CS 3710 Advanced Topics in AI
Lecture 10

Review of exact inference methods

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Markov random fields

- **Probabilistic models with symmetric dependences:**
  - **Full joint for the variables defined as:**

$$P(\mathbf{x}) = \frac{1}{Z} \prod_{c \in Factors\ (X)} f_c(\mathbf{x}_c)$$

$f_c(\mathbf{x}_c)$ - A potential function (defined over factors)

$$Z = \sum_{x \in \{x\}} \prod_{c \in Factors\ (X)} f_c(\mathbf{x}_c)$$ - A partition function

$$P(x) = \frac{1}{Z} \exp\left(- \sum_{c \in cl(x)} \phi_c(x_c) \right)$$

- Gibbs (Boltzman) distribution

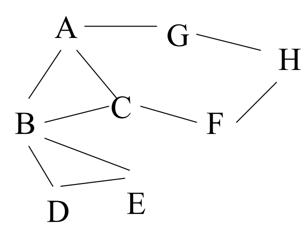
Graphical representation of MRFs

MRF representation:

- **An undirected network (also called independence graph)**
- **Variables in factors are represented by cliques**

Example:

- variables A,B ..H
- Assume the full joint of MRF



$$P(A,B,\ldots H) = \phi_1(A,B,C)\phi_2(B,D,E)\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

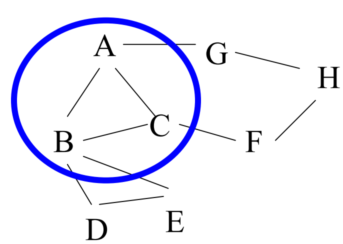
Graphical representation of MRFs

MRF representation:

- **An undirected network (also called independence graph)**
- **Variables in factors are represented by cliques**

Example:

- variables A,B ..H
- Assume the full joint of MRF

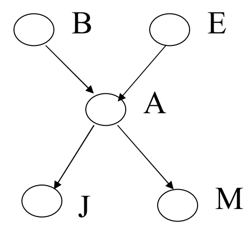


$$P(A,B,\ldots H) = \phi_1(A,B,C)\phi_2(B,D,E)\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

Bayesian belief networks

Two components:

- **Directed acyclic graph**
  - Nodes correspond to random variables
  - (Missing) links encode independences



- **Parameters**
  - Local conditional probability distributions for every variable-parent configuration

    $$\mathbf{P}(X_i \mid pa(X_i))$$

  Where:

  $pa(X_i)$ - stand for parents of X_i

P(A|B,E)

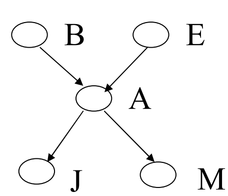
B	E	**T**	**F**
T	T	0.95	0.05
T	F	0.94	0.06
F	T	0.29	0.71
F	F	0.001	0.999

Bayesian Belief Networks

Full joint distribution is defined in terms of local conditional distributions:

$$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

Example:



$$P(B,E,A,J,M) =$$
$$P(B)P(E)P(A \mid B,E)P(J \mid A)P(M \mid A)$$

Conversion of BBNs to MRFs

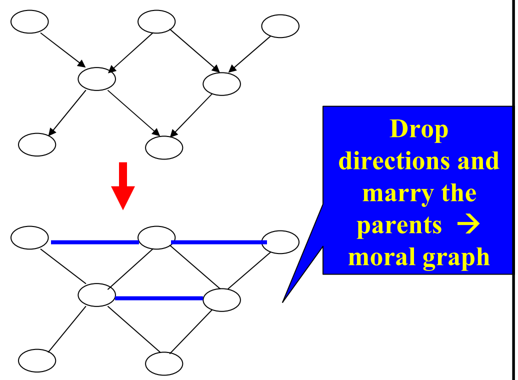
BBN: $$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \mathbf{P}(X_i \mid pa(X_i))$$

MRF: $$\mathbf{P}(X_1, X_2, .., X_n) = \prod_{i=1,..n} \phi_i(X_i, pa(X_i))$$

Graphically:

Directed graph

Undirected graph



Factors

- **Factor:** is a function that maps value assignments for a subset of random variables to $\Re$ (reals)
- **The scope of the factor:**
  - a set of variables defining the factor
- **Example:**
  - Assume discrete random variables x (with values a1,a2, a3) and y (with values b1 and b2)
  - Factor:

$$\phi(x, y)$$

a1	b1	0.5
a1	b2	0.2
a2	b1	0.1
a2	b2	0.3
a3	b1	0.2
a3	b2	0.4

  - Scope of the factor:

$$\{x, y\}$$

Factor Product

$$\phi_1(x,y)\phi_2(y,z) = \tau(x,y,z)$$

a1	b1	0.5
a1	b2	0.2
a2	b1	0.1
a2	b2	0.3
a3	b1	0.2
a3	b2	0.4

•

b1	c1	0.1
b1	c2	0.6
b2	c1	0.3
b2	c2	0.4

=

a1	b1	c1	0.5*0.1
a1	b1	c2	0.5*0.6
a1	b2	c1	0.2*0.3
a1	b2	c2	0.2*0.4
a2	b1	c1	0.1*0.1
a2	b1	c2	0.1*0.6
a2	b2	c1	0.3*0.3
a2	b2	c2	0.3*0.4
a3	b1	c1	0.2*0.1
a3	b1	c2	0.2*0.6
a3	b2	c1	0.4*0.3
a3	b2	c2	0.4*0.4

Factor Product

$$\phi_1(x,y)\phi_2(y,z) = \tau(x,y,z)$$

a1	b1	0.5
a1	b2	0.2
a2	b1	0.1
a2	b2	0.3
a3	b1	0.2
a3	b2	0.4

•

b1	c1	0.1
b1	c2	0.6
b2	c1	0.3
b2	c2	0.4

=

a1	b1	c1	0.5*0.1
a1	b1	c2	0.5*0.6
a1	b2	c1	0.2*0.3
a1	b2	c2	0.2*0.4
a2	b1	c1	0.1*0.1
a2	b1	c2	0.1*0.6
a2	b2	c1	0.3*0.3
a2	b2	c2	0.3*0.4
a3	b1	c1	0.2*0.1
a3	b1	c2	0.2*0.6
a3	b2	c1	0.4*0.3
a3	b2	c2	0.4*0.4

Factor Sum (marginalization)

$$\sum_{y} \phi(x, y, z) = \tau(x, z)$$

$\sum$

a1	b1	c1	0.2
a1	b1	c2	0.35
a1	b2	c1	0.4
a1	b2	c2	0.15
a2	b1	c1	0.5
a2	b1	c2	0.1
a2	b2	c1	0.3
a2	b2	c2	0.2
a3	b1	c1	0.25
a3	b1	c2	0.45
a3	b2	c1	0.15
a3	b2	c2	0.25

