maps and P-maps??

Minimal I-maps

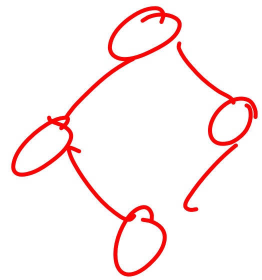
- For BNs: Minimal I-map not unique



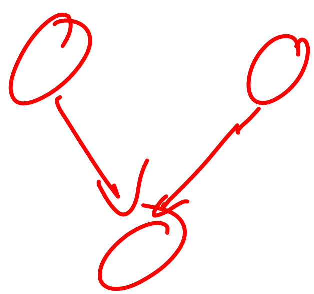
- For MNs: For positive P, minimal I-map is unique!!

P-maps

- Do P-maps always exist?
- For BNs: no



- How about Markov Nets?



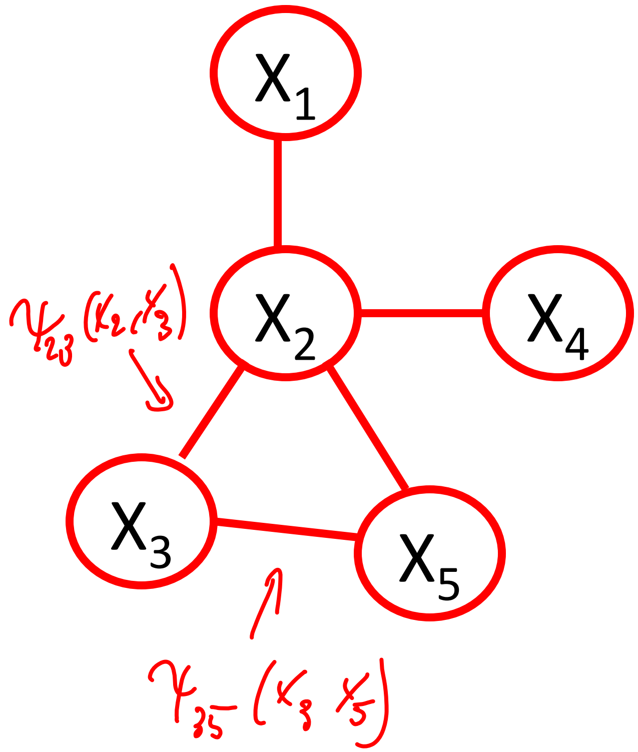
does not have MN P-map!

Exact inference in MNs

- Variable elimination and junction tree inference work exactly the same way!
  - Need to construct junction trees by obtaining chordal graph through triangulation

Pairwise MNs

- A pairwise MN is a MN where all factors are defined over single variables or pairs of variables
- Can reduce any MN to pairwise MN!



Tasks

- Read Koller & Friedman Chapters 10 and 4.1-4.5