CS 3750 Machine Learning
Lecture 4

Markov Random Fields

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

Markov random fields

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

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

$\phi_c(x_c)$ - A potential function (defined over factors)

- If $\phi_c(x_c)$ is strictly positive we can rewrite the definition as:

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

- Gibbs (Boltzman) distribution

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

Graphical representation of MRFs

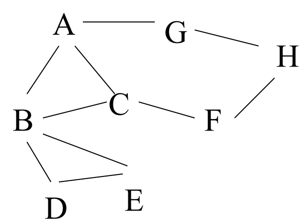
An undirected network (also called independence graph)

- 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:

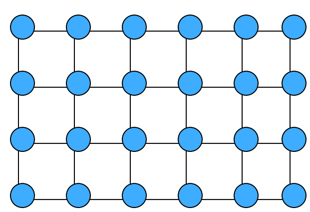
- variables A,B ..H
- Assume the full joint of MRF



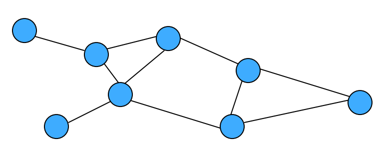
$$P(A, B, \ldots H) \sim \phi_1(A, B, C)\phi_2(B, D, E)\phi_3(A, G)\phi_4(C, F)\phi_5(G, H)\phi_6(F, H)$$

Markov random fields

- **regular lattice (Ising model)**

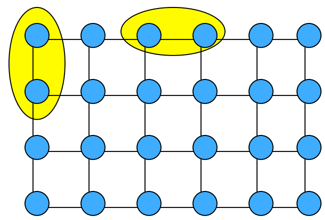


- **Arbitrary graph**

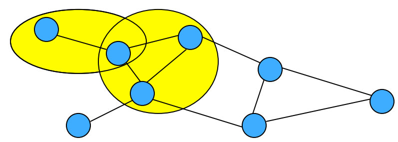


Markov random fields

- **regular lattice (Ising model)**

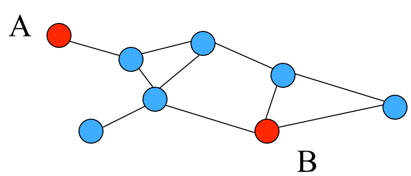


- **Arbitrary graph**



Markov random fields

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



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

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

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

Types of Markov random fields

- **MRFs with discrete random variables**
  - Clique potentials can be defined by mapping all clique-variable instances to R
  - Example: Assume two binary variables A,B with values {a1,a2,a3} and {b1,b2} are in the same clique c. Then:

$\phi_c(A,B) \cong$

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

Types of Markov random fields

- **Gaussian Markov Random Field**

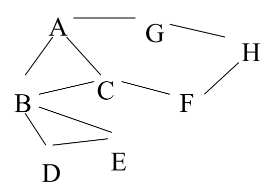
$$\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$p(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

- **Precision matrix** $\boldsymbol{\Sigma}^{-1}$
- **Variables in x are connected in the network only if they have a nonzero entry in the precision matrix**
  - All zero entries are not directly connected
  - Why?

MRF variable elimination inference

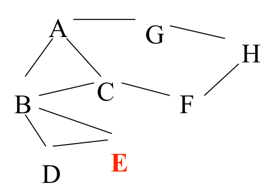
Example:



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

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

Eliminate E



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

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:

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

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

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

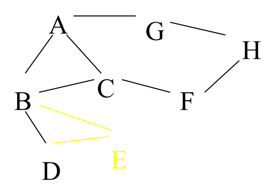
a1	b1	c1	0.2
a1	b1	c2	0.35
a1	b2	c1	0.4
a1	b2	c2	0.15
a2	b1	c1	0.5
a2	b1	c2	0.1
a2	b2	c1	0.3
a2	b2	c2	0.2
a3	b1	c1	0.25
a3	b1	c2	0.45
a3	b2	c1	0.15
a3	b2	c2	0.25

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

MRF variable elimination inference

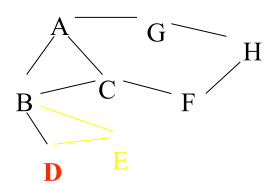
Example (cont):