=

a1	c1	0.6
a1	c2	0.5
a2	c1	0.8
a2	c2	0.3
a3	c1	0.4
a3	c2	0.7

Factor Sum (marginalization)

$$\sum_{y} \phi(x, y, z) = \tau(x, z)$$

$\sum$

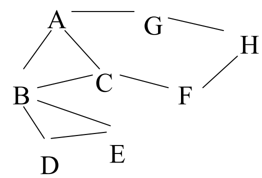
a1	b1	c1	0.2
a1	b1	c2	0.35
a1	b2	c1	0.4
a1	b2	c2	0.15
a2	b1	c1	0.5
a2	b1	c2	0.1
a2	b2	c1	0.3
a2	b2	c2	0.2
a3	b1	c1	0.25
a3	b1	c2	0.45
a3	b2	c1	0.15
a3	b2	c2	0.25

=

a1	c1	0.6
a1	c2	0.5
a2	c1	0.8
a2	c2	0.3
a3	c1	0.4
a3	c2	0.7

MRF variable elimination inference

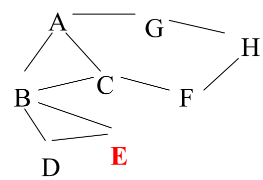
Example:



$$P(B) = \sum_{A,C,D,..H} P(A,B,\ldots H)$$

$$= \sum_{A,C,D,..H} \phi_1(A,B,C)\phi_2(B,D,E)\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

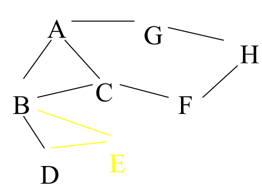
Eliminate E



$$= \sum_{A,C,D,F,G,H} \phi_1(A,B,C)\underbrace{\left[\sum_{E} \phi_2(B,D,E)\right]}_{\tau_1(B,D)}\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

MRF variable elimination inference

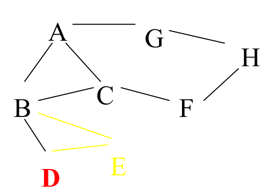
Example (cont):



$$P(B) = \sum_{A,C,D,..H} P(A,B,\ldots H)$$

$$= \sum_{A,C,D,F,G,H} \phi_1(A,B,C)\tau_1(B,D)\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

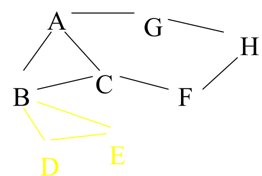
Eliminate D



$$= \sum_{A,C,F,G,H} \phi_1(A,B,C)\underbrace{\left[\sum_{D} \tau_1(B,D)\right]}_{\tau_2(B)}\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

MRF variable elimination inference

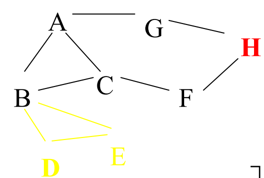
Example (cont):



$$P(B) = \sum_{A,C,D,..H} P(A,B,...H)$$

$$= \sum_{A,C,F,G,H} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

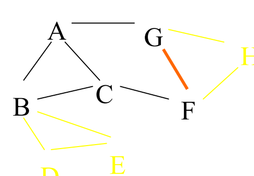
Eliminate H



$$= \sum_{A,C,F,G} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\phi_4(C,F)\underbrace{\left[\sum_{H} \underbrace{\phi_5(G,H)\phi_6(F,H)}_{\tau_3(F,G,H)}\right]}_{\tau_4(F,G)}$$

MRF variable elimination inference

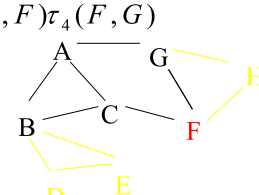
Example (cont):



$$P(B) = \sum_{A,C,D,..H} P(A,B,...H)$$

$$= \sum_{A,C,F,G} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\phi_4(C,F)\tau_4(F,G)$$

Eliminate F

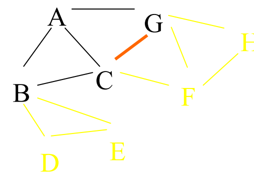


$$= \sum_{A,C,G} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\underbrace{\left[\sum_{F} \underbrace{\phi_4(C,F)\tau_4(F,G)}_{\tau_5(C,F,G)}\right]}_{\tau_6(G,C)}$$

MRF variable elimination inference

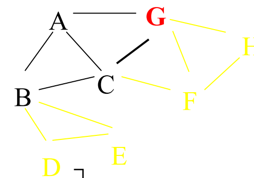
Example (cont):

$$P(B) = \sum_{A,C,D,..H} P(A,B,...H)$$



$$= \sum_{A,C,G} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\tau_6(C,G)$$

Eliminate G

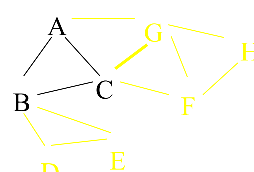


$$= \sum_{A,C} \phi_1(A,B,C)\tau_2(B)\underbrace{\left[\underbrace{\sum_{F} \phi_3(A,G)\tau_6(C,G)}_{\tau_7(A,C,G)}\right]}_{\tau_8(A,C)}$$

MRF variable elimination inference

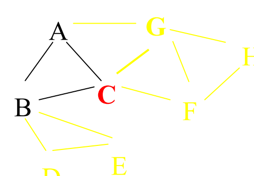
Example (cont):

$$P(B) = \sum_{A,C,D,..H} P(A,B,...H)$$



$$= \sum_{A,C} \phi_1(A,B,C)\tau_2(B)\tau_8(A,C)$$

Eliminate C



$$= \sum_{A} \tau_2(B)\underbrace{\left[\underbrace{\sum_{C} \phi_1(A,B,C)\tau_8(A,C)}_{\tau_9(A,B,C)}\right]}_{\tau_{10}(A,B)}$$

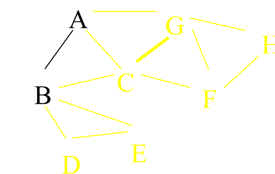
MRF variable elimination inference

Example (cont):

