CS 3710 Advanced Topics in AI
Lecture 6

Undirected graphical models and factors

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

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:

$$\phi(x, y)$$

  - Scope of the factor:

$$\{x, y\}$$

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

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

Markov random fields

- **Probabilistic models with symmetric dependences.**
  - Typically models of 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)

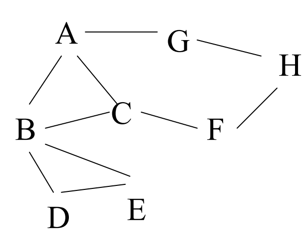
- G = (S, E)
  - S=1, 2, .. N correspond to random variables
  - $(i, j) \in E \Leftrightarrow \exists c : \{i, j\} \subset c$

    or x_i and x_j appear within the same factor c

Example:

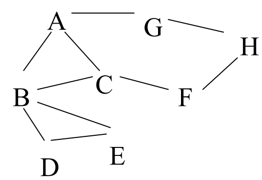
- variables A,B ..H
- Assume the full joint of MRF

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



MRF variable elimination inference

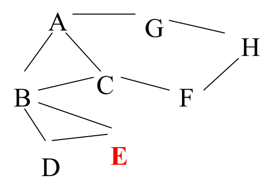
Example:



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

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

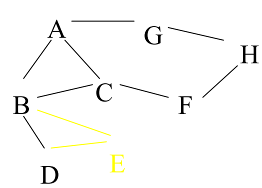
Eliminate E



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

MRF variable elimination inference

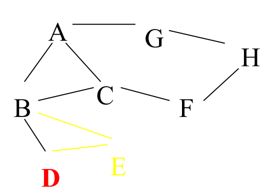
Example (cont):



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

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

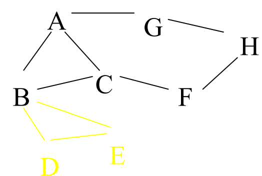
Eliminate D



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

MRF variable elimination inference

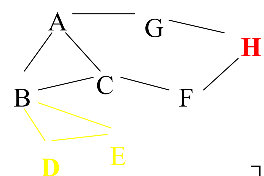
Example (cont):



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

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

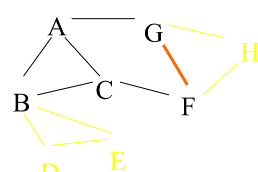
Eliminate H



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

MRF variable elimination inference

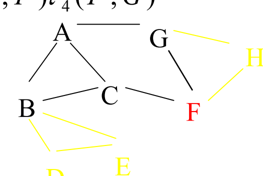
Example (cont):



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

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

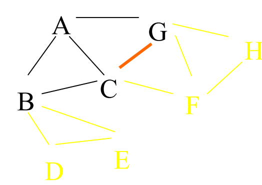
Eliminate F



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

MRF variable elimination inference

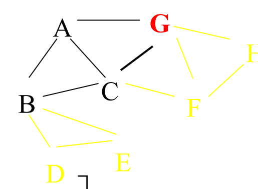
Example (cont):



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

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

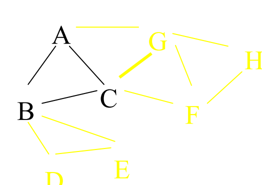
Eliminate G



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

MRF variable elimination inference

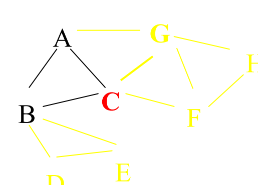
Example (cont):



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

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

Eliminate C



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

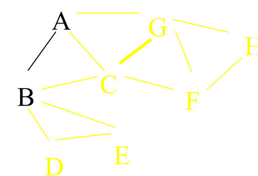
MRF variable elimination inference

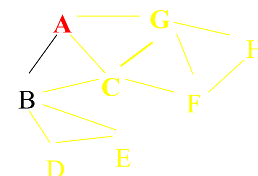
Example (cont):

