Graph Decompositions and Junction Trees

Lecture 3
Saint Flour Summerschool, July 6, 2006

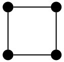
Steffen L. Lauritzen, University of Oxford

Overview of lectures

1. Conditional independence and Markov properties
2. More on Markov properties
3. Graph decompositions and junction trees
4. Probability propagation and similar algorithms
5. Log-linear and Gaussian graphical models
6. Conjugate prior families for graphical models
7. Hyper Markov laws
8. Structure learning and Bayes factors
9. More on structure learning.

Some motivation

- *Perfect DAGs* are simple, because their directions can be ignored as they are Markov equivalent to their skeleton;
- Undirected graphs which can occur as *skeletons of perfect DAGs* are therefore particularly simple;
- An n-cycle with $n \geq 4$ *cannot be oriented* to form a perfect DAG:



- The important simplifying idea is that of *graph decomposition* and *decomposability*.

Graph decomposition

Consider an *undirected* graph $\mathcal{G} = (V, E)$. A partitioning of V into a triple (A, B, S) of subsets of V forms a *decomposition* of $\mathcal{G}$ if

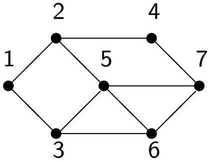
$$A \perp_{\mathcal{G}} B \mid S \text{ and } S \text{ is complete.}$$

The decomposition is *proper* if $A \neq \emptyset$ and $B \neq \emptyset$.

The *components* of $\mathcal{G}$ are the induced subgraphs $\mathcal{G}_{A \cup S}$ and $\mathcal{G}_{B \cup S}$.

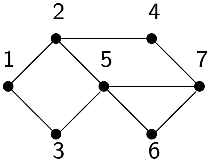
A graph is *prime* if no proper decomposition exists.

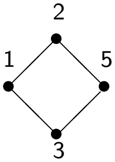
Examples

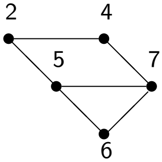


The graph to the left is prime

Decomposition with $A = \{1, 3\}$, $B = \{4, 6, 7\}$ and $S = \{2, 5\}$







Decomposition of Markov properties

Suppose P satisfies (F) w.r.t. $\mathcal{G}$ and (A, B, S) is a decomposition. Then

(i) *$P_{A\cup S}$ and $P_{B\cup S}$ satisfy (F) w.r.t. $\mathcal{G}_{A\cup S}$ and $\mathcal{G}_{B\cup S}$ respectively;*

(ii)
$$f(x)f_S(x_S) = f_{A\cup S}(x_{A\cup S})f_{B\cup S}(x_{B\cup S}).$$

The first part of the statement is true when (F) is replaced by (G).

The second is also true for (G) if the relevant densities exist.

Markov combination

Let Q and R be distributions on $\mathcal{X}_{A\cup S}$ and $\mathcal{X}_{B\cup S}$ resp. and assume Q and R are *consistent*, i.e. $Q_S = R_S$.

Then *there is a unique distribution* $P = Q * R$ so that

(i) $P_{A\cup S} = Q$ and $P_{B\cup S} = R$;

(ii) $A \perp\!\!\!\perp_P B \mid S$.

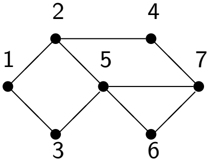
$Q * R$ is the *Markov combination* of Q and R. If Q and R have densities q and r, *so has* P and

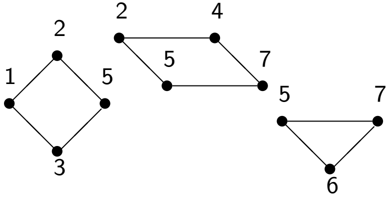
$$p(x)q_S(x_S) = p(x)r_S(x_S) = q(x_{A\cup S})r(x_{B\cup S}).$$

The Markov combination *maximizes entropy* among measures satisfying (i).

Decomposability

Any graph can be recursively decomposed into its maximal prime subgraphs:





A graph is *decomposable* (or rather fully decomposable) if it is complete or admits a proper decomposition into *decomposable* subgraphs.

