Readings:
K&F: 9.1, 9.2, 9.3, 9.4
K&F: 5.1, 5.2, 5.3, 5.4, 5.5

Clique Trees 2

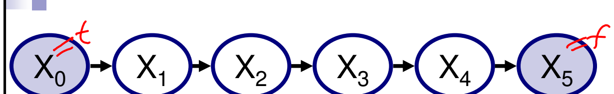
Undirected Graphical Models

Here the couples get to swing!

Graphical Models – 10708
Carlos Guestrin
Carnegie Mellon University

October 18th, 2006

What if I want to compute $P(X_i|x_0,x_{n+1})$ for each i?



Compute:

$$P(X_i \mid x_0 = t, x_{n+1} = t)$$

Variable elimination for each i? e.g., $X_1 \dots X_{i-1}, X_{i+1} \dots X_n$

eliminate X_1 $g_1(X_2) = \sum_{x_1} P(x_0) . P(X_1|x_0) . P(X_2|x_1)$

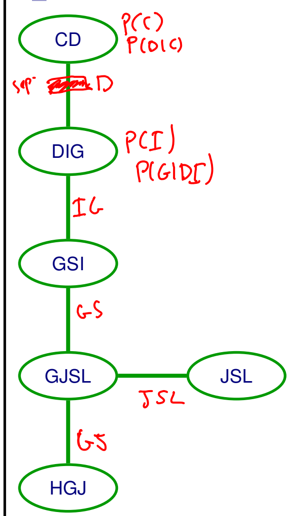
Complexity of $P(X_i | x_0, x_{n+1}) = O(n)$

Variable elimination for every i, what's the complexity?

run VE n times:

naive $O(n^2)$!

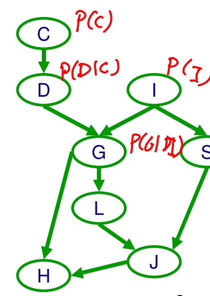
Cluster graph



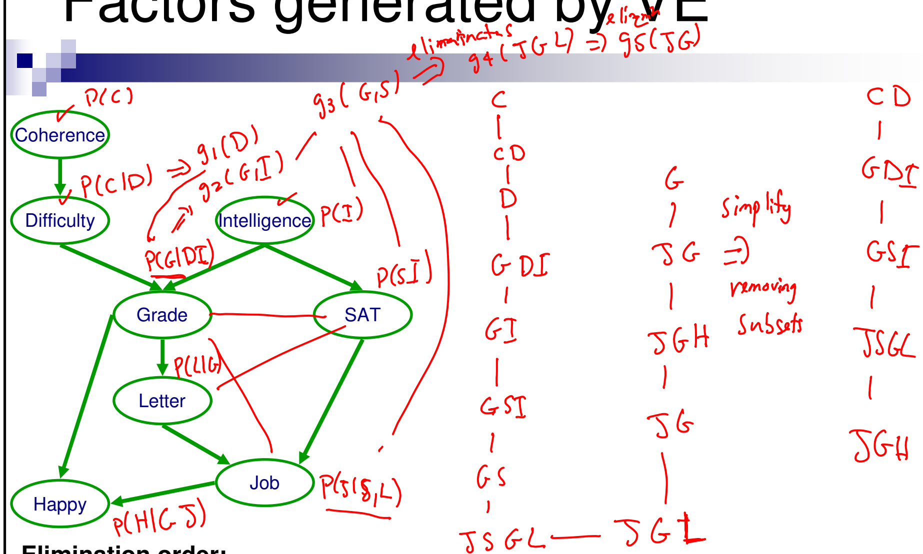
- **Cluster graph**: For set of factors *F*
  - Undirected graph
  - Each node i associated with a cluster $\mathbf{C}_i$
  - *Family preserving*: for each factor $f_j \in F$, $\exists$ node i such that scope$[f_j] \subseteq \mathbf{C}_i$
  - Each edge i – j is associated with a separator $\mathbf{S}_{ij} = \mathbf{C}_i \cap \mathbf{C}_j$

