4
Graph Theory

The power and appeal of probabilistic networks stems from the *pictorial representation* they provide of the structural inter-relationships and dependencies between the variables of a problem, and the fact that these pictures have a formal definition as *graphs*. Many people find this form easy to understand and manipulate. But the graphs also serve as a precise and compact way of communicating these relations to a computer, paving the way for the use of efficient computational algorithms.

Problems that can be usefully tackled by probabilistic networks are those for which the graphs are relatively sparse. Such structures exhibit many independence relationships, thereby facilitating local inference, involving only a few variables at any one time. This chapter collects together a number of ideas and results from graph theory, and primarily contains definitions, notation, properties, and algorithms. Readers may prefer to skip this chapter on first reading and refer back to it only to clarify unfamiliar terms.

4.1 Basic concepts

Graph theory can be developed as a purely abstract mathematical subject. However, much of the immediate power of graph theory when applied to probabilistic expert systems lies in its ability to present a *visual* summary of expert knowledge or opinion about some subject. Accordingly, we shall develop graphical ideas and theorems making liberal use of pictorial representations. There are many variants of definitions and notations in use,

hence the need here to state explicitly those that will be used throughout the book.

We define a *graph* $\mathcal{G}$ to be a pair $\mathcal{G} = (V, E)$, where V is a finite set of *vertices*, also called *nodes*, of $\mathcal{G}$, and E is a subset of the set $V \times V$ of ordered pairs of vertices, called the *edges* or *links* of $\mathcal{G}$. Thus, as E is a set, the graph $\mathcal{G}$ has no multiple edges. We further require that E consist of pairs of distinct vertices so that there are no loops.

If both ordered pairs (α, β) and (β, α) belong to E, we say that we have an *undirected* edge between α and β, and write $\alpha \sim \beta$ (or $\alpha \sim_{\mathcal{G}} \beta$ to indicate the relevant graph $\mathcal{G}$; similar elaborations may be made to the other notation introduced below). We also say that α and β are *neighbours*, α is a neighbour of β, or β is a neighbour of α. The set of neighbours of a vertex β is denoted by $\mathrm{ne}(\beta)$.

If $(\alpha, \beta) \in E$ but $(\beta, \alpha) \notin E$, we call the edge *directed*, and write $\alpha \to \beta$. We also say that α is a *parent* of β, and that β is a *child* of α. The set of parents of a vertex β is denoted by $\mathrm{pa}(\beta)$, and the set of children of a vertex α by $\mathrm{ch}(\alpha)$. The *family* of β, denoted $\mathrm{fa}(\beta)$, is $\mathrm{fa}(\beta) = \{\beta\} \cup \mathrm{pa}(\beta)$.

If $(\alpha, \beta) \in E$ or $(\beta, \alpha) \in E$ we say that α and β are *joined*. Then $\alpha \not\sim \beta$ indicates that α and β are not joined, i.e., both $(\alpha, \beta) \notin E$ and $(\beta, \alpha) \notin E$. We also write $\alpha \not\to \beta$ if $(\alpha, \beta) \notin E$.

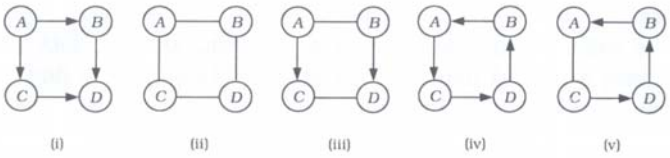


FIGURE 4.1. Examples of valid graphs on four vertices.

Figure 4.1 illustrates some graphs permitted by our definitions. Vertices are visually represented by (possibly) labelled circles, directed edges by arrows, and undirected edges by lines. In (iii) $A \sim B$ and $A \to C$, but $A \not\sim D$. In contrast, Figure 4.2 shows some 'graphs' which fail our definition, but which may pass definitions used by other authors for different purposes.

If $A \subset V$, the expressions $\mathrm{pa}(A)$, $\mathrm{ne}(A)$ and $\mathrm{ch}(A)$ will denote the collection of parents, children, and neighbours, respectively, of the elements of A, but exclude any element in A:

$$\begin{aligned} \mathrm{pa}(A) &= \textstyle\bigcup_{\alpha \in A} \mathrm{pa}(\alpha) \setminus A \\ \mathrm{ne}(A) &= \textstyle\bigcup_{\alpha \in A} \mathrm{ne}(\alpha) \setminus A \\ \mathrm{ch}(A) &= \textstyle\bigcup_{\alpha \in A} \mathrm{ch}(\alpha) \setminus A. \end{aligned}$$

Referring to Figure 4.1(iii), A is a parent of C, and thus C a child of A. Also, B is a parent of D, so that D is a child of B. In addition, C and

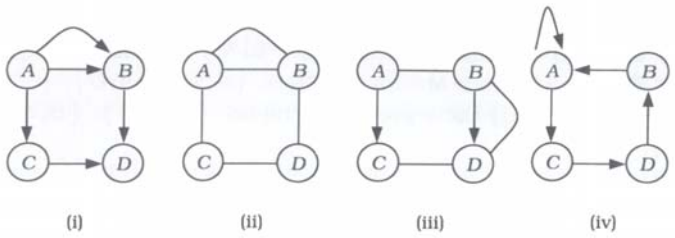


FIGURE 4.2. Examples of invalid graphs on four vertices. Examples (i), (ii) and (iii) exhibit illegal multiple edges of various types; example (iv) exhibits a loop as the node A is connected to itself.

D are neighbours, as are A and B. Finally, we have that $\mathrm{pa}(\{A, B\}) = \emptyset$, while $\mathrm{pa}(\{C, D\}) = \{A, B\}$.