Definition is recursive. Alternatively this means that *all maximal prime subgraphs are cliques.*

Factorization of Markov distributions

Recursive decomposition of a decomposable graph into cliques yields the formula:

$$f(x)\prod_{S\in\mathcal{S}} f_S(x_S)^{\nu(S)} = \prod_{C\in\mathcal{C}} f_C(x_C).$$

Here $\mathcal{S}$ is the set of *minimal complete separators* occurring in the decomposition process and $\nu(S)$ the number of times such a separator appears in this process.

Combinatorial consequences

Note that if we let $\mathcal{X}_v = \{0, 1\}$ and f be uniform, this yields

$$2^{-|V|} \prod_{S \in \mathcal{S}} 2^{-|S|\nu(S)} = \prod_{C \in \mathcal{C}} 2^{-|C|}$$

and hence we must have

$$\sum_{C \in \mathcal{C}} |C| - \sum_{S \in \mathcal{S}} |S|\nu(S) = |V|.$$

It also holds that

$$\sum_{S \in \mathcal{S}} \nu(S) = |V| - 1.$$

Properties associated with decomposability

A numbering $V = \{1, \ldots, |V|\}$ of the vertices of an undirected graph is *perfect* if the induced oriented graph is a perfect DAG or, equivalently, if

$$\forall j = 2, \ldots, |V| : \mathrm{bd}(j) \cap \{1, \ldots, j-1\} \text{ is complete in } \mathcal{G}.$$

An undirected graph $\mathcal{G}$ is *chordal* if it has no chordless n-cycles with $n \geq 4$.

These graphs are also known as *rigid circuit* graphs or *triangulated* graphs.

A set S is an (α, β)*-separator* if $\alpha \perp_{\mathcal{G}} \beta \,|\, S$,

Characterizing chordal graphs

The following are equivalent for any undirected graph $\mathcal{G}$.

(i) *$\mathcal{G}$ is chordal;*

(ii) *$\mathcal{G}$ is decomposable;*

(iii) *All maximal prime subgraphs of $\mathcal{G}$ are cliques;*

(iv) *$\mathcal{G}$ admits a perfect numbering;*

(v) *Every minimal (α, β)-separator are complete.*

Trees are chordal graphs and thus decomposable.

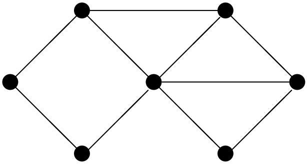
Identifying chordal graphs

Here is a (greedy) algorithm for checking chordality:

1. Look for a vertex v^* with $\mathrm{bd}(v^*)$ complete. *If no such vertex exists, the graph is not chordal.*
2. Form the subgraph $\mathcal{G}_{V \setminus v^*}$ and let $v^* = |V|$;
3. Repeat the process under 1;
4. *If the algorithm continues until only one vertex is left, the graph is chordal and the numbering is perfect.*

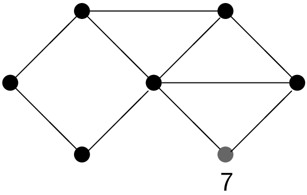
The complexity of this algorithm is $O(|V|^2)$.

Greedy algorithm



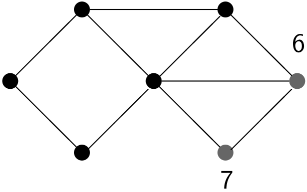
Is this graph chordal?

Greedy algorithm



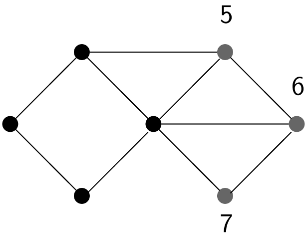
Is this graph chordal?

Greedy algorithm



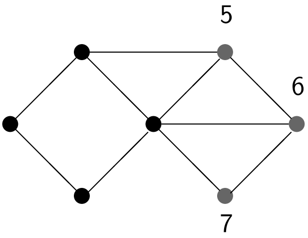
Is this graph chordal?

Greedy algorithm



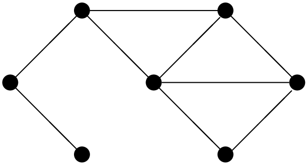
Is this graph chordal?