CPTs

covers

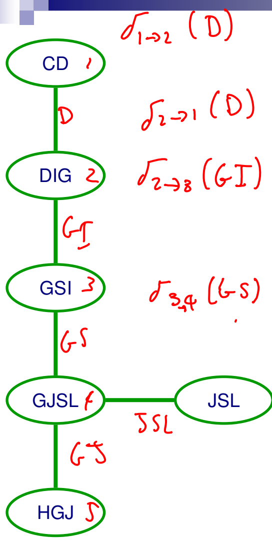


Factors generated by VE



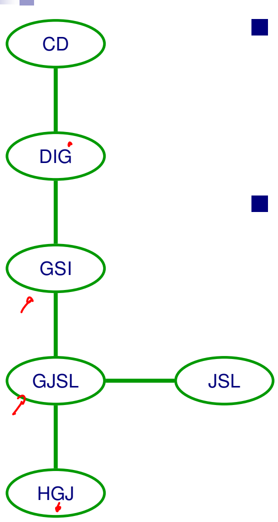
Elimination order:
{C,D,I,S,L,H,J,G}

Cluster graph for VE



- **VE generates cluster tree!**
  - One clique for each factor used/generated
  - Edge i – j, if f_i used to generate f_j
  - "Message" from i to j generated when marginalizing a variable from f_i
  - Tree because factors only used once
- **Proposition**:
  - "Message" δ_{ij} from i to j
  - Scope[δ_{ij}] $\subseteq$ $\mathbf{S}_{ij}$

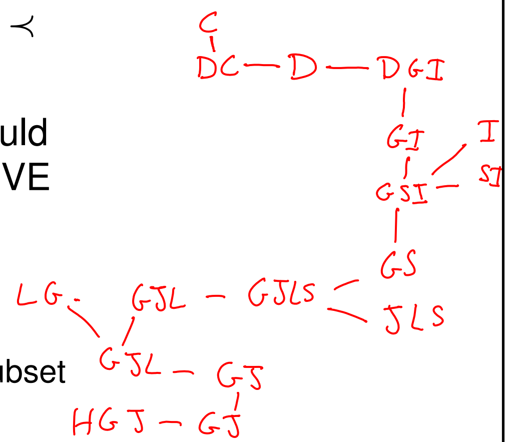
Running intersection property



- **Running intersection property (RIP)**
  - Cluster tree satisfies RIP if whenever X∈ $\mathbf{C}_i$ and X∈ $\mathbf{C}_j$ then X is in every cluster in the (unique) path from $\mathbf{C}_i$ to $\mathbf{C}_j$
- **Theorem**:
  - Cluster tree generated by VE satisfies RIP

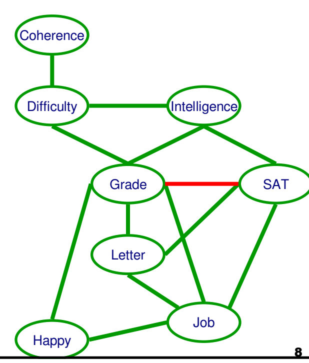
Constructing a clique tree from VE

- Select elimination order $\prec$
- Connect factors that would be generated if you run VE with order $\prec$
- Simplify!
  - Eliminate factor that is subset of neighbor

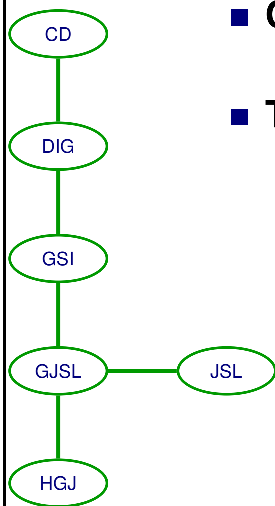


Find clique tree from chordal graph

- Triangulate moralized graph to obtain chordal graph
- Find maximal cliques
  - NP-complete in general
  - Easy for chordal graphs
  - Max-cardinality search
- Maximum spanning tree finds clique tree satisfying RIP!!!
  - Generate weighted graph over cliques
  - Edge weights (i,j) is separator size – $|\mathbf{C}_i \cap \mathbf{C}_j|$