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



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

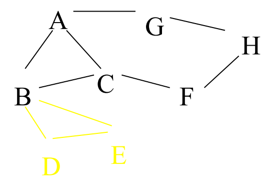
Eliminate D



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

MRF variable elimination inference

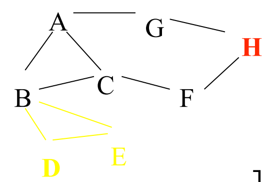
Example (cont):



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

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

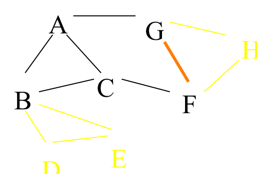
Eliminate H



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

MRF variable elimination inference

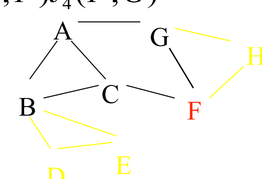
Example (cont):



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

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

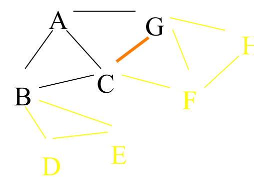
Eliminate F



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

MRF variable elimination inference

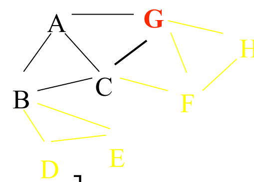
Example (cont):



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

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

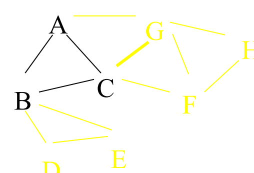
Eliminate G



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

MRF variable elimination inference

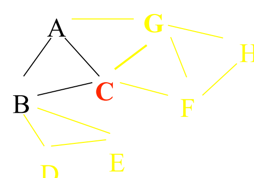
Example (cont):



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

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

Eliminate C



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

MRF variable elimination inference

Example (cont):

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

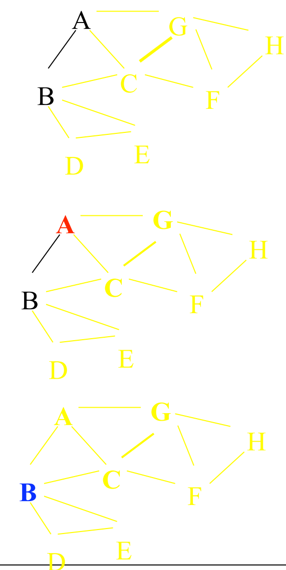
$$= \sum_{A} \tau_2(B)\tau_{10}(A,B)$$

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

Eliminate A

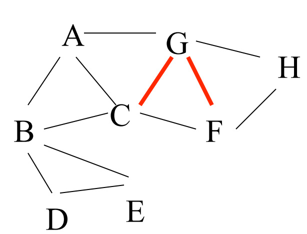
$$= \tau_2(B)\underbrace{\sum_{A} \tau_{10}(A,B)}_{\tau_{11}(B)}$$