Greedy algorithm



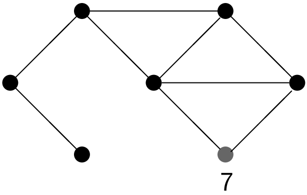
This graph is *not* chordal, as there is no candidate for number 4.

Greedy algorithm



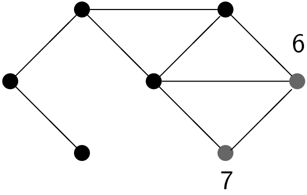
Is this graph chordal?

Greedy algorithm



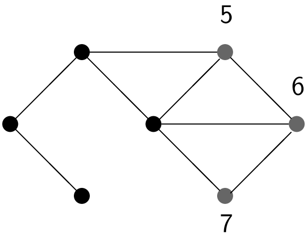
Is this graph chordal?

Greedy algorithm



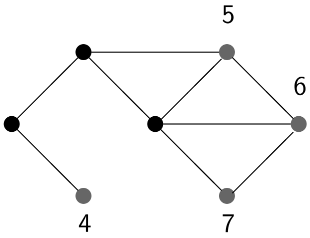
Is this graph chordal?

Greedy algorithm



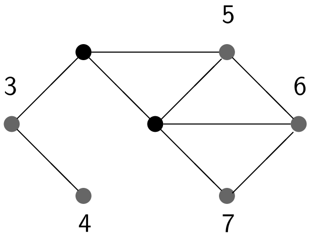
Is this graph chordal?

Greedy algorithm



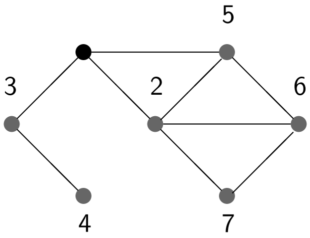
Is this graph chordal?

Greedy algorithm



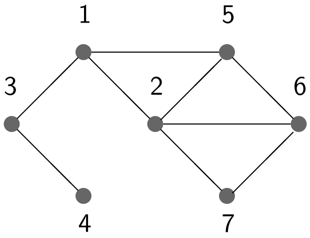
Is this graph chordal?

Greedy algorithm



Is this graph chordal?

Greedy algorithm



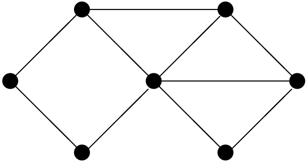
This graph is chordal!

Maximum cardinality search

This simple algorithm has complexity $O(|V| + |E|)$:

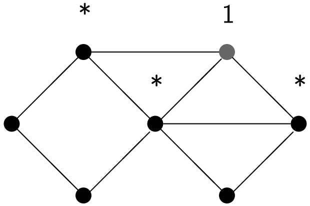
1. Choose $v_0 \in V$ arbitrary and let $v_0 = 1$;
2. When vertices $\{1, 2, \dots, j\}$ have been identified, choose $v = j + 1$ among $V \setminus \{1, 2, \dots, j\}$ with highest cardinality of its numbered neighbours;
3. *If* $\mathrm{bd}(j+1) \cap \{1, 2, \dots, j\}$ *is not complete,* $\mathcal{G}$ is not chordal;
4. Repeat from 2;
5. *If the algorithm continues until only one vertex is left, the graph is chordal and the numbering is perfect.*

Maximum Cardinality Search



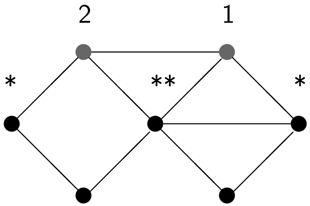
Is this graph chordal?

Maximum Cardinality Search



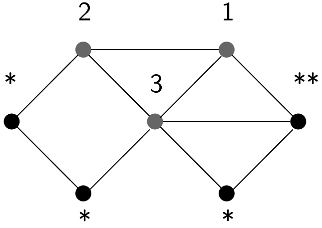
Is this graph chordal?

Maximum Cardinality Search



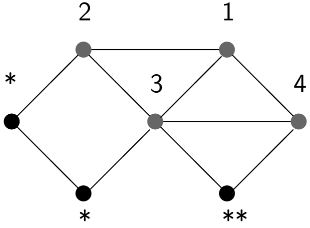
Is this graph chordal?