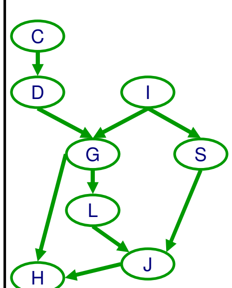


Clique tree & Independencies



- **Clique tree (or Junction tree)**
  - A cluster tree that satisfies the RIP
- **Theorem**:
  - Given some BN with structure *G* and factors *F*
  - For a clique tree *T* for *F* consider $\mathbf{C}_i$ – $\mathbf{C}_j$ with separator $\mathbf{S}_{ij}$:
    - **X** – any set of vars in $\mathbf{C}_i$ side of the tree
    - **Y** – any set of vars in $\mathbf{C}_j$ side of the tree
  - Then, $(\mathbf{X} \perp \mathbf{Y} \mid \mathbf{S}_{ij})$ in BN
  - Furthermore, $I(T) \subseteq I(G)$

Variable elimination in a clique tree 1

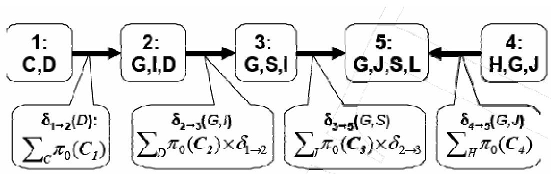


- **Clique tree for a BN**
  - Each CPT assigned to a clique
  - Initial potential $\pi_0(\mathbf{C}_i)$ is product of CPTs

Variable elimination in a clique tree 2

- **VE in clique tree to compute P(X_i)**
  - Pick a root (any node containing X_i)
  - Send messages recursively from leaves to root
    - Multiply incoming messages with initial potential
    - Marginalize vars that are not in separator
  - Clique *ready* if received messages from all neighbors

Belief from message

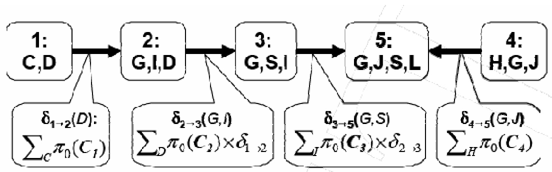


- **Theorem**: When **clique C_i is ready**
  - Received messages from all neighbors
  - Belief π_i(**C**$_i$) is product of initial factor with messages:

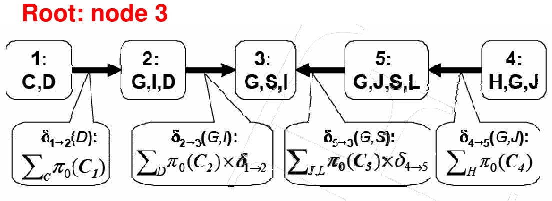
Choice of root

- **Message does not depend on root!!!**

Root: node 5

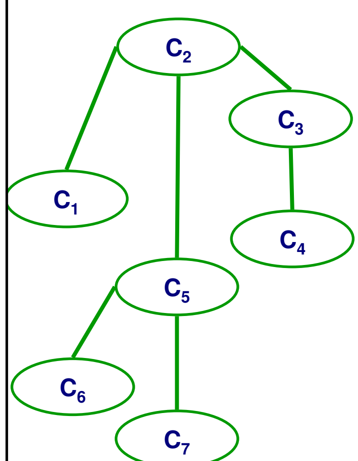


Root: node 3



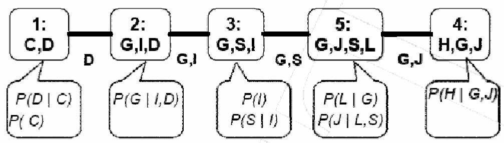
"Cache" computation: Obtain belief for all roots in linear time!!

Shafer-Shenoy Algorithm (a.k.a. VE in clique tree for all roots)



- Clique $\mathbf{C}_i$ *ready to transmit* to neighbor $\mathbf{C}_j$ if received messages from all neighbors but j
  - Leaves are always ready to transmit
- While $\exists$ $\mathbf{C}_i$ ready to transmit to $\mathbf{C}_j$
  - Send message $\delta_{i\to j}$
- Complexity: Linear in # cliques
  - One message sent each direction in each edge
- **Corollary**: At convergence
  - Every clique has correct belief

Calibrated Clique tree



- Initially, neighboring nodes don't agree on "distribution" over separators
- **Calibrated clique tree**:
  - At convergence, tree is *calibrated*
  - Neighboring nodes agree on distribution over separator

Answering queries with clique trees

- Query within clique
- Incremental updates – Observing evidence Z=z
  - Multiply some clique by indicator **1**(Z=z)
- Query outside clique
  - Use variable elimination!

