CS 3750 Machine Learning

Lecture 3

Graphical models and inference II

Milos Hauskrecht
milos@pitt.edu
5329 Sennott Square, x4-8845

http://www.cs.pitt.edu/~milos/courses/cs3750-Spring2020/

Challenges for modeling complex multivariate distributions

How to model/parameterize complex multivariate distributions $P(\mathbf{X})$ with a large number of variables?

One solution:

- Decompose the distribution. Reduce the number of parameters, using some form of independence.

Two models:

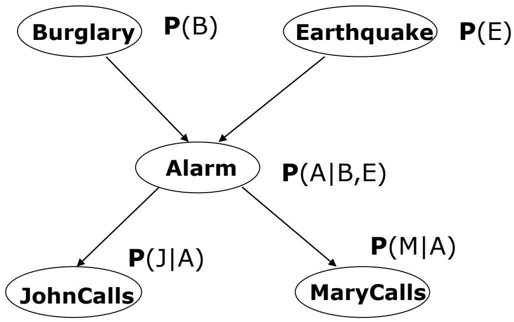
- **Bayesian belief networks (BBNs)**
- **Markov Random Fields (MRFs)**

- **Learning of these models** relies on the decomposition.

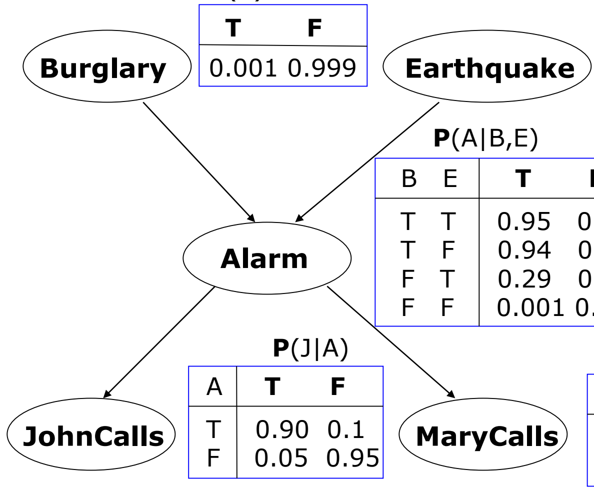
Bayesian belief network

Directed acyclic graph

- **Nodes** = random variables
- **Links** = direct (causal) dependencies
  Missing links encode different marginal and conditional independences



Bayesian belief network



P(B)

T	**F**
0.001	0.999

P(E)

T	**F**
0.002	0.998

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

P(J|A)

A	**T**	**F**
T	0.90	0.1
F	0.05	0.95

P(M|A)

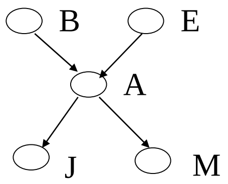
A	T	F
T	0.7	0.3
F	0.01	0.99

Full joint distribution in BBNs

The full joint distribution is defined as a product 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:



Assume the following assignment of values to random variables

$B = T, E = T, A = T, J = T, M = F$

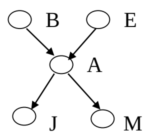
Then its probability is:

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

Inference in Bayesian networks

- Full joint uses the decomposition
- **Calculation of marginals:**
  - Requires summation over variables we want to take out

$$P(J = T) =$$
$$= \sum_{b \in T,F} \sum_{e \in T,F} \sum_{a \in T,F} \sum_{m \in T,F} P(B = b, E = e, A = a, J = T, M = m)$$



- How to compute sums and products more efficiently?

$$\sum_x af(x) = a \sum_x f(x)$$

Variable elimination

Assume order: **M, E, B, A** to calculate $P(J=T)$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F} P(J=T\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)\left[\sum_{m\in T,F} P(M=m\mid A=a)\right]$$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F} P(J=T\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)\quad 1$$

$$= \sum_{a\in T,F}\sum_{b\in T,F} P(J=T\mid A=a)P(B=b)\left[\sum_{e\in T,F} P(A=a\mid B=b,E=e)P(E=e)\right]$$

$$= \sum_{a\in T,F}\sum_{b\in T,F} P(J=T\mid A=a)P(B=b)\,\tau_1(A=a,B=b)$$

$$= \sum_{a\in T,F} P(J=T\mid A=a)\left[\sum_{e\in T,F} P(B=b)\,\tau_1(A=a,B=b)\right]$$

$$= \sum_{a\in T,F} P(J=T\mid A=a)\quad \tau_2(A=a) \quad = \quad \boxed{P(J=T)}$$