Maximum Cardinality Search



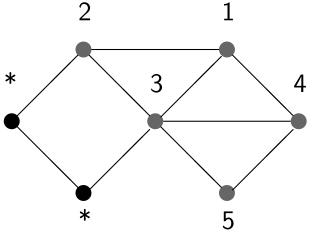
Is this graph chordal?

Maximum Cardinality Search



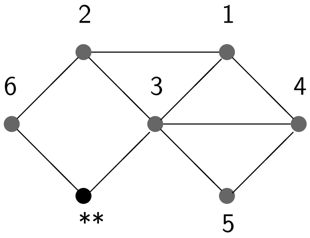
Is this graph chordal?

Maximum Cardinality Search



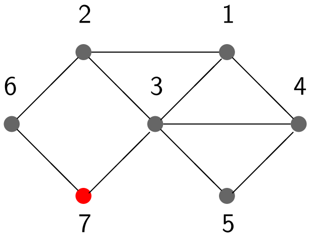
Is this graph chordal?

Maximum Cardinality Search



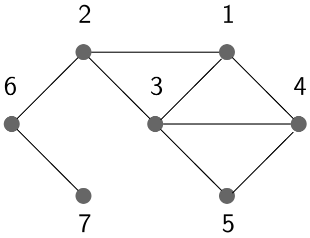
Is this graph chordal?

Maximum Cardinality Search



The graph is not chordal! because 7 does not have a complete boundary.

Maximum Cardinality Search



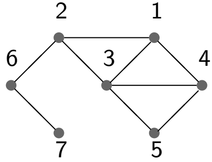
MCS numbering for the chordal graph. Algorithm runs essentially as before.

Finding the cliques of a chordal graph

From an MCS numbering $V = \{1, \ldots, |V|\}$, let

$$S_\lambda = \mathrm{bd}(\lambda) \cap \{1, \ldots, \lambda - 1\}$$

and $\pi_\lambda = |S_\lambda|$. Call λ a *ladder vertex* if $\lambda = |V|$ or if $\pi_{\lambda+1} < \pi_\lambda + 1$ and let Λ be the set of ladder vertices.



π_λ: 0,1,2,2,2,1,1. *The cliques are* $C_\lambda = \{\lambda\} \cup S_\lambda, \lambda \in \Lambda.$

Junction tree

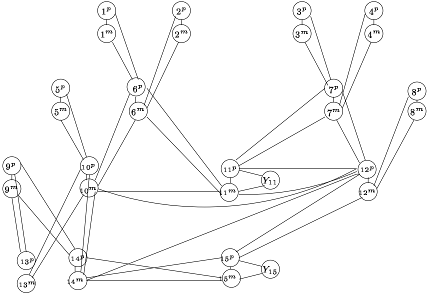
Let $\mathcal{A}$ be a collection of finite subsets of a set V. A *junction tree* $\mathcal{T}$ of sets in $\mathcal{A}$ is an undirected tree with $\mathcal{A}$ as a vertex set, satisfying the *junction tree property:*

If $A, B \in \mathcal{A}$ and C is on the unique path in $\mathcal{T}$ between A and B *it holds that* $A \cap B \subset C$.

If the sets in $\mathcal{A}$ are pairwise incomparable, *they can be arranged in a junction tree if and only if* $\mathcal{A} = \mathcal{C}$ *where* $\mathcal{C}$ *are the cliques of a chordal graph.*

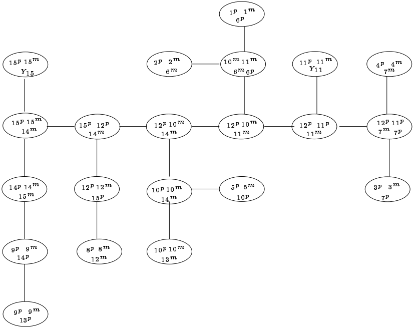
The junction tree can be *constructed directly from the MCS ordering* $C_\lambda, \lambda \in \Lambda$.

A chordal graph



This graph is chordal, but it might not be that easy to see. . . Maximum Cardinality Search is handy!

Junction tree



Cliques of graph arranged into a tree with $C_1 \cap C_2 \subseteq D$ for all cliques D on path between C_1 and C_2.