Decomposition of log-linear models

Steffen Lauritzen, University of Oxford

Graphical Models, Lecture 5, Michaelmas Term 2010

October 29, 2010

A density f *factorizes* w.r.t. $\mathcal{A}$ if there exist functions $\psi_a(x)$ which depend on x_a only so that

$$f(x) = \prod_{a \in \mathcal{A}} \psi_a(x).$$

The set of distributions $\mathcal{P}_{\mathcal{A}}$ which factorize w.r.t. $\mathcal{A}$ is the *hierarchical log–linear model* generated by $\mathcal{A}$.

$\mathcal{A}$ is the *generating class* of the log–linear model.

For any generating class $\mathcal{A}$ we construct the dependence graph $G(\mathcal{A}) = G(\mathcal{P}_{\mathcal{A}})$ of the log–linear model $\mathcal{P}_{\mathcal{A}}$.

The dependence graph is determined by the relation

$$\alpha \sim \beta \iff \exists a \in \mathcal{A} : \alpha, \beta \in a.$$

For sets in $\mathcal{A}$ are clearly complete in $G(\mathcal{A})$ and therefore *distributions in $\mathcal{P}_{\mathcal{A}}$ do factorize according to $G(\mathcal{A})$*.

They are thus also global, local, and pairwise Markov w.r.t. $G(\mathcal{A})$.

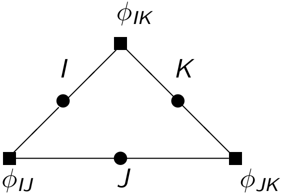
As a generating class defines a dependence graph $G(\mathcal{A})$, the reverse is also true.

The set $\mathcal{C}(\mathcal{G})$ of *cliques* (maximal complete subsets) of $\mathcal{G}$ is a generating class for the log–linear model of distributions which factorize w.r.t. $\mathcal{G}$.

If the dependence graph completely summarizes the restrictions imposed by $\mathcal{A}$, i.e. if

$$\mathcal{A} = \mathcal{C}(G(\mathcal{A})),$$

$\mathcal{A}$ is *conformal*.



The *factor graph* of $\mathcal{A}$ is the bipartite graph with vertices $V \cup \mathcal{A}$ and edges define by

$$\alpha \sim a \iff \alpha \in a.$$

Using this graph even non-conformal log–linear models admit a simple visual representation.

The maximum likelihood estimate $\hat{p}$ of p is the unique element of $\overline{\mathcal{P}_{\mathcal{A}}}$ which satisfies the system of equations

$$n\hat{p}(x_a) = n(x_a), \forall a \in \mathcal{A}, x_a \in \mathcal{X}_a. \quad (1)$$

Here $g(x_a) = \sum_{y:y_a=x_a} g(y)$ is the *a-marginal* of the function g.

The system of equations (1) expresses the *fitting of the marginals* in $\mathcal{A}$.

There is a *convergent* algorithm which solves the likelihood equations. This cycles (repeatedly) through all the a-marginals in $\mathcal{A}$ and fit them one by one.

For $a \in \mathcal{A}$ define the following *scaling* operation on p:

$$(T_a p)(x) \leftarrow p(x)\frac{n(x_a)}{np(x_a)}, \quad x \in \mathcal{X}$$

where $0/0 = 0$ and $b/0$ is undefined if $b \neq 0$.

Make an ordering of the generators $\mathcal{A} = \{a_1, \ldots, a_k\}$. Define S by a full cycle of scalings

$$Sp = T_{a_k} \cdots T_{a_2} T_{a_1}.$$

Define the iteration

$$p_0(x) \leftarrow 1/|\mathcal{X}|, \quad p_n = Sp_{n-1}, n = 1, \ldots.$$

It then holds that

$$\lim_{n\to\infty} p_n = \hat{p}$$

where $\hat{p}$ is the unique maximum likelihood estimate of $p \in \overline{\mathcal{P}_{\mathcal{A}}}$, i.e. the solution of the equation system (1).

In some cases the IPS algorithm converges after a finite number of cycles. An explicit formula is then available for the MLE of $p \in \mathcal{P}_{\mathcal{A}}$.