If all the edges of a graph are directed, we say that it is a *directed graph.* Conversely, if all the edges of a graph are undirected, we say that it is an *undirected graph.* Referring again to Figure 4.1, graphs (i) and (iv) are directed graphs and (ii) is an undirected graph; neither of graphs (iii) nor (v) specialize to either of these two categories.

The *boundary* $\mathrm{bd}(\alpha)$ of a vertex α is the set of parents and neighbours of α; the boundary $\mathrm{bd}(A)$ of a subset $A \subset V$ is the set of vertices in $V \setminus A$ that are parents or neighbours to vertices in A, i.e., $\mathrm{bd}(A) = \mathrm{pa}(A) \cup \mathrm{ne}(A)$. The *closure* of A is $\mathrm{cl}(A) = A \cup \mathrm{bd}(A)$. Hence, in Figure 4.1(i), $\mathrm{bd}(A) = \emptyset$, while in (iii), $\mathrm{bd}(C) = \{A, D\}$.

The *undirected version* $\mathcal{G}^{\sim}$ of a graph $\mathcal{G}$ is the undirected graph obtained by replacing the directed edges of $\mathcal{G}$ by undirected edges. For example, Figure 4.1(ii) is the undirected version of the other four graphs.

We call $\mathcal{G}_A = (A, E_A)$ a *subgraph* of $\mathcal{G} = (V, E)$ if $A \subseteq V$ and $E_A \subseteq E \cap (A \times A)$. Thus, it may contain the same vertex set but possibly fewer edges. If, in addition, $E_A = E \cap (A \times A)$, we say that $\mathcal{G}_A$ is the subgraph of $\mathcal{G}$ *induced* by the vertex set A. This is illustrated in Figure 4.3.

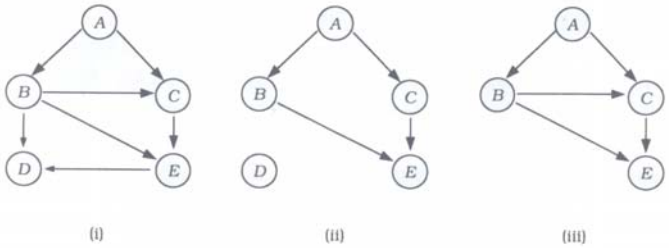


FIGURE 4.3. A graph (i), a subgraph (ii), and a vertex induced subgraph (iii).

A graph is called *complete* if every pair of vertices is joined. Figure 4.4 shows the complete undirected graph on five vertices. We say that a subset

of vertices of $\mathcal{G}$ is *complete* if it induces a complete subgraph. A complete subgraph which is maximal (with respect to $\subseteq$) is called a *clique*. In all graphs of Figure 4.1 there are four cliques: $\{A, B\}$, $\{A, C\}$, $\{C, D\}$, and $\{B, D\}$. In Figure 4.3(i) there are three cliques: $\{A, B, C\}$, $\{B, C, E\}$, and $\{B, D, E\}$.

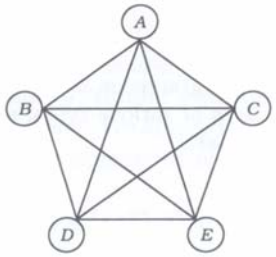


FIGURE 4.4. The complete undirected graph on five vertices.

A *path* of length n from α to β is a sequence $\alpha = \alpha_0, \ldots, \alpha_n = \beta$ of distinct vertices such that $(\alpha_{i-1}, \alpha_i) \in E$ for all $i = 1, \ldots, n$. Thus, a path can never cross itself and movement along a path never goes against the directions of arrows.

If the path of length n from α to β given by the sequence $\alpha = \alpha_0, \ldots, \alpha_n = \beta$ is such that for at least one $i \in \{1, \ldots, n\}$ there is a directed edge $\alpha_{i-1} \to \alpha_i$, we say that the path is *directed*.

We write $\alpha \mapsto \beta$ if there is a path from α to β, and say that α *leads to* β. If $\alpha \mapsto \beta$ and $\beta \mapsto \alpha$ we say that α and β *connect*, and write $\alpha \rightleftharpoons \beta$. This is clearly an equivalence relation which induces equivalence classes $[\alpha]$, where

$$\beta \in [\alpha] \Leftrightarrow \alpha \rightleftharpoons \beta.$$

We call the equivalence classes the *strong components* of $\mathcal{G}$. If $\alpha \in A \subseteq V$, the symbol $[\alpha]_A$ denotes the strong component of α in $\mathcal{G}_A$. In Figure 4.1(iii) the strong components are $\{A, B\}$ and $\{C, D\}$.

A graph $\mathcal{G}$ is said to be *connected* if there is a path between every pair of vertices in its undirected version $\mathcal{G}^{\sim}$. Any graph can be decomposed into a union of its *connected components*. The connected components are the strong components of $\mathcal{G}^{\sim}$.

An *n-cycle* is a path of length n with the modification that the end points are identical. Similarly a *directed n-cycle* is a directed path with the modification that the end points are identical. We say that a graph is *acyclic* if it does not possess any cycles.

A directed graph which is acyclic is called a *directed acyclic graph*, or DAG. A graph that has no directed cycles is called a *chain graph*. Thus, undirected graphs and directed acyclic graphs are both special cases of chain graphs.

For example, in Figure 4.1 graphs (i) and (iv) are directed graphs, but (iv) has a directed cycle and so is not a directed acyclic graph, whereas (i) is a DAG. Similarly, neither of the graphs (iv) and (v) are chain graphs, as they contain directed cycles, whereas graphs (i), (ii), and (iii) are chain graphs.