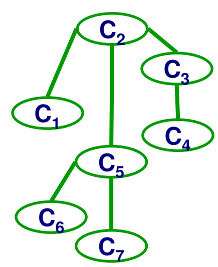
Message passing with division

- Computing messages by multiplication:

- Computing messages by division:

Lauritzen-Spiegelhalter Algorithm (a.k.a. belief propagation)

Simplified description see reading for details



- Initialize all separator potentials to 1
  - $\mu_{ij} \leftarrow 1$
- All messages ready to transmit
- While $\exists\ \delta_{i\rightarrow j}$ ready to transmit
  - $\mu_{ij}' \leftarrow$
  - If $\mu_{ij}' \neq \mu_{ij}$
    - $\delta_{i\rightarrow j} \leftarrow$
    - $\pi_j \leftarrow \pi_j \times \delta_{i\rightarrow j}$
    - $\mu_{ij} \leftarrow \mu_{ij}'$
    - $\forall$ neighbors k of j, k$\neq$ i, $\delta_{j\rightarrow k}$ ready to transmit
- Complexity: Linear in # cliques
  - for the “right” schedule over edges (leaves to root, then root to leaves)
- **Corollary**: At convergence, every clique has correct belief

VE versus BP in clique trees

- VE messages (the one that multiplies)

- BP messages (the one that divides)

Clique tree invariant

- **Clique tree potential**:
  - Product of clique potentials divided by separators potentials

- Clique tree **invariant**:
  - P(**X**) = π_T(**X**)

Belief propagation and clique tree invariant

- **Theorem**: Invariant is maintained by BP algorithm!
- BP reparameterizes clique potentials and separator potentials
  - At convergence, potentials and messages are marginal distributions

Subtree correctness

- **Informed message** from i to j, if all messages into i (other than from j) are informed
  - Recursive definition (leaves always send informed messages)
- **Informed subtree**:
  - All incoming messages informed
- **Theorem**:
  - Potential of connected informed subtree *T'* is marginal over scope[*T'*]
- **Corollary**:
  - At convergence, clique tree is *calibrated*
    - $\pi_i = P(\text{scope}[\pi_i])$
    - $\mu_{ij} = P(\text{scope}[\mu_{ij}])$

Clique trees versus VE

- Clique tree advantages
  - Multi-query settings
  - Incremental updates
  - Pre-computation makes complexity explicit

- Clique tree disadvantages
  - Space requirements – no factors are “deleted”
  - Slower for single query
  - Local structure in factors may be lost when they are multiplied together into initial clique potential

Clique tree summary

- Solve marginal queries for all variables in only twice the cost of query for one variable
- Cliques correspond to maximal cliques in induced graph
- Two message passing approaches
  - VE (the one that multiplies messages)
  - BP (the one that divides by old message)
- Clique tree invariant
  - Clique tree potential is always the same
  - We are only reparameterizing clique potentials
- Constructing clique tree for a BN
  - from elimination order
  - from triangulated (chordal) graph
- Running time (only) exponential in size of largest clique
  - Solve **exactly** problems with thousands (or millions, or more) of variables, and cliques with tens of nodes (or less)

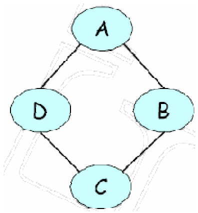
Announcements

- Recitation tomorrow, don't miss it!!!
  - Ajit on Junction Trees

Swinging Couples revisited

- This is no perfect map in BNs
- But, an undirected model will be a perfect map

Potentials (or Factors) in Swinging Couples



$\pi_1[A,B]$			$\pi_2[B,C]$			$\pi_3[C,D]$			$\pi_4[D,A]$		
a^0	b^0	30	b^0	c^0	100	c^0	d^0	1	d^0	a^0	100
a^0	b^1	5	b^0	c^1	1	c^0	d^1	100	d^0	a^1	1
a^1	b^0	1	b^1	c^0	1	c^1	d^0	100	d^1	a^0	1
a^1	b^1	10	b^1	c^1	100	c^1	d^1	1	d^1	a^1	100

Computing probabilities in Markov networks v. BNs

- In a BN, can compute prob. of an instantiation by multiplying CPTs
- In an Markov networks, can only compute ratio of probabilities directly