Consider first the case of a generating class with only two elements: $\mathcal{A} = \{a, b\}$ and thus $V = a \cup b$. Let $c = a \cap b$. Recall that the MLE is the unique solution to

$$n\hat{p}(x_a) = n(x_a), \forall a \in \mathcal{A}, x_a \in \mathcal{X}_a.$$

Let

$$p^*(x) = \frac{n(x_a)n(x_b)}{n(x_c)n}.$$

$$p^*(x) = \frac{n(x_a)n(x_b)}{n(x_c)n}.$$

This satisfies (1) since e.g.

$$\begin{aligned} np^*(x_a) &= \sum_{y:y_a=x_a} \frac{n(y_a)n(y_b)}{n(y_c)} = \sum_{y:y_a=x_a} \frac{n(x_a)n(y_b)}{n(x_c)} \\ &= \frac{n(x_a)}{n(x_c)} \sum_{y:y_a=x_a} n(y_b) = \frac{n(x_a)}{n(x_c)} n(x_c) = n(x_a) \end{aligned}$$

and similarly with the other marginal. Hence we have $\hat{p} = p^*$.

The generating class $\mathcal{A} = \{a, b\}$ is conformal. Its dependence graph $\mathcal{G}$ has exactly two cliques a and b.

The graph is *chordal*, meaning that any cycle of length ≥ 4 has a chord.

$\mathcal{A}$ is called *decomposable* if $\mathcal{A}$ is conformal, i.e. $\mathcal{A} = \mathcal{C}(\mathcal{G})$, and $\mathcal{G}$ is chordal.

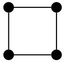
The IPS-algorithm converges after a finite number of cycles (at most two) if and only if $\mathcal{A}$ is decomposable.

Non-decomposable generating classes

A generating class can be non-decomposable in different ways.

The generating class $\mathcal{A} = \{\{1,2\},\{2,3\},\{1,3\}\}$ is the smallest non-decomposable generating class. This is non-conformal.

The graph below is the smallest non-chordal graph and its generating class is non-decomposable:

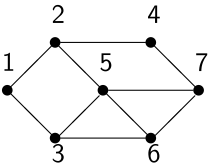


Consider an *undirected* graph $\mathcal{G} = (V, E)$. A partitioning of V into a triple (A, B, S) of subsets of V forms a *decomposition* of $\mathcal{G}$ if

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

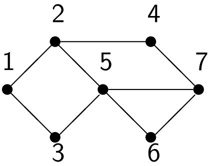
The decomposition is *proper* if $A \neq \emptyset$ and $B \neq \emptyset$.
The *components* of $\mathcal{G}$ are the induced subgraphs $\mathcal{G}_{A \cup S}$ and $\mathcal{G}_{B \cup S}$.
A graph is *prime* if no proper decomposition exists.

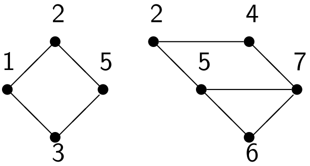
Examples



The graph to the left is prime

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





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

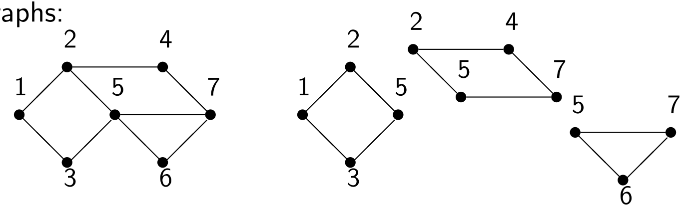
(i) *$P_{A\cup S}$ and $P_{B\cup S}$ satisfy (F) w.r.t. $\mathcal{G}_{A\cup S}$ and $\mathcal{G}_{B\cup S}$ respectively;*

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

The converse also holds in the sense that *if (i) and (ii) hold, and (A, B, S) is a decomposition of $\mathcal{G}$, then P factorizes w.r.t. $\mathcal{G}$.*

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.*

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.

As we have a particularly simple factorization of the density, we have a similar factorization of the maximum likelihood estimate for a decomposable log-linear model.

