CS 3750 Machine Learning
Lecture 6

Graphical models
Inference

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

Markov random fields

- **Probabilistic models with symmetric dependences.**
  - Typically models spatially varying quantities

$$P(x) \propto \prod_{c \in cl(x)} f_c(x_c)$$

$f_c(x_c)$ - A potential function (defined over factors)

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

- Gibbs (Boltzman) distribution

$$Z = \sum_{x \in \{x\}} \exp\left(- \sum_{c \in cl(x)} \phi_c(x_c) \right)$$ - A partition function

Graphical representation of MRFs

An undirected network (also called independence graph)

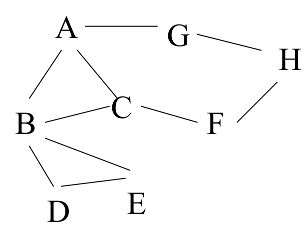
- G = (S, E)
  - S=1, 2, .. N correspond to random variables
  - $(i, j) \in E \Leftrightarrow \exists c : \{i, j\} \subset c$

    or x_i and x_j appear within the same factor c

Example:

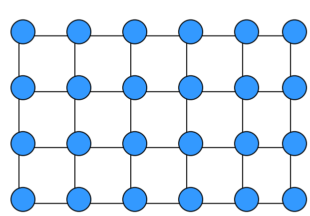
- variables A,B ..H
- Assume the full joint of MRF

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

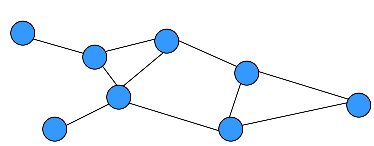


Markov random fields

- **regular lattice (Ising model)**

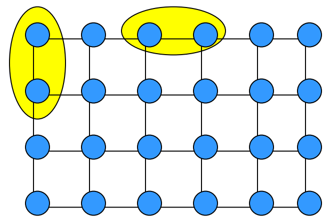


- **Arbitrary graph**

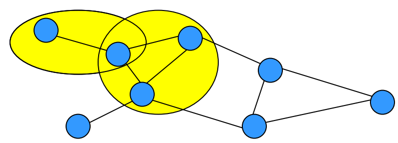


Markov random fields

- **regular lattice (Ising model)**

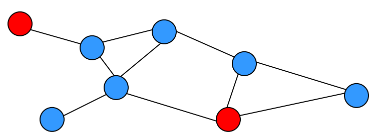


- **Arbitrary graph**



Markov random fields

- **Pairwise Markov property**
  - Two nodes in the network that are not directly connected can be made independent given all other nodes



$$P(x_A, x_B \mid x_r) = \frac{P(x_A, x_B, x_r)}{P(x_r)} \propto \exp\left(- \sum_{c: c \cap A \neq \{\}} \phi_c(x_c) - \sum_{c: c \cap A = \{\}, c \cap B \neq \{\}} \phi_c(x_c) - \sum_{c: c \cap A = \{\}, c \cap B = \{\}} \phi_c(x_c) \right)$$

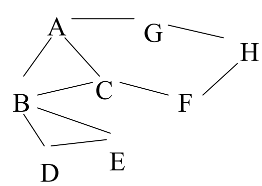
$$\propto \exp\left(- \sum_{c: c \cap A \neq \{\}} \phi_c(x_c) \right) \exp\left(- \sum_{c: c \cap A = \{\}, c \cap B \neq \{\}} \phi_c(x_c) \right) \approx P(x_A \mid x_r) P(x_B \mid x_r)$$

Markov random fields

- **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

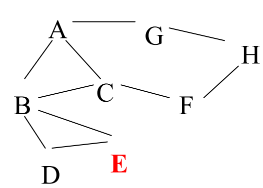
Example:



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

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

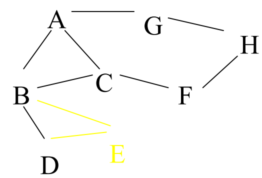
Eliminate E



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

MRF variable elimination inference

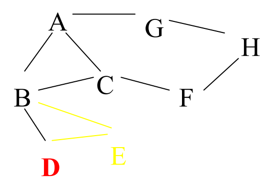
Example (cont):