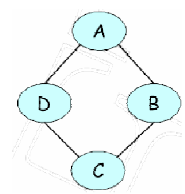
$\pi_1[A,B]$			$\pi_2[B,C]$			$\pi_3[C,D]$			$\pi_4[D,A]$		
a^0	b^0	30	b^0	c^0	100	c^0	d^0	1	d^0	a^0	100
a^0	b^1	5	b^0	c^1	1	c^0	d^1	100	d^0	a^1	1
a^1	b^0	1	b^1	c^0	1	c^1	d^0	100	d^1	a^0	1
a^1	b^1	10	b^1	c^1	100	c^1	d^1	1	d^1	a^1	100

Normalization for computing probabilities

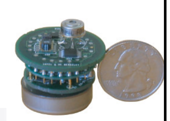
- To compute actual probabilities, must compute normalization constant (also called partition function)

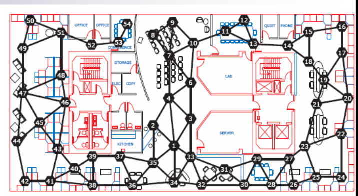
Assignment				Unnormalized	Normalized
a^0	b^0	c^0	d^0	300000	0.04
a^0	b^0	c^0	d^1	300000	0.04
a^0	b^0	c^1	d^0	300000	0.04
a^0	b^0	c^1	d^1	30	$4.1 \cdot 10^{-6}$
a^0	b^1	c^0	d^0	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^0	d^1	500	$6.9 \cdot 10^{-5}$
a^0	b^1	c^1	d^0	5000000	0.69
a^0	b^1	c^1	d^1	500	$6.9 \cdot 10^{-5}$
a^1	b^0	c^0	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^0	d^1	1000000	0.14
a^1	b^0	c^1	d^0	100	$1.4 \cdot 10^{-5}$
a^1	b^0	c^1	d^1	100	$1.4 \cdot 10^{-5}$
a^1	b^1	c^0	d^0	10	$1.4 \cdot 10^{-6}$
a^1	b^1	c^0	d^1	100000	0.014
a^1	b^1	c^1	d^0	100000	0.014
a^1	b^1	c^1	d^1	100000	0.014

- Computing partition function is hard! → Must sum over all possible assignments



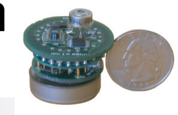
Factorization in Markov networks

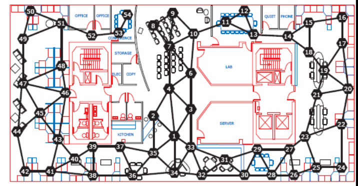




- Given an undirected graph H over variables $\mathbf{X}=\{X_1,...,X_n\}$
- A distribution P **factorizes** over H if ∃
  - subsets of variables $\mathbf{D}_1 \subseteq \mathbf{X}, \ldots, \mathbf{D}_m \subseteq \mathbf{X}$, such that the $\mathbf{D}_i$ are *fully connected* in H
  - *non-negative potentials* (or factors) $\pi_1(\mathbf{D}_1), \ldots, \pi_m(\mathbf{D}_m)$
    - also known as clique potentials
  - such that
- Also called Markov random field H, or Gibbs distribution over H

Global Markov assumption in Markov networks





- A path $X_1 - \ldots - X_k$ is **active** when set of variables **Z** are observed if none of $X_i \in \{X_1,\ldots,X_k\}$ are observed (are part of **Z**)
- Variables **X** are **separated** from **Y** given **Z** in graph *H*, $sep_H(\mathbf{X};\mathbf{Y}|\mathbf{Z})$, if there is no active path between any $X \in \mathbf{X}$ and any $Y \in \mathbf{Y}$ given **Z**
- The **global Markov assumption** for a Markov network *H* is

The BN Representation Theorem

If conditional independencies in BN are subset of conditional independencies in *P*

Obtain

Joint probability distribution:

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P\left(X_i \mid \mathbf{Pa}_{X_i}\right)$$

Important because:
Independencies are sufficient to obtain BN structure *G*

If joint probability distribution:

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P\left(X_i \mid \mathbf{Pa}_{X_i}\right)$$

Obtain

Then conditional independencies in BN are subset of conditional independencies in *P*

Important because:
Read independencies of *P* from BN structure *G*

Markov networks representation Theorem 1

If joint probability distribution *P*:

$$P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\pi_i(\mathbf{D}_i)$$

Then

H **is an I-map for *P***

- If you can write distribution as a normalized product of factors ⇒ Can read independencies from graph

What about the other direction for Markov networks ?