The MLE for p under the log-linear model with generating class $\mathcal{A} = \mathcal{C}(\mathcal{G})$ *for a chordal graph* $\mathcal{G}$ *is*

$$\hat{p}(x) = \frac{\prod_{C \in \mathcal{C}} n(x_C)}{n \prod_{S \in \mathcal{S}} n(x_S)^{\nu(S)}}$$

where $\nu(S)$ *is the number of times* S *appears as a separator in the total decomposition of its dependence graph.*

Perfect numbering

A numbering $V = \{1, \ldots, |V|\}$ of the vertices of an undirected graph is *perfect* if

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

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

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.*

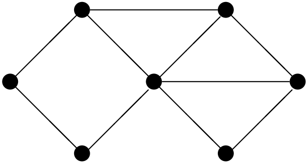
Trees are chordal graphs and thus decomposable.

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.*

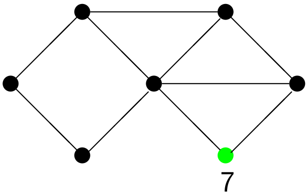
The complexity of this algorithm is $O(|V|^2)$.

Greedy algorithm



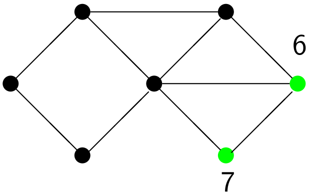
Is this graph chordal?

Greedy algorithm



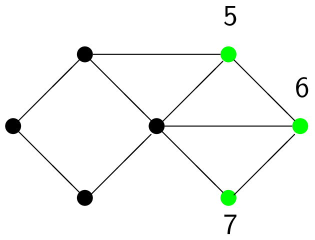
Is this graph chordal?

Greedy algorithm



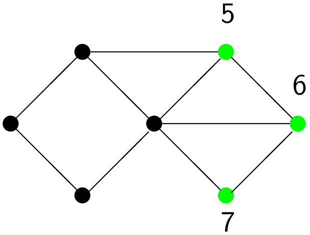
Is this graph chordal?

Greedy algorithm



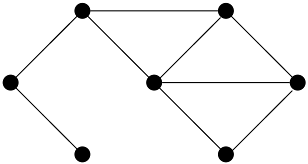
Is this graph chordal?

Greedy algorithm



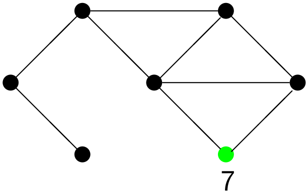
This graph is *not* chordal, as there is no candidate for number 4.

Greedy algorithm



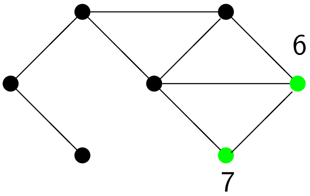
Is this graph chordal?

Greedy algorithm



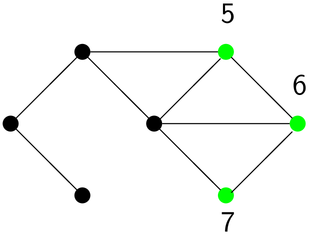
Is this graph chordal?

Greedy algorithm



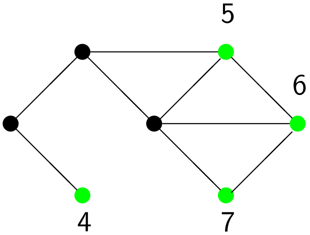
Is this graph chordal?

Greedy algorithm



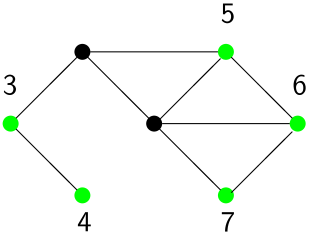
Is this graph chordal?

Greedy algorithm



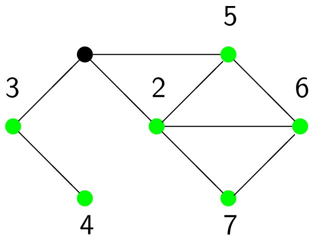
Is this graph chordal?

Greedy algorithm



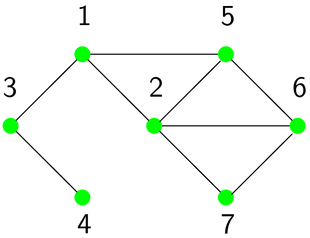
Is this graph chordal?

Greedy algorithm



Is this graph chordal?