A *trail* of length n from α to β is a sequence $\alpha = \alpha_0, \ldots, \alpha_n = \beta$ of distinct vertices such that $\alpha_{i-1} \to \alpha_i$, or $\alpha_i \to \alpha_{i-1}$, or $\alpha_{i-1} \sim \alpha_i$ for all $i = 1, \ldots, n$. Thus, movement along a trail could go against the direction of the arrows, in contrast to the case of a path. In other words, a trail in $\mathcal{G}$ is a sequence of vertices that form a path in the undirected version $\mathcal{G}^{\sim}$ of $\mathcal{G}$.

If $\mathcal{K}$ is a chain graph, let $\mathcal{K}^{\not\to}$ denote the same graph but with the directed edges removed. Then each connected component of $\mathcal{K}^{\not\to}$ is called a *chain component* of $\mathcal{K}$. The strong components of a chain graph $\mathcal{K}$ are exactly its chain components. In fact, a graph is a chain graph if and only if its strong components induce undirected subgraphs.

As a special case, each node of a DAG $\mathcal{D}$ forms a chain component of $\mathcal{D}$. Figure 4.5 shows a six-vertex chain graph having five chain components.

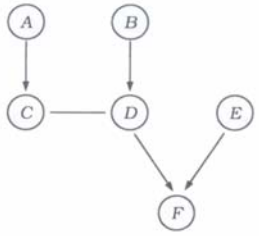


FIGURE 4.5. A six-vertex chain graph. The chain components are A, B, $\{C, D\}$, E, and F.

It is always possible to *well-order* the nodes of a DAG by a linear ordering or numbering such that, if two nodes are connected, the edge points from the lower to the higher of the two nodes with respect to the ordering. For example, the graph in Figure 4.3(i) has the unique well-ordering (A, B, C, E, D). Note that a DAG may not have a unique well-ordering. If a DAG is well-ordered, the *predecessors* of a node α, denoted by $\mathrm{pr}(\alpha)$, are those nodes that have a lower number than α.

A simple method to construct such a well-ordering is the following:

Algorithm 4.1 [TOPOLOGICAL SORT]

- Begin with all vertices unnumbered.
- Set counter $i := 1$.
- While any vertices remain:

- Select any vertex that has no parents;
- number the selected vertex as i;
- delete the numbered vertex and all its adjacent edges from the graph;
- increment i by 1. □

Alternatively, we can use the dual version of this algorithm, which recursively selects and deletes childless vertices, while numbering downward.

The above algorithm, or its dual, extends to well-ordering the chain components of a chain graph as follows. Let $\mathcal{K}$ be a chain graph having the set of chain components K. Then, instead of selecting a parentless node for deletion, one selects a parentless chain component, that is a chain component none of whose nodes have parents. The result is a well-ordering of the chain components. One possible well-ordering of the chain components of Figure 4.5 is $(A, B, \{C, D\}, E, F)$; yet another is $(E, B, A, \{C, D\}, F)$.

Given a chain graph, the set of vertices α such that $\alpha \mapsto \beta$ but not $\beta \mapsto \alpha$ is the set $\mathrm{an}(\beta)$ of the *ancestors* of β, and the *descendants* $\mathrm{de}(\alpha)$ of α are the vertices β such that $\alpha \mapsto \beta$ but not $\beta \mapsto \alpha$. The *nondescendants* $\mathrm{nd}(\alpha)$ of α is the set $V \setminus (\mathrm{de}(\alpha) \cup \alpha)$. If $\mathrm{bd}(\alpha) \subseteq A$ for all $\alpha \in A$, we say that A is an *ancestral* set. The symbol $\mathrm{An}(A)$ denotes the smallest ancestral set containing A. Note that in general $\mathrm{An}(A) \neq A \cup_{\alpha \in A} \mathrm{an}(\alpha)$. Thus, in Figure 4.5 the set of ancestors of F consists of all remaining nodes. The ancestors of C are $\{A, B\}$, F is the only descendant of C, $\{A, B, C, D\}$ is an ancestral set, and $\mathrm{An}(C) = \{A, B, C, D\}$. Node E has no ancestors, and $\{E\}$ is an ancestral set. The set of nondescendants of D is $\{A, B, C, E\}$.

A subset $C \subseteq V$ is said to be an (α, β)-*separator* if all trails from α to β intersect C. The subset C is said to *separate* A from B if it is an (α, β)-separator for every $\alpha \in A$ and $\beta \in B$. An (α, β)-separator C is said to be *minimal* if no proper subset of C is itself an (α, β)-separator. In Figure 4.5 the set $\{C, D\}$ is an (A, F)-separator; moreover, both C and D are each minimal (A, F)-separators. In Figure 4.3(i) both $\{B, C\}$ and $\{B, E\}$ are minimal (A, D)-separators.

An important class of graphs is that of the trees. We say that a graph $\mathcal{G}$ is a *tree* if it is connected and its undirected version $G^{\sim}$ has no cycles; thus, there is a unique trail in a tree between any two vertices. We use the symbol $\mathcal{T}$ to denote a tree graph. A *rooted* tree is a tree with a designated vertex ρ called the *root*. A *leaf* of a tree is a node that is joined to at most one other node. A tree that has more than one node thus has at least two leaves. The *diameter* of a tree is the length of longest trail between two leaf nodes. A *forest* is a graph having no cycles, that is, its connected components are all trees. The graph of Figure 4.5 is a tree.

Given a chain graph $\mathcal{K}$, we define the *moral graph* of $\mathcal{K}$ to be the undirected graph $\mathcal{K}^m$ obtained from $\mathcal{K}$ by first adding undirected edges between all pairs of vertices that have children in a common chain component and

that are not already joined, and then forming the undirected version of the resulting graph.

For the special case in which $\mathcal{K}$ is a DAG, this process of *moralization* involves adding undirected edges between all pairs of parents of each vertex which are not already joined, and then making all edges undirected. Figure 4.6 shows the moral graph of Figure 4.5, obtained by adding the two undirected edges $A \sim B$ (common parents of the chain component $\{C, D\}$) and $D \sim E$ (the parents of the chain component F), and then forming the undirected version. This moralization procedure is an important first step in constructing the inference engine for a probabilistic network specified by a chain graph.