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

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

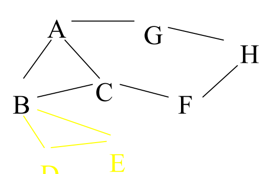
Eliminate D



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

MRF variable elimination inference

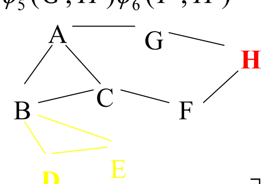
Example (cont):



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

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

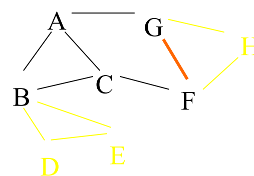
Eliminate H



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

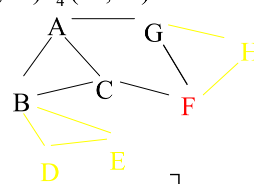
MRF variable elimination inference

Example (cont):



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

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

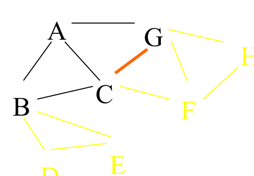


Eliminate F

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

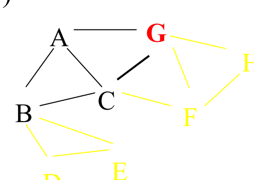
MRF variable elimination inference

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

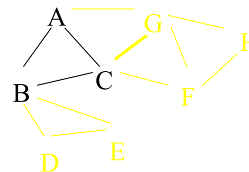
$$= \sum_{A,C} \phi_1(A,B,C)\tau_2(B)\underbrace{\left[\sum_F \underbrace{\phi_3(A,G)\tau_6(C,G)}_{\tau_7(A,C,G)}\right]}_{\tau_8(A,C)}$$

MRF variable elimination inference

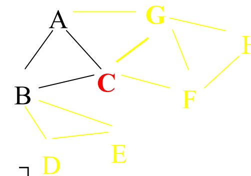
Example (cont):

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

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



Eliminate C



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

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

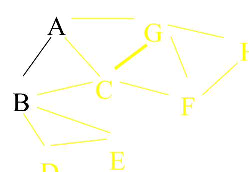
MRF variable elimination inference

Example (cont):

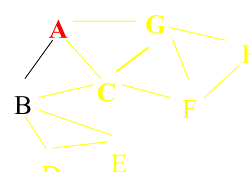
$$P(B) = \sum_{A,C,D,..H} P(A,B,...H)$$

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

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

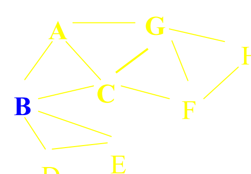


Eliminate A



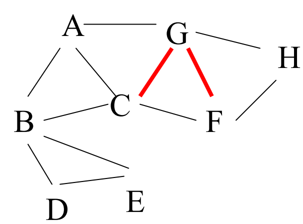
$$= \tau_2(B)\underbrace{\sum_{A} \tau_{10}(A,B)}_{\tau_{11}(B)}$$

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



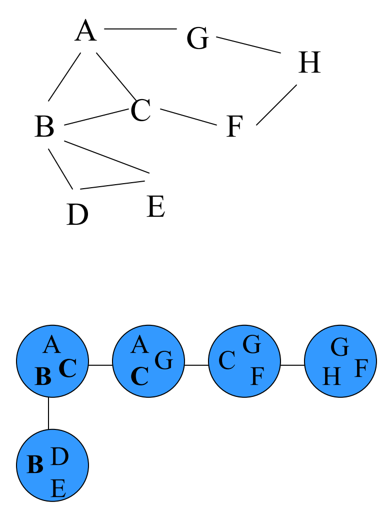
Induced graph

- **A graph induced by a specific variable elimination order:**
- **a graph G extended by links that represent intermediate factors**
  - .