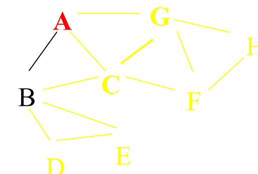
$$P(B) = \sum_{A,C,D,..H} P(A,B,...H)$$

$$= \sum_{A} \tau_2(B)\tau_{10}(A,B)$$

$$= \tau_2(B)\sum_{A} \tau_{10}(A,B)$$

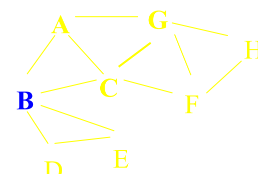


Eliminate A



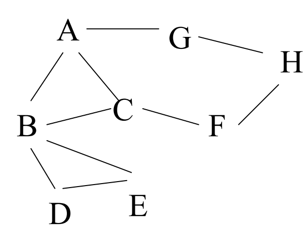
$$= \tau_2(B)\underbrace{\sum_{A} \tau_{10}(A,B)}_{\tau_{11}(B)}$$

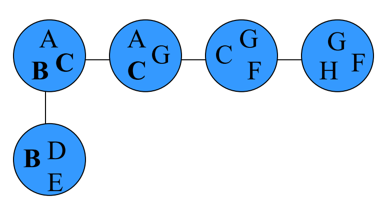
$$= \tau_2(B)\tau_{11}(B)$$



Tree decomposition of the graph

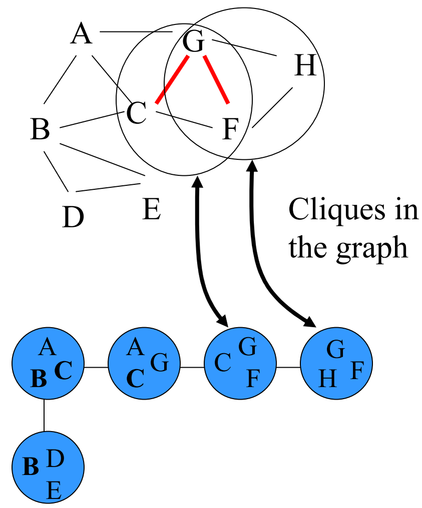
- **A tree decomposition of a graph G:**
  - A tree *T* with a vertex set associated to every node.
  - For all edges $\{v,w\} \in$ G: there is a set containing both *v* and *w* in *T*.
  - **Running intersection:** For every $v \in$ G : the nodes in T that contain *v* form a connected subtree.





Tree decomposition of the graph

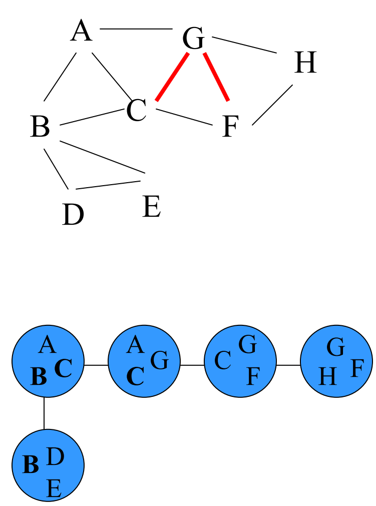
- **A tree decomposition of a graph G:**
  - A tree *T* with a vertex set associated to every node.
  - For all edges $\{v,w\} \in G$: there is a set containing both *v* and *w* in *T*.
  - **Running intersection:** For every $v \in G$: the nodes in T that contain *v* form a connected subtree.



Triangulation

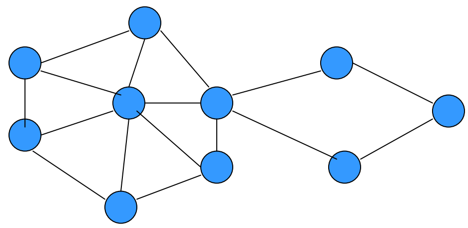
A way to build a tree decomposition T of a graph G

- Add undirected links to G so that cycles of 4 or more are broken
- Make cliques in the new G the clusters of the tree T



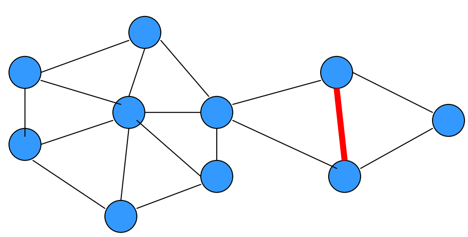
Triangulation

Is this graph triangulated?



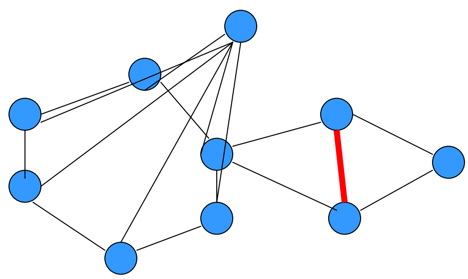
Triangulation

Is this graph triangulated?



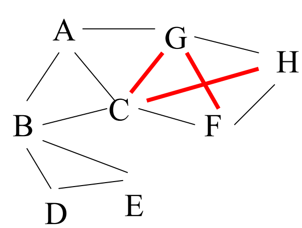
Triangulation

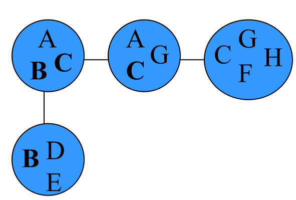
Is this graph triangulated?



Tree decomposition of the graph

- Many tree decompositions of a graph G exist



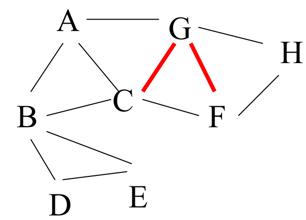


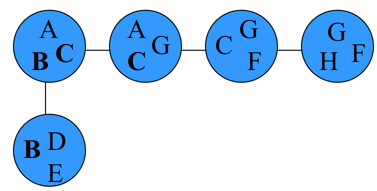
Treewidth of the graph

- **Width** of the tree decomposition:

  $$\max_{i \in I} |X_i| - 1$$

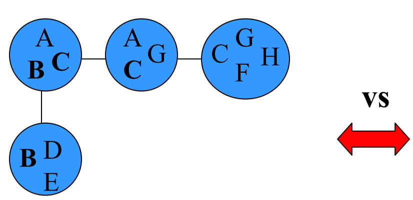
- **Treewidth** of a graph G: tw(G)= minimum width over all tree decompositions of G.