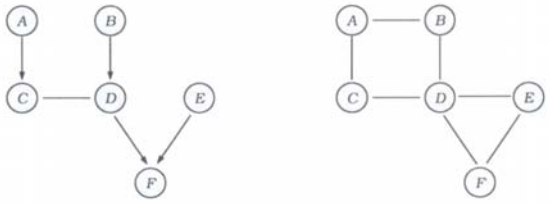


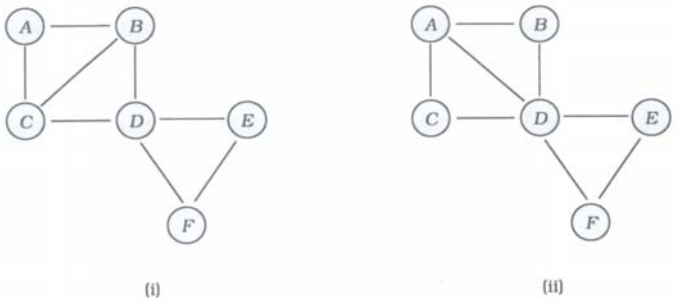
FIGURE 4.6. The graph of Figure 4.5 and its moral graph.

4.2 Chordal and decomposable graphs

An important type of graph is the *decomposable graph*, which we characterize below. This is basic to the analysis of probabilistic networks. The qualitative and quantitative expertise encoded within a probabilistic network can be transformed into a decomposable graph, using well-defined graphical algorithms and without loss of information, and then further into an associated *junction tree* which supports efficient computational algorithms. This section deals with the properties of decomposable graphs. Junction trees and their relation to decomposable graphs are discussed in Section 4.3 below, while Section 4.4 describes algorithms for constructing a junction tree.

Let $\mathcal{G}$ be an undirected graph with vertex set V. Recall that in this case an n-cycle in $\mathcal{G}$ is a sequence $(\alpha_0, \alpha_1, \ldots, \alpha_n)$ of vertices in V, distinct except that $\alpha_0 = \alpha_n$, and such that $\alpha_i \sim \alpha_{i+1}$ for all i. Let σ be an n-cycle in $\mathcal{G}$. A *chord* of this cycle is a pair (α_i, α_j) of non-consecutive vertices in σ such that $\alpha_i \sim \alpha_j$ in $\mathcal{G}$. The undirected graph $\mathcal{G}$ is called *chordal* or *triangulated* if every one of its cycles of length ≥ 4 possesses a chord. A definition such as this is a so-called 'forbidden path' definition, which has several consequences. For example, the property is stable under taking vertex-induced

subgraphs, i.e., if $\mathcal{G}$ is chordal and $A \subset V$, then $\mathcal{G}_A$ is also chordal. The moral graph of Figure 4.6 is clearly not chordal. Figure 4.7 shows two possible chordal graphs obtained from the moral graph in Figure 4.6 by adding one undirected edge.



(i) (ii)

FIGURE 4.7. Two chordal graphs which can be derived from the moral graph in Figure 4.6, by either (i) adding the edge $B \sim C$, or (ii) adding the edge $A \sim D$.

An important concept that forms the basis of localizing computation in a probabilistic expert system is that of a decomposition of a graph, as defined below.

Definition 4.2 [DECOMPOSITION]
A triple (A, B, C) of disjoint subsets of the vertex set V of an undirected graph $\mathcal{G}$ is said to form a *decomposition* of $\mathcal{G}$, or to *decompose* $\mathcal{G}$, if $V = A \cup B \cup C$, and the following two conditions hold:

1. C separates A from B;
2. C is a complete subset of V. □

Note that we allow any of the sets A, B, and C to be empty. If both A and B are non-empty, we say that the decomposition is *proper*.

Definition 4.3 [DECOMPOSABLE GRAPH]
We say that an undirected graph $\mathcal{G}$ is *decomposable* if either: (i) it is complete, or (ii) it possesses a proper decomposition (A, B, C) such that both subgraphs $\mathcal{G}_{A\cup C}$ and $\mathcal{G}_{B\cup C}$ are decomposable. □

Note that this is a recursive definition, which is permissible because the decomposition (A, B, C) is required to be proper, so that each of $\mathcal{G}_{A\cup C}$ and $\mathcal{G}_{B\cup C}$ has fewer vertices than the original graph $\mathcal{G}$.

There is a strong connection between decomposability and chordality, as captured by the following theorem:

Theorem 4.4 *The following conditions are equivalent for an undirected graph $\mathcal{G}$:*

1. *$\mathcal{G}$ is decomposable;*

2. *$\mathcal{G}$ is chordal;*

3. *Every minimal (α, β)-separator is complete.*

Proof. The result is well-known (Berge 1973; Golumbic 1980). The present proof is taken from Lauritzen (1996).

We proceed by induction on the number of vertices $|V|$ of $\mathcal{G}$. The result is trivial for a graph with no more than three vertices since the three conditions are then all automatically fulfilled. So assume the result to hold for all graphs with $|V| \leq n$ and consider a graph $\mathcal{G}$ with $n+1$ vertices.

First we show $1 \Rightarrow 2$. Suppose that $\mathcal{G}$ is decomposable. If it is complete, it is obviously chordal. Otherwise it has a decomposition (A, B, C) into decomposable subgraphs $\mathcal{G}_{A\cup C}$ and $\mathcal{G}_{B\cup C}$, both with fewer vertices. By the inductive hypothesis these are chordal. Thus, the only possibility for a chordless cycle is one that intersects both A and B. But because C separates A from B, such a cycle must intersect C at least twice. But then it contains a chord because C is complete.