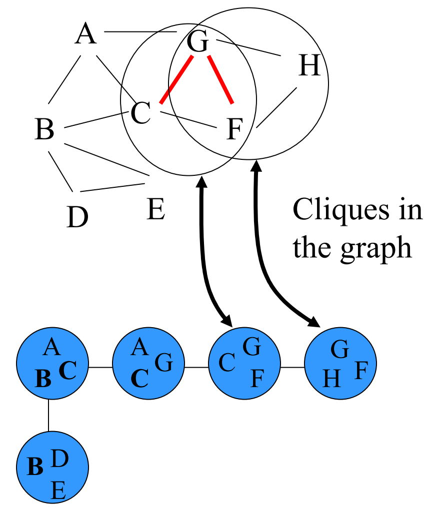
Tree decomposition of the graph

- **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.
  - For every $v \in G$: the nodes in T that contain v form a connected subtree.



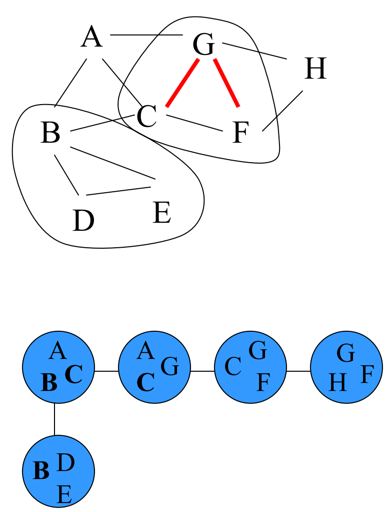
Tree decomposition of the graph

- **A tree decomposition of a graph G:**
  - A tree *T* with a vertex set associated to every node.
  - For all edges {*v*,*w*}∈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.



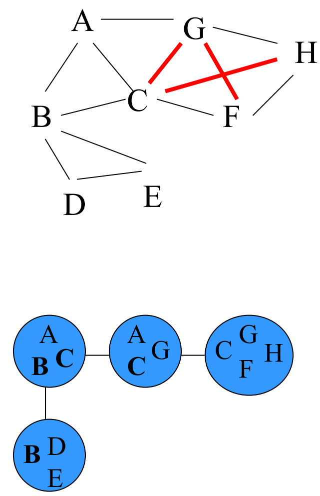
Tree decomposition of the graph

- **A tree decomposition of a graph G:**
  - A tree *T* with a vertex set associated to every node.
  - 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.



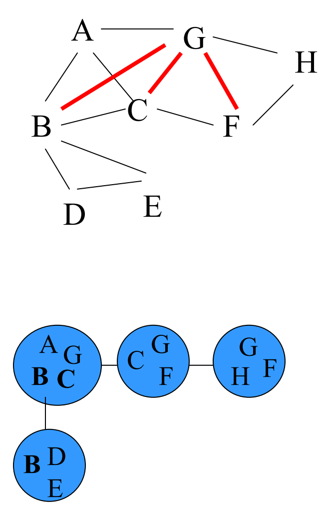
Tree decomposition of the graph

- **Another 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*.
  - For every $v \in$ G : the nodes in T that contain v form a connected subtree.



Tree decomposition of the graph

- **Another 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*.
  - For every $v \in$ G : the nodes in T that contain v form a connected subtree.

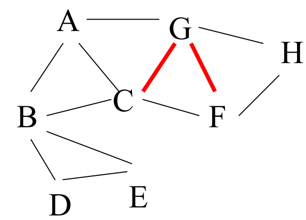


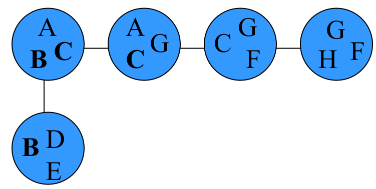
Treewidth of the graph

- **Width** of the tree decomposition:

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

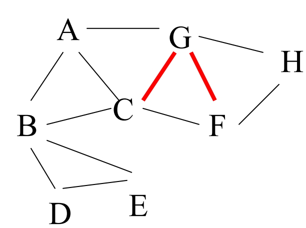
- **Treewidth** of a graph *G*: tw(*G*)= minimum width over all tree decompositions of *G*.