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



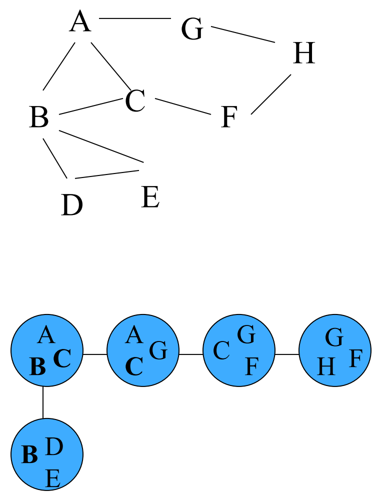
Induced graph

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



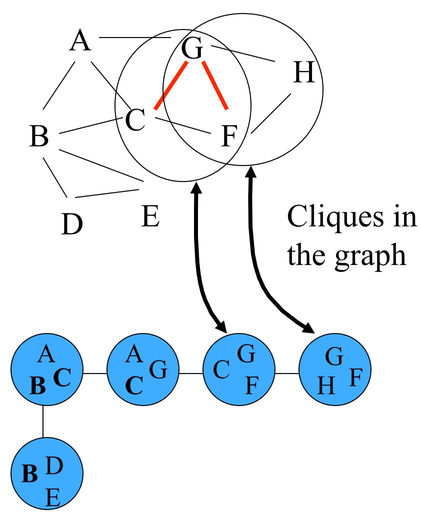
Tree decomposition of the graph

- **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

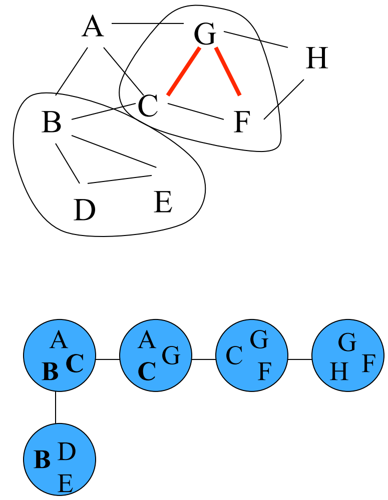
- **A tree decomposition of a graph G:**
  - A tree *T* with a vertex set associated to every node.
  - For all edges $\{v,w\} \in G$: there is a set containing both *v* and *w* in *T*.
  - For every $v \in G$: the nodes in T that contain *v* form a connected subtree.



CS 3750 Advanced Machine Learning

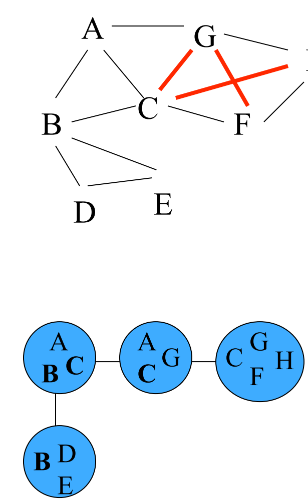
Tree decomposition of the graph

- **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.



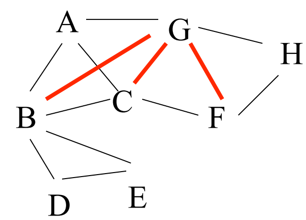
Tree decomposition of the graph

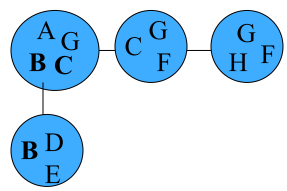
- **Another tree decomposition of a graph G:**
  - A tree *T* with a vertex set associated to every node.
  - For all edges $\{v,w\} \in G$: there is a set containing both v and w in T.
  - For every $v \in G$: the nodes in T that contain v form a connected subtree.



Tree decomposition of the graph

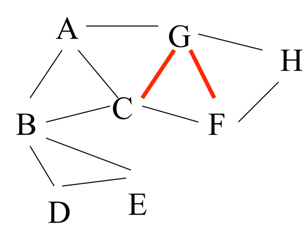
- **Another tree decomposition of a graph G:**
  - A tree *T* with a vertex set associated to every node.
  - For all edges $\{v,w\} \in G$: there is a set containing both *v* and *w* in *T*.
  - For every $v \in G$: the nodes in T that contain *v* form a connected subtree.

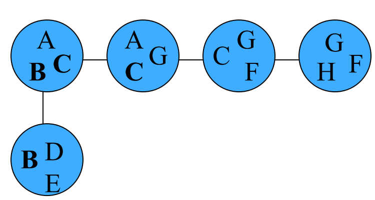




Treewidth of the graph

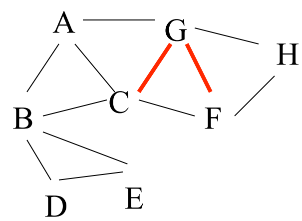
- **Width** of the tree decomposition:
  $$\max_{i \in I} |X_i| - 1$$
- **Treewidth** of a graph *G*: tw(*G*)= minimum width over all tree decompositions of *G*.