Variable elimination

Assume order: **M, E, B, A** to calculate $P(J=T)$

$$= \sum_{b\in T,F}\sum_{e\in T,F}\sum_{a\in T,F}\sum_{m\in T,F} P(J=T\mid A=a)P(M=m\mid A=a)P(A=a\mid B=b,E=e)P(B=b)P(E=e)$$

$$= \sum_{B\in T,F}\sum_{E\in T,F}\sum_{A\in T,F}\sum_{M\in T,F} f_1(A)\, f_2(M,A)\, f_3(A,B,E)\, f_4(B)\, f_4(E)$$

Conditional probabilities defining the joint = factors

Variable elimination inference can be cast in terms of operations defined over factors

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:

$$\phi(x, y) \longrightarrow$$

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

Variables: A,B,C

$$\phi(A,B,C) = \phi(B,C) \circ \phi(A,B)$$

$\phi(B,C)$

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

$\phi(A,B)$

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

$\phi(A,B,C)$

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 Marginalization

Variables: A,B,C

$$\phi(A,C) = \sum_{B} \phi(A,B,C)$$

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	0.2+0.4=0.6
a1	c2	0.35+0.15=0.5
a2	c1	0.8
a2	c2	0.3
a3	c1	0.4
a3	c2	0.7

Factor division

A=1	B=1	0.5
A=1	B=2	0.4
A=2	B=1	0.8
A=2	B=2	0.2
A=3	B=1	0.6
A=3	B=2	0.5

A=1	0.4
A=2	0.4
A=3	0.5

A=1	B=1	0.5/0.4=1.25
A=1	B=2	0.4/0.4=1.0
A=2	B=1	0.8/0.4=2.0
A=2	B=2	0.2/0.4=2.0
A=3	B=1	0.6/0.5=1.2
A=3	B=2	0.5/0.5=1.0

Inverse of a factor product

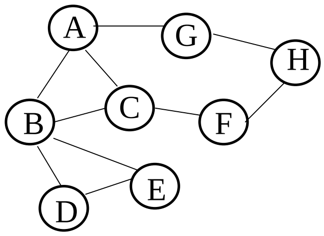
Markov random fields

An undirected network (also called independence graph)

- **Probabilistic models with symmetric dependences**
- G = (S, E)
  - S set of random variables
  - Undirected edges E that define dependences between pairs of variables

Example:

variables A,B ..H



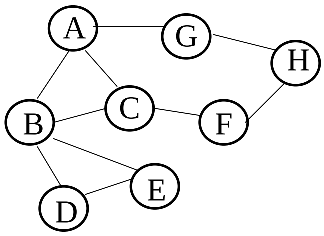
Markov random fields

The full joint of the MRF is defined

$$P(\mathbf{x}) \propto \prod_{c \in cl(x)} \phi_c(\mathbf{x}_c)$$

$\phi_c(x_c)$ - A potential function (defined over variables in cliques/factors)

Example:



Full joint:

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

$\phi_c(x_c)$ - A potential function (defined over a clique of the graph)

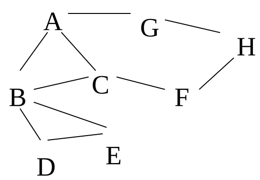
Markov random fields: independence relations

- **Pairwise Markov property**
  - Two nodes in the network that are not directly connected can be made independent given all other nodes
- **Local Markov property**
  - A set of nodes (variables) can be made independent from the rest of nodes variables given its immediate neighbors
- **Global Markov property**
  - A vertex set A is independent of the vertex set B (A and B are disjoint) given set C if all chains in between elements in A and B intersect C

MRF variable elimination inference

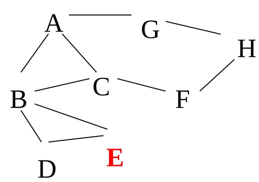
Example:

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



$$= \frac{1}{Z} \sum_{A,C,D,..H} \phi_1(A,B,C)\phi_2(B,D,E)\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

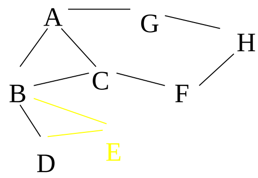
Eliminate E



$$= \frac{1}{Z} \sum_{A,C,D,F,G,H} \phi_1(A,B,C)\underbrace{\left[\sum_{E} \phi_2(B,D,E)\right]}_{\tau_1(B,D)}\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H$$

MRF variable elimination inference

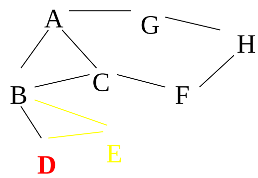
Example (cont):