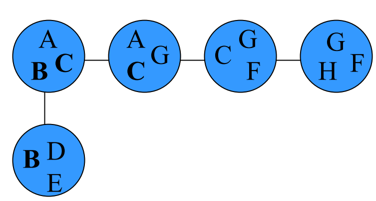




Treewidth of the graph

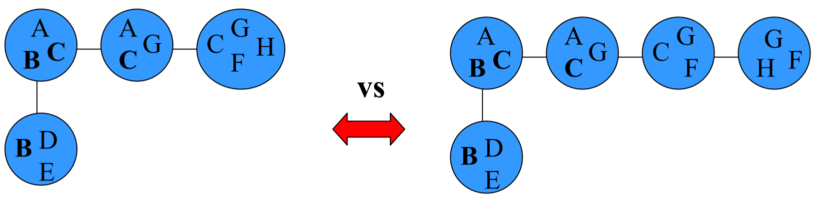
- **Why it matters?** The decomposition affects probabilistic calculations
- **Treewidth** gives the best case complexity
- **Caveat:** finding the best tree decomposition is NP-hard



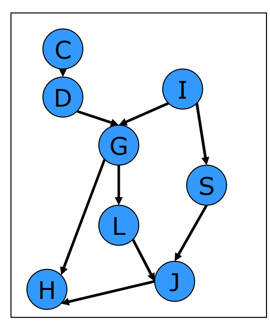


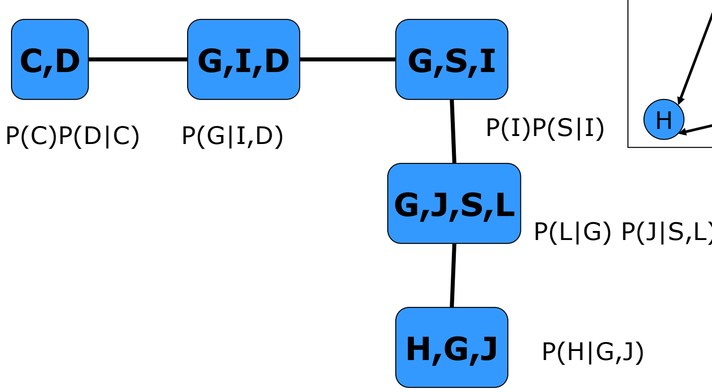
Variable elimination and tree decompositions

- Variable elimination on linear structures is easy
- Sum things out according to the tree structure
- Clique trees (or cluster graphs) introduce the elimination order



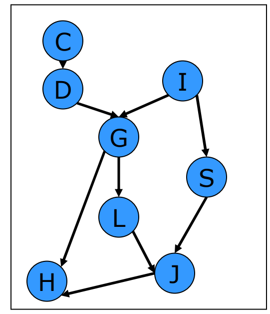
Clique trees

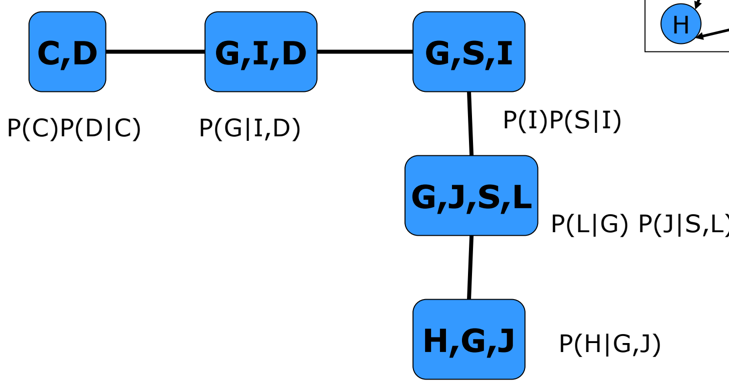




Clique trees

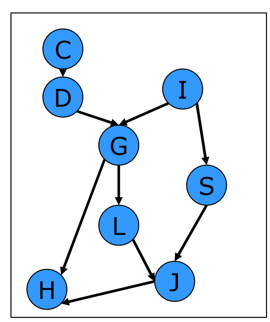
Assume we want to calculate: P(J)
We need to sum out C,D, …