Now we show $2 \Rightarrow 3$. Assume that $\mathcal{G}$ is chordal and let C be a minimal (α, β)-separator. If C has only one vertex, it is complete. If not, it contains at least two vertices, γ_1 and γ_2 say. Since C is a minimal separator, there will be paths from α to β via γ_1 and back via γ_2. The sequence

$$(\alpha, \ldots, \gamma_1, \ldots, \beta, \ldots, \gamma_2, \ldots, \alpha)$$

forms a cycle, with the modification that it can have repeated points. These, and chords other than a link between γ_1 and γ_2, can be used to shorten the cycle, still leaving at least one vertex in the component $[\alpha]_{V\setminus C}$ and one in $[\beta]_{V\setminus C}$, where these symbols denote the connected components of the graph $\mathcal{G}_{V\setminus C}$ containing α and β respectively. This produces eventually a cycle of length at least 4, which must have a chord, whereby we get that $\gamma_1 \sim \gamma_2$. Repeating the argument for all pairs of vertices in C gives that C is complete.

Finally we show that $3 \Rightarrow 1$. Suppose that every minimal (α, β)-separator is complete. If $\mathcal{G}$ is complete there is nothing to show. Otherwise it has at least two non-adjacent vertices, α and β. Assume that the result has been established for every proper subgraph of $\mathcal{G}$. Let C be a minimal (α, β)-separator and partition the vertex set into $[\alpha]_{V\setminus C}$, $[\beta]_{V\setminus C}$, C and D (where D is the set of remaining vertices). Then, since C is complete, the triple (A, B, C), where $A = [\alpha]_{V\setminus C} \cup D$, and $B = [\beta]_{V\setminus C}$, forms a decomposition of $\mathcal{G}$. But each of the subgraphs $\mathcal{G}_{A\cup C}$ and $\mathcal{G}_{B\cup C}$ must be decomposable. For if C_1 is a minimal (α_1, β_1)-separator in $\mathcal{G}_{A\cup C}$, it is also a minimal separator in $\mathcal{G}$ and therefore complete by assumption. The inductive assumption implies that $\mathcal{G}_{A\cup C}$ is decomposable, and similarly with $\mathcal{G}_{B\cup C}$. Thus, we have decomposed $\mathcal{G}$ into decomposable subgraphs. □

The smallest graph that is not decomposable is a 4-cycle, as displayed in Figure 4.1(ii).

A directed acyclic graph for which the parents of every node form a complete set is called *perfect*. For example, Figure 4.3(i) is a perfect directed graph. If $\mathcal{G}$ is an undirected graph, then a numbering of its vertices, $(v_1, \ldots, v_k)$ say, is called *perfect* if the neighbours of any node that have lower numbers, i.e., $\mathrm{ne}(v_j) \cap \{v_1, \ldots, v_{j-1}\}$, induce a complete subgraph.

For example, in the graph of Figure 4.7(i), $(A_1, B_2, C_3, D_4, E_5, F_6)$ is a perfect numbering, but this is not the case for $(A_1, B_2, C_3, E_4, F_5, D_6)$. For, in the latter numbering, the previously numbered neighbours of D, i.e., (B, C, E, F), do not induce a complete graph. Note that any vertex numbering of a complete undirected graph is perfect.

Given a well-ordered perfect directed graph $\mathcal{G}$, its undirected version $\mathcal{G}^\sim$ is a chordal graph for which the ordering $(v_1, v_2, \ldots, v_k)$ constitutes a perfect numbering. This is easily seen by induction using the fact that for all j the triple (W_j, V_{j-1}, S_j) forms a decomposition of $\mathcal{G}^\sim_{V_j}$, where $V_j = \{v_1, \ldots, v_j\}$, $W_j = \mathrm{cl}^\sim(v_j) \cap V_j$, and $S_j = W_j \cap V_{j-1}$. Here $\mathrm{cl}^\sim$ denotes closure relative to the undirected graph $\mathcal{G}^\sim$.

Conversely, given an undirected graph $\mathcal{G}$ and a perfect numbering of its vertices, $(v_1, \ldots, v_k)$, one can construct a perfect directed graph simply by directing the edges from lower to higher numbered vertices. It follows that the graph $\mathcal{G}$ must be chordal for such a perfect numbering to exist. More precisely, the following result holds true:

Theorem 4.5 *An undirected graph is chordal if and only if it admits a perfect numbering.*

Proof. See Lauritzen (1996), Proposition 2.17. □

It is worth noting the slightly stronger result that if $\mathcal{G}$ is chordal and v is an arbitrary node of $\mathcal{G}$, then a perfect numbering of $\mathcal{G}$ exists with $v_1 = v$: see Algorithm 4.9 below.

4.3 Junction trees

In this section we summarize some of the important properties of junction trees and their relationship to decomposable graphs.

Let $\mathcal{C}$ be a collection of subsets of a finite set V and $\mathcal{T}$ a tree with $\mathcal{C}$ as its node set. Then $\mathcal{T}$ is said to be a *junction tree* if any intersection $C_1 \cap C_2$ of a pair C_1, C_2 of sets in $\mathcal{C}$ is contained in every node on the unique path in $\mathcal{T}$ between C_1 and C_2. Equivalently, for any vertex v in $\mathcal{G}$, the set of subsets in $\mathcal{C}$ containing v induces a connected subtree of $\mathcal{T}$. Junction trees also appear under other names in the literature, e.g., *join trees* in relational databases (see Section 4.5).