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

$$= \frac{1}{Z} \sum_{A,C,D,F,G,H} \phi_1(A,B,C)\tau_1(B,D)\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

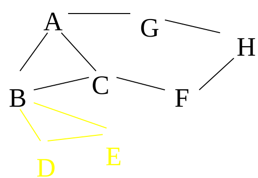
Eliminate D



$$= \frac{1}{Z} \sum_{A,C,F,G,H} \phi_1(A,B,C)\underbrace{\left[\sum_{D} \tau_1(B,D)\right]}_{\tau_2(B)}\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

MRF variable elimination inference

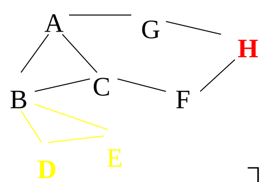
Example (cont):



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

$$= \frac{1}{Z} \sum_{A,C,F,G,H} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\phi_4(C,F)\phi_5(G,H)\phi_6(F,H)$$

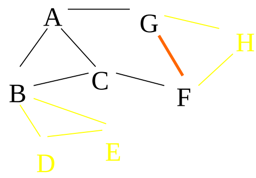
Eliminate H



$$= \frac{1}{Z} \sum_{A,C,F,G} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\phi_4(C,F)\underbrace{\left[\sum_{H} \underbrace{\phi_5(G,H)\phi_6(F,H)}_{\tau_3(F,G,H)}\right]}_{\tau_4(F,G)}$$

MRF variable elimination inference

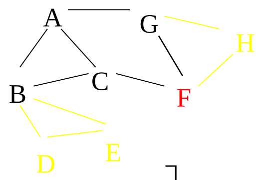
Example (cont):



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

$$= \frac{1}{Z} \sum_{..,C,F,G} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\phi_4(C,F)\tau_4(F,G)$$

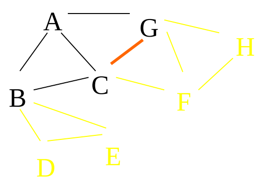
Eliminate F



$$= \frac{1}{Z} \sum_{A,C,G} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\underbrace{\left[\underbrace{\sum_F \phi_4(C,F)\tau_4(F,G)}_{\tau_5(C,F,G)}\right]}_{\tau_6(G,C)}$$

MRF variable elimination inference

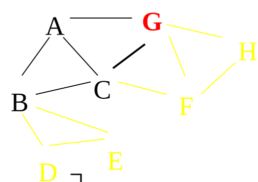
Example (cont):



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

$$= \frac{1}{Z} \phi_1(A,B,C)\tau_2(B)\phi_3(A,G)\tau_6(C,G)$$

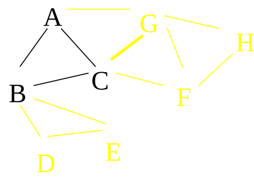
Eliminate G



$$= \frac{1}{Z} \sum_{A,C} \phi_1(A,B,C)\tau_2(B)\underbrace{\left[\underbrace{\sum_F \phi_3(A,G)\tau_6(C,G)}_{\tau_7(A,C,G)}\right]}_{\tau_8(A,C)}$$

MRF variable elimination inference

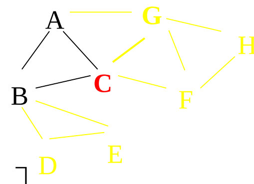
Example (cont):



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

$$= \frac{1}{Z} \sum_{A,C} \phi_1(A,B,C)\tau_2(B)\tau_8(A,C)$$

Eliminate C

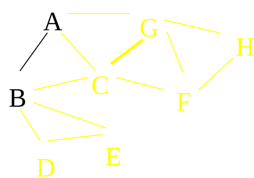


$$= \frac{1}{Z} \sum_{A} \tau_2(B)\left[\sum_{C} \underbrace{\phi_1(A,B,C)\tau_8(A,C)}_{\tau_9(A,B,C)}\right]$$

$$\underbrace{\qquad\qquad}_{\tau_{10}(A,B)}$$

MRF variable elimination inference

Example (cont):

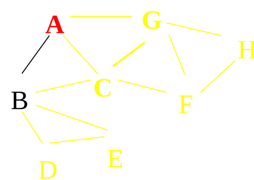


$$P(B) = \sum_{A,C,D,..} P(A,B,...H)$$

$$= \frac{1}{Z} \sum \tau_2(B)\tau_{10}(A,B)$$

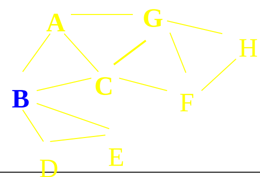
$$= \frac{1}{Z} \tau_2(B)\sum_{A} \tau_{10}(A,B)$$

Eliminate A



$$= \frac{1}{Z} \tau_2(B)\underbrace{\sum_{A} \tau_{10}(A,B)}_{\tau_{11}(B)}$$

$$= \frac{1}{Z} \tau_2(B)\tau_{11}(B)$$



Are BBNs and MRFs different?