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

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

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

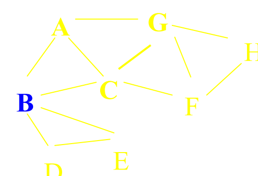




Eliminate A

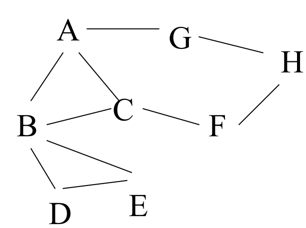
$$= \tau_2(B)\underbrace{\sum_{A} \tau_{10}(A,B)}_{\tau_{11}(B)}$$

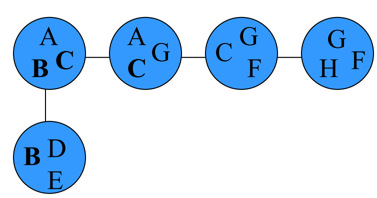
$$= \tau_2(B)\tau_{11}(B)$$



Tree decomposition of the graph

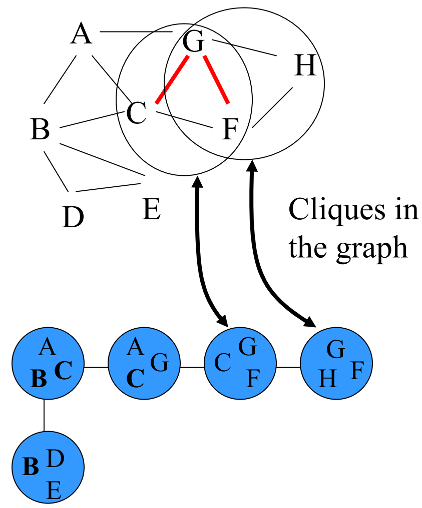
- **A tree decomposition of a graph G:**
  - A tree *T* with a vertex set associated to every node.
  - For all edges $\{v,w\} \in$ G: there is a set containing both *v* and *w* in *T*.
  - For every $v \in$ G : the nodes in T that contain *v* form a connected subtree.





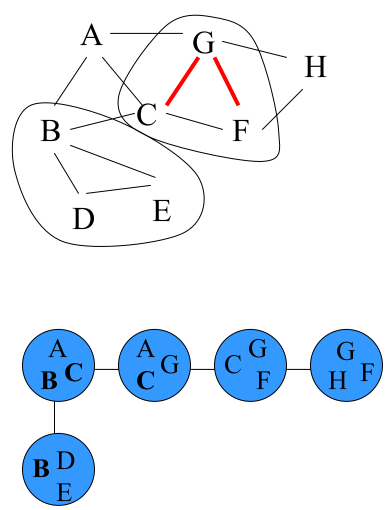
Tree decomposition of the graph

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



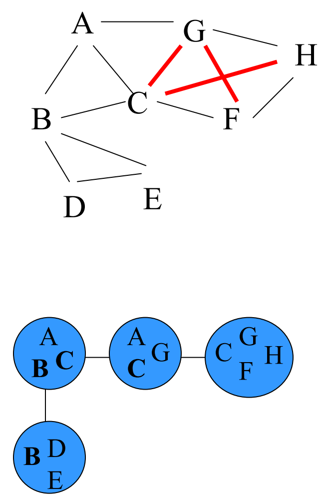
Tree decomposition of the graph

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



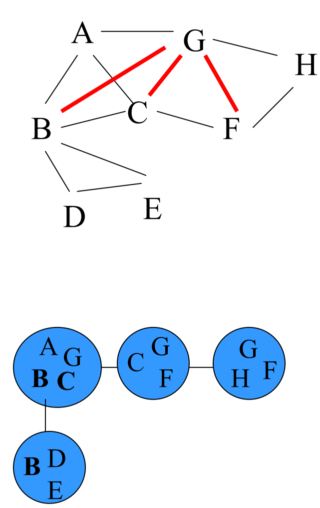
Tree decomposition of the graph

- **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.



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.

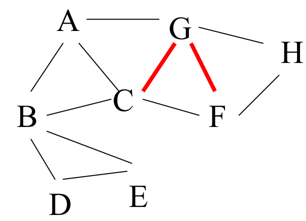


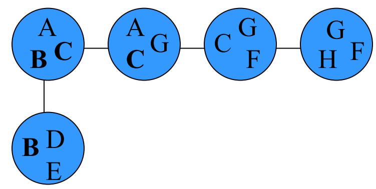
Treewidth of the graph

- **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