Now let $\mathcal{G}$ be an undirected graph, and $\mathcal{C}$ the family of its cliques. If $\mathcal{T}$ is a junction tree with $\mathcal{C}$ as its node set, we say that $\mathcal{T}$ is a junction tree (of cliques) for the graph $\mathcal{G}$. We have

Theorem 4.6 *There exists a junction tree $\mathcal{T}$ of cliques for the graph $\mathcal{G}$ if and only if $\mathcal{G}$ is decomposable.*

Proof. The theorem clearly holds if $\mathcal{G}$ contains at most two cliques. Suppose that the theorem holds for all graphs with at most k cliques and let $\mathcal{G}$ have $k+1$ cliques.

Assume $\mathcal{T}$ is a junction tree of cliques for $\mathcal{G}$. Take C_1 and C_2 adjacent in $\mathcal{T}$. On cutting the link $C_1 \sim C_2$, $\mathcal{T}$ separates into two subtrees, $\mathcal{T}_1$ and $\mathcal{T}_2$. Let V_i be the union of the nodes in $\mathcal{T}_i$ for $i = 1, 2$, and let $\mathcal{G}_i = \mathcal{G}_{V_i}$. The nodes in $\mathcal{T}_i$ are then the cliques of $\mathcal{G}_i$, and $\mathcal{T}_i$ is a junction tree for $\mathcal{G}_i$. By the inductive hypothesis, $\mathcal{G}_1$ and $\mathcal{G}_2$ are both decomposable. Thus, we are done if we can show that $S := V_1 \cap V_2$ is complete and separates V_1 from V_2. Suppose $v \in V_1 \cap V_2$. Then there exists a clique C_i' of $\mathcal{G}_i$ for each $i = 1, 2$, with $v \in C_i'$. Clearly the path in $\mathcal{T}$ joining C_1' and C_2' passes through both C_1 and C_2. Therefore, $v \in C_1 \cap C_2$ and so we must have $V_1 \cap V_2 \subseteq C_1 \cap C_2$. Since clearly $C_1 \cap C_2 \subseteq V_1 \cap V_2$, we must have that $S = C_1 \cap C_2$ and that S is complete.

Now take $u \in V_1 \setminus S$ and $v \in V_2 \setminus S$, and suppose there exists a path $u, w_1, w_2, \ldots, w_k, v$ with each $w_i \notin S$. Then there exists a clique C containing the complete set $\{u, w_1\}$. Clearly $C \subseteq V_1$, so $w_1 \in V_1$, whence $w_1 \in V_1 \setminus S$. Repeat the argument to deduce $w_2 \in V_1 \setminus S, \ldots, v \in V_1 \setminus S$. This is a contradiction, hence S separates V_1 from V_2, and (V_1, V_2, S) is a decomposition of $\mathcal{G}$. We have now decomposed $\mathcal{G}$ into subgraphs that possess junction trees and thus are decomposable by the inductive assumption.

Conversely, assume that $\mathcal{G}$ is decomposable, and let (W_1, W_2, S) be a decomposition of $\mathcal{G}$ into proper decomposable subgraphs $\mathcal{G}_{V_1}$, $\mathcal{G}_{V_2}$, where $V_i = W_i \cup S$. Then at least one of V_1 and V_2 — say V_1 — has the form $\bigcup_{C \in \mathcal{C}_1} C$, with $\mathcal{C}_1 \subset \mathcal{C}$; and then we can, if necessary, redefine $V_2 = \bigcup_{C \in \mathcal{C}_2} C$ (with $\mathcal{C}_2 = \mathcal{C} \backslash \mathcal{C}_1$) and still have a decomposition. Let $C_i \in \mathcal{C}_i$ satisfy $S \subseteq C_i$. By hypothesis, we have a junction tree $\mathcal{T}_i$ for $\mathcal{G}_i$ where, as before, $\mathcal{G}_i = \mathcal{G}_{V_i}$. Form $\mathcal{T}$ by linking C_1 in $\mathcal{T}_1$ to C_2 in $\mathcal{T}_2$.

Let $v \in V$. If $v \notin V_2$, then all cliques containing v are in $\mathcal{C}_1$, and so connected in $\mathcal{T}_1$, hence in $\mathcal{T}$. If $v \notin V_1$, then similarly for $\mathcal{T}_2$. Otherwise $v \in S$. The cliques in $\mathcal{C}_i$ containing v are connected in $\mathcal{T}_i$, and include C_i. Since C_1 and C_2 are connected in $\mathcal{T}$, the result follows. □

The above proof demonstrates that an intersection $S = C_1 \cap C_2$ between two neighbouring nodes in a junction tree of cliques separates the decomposable graph $\mathcal{G}$ (in fact, is a minimal separator). We therefore call S the *separator* associated with the edge between C_1 and C_2 of the junction tree; we use the term separator also in the case where the nodes of the junction tree are not all cliques. It is possible that distinct edges may have identical

separators. The set of all separators, including any such repetitions, will be denoted by $\mathcal{S}$. When $\mathcal{G}$ admits more than one junction tree of cliques, it can be shown that $\mathcal{S}$ will be the same for all of them.

The separators are often displayed as labels on the edges of a junction tree. They play an important role in the propagation algorithms discussed in Chapters 6 to 8.

A clique $C^* \in \mathcal{C}$ is called *extremal* if, with $V_2 = \bigcup_{C \in \mathcal{C} \setminus \{C^*\}} C$, the triple $(C^* \setminus V_2, V_2 \setminus C^*, C^* \cap V_2)$ is a decomposition of $\mathcal{G}$. We have:

Corollary 4.7 *If a chordal graph $\mathcal{G}$ has at least two cliques, it has at least two extremal cliques.*

Proof. Any junction tree of $\mathcal{G}$ has at least two leaves. □

Figure 4.8 shows junction trees constructed from the chordal graphs of Figure 4.7, where the separators are displayed on the links as rectangular. There are two possible junction tree structures for Figure 4.7(ii), the difference not being in their cliques but in the way they are connected.