Both models represent independences that hold among variables or sets of variables?

- Are the two the same in terms of independences they can represent?
- Or, are they different?

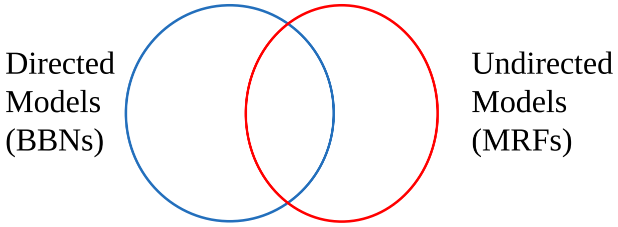
Are BBNs and MRFs different?

Both models represent independences that hold among variables or sets of variables?

- Are the two the same in terms of independences they can represent?
- Or, are they different?

Answer: MRFs are different from BBNs

- There are independences that can be represented by one model but not the other

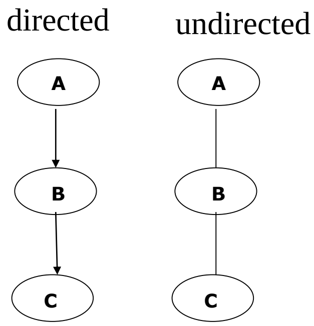


Are BBNs and MRFs different?

MRFs are different from BBNs

- There are independences that can be represented by one model but not the other

Analysis:



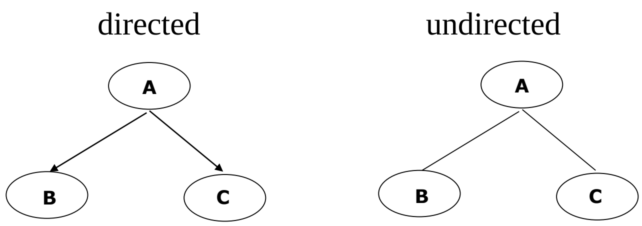
A is independent of C given B

Are BBNs and MRFs different?

MRFs are different from BBNs

- There are independences that can be represented by one model but not the other

Analysis:



B is independent of C given A

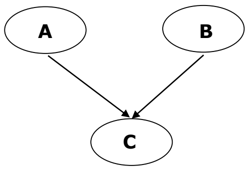
Are BBNs and MRFs different?

MRFs are different from BBNs

- There are independences that can be represented by one model but not the other

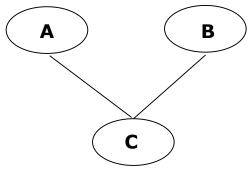
Analysis:

directed

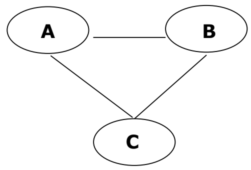


≠

undirected



Fix to undirected (moralization)



A and B are marginally independent

A and B are independent given C

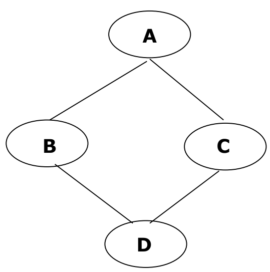
A, B, C are all dependent
No false independence

Are BBNs and MRFs different?

MRFs are different from BBNs

- There are independences that can be represented by one model but not the other

Analysis: undirected



No directed graph can represent the same set of independences

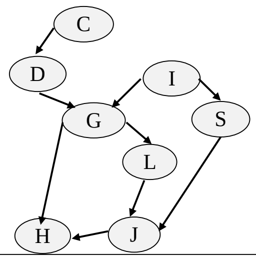
B and C are independent given A,D

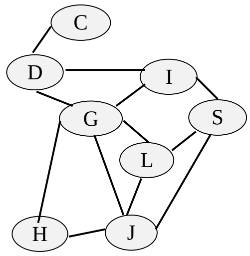
A and D are independent given B,C

Converting BBNs to MRFs

Moral-graph H[G]: of a Bayesian network over X is an undirected graph over X that contains an edge between x and y if:

- There exists a directed edge between them in G.
- They are both parents of the same node in G.