If *H* is an I-map for *P*

Then

joint probability distribution *P*:

$$P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\pi_i(\mathbf{D}_i)$$

- Counter-example: $X_1,\ldots,X_4$ are binary, and only eight assignments have positive probability: (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)
- For example, $X_1 \perp X_3 | X_2, X_4$:
- But distribution doesn't factorize!!!

Markov networks representation Theorem 2 (Hammersley-Clifford Theorem)

If H is an I-map for P
and
P is a positive distribution

Then

joint probability
distribution P:

$$P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\pi_i(\mathbf{D}_i)$$

- Positive distribution and independencies $\Rightarrow$ P factorizes over graph

Representation Theorem for Markov Networks

If joint probability
distribution P:

$$P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\pi_i(\mathbf{D}_i)$$

Then

H is an I-map for P

If H is an I-map for P
and
P is a positive distribution

Then

joint probability
distribution P:

$$P(X_1,\ldots,X_n) = \frac{1}{Z}\prod_{i=1}^{m}\pi_i(\mathbf{D}_i)$$

Completeness of separation in Markov networks

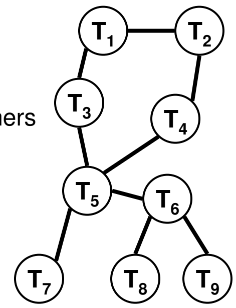
- **Theorem: Completeness of separation**
  - For "almost all" distributions that P factorize over Markov network H, we have that $I(H) = I(P)$
  - *"almost all" distributions*: except for a set of measure zero of parameterizations of the Potentials (assuming no finite set of parameterizations has positive measure)
- **Analogous to BNs**

What are the "local" independence assumptions for a Markov network?

- In a BN G:
  - local Markov assumption: variable independent of non-descendants given parents
  - d-separation defines global independence
  - Soundness: For all distributions:
- In a Markov net H:
  - **Separation** defines global independencies
  - What are the notions of local independencies?

Local independence assumptions for a Markov network

- **Separation** defines global independencies
- **Pairwise Markov Independence**:
  - Pairs of non-adjacent variables are independent given all others
- **Markov Blanket**:
  - Variable independent of rest given its neighbors



Equivalence of independencies in Markov networks

- **Soundness Theorem**: For all positive distributions *P*, the following three statements are equivalent:
  - *P* entails the global Markov assumptions
  - *P* entails the pairwise Markov assumptions
  - *P* entails the local Markov assumptions (Markov blanket)

Minimal I-maps and Markov Networks

- A fully connected graph is an I-map
- Remember minimal I-maps?
  - A "simplest" I-map → Deleting an edge makes it no longer an I-map
- In a BN, there is no unique minimal I-map
- Theorem: **In a Markov network, minimal I-map is unique!!**
- Many ways to find minimal I-map, e.g.,
  - Take pairwise Markov assumption:
  - If P doesn't entail it, add edge:

How about a perfect map?

- Remember perfect maps?
  - independencies in the graph are exactly the same as those in P
- For BNs, doesn't always exist
  - counter example: Swinging Couples
- How about for Markov networks?

Unifying properties of BNs and MNs

- BNs:
  - give you: V-structures, CPTs are conditional probabilities, can directly compute probability of full instantiation
  - but: require acyclicity, and thus no perfect map for swinging couples
- MNs:
  - give you: cycles, and perfect maps for swinging couples
  - but: don't have V-structures, cannot interpret potentials as probabilities, requires partition function
- Remember PDAGS???
  - skeleton + immoralities
  - provides a (somewhat) unified representation
  - see book for details

What you need to know so far about Markov networks

- Markov network representation:
  - undirected graph
  - potentials over cliques (or sub-cliques)
  - normalize to obtain probabilities
  - need partition function
- Representation Theorem for Markov networks
  - if P factorizes, then it's an I-map
  - if P is an I-map, only factorizes for positive distributions
- Independence in Markov nets:
  - active paths and separation
  - pairwise Markov and Markov blanket assumptions
  - equivalence for positive distributions
- Minimal I-maps in MNs are unique
- Perfect maps don't always exist