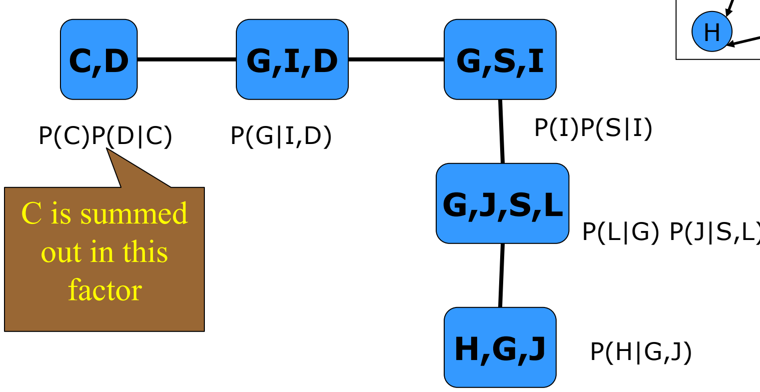


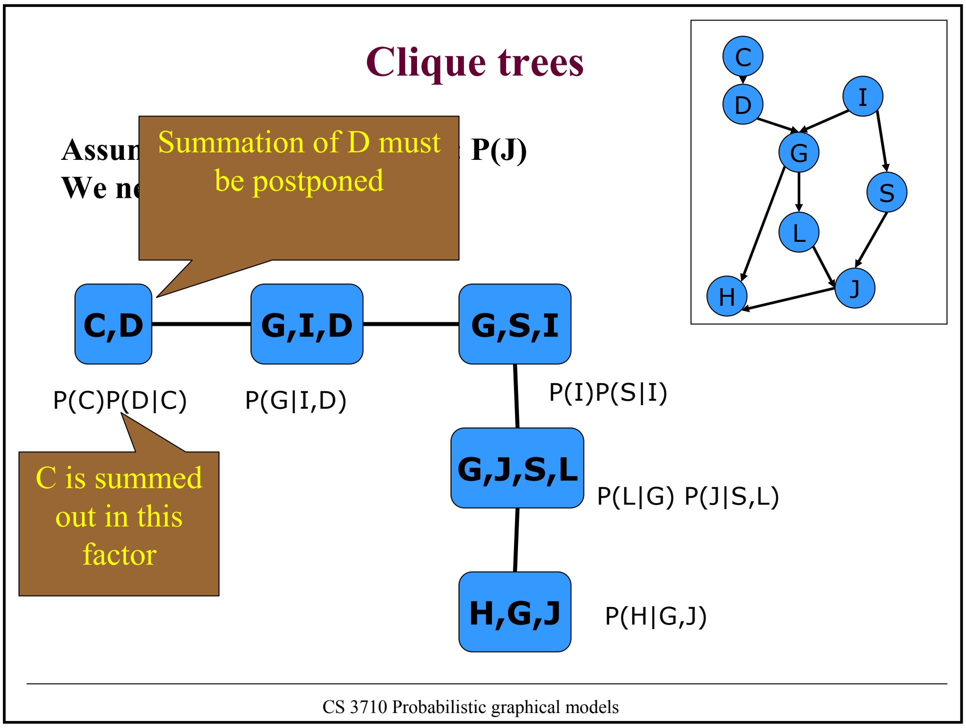


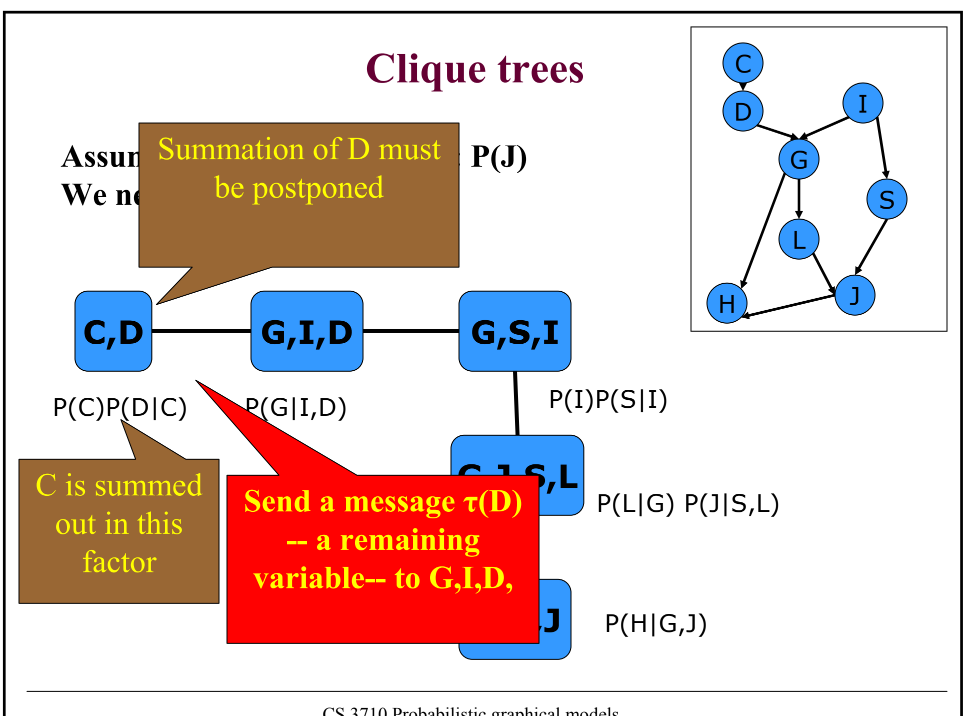
Clique trees

Assume we want to calculate: P(J)
We need to sum out C,D, …



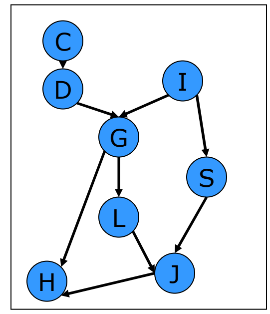


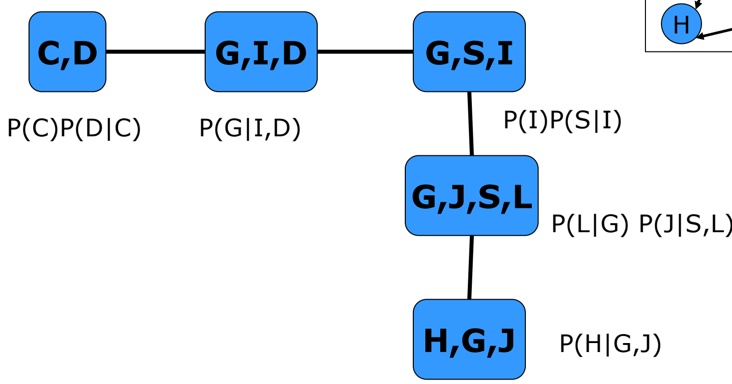

Clique trees
Assu
We n
Summation of D must be postponed
P(J)
C,D
G,I,D
G,S,I
G,J,S,L
H,G,J
P(C)P(D|C)
P(G|I,D)
P(I)P(S|I)
P(L|G) P(J|S,L)
P(H|G,J)
C is summed out in this factor
C
D
I
G
S
L
H
J
CS 3710 Probabilistic graphical models


Clique trees
Assu
We n
Summation of D must be postponed
P(J)
C,D
G,I,D
G,S,I
P(C)P(D|C)
P(G|I,D)
P(I)P(S|I)
P(L|G) P(J|S,L)
P(H|G,J)
C is summed out in this factor
Send a message τ(D) -- a remaining variable-- to G,I,D,
C
D
I
G
S
L
H
J
CS 3710 Probabilistic graphical models

Clique trees

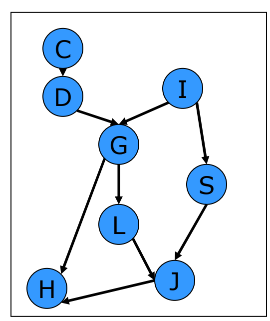
Assume we want to calculate: P(J,C=c)
We need to sum out D, …