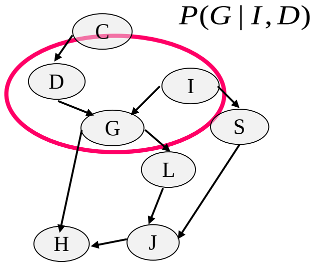


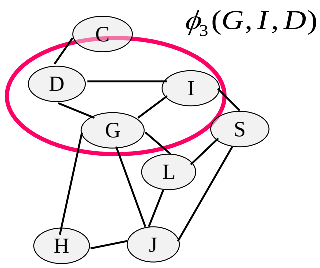


Moral Graphs: define MRFs

Why moralization?

$$P(C, D, G, I, S, L, J, H) =$$
$$= P(C)P(D \mid C)P(G \mid I, D)P(S \mid I)P(L \mid G)P(J \mid L, S)P(H \mid G, J)$$
$$= \phi_1(C)\phi_2(D, C)\phi_3(G, I, D)\phi_4(S, I)\phi_5(L, G)\phi_6(J, L, S)\phi_7(H, G, J)$$



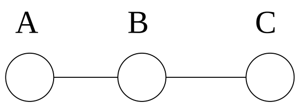


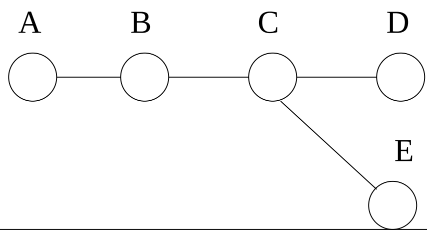
Inference

Variable elimination: Depends on the order of variables to eliminate

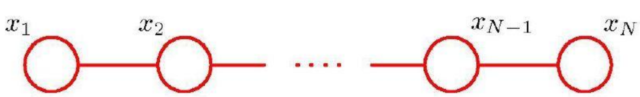
Question: can we optimize the structures ahead of times so that we can make inferences efficiently and without worrying about the specific variable order?

- **Structures that support efficient inferences: Chains, and trees**





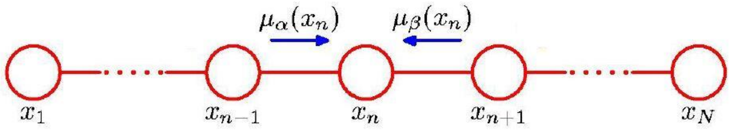
Inference on a Chain



$$p(\mathbf{x}) = \frac{1}{Z}\psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\cdots\psi_{N-1,N}(x_{N-1}, x_N)$$

$$p(x_n) = \sum_{x_1}\cdots\sum_{x_{n-1}}\sum_{x_{n+1}}\cdots\sum_{x_N} p(\mathbf{x})$$

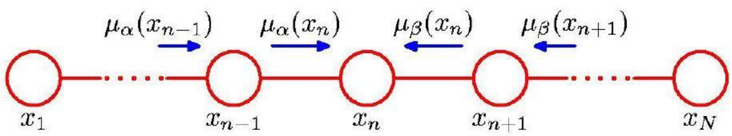
Inference on a Chain



$$p(x_n) = \frac{1}{Z} \underbrace{\left[\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_1} \psi_{1,2}(x_1, x_2) \right] \cdots \right]}_{\mu_\alpha(x_n)}$$

$$\underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \right] \cdots \right]}_{\mu_\beta(x_n)}$$

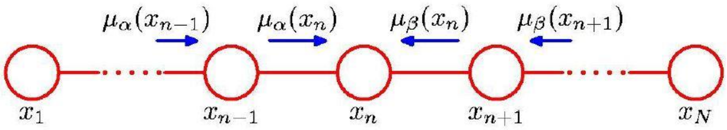
Inference on a Chain



$$\begin{aligned} \mu_\alpha(x_n) &= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \left[\sum_{x_{n-2}} \cdots \right] \\ &= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \mu_\alpha(x_{n-1}). \end{aligned}$$

$$\begin{aligned} \mu_\beta(x_n) &= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \left[\sum_{x_{n+2}} \cdots \right] \\ &= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \mu_\beta(x_{n+1}). \end{aligned}$$

Inference on a Chain



$$\mu_\alpha(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2) \qquad \mu_\beta(x_{N-1}) = \sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)$$

$$Z = \sum_{x_n} \mu_\alpha(x_n)\mu_\beta(x_n)$$

Slides by C. Bishop

Inference on a Chain

To compute local marginals:

- Compute and store all forward messages, $\mu_\alpha(x_n)$.
- Compute and store all backward messages, $\mu_\beta(x_n)$.
- Compute Z at any node x_m
- Compute

$$p(x_n) = \frac{1}{Z}\mu_\alpha(x_n)\mu_\beta(x_n)$$

for all variables required.