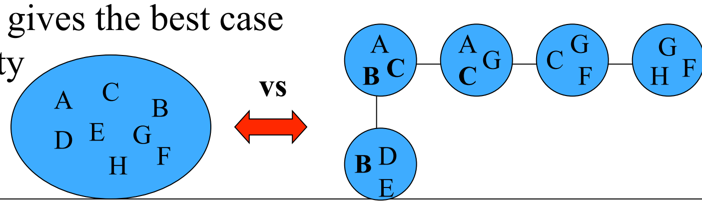




Treewidth of the graph

- **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

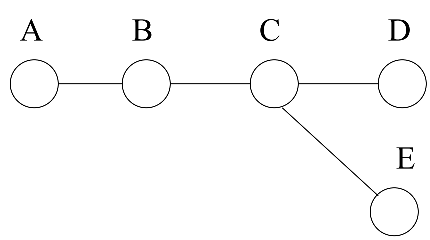




Trees

Why do we like trees?

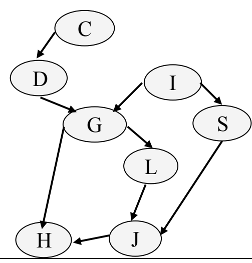
- Inference in trees structures can be done in time **linear in the number of nodes in the tree**

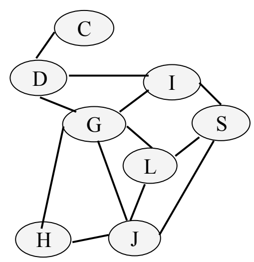


Converting BBNs to MRFs

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.

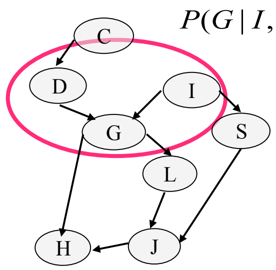


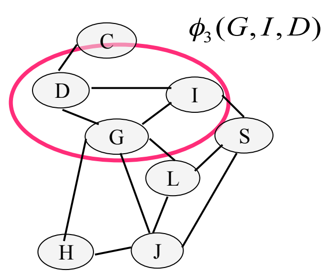


Moral Graphs

Why moralization?

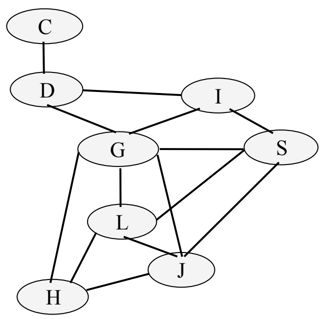
$$P(C,D,G,I,S,L,J,H) =$$
$$= P(C)P(D|C)P(G|I,D)P(S|I)P(L|G)P(J|L,S)P(H|G,J)$$
$$= \phi_1(C)\phi_2(D,C)\phi_3(G,I,D)\phi_4(S,I)\phi_5(L,G)\phi_6(J,L,S)\phi_7(H,G,J)$$



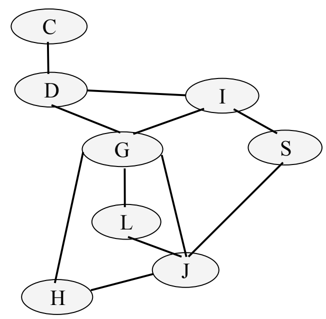


Chordal graphs

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



Chordal.



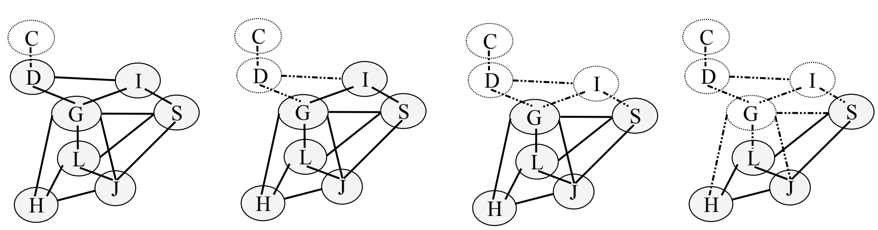
Not Chordal.

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*.

Inference: conclusions

- We cannot escape **exponential costs in the treewidth.**

- But in many graphs the treewidth is much smaller than the total number of variables

- Still a problem: Finding the optimal decomposition is hard
  - But, paying the cost up front may be worth it.
  - Triangulate once, query many times.
  - Real cost savings if not a bounded one.