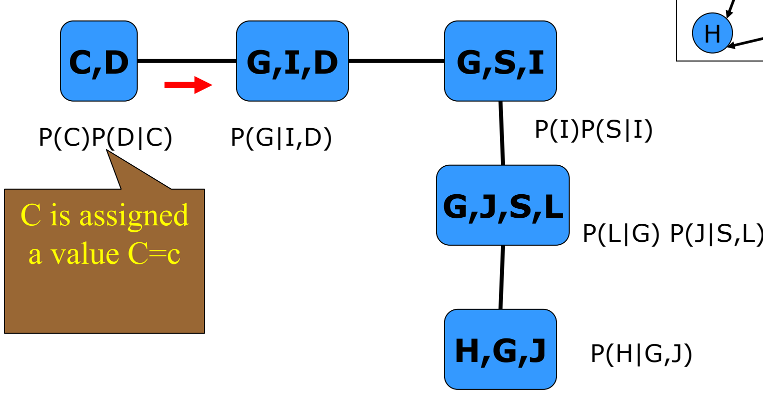


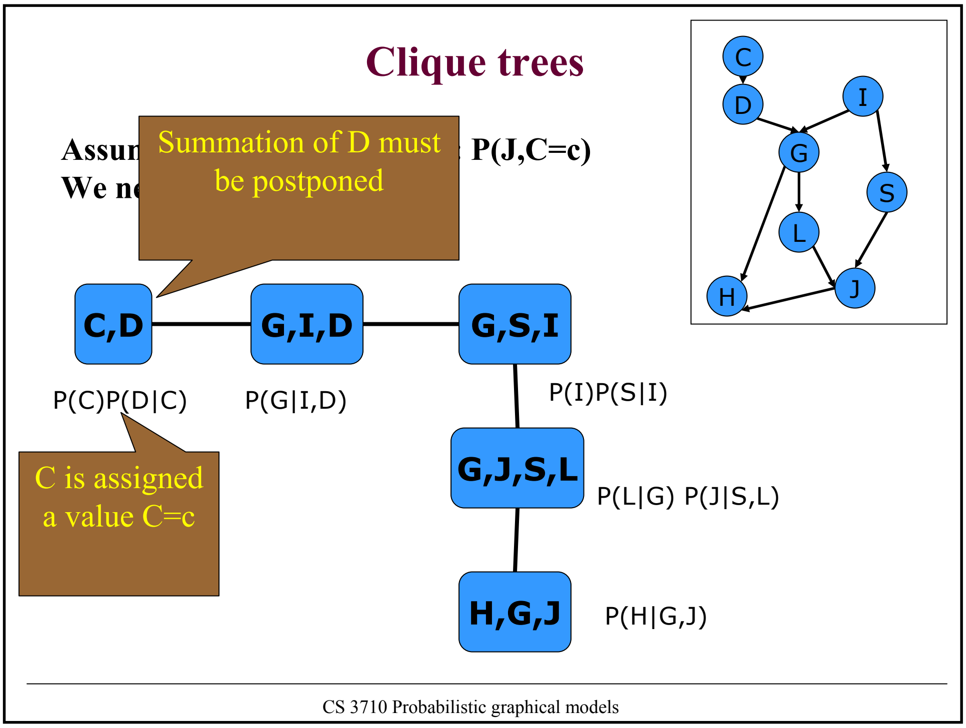


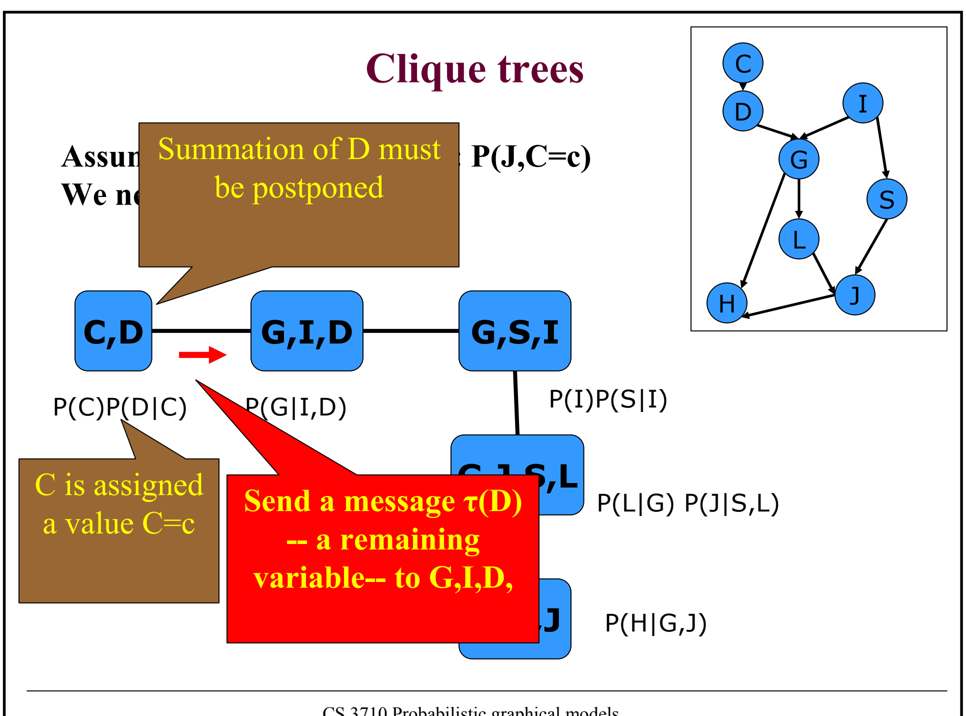
Clique trees

Assume we want to calculate: P(J,C=c)
We need to sum out D, …

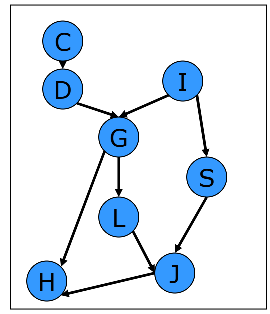




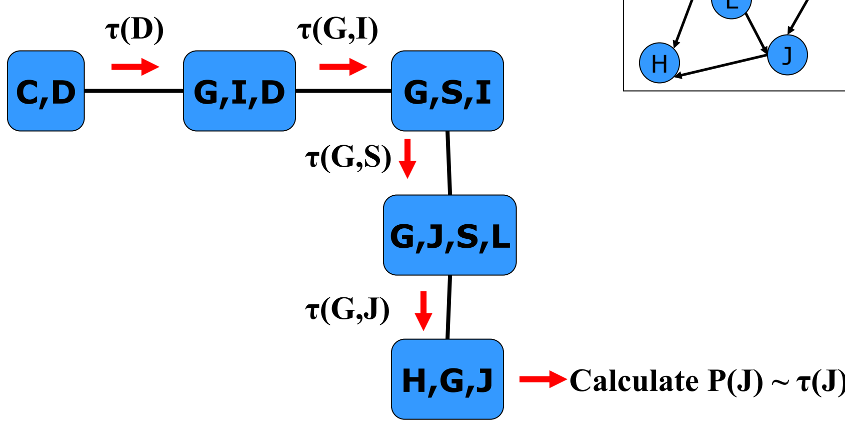

Clique trees
Assun
We ne
Summation of D must be postponed
P(J,C=c)
C,D
G,I,D
G,S,I
G,J,S,L
H,G,J
P(C)P(D|C)
P(G|I,D)
P(I)P(S|I)
P(L|G) P(J|S,L)
P(H|G,J)
C is assigned a value C=c
C
D
I
G
S
L
H
J
CS 3710 Probabilistic graphical models


Clique trees
Assun
We ne
Summation of D must be postponed
P(J,C=c)
C,D
G,I,D
G,S,I
P(C)P(D|C)
P(G|I,D)
P(I)P(S|I)
P(L|G) P(J|S,L)
P(H|G,J)
C is assigned a value C=c
Send a message τ(D) -- a remaining variable-- to G,I,D,
C
D
I
G
S
L
H
J
CS 3710 Probabilistic graphical models