Slides by C. Bishop

Inference

Many BBNs or MRFs are not tree structured

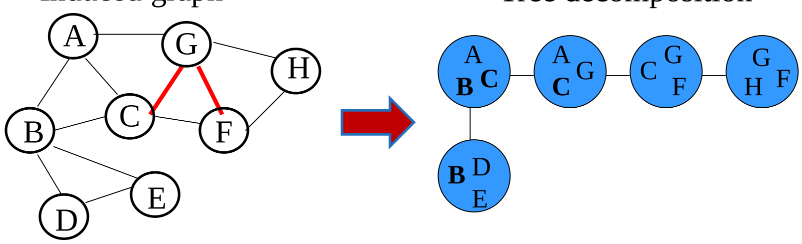
- Can we optimize the structures ahead of times so that we can make inferences efficiently and without worrying about the specific variable order?
- **Idea: Convert to trees that support efficient inference**
- **Next:** two approaches to convert MRFs (or BBNs) to tree structures

Induced graph

A graph induced by a specific variable elimination order that covers all variables:

- **a graph G extended by links that represent intermediate factors**

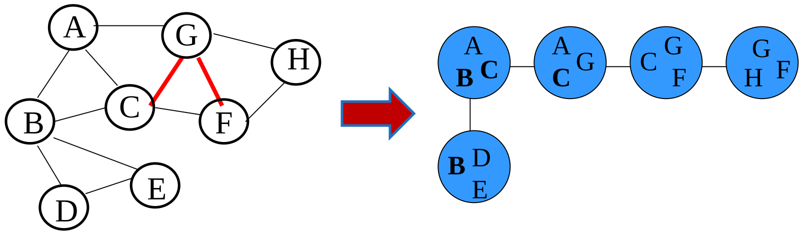
Induced graph Tree decomposition



- **Induced graph defines a tree decomposition of a graph G (or a clique tree)**

Tree decomposition

Induced graph (G) Tree decomposition (T)

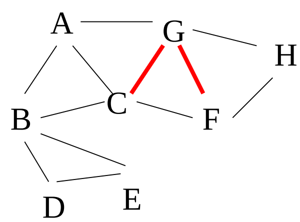


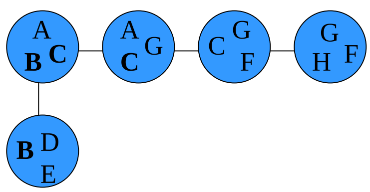
- A node in tree *T* is formed by a set of vertices corresponding to maximum cliques in G
- For all edges {*v*,*w*}∈G: there is a set containing both *v* and *w* in *T*
- For every *v* ∈G : the nodes in T that contain *v* form a connected subtree.

Tree decomposition of the graph

A tree decomposition of a graph G (clique tree):

- A node in tree *T* is formed by a set of vertices corresponding to maximum cliques in G
- For all edges {*v*,*w*}∈G: there is a set containing both *v* and *w* in *T*.
- For every *v* ∈G : the nodes in T that contain *v* form a connected subtree.

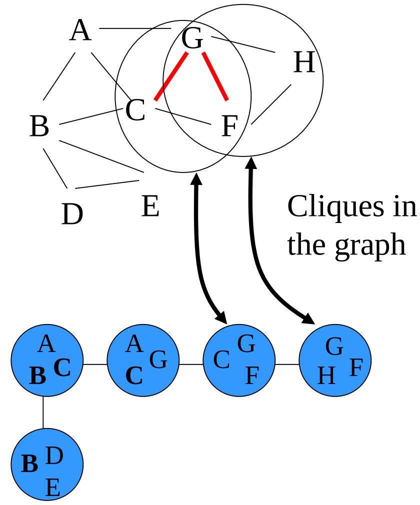




Tree decomposition of the graph

A tree decomposition of a graph G (clique tree):

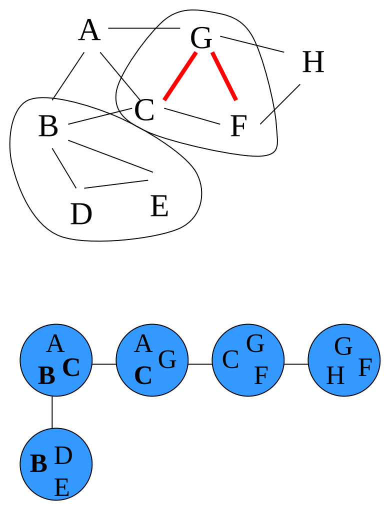
- A node in tree *T* is formed by a set of vertices corresponding to maximum cliques in G
- For all edges $\{v,w\} \in G$: there is a set containing both v and w in T.
- For every $v \in G$: the nodes in T that contain v form a connected subtree.