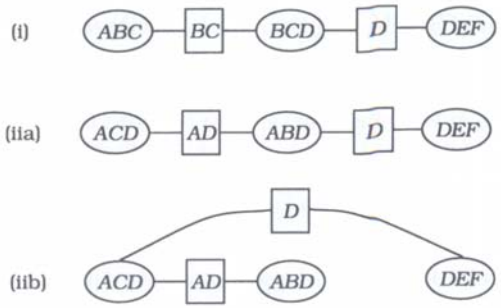


FIGURE 4.8. Junction trees of the chordal graphs of Figure 4.7.

A sequence $(C_1, C_2, \ldots, C_k)$ of sets is said to have the *running intersection property* if, for all $1 < j \leq k$, there is an $i < j$ such that $C_j \cap (C_1 \cup \cdots \cup C_{j-1}) \subseteq C_i$.

The cliques of a decomposable graph can be ordered to satisfy this property simply by well-ordering the junction tree. Conversely, if the cliques have been ordered to satisfy the running intersection property, a junction tree (of cliques) can be built using the following algorithm.

Algorithm 4.8 [JUNCTION TREE CONSTRUCTION]
From the cliques $(C_1, \ldots, C_p)$ of a chordal graph ordered to have running intersection property:

1. Associate a node of the tree with each clique C_i.
2. For $i = 2, \ldots, p$, add an edge between C_i and C_j where j is any one value in $\{1, \ldots, i-1\}$ such that

$$C_i \cap (C_1 \cup \cdots \cup C_{i-1}) \subseteq C_j. \qquad \square$$

4.4 From chain graph to junction tree

Advances in the computational analysis of probabilistic networks have come about through the realization that the joint distribution of a probabilistic network can be represented and manipulated efficiently using a junction tree derived from the original graph. This section collects together some of the algorithms for effecting this transformation.

Suppose that a probabilistic network has a chain graph structure $\mathcal{K}$ (we include as a possibility that $\mathcal{K}$ may be a directed or undirected graph). In Chapter 6 we shall see that the first stage in passing to the inference structure is to form the moral graph $\mathcal{K}^m$. The moral graph is undirected, but it may not be a chordal graph. Tarjan and Yannakakis (1984) gave the following efficient algorithm, and proved its correctness, for deciding whether a given undirected graph $\mathcal{G} = (V, E)$ is chordal or not; they also showed that it can be implemented to run in $O(n+e)$ time where $n = |V|$ is the number of nodes and $e = |E|$ the number of edges:

Algorithm 4.9 [MAXIMUM CARDINALITY SEARCH]

- Set **Output**:= '$\mathcal{G}$ is chordal'.
- Set counter $i := 1$.
- Set $L = \emptyset$.
- For all $v \in V$, set $c(v) := 0$.
- While $L \neq V$:
  - Set $U := V \setminus L$.
  - Select any vertex v maximizing $c(v)$ over $v \in U$ and label it i.
  - If $\Pi_{v_i} := \mathrm{ne}(v_i) \cap L$ is not complete in $\mathcal{G}$:
    Set **Output**:= '$\mathcal{G}$ is not chordal'.
  - Otherwise, set $c(w) = c(w) + 1$ for each vertex $w \in \mathrm{ne}(v_i) \cap U$.

- Set $L = L \cup \{v_i\}$.
- Increment i by 1.

• Report **Output**. □

At each stage, L consists of all previously labelled vertices. The algorithm recursively labels vertices in such a way as to maximize the cardinality of the set of previously labelled neighbours. If at any stage this set is not complete, $\mathcal{G}$ is not chordal. The process could be aborted at this stage.

If $\mathcal{G}$ passes the maximum cardinality search, the vertex numbering found will be perfect, as for any node v the set Π_v of its previously numbered neighbours will be complete, and thus $\mathcal{G}$ must be chordal. The converse is also true:

Theorem 4.10 *If $\mathcal{G}$ is chordal, then maximum cardinality search will provide a perfect numbering of $\mathcal{G}$.*

Proof. See Tarjan and Yannakakis (1984). □

If a graph is chordal and its vertices have been numbered by maximum cardinality search, its cliques can be identified in a simple fashion using the algorithm described below, which simultaneously provides an ordering of the cliques having the running intersection property.

Algorithm 4.11 [FINDING THE CLIQUES OF A CHORDAL GRAPH]
Starting from a numbering $(v_1, \ldots, v_k)$ obtained by maximum cardinality search, we can find the cliques of a chordal graph as follows. Denote the cardinality of Π_{v_i} by π_i. Call node v_i a *ladder node* if $i = k$, or if $i < k$ and $\pi_{i+1} < 1 + \pi_i$. Let the jth ladder node, in ascending order, be λ_j, and define $C_j = \{\lambda_j\} \cup \Pi_{\lambda_j}$. □

Theorem 4.12 *There is a one-to-one correspondence between the ladder nodes and the cliques of $\mathcal{G}$, the clique associated with ladder node λ_j being C_j. The clique ordering $(C_1, C_2, \ldots)$ will possess the running intersection property.*

Proof. Again we may suppose that $\mathcal{G}$ is connected. We argue by induction. If $|V| = 1$ there is nothing to prove. Suppose the algorithm works for $|V| \leq n$, and consider a case having $|V| = n + 1$. Let $v^* = v_{n+1}$, $\Pi^* = \Pi_{v_{n+1}} = \mathrm{bd}(v^*)$, $\pi^* = \pi_{n+1} = |\Pi^*|$. The first n nodes numbered induce a subgraph $\mathcal{G}'$ on $V' = V \setminus \{v^*\}$, which can itself be regarded as having been numbered by the same algorithm. Consequently, by the inductive hypothesis we can suppose that the algorithm has supplied a clique-ordering for $\mathcal{G}'$, $(C'_1, \ldots, C'_p)$ say, with the running intersection property.