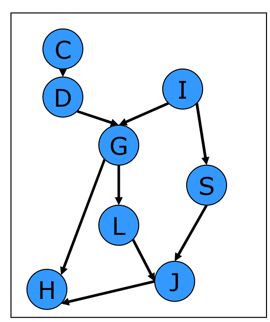
Clique trees



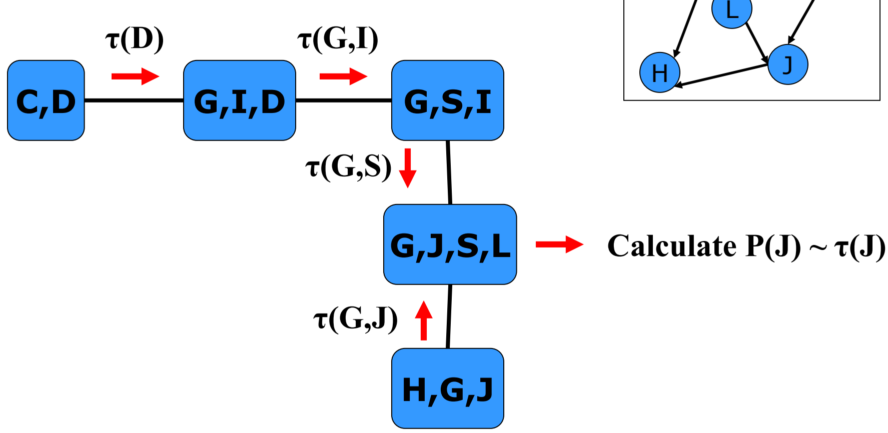
Assume we want to calculate: P(J)
We need to sum out C,D, …



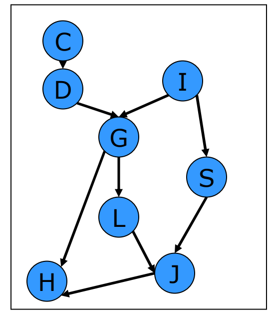
Clique trees



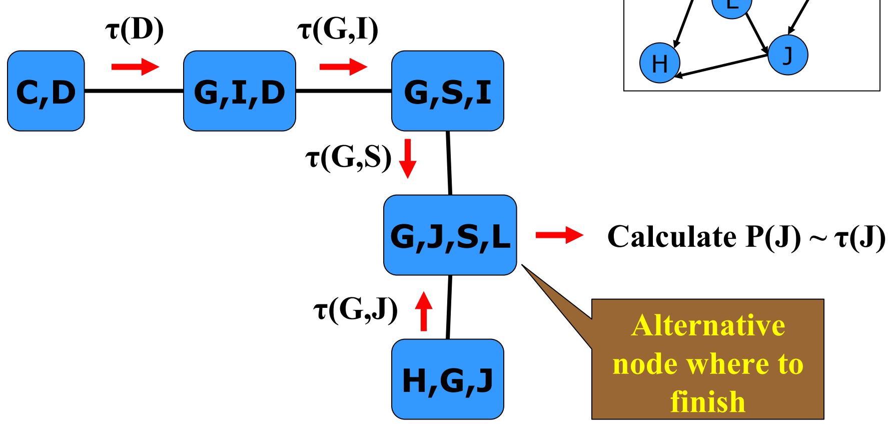
Assume we want to calculate: P(J)
We need to sum out C,D, …



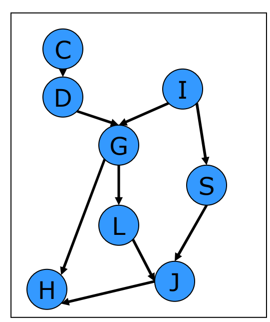
Clique trees



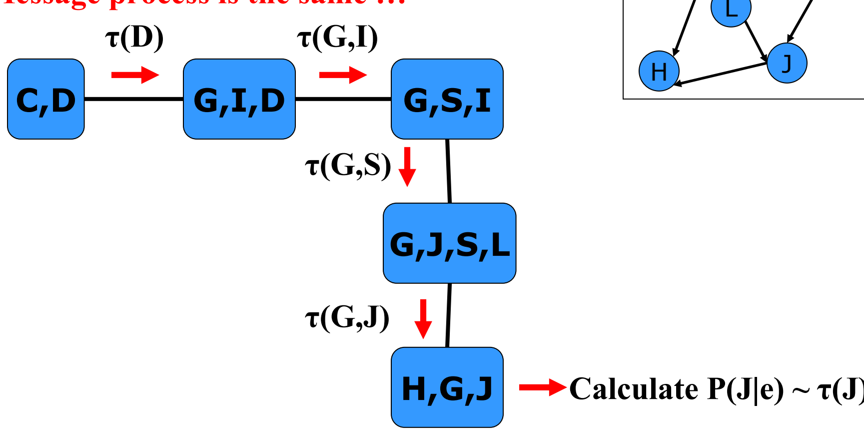
Assume we want to calculate: P(J)
We need to sum out C,D, …



Clique trees

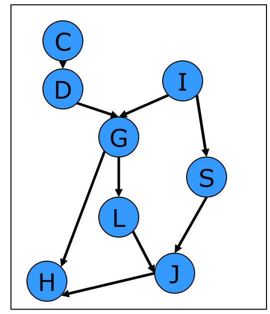


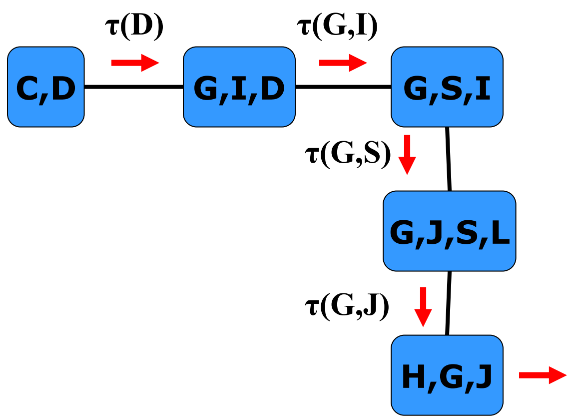
Assume we want to calculate: P(J,C=c,S=s)
We need to sum out D, …
Message process is the same !!!