Tree decomposition of the graph

A tree decomposition of a graph G (clique tree):

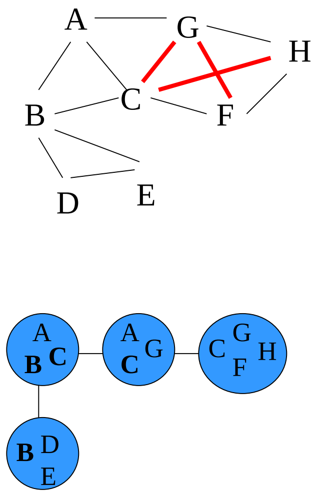
- A node in tree *T* is formed by a set of vertices corresponding to maximum cliques in G
- For all edges $\{v,w\} \in G$: there is a set containing both v and w in T.
- For every $v \in G$: the nodes in T that contain v form a connected subtree.



Tree decomposition of the graph

A tree decomposition of a graph G (clique tree):

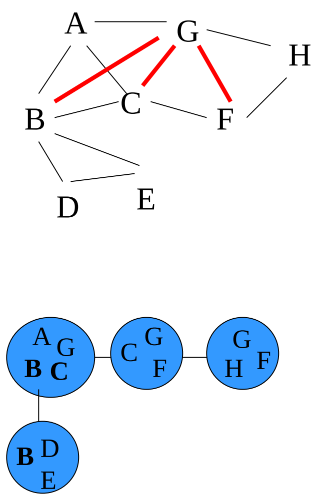
- A node in tree *T* is formed by a set of vertices corresponding to maximum cliques in G
- For all edges {*v*,*w*}∈G: there is a set containing both *v* and *w* in *T*.
- For every *v* ∈G : the nodes in T that contain *v* form a connected subtree.



Tree decomposition of the graph

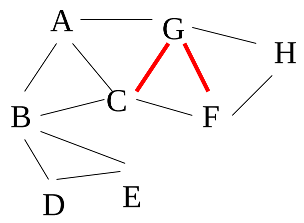
Another decomposition of a graph G:

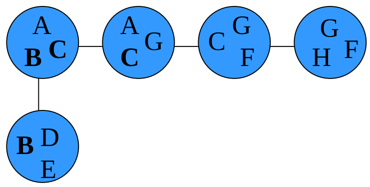
- A node in tree *T* is formed by a set of vertices corresponding to maximum cliques in G
- For all edges {*v*,*w*}∈G: there is a set containing both *v* and *w* in *T*.
- For every *v* ∈G : the nodes in T that contain *v* form a connected subtree.



Treewidth of the graph

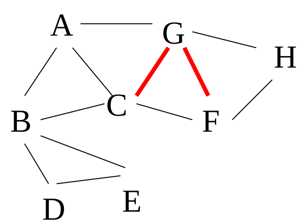
- **Width** of the tree decomposition: $\max_{i \in I} |X_i| - 1$
- **Treewidth** of a graph G: tw(G)= minimum width over all tree decompositions of G.

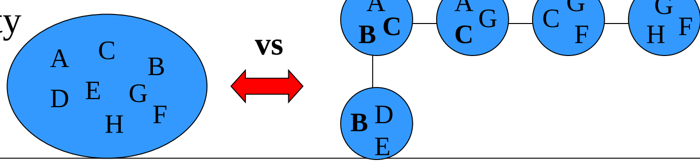




Treewidth of the graph

- **Treewidth** of a graph G: tw(G)= minimum width over all tree decompositions of G
- Why is it important?
- Many calculations can take advantage of the structure and be performed more efficiently
- treewidth gives the best case complexity

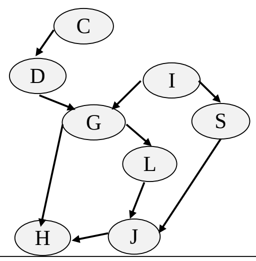


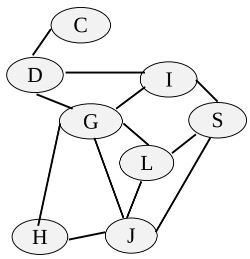


Converting BBNs to MRFs

Moral-graph H[G]: of a Bayesian network over X is an undirected graph over X that contains an edge between x and y if:

- There exists a directed edge between them in G.
- They are both parents of the same node in G.