Greedy algorithm

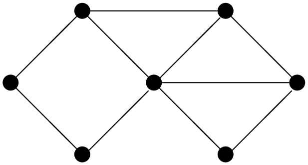


This graph is chordal!

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

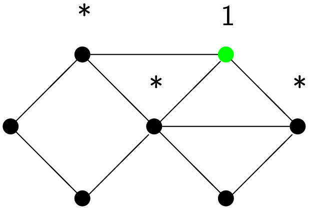
1. Choose $v_0 \in V$ arbitrary and let $v_0 = 1$;
2. When vertices $\{1, 2, \ldots, j\}$ have been identified, choose $v = j + 1$ among $V \setminus \{1, 2, \ldots, j\}$ with highest cardinality of its numbered neighbours;
3. *If* $\mathrm{bd}(j + 1) \cap \{1, 2, \ldots, j\}$ *is not complete,* $\mathcal{G}$ 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



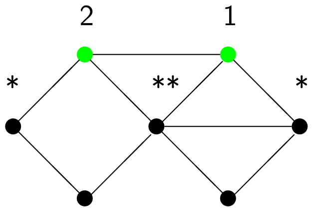
Is this graph chordal?

Maximum Cardinality Search



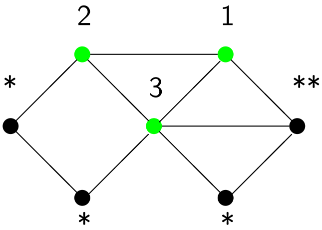
Is this graph chordal?

Maximum Cardinality Search



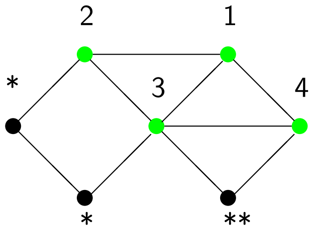
Is this graph chordal?

Maximum Cardinality Search



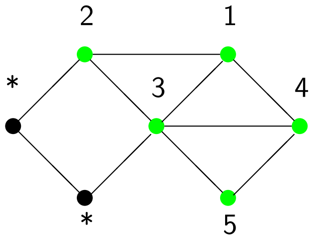
Is this graph chordal?

Maximum Cardinality Search



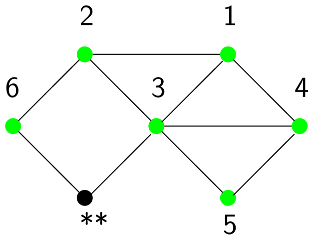
Is this graph chordal?

Maximum Cardinality Search



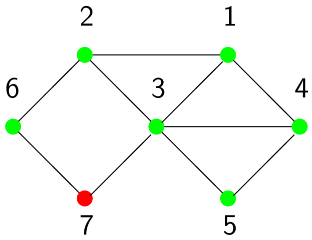
Is this graph chordal?

Maximum Cardinality Search



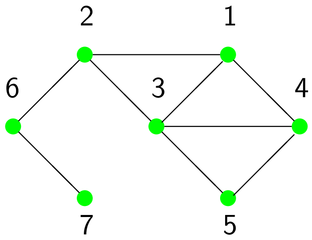
Is this graph chordal?

Maximum Cardinality Search



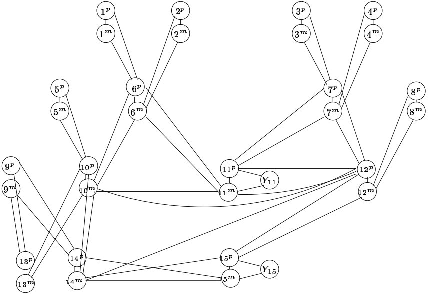
The graph is not chordal! because 7 does not have a complete boundary.

Maximum Cardinality Search



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

A chordal graph



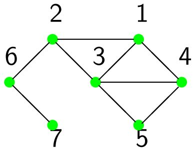
This graph is chordal, but it might not be that easy to see. . . Maximum Cardinality Search is handy!

Finding the cliques of a chordal graph

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

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

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



π_λ: 0,1,2,2,2,1,1.

The cliques are $C_\lambda = \{\lambda\} \cup B_\lambda, \lambda \in \Lambda$.