Message passing

Assume we want to calculate marginal for every variable



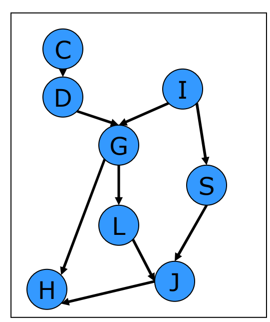


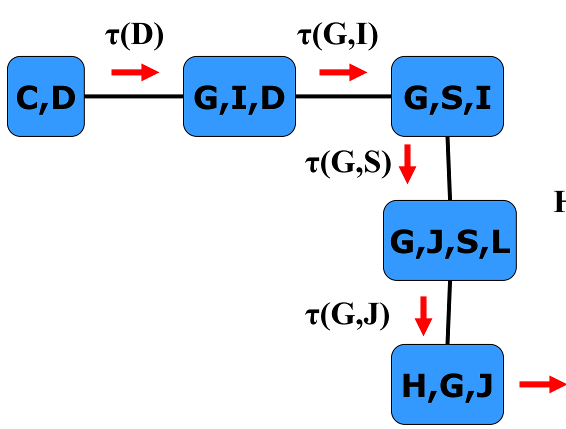
Calculate P(J) ~ τ(J)

Calculate P(G) ~ τ(G)

Message passing

Assume we want to calculate marginal for every variable





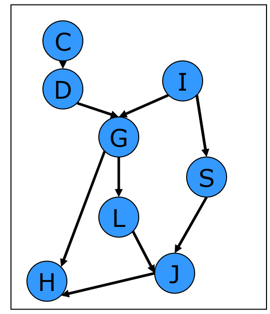
How to calculate P(S)?

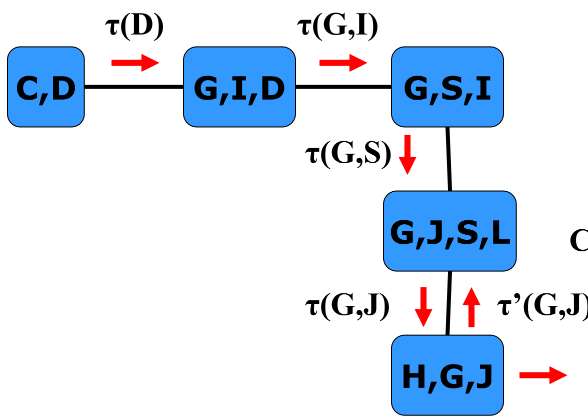
Calculate P(J) ~ τ(J)

Calculate P(G) ~ τ(G)

Message passing

Assume we want to calculate marginal for every variable



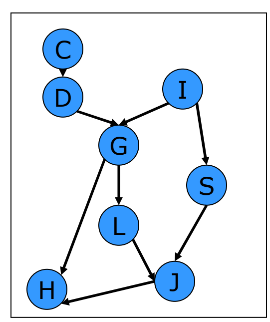


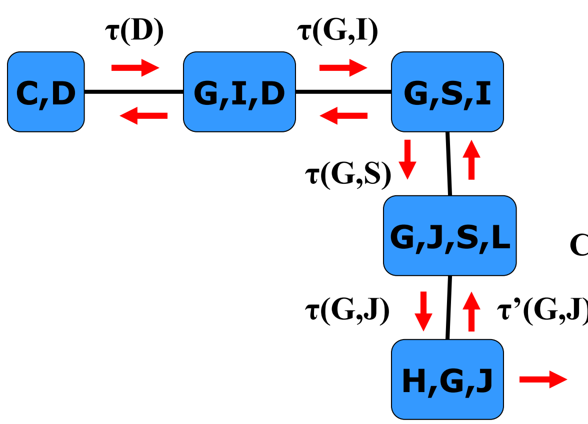
Calculate P(S) ~ τ(S)

Calculate P(J) ~ τ(J)
Calculate P(G) ~ τ(G)

Message passing

Assume we want to calculate marginal for every variable
We need messages to be sent from all parts

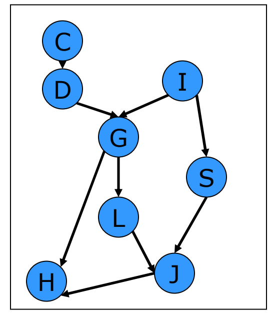




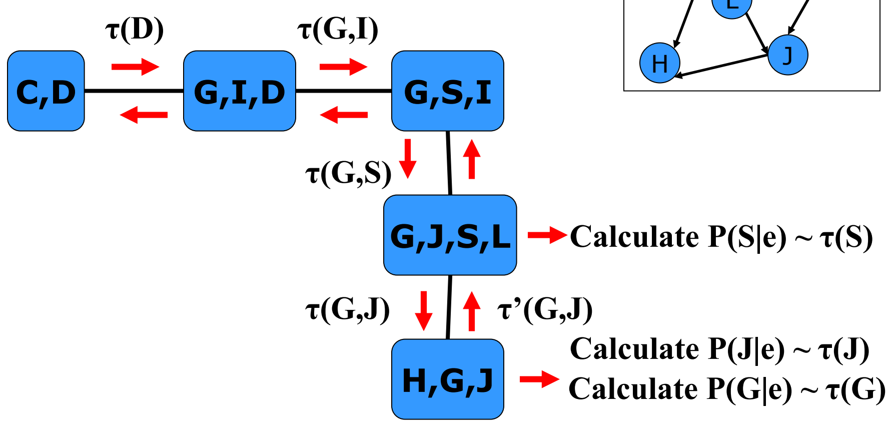
Calculate P(S) ~ τ(S)

Calculate P(J) ~ τ(J)
Calculate P(G) ~ τ(G)

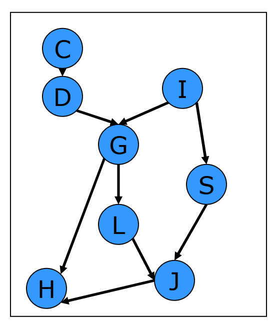
Message passing



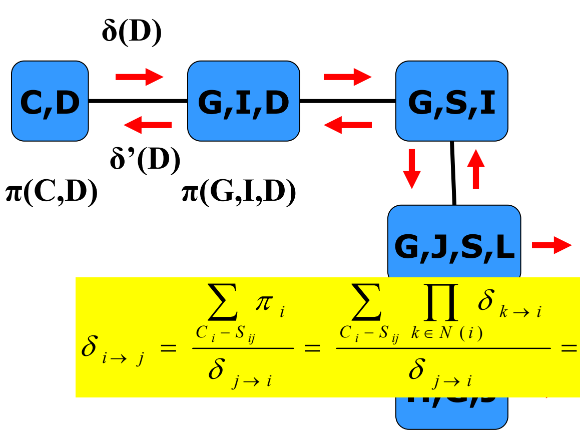
The same thing applies if we have some fixed Evidence, say C



Belief propagation



Keep sending messages back and forth
Summation



$$\delta_{i \to j} = \frac{\sum_{C_i - S_{ij}} \pi_i}{\delta_{j \to i}} = \frac{\sum_{C_i - S_{ij}} \prod_{k \in N(i)} \delta_{k \to i}}{\delta_{j \to i}} = \sum_{C_i - S_{ij}} \prod_{k \in N(i) \setminus j} \delta_{k \to i}$$