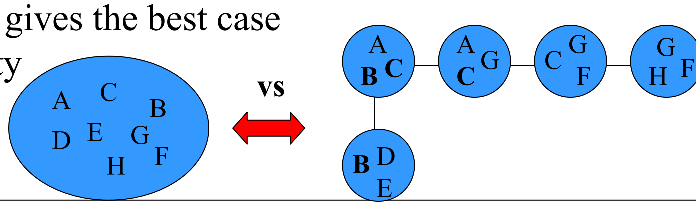




Treewidth of the graph

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

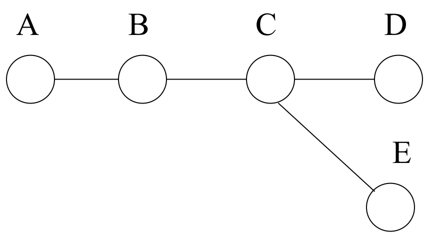




Trees

Why do we like trees?

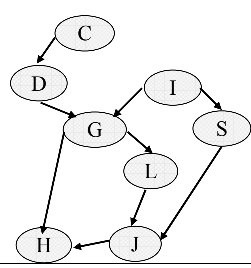
- Inference in trees structures can be done in time **linear in the number of nodes**

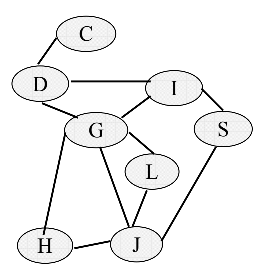


Converting BBNs to MRFs

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.

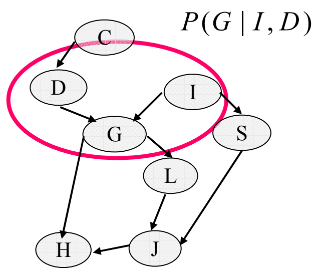


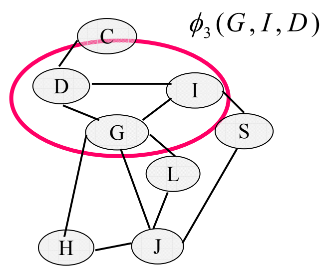


Moral Graphs

Why moralization?

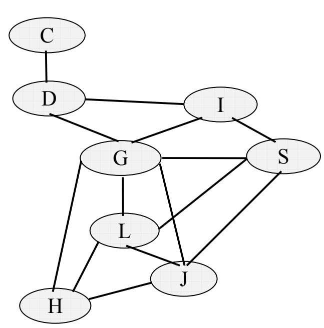
$$P(C,D,G,I,S,L,J,H) =$$
$$= P(C)P(D|C)P(G|I,D)P(S|I)P(L|G)P(J|L,S)P(H|G,J)$$
$$= \phi_1(C)\phi_2(D,C)\phi_3(G,I,D)\phi_4(S,I)\phi_5(L,G)\phi_6(J,L,S)\phi_7(H,G,J)$$



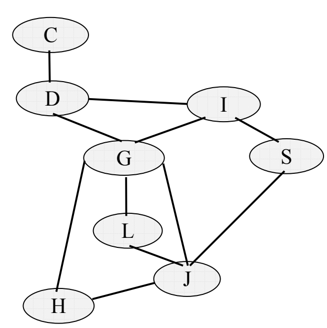


Chordal graphs

Chordal Graph: an undirected graph *G* whose minimum cycle contains 3 verticies.



Chordal.

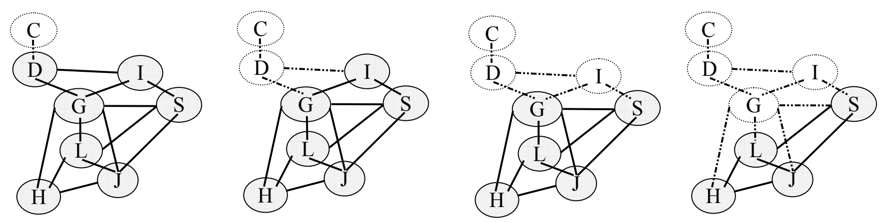


Not Chordal.

Chordal Graphs

Properties:

– There exists an elimination ordering that adds no edges.
– The minimal induced treewidth 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 (treewidth) optimal.

- There exist exact algorithms for **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.