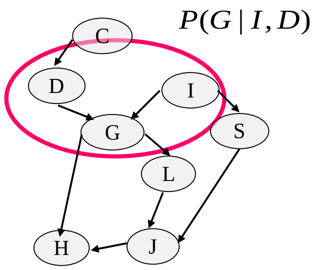
Moral Graphs: define MRFs

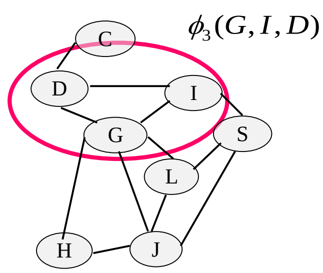
Why moralization?

$$P(C,D,G,I,S,L,J,H) =$$

$$= \quad P(C)P(D|C)P(G|I,D)P(S|I)P(L|G)P(J|L,S)P(H|G,J)$$

$$= \quad \phi_1(C)\phi_2(D,C)\phi_3(G,I,D)\phi_4(S,I)\phi_5(L,G)\phi_6(J,L,S)\phi_7(H,G,J)$$

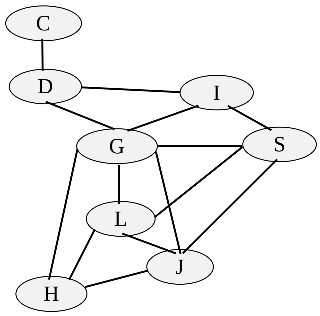




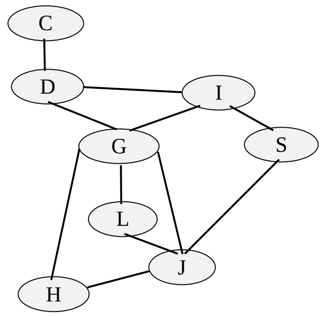
Chordal graphs

Chordal Graph: an undirected graph *G*

- all cycles of four or more vertices have a chord (another edge breaking the cycle)
- minimum cycle for every vertex in a cycle is 3 (contains 3 verticies*)*



Chordal.

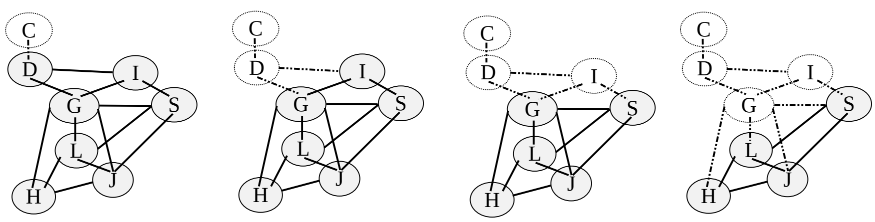


Not Chordal

Chordal Graphs

Properties:

- **There exists an elimination ordering that adds no edges.**
- The minimal induced tree-width of the graph is equal to the size of the largest clique - 1



Triangulation

The process of converting a graph G into a chordal graph is called **Triangulation**

A new graph obtained via triangulation is:

1) Guaranteed to be chordal.
2) Not guaranteed to be (tree-width) optimal.

- There exist exact algorithms for finding the **minimal chordal graphs**, and heuristic methods with a guaranteed upper bound

Chordal Graphs

- Given a minimum triangulation for a graph *G*, we can carry out the variable-elimination algorithm in the minimum possible time.

- **Complexity** of the optimal triangulation:
  – Finding the minimal triangulation is **NP-Hard**.

- **The inference limit:**
  – Inference time is exponential in terms of the largest clique (factor) in *G*.

CS 3750 Advanced Machine Learning

Conversion of an MRF (BBN) to a clique tree

MRF conversions to clique trees:

Option 1:

- Via triangulation to form a chordal graph
- Cliques in the chordal graph define the clique tree

Option 2:

- From the induced graph built by running the variable elimination (VE) procedure
- Cliques are defined by factors generated during the VE procedure

BBN conversion:

- Convert the BBN to an MRF – a moral graph
- Apply MRF conversion

Conclusions on inference complexity

We cannot escape **exponential costs of the tree-width of the graph**

- **Recall: Tree-width = the width of the optimal tree decomposition (or the optimal clique tree)**

Good news:

- For many graphs the **tree-width** is much smaller than the total number of variables !!!

Still a problem: Finding the optimal clique tree is hard (NP hard)

  – But, paying the cost up front may be worth it
  – Triangulate once, query many times.
  – Real cost savings if not a bounded one