Since, by Theorem 4.10, Π^* is complete, there exists a clique C'_m of $\mathcal{G}'$ such that $\Pi^* \subseteq C'_m$. Let v_i be its corresponding ladder node. We distinguish two cases, according as whether (i) $\Pi^* \neq C'_m$, or (ii) $\Pi^* = C'_m$. Note that

if $m < p$ we must be in case (i), since otherwise the maximum cardinality property would have selected v^* over v_{i+1}.

In case (i), $C_{p+1} := \{v^*\} \cup \Pi^*$ does not properly contain any clique of $\mathcal{G}'$. Taking $C_j = C'_j$ for $j = 1, \ldots, p$, it is easily seen that the algorithm behaves as asserted for the full graph $\mathcal{G}'$, delivering cliques $(C_j : j = 1, \ldots, p+1)$ having the running intersection property.

Otherwise, if we are in case (ii), we must have $p = m$. It readily follows that, in the final numbering of V, v_n is no longer a ladder node, while $v^* = v_{n+1}$ is; and that the algorithm applied to the full graph $\mathcal{G}'$ has again behaved as asserted, delivering cliques $C_j = C'_j$ for $j < p$, $C_p = C'_p \cup \{v^*\}$, having the running intersection property. □

Algorithm 4.11 is essentially the same as ordering the cliques by their largest numbered nodes in a maximum cardinality ordering, as described, for example, by Leimer (1989). However, Algorithm 4.11 can be carried out 'online' during the execution of Algorithm 4.9. Then, if $\mathcal{G}$ is chordal, one pass of this combined algorithm will not only verify the fact, but also identify its cliques, together with a running intersection ordering for them.

Note that Algorithm 4.11 need not work for an arbitrary perfect numbering if it is not generated by maximum cardinality search. A counterexample is given by the graph in Figure 4.9.

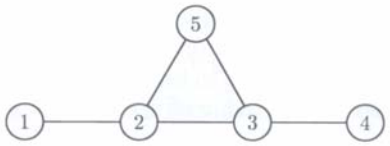


FIGURE 4.9. The numbering of the vertices is perfect, but the cliques, numbered as $(\{1,2\}, \{3,4\}, \{2,3,5\})$, do not have the running intersection property. This numbering could not have been generated by maximum cardinality search (which at least would reverse 4 and 5).

4.4.1 Triangulation

If the graph $\mathcal{G} = (V, E)$ is not chordal, it can always be made so by adding extra edges F in a suitable way to form $\mathcal{G}' = (V, E')$, where $E' = E \cup F$. The edges in F are referred to as *fill-in* edges. If $\mathcal{G}'$ is chordal, we refer to it as a *triangulation* of $\mathcal{G}$.

In general, given any ordering, say $(v_1, \ldots, v_k)$, of the nodes of an undirected graph $\mathcal{G}$, one can triangulate $\mathcal{G}$ by recursively examining each node v_j in turn in reverse order, beginning with v_k, and joining those pairs of neighbours that appear earlier in the ordering and are not already joined. The end result is a chordal graph $\mathcal{G}'$. Clearly the given ordering $(v_1, \ldots, v_k)$

is then a perfect numbering for the triangulation $\mathcal{G}'$ of $\mathcal{G}$. The problem of obtaining a good triangulation is thus one of finding a good ordering, but the general problem of finding optimal triangulations for undirected graphs is *NP-hard* (Yannakakis 1981), so heuristic algorithms must be developed.

Kjærulff (1992) examined various algorithms for triangulating a non-chordal graph. For problems in which large cliques are unavoidable the method of simulated annealing performs well. Although using simulated annealing to find a triangulation of a graph may be time-consuming, for any given probabilistic network it only needs to be performed once, so it can be a worthwhile investment of time for some problems. Kjærulff (1992) also looked at a number of heuristic algorithms that involve selecting the next node v on the basis of some *optimality criterion* $c(v)$, for example, either maximizing or minimizing some cost or utility function which depends upon the node being selected. The basic algorithm is described by Olmsted (1983) and Kong (1986) and runs as follows for an undirected graph $\mathcal{G}$ with k vertices.

Algorithm 4.13 [ONE-STEP LOOK AHEAD TRIANGULATION]

- Start with all vertices unnumbered, set counter $i := k$.
- While there are still some unnumbered vertices:
  - Select an unnumbered vertex v to optimize the criterion $c(v)$.
  - Label it with the number i, i.e., let $v_i := v$.
  - Form the set C_i consisting of v_i and its unnumbered neighbours.
  - Fill in edges where none exist between all pairs of vertices in C_i.
  - Eliminate v_i and decrement i by 1. □

Note that this algorithm operates with a numbering strategy opposite to that of maximum cardinality search. The quality of the triangulation, with regard to computational efficiency in applications to probabilistic networks, will depend upon the optimality criterion $c(v)$ used to select vertices.

For models with discrete random variables, selecting $c(v_j)$ to be the cardinality of the joint state space for the variables in the set C_j usually yields good results. Another possibility, which does not depend upon the interpretation of the variables, is to take $c(v)$ to be the number of fill-in edges required if v were to be selected for labelling. For other methods and comparisons between them, see Kjærulff (1992).

Although the maximum cardinality search algorithm is efficient for testing the chordality of a graph, as a numbering method for generating a chordal graph from a non-chordal graph it tends to introduce many more fill-in edges than are necessary, which in turn leads to larger than necessary cliques and reduces the efficiency of the algorithms for probabilistic computations.

4.4.2 *Elimination tree*

For a given numbering $(v_1, \ldots, v_k)$ of the nodes of a graph $\mathcal{G}$ one can associate the sequence of sets $(C_1, \ldots, C_k)$ defined by Algorithm 4.13. By construction, each set C_j has the following properties: it contains v_j; the indices of any remaining nodes in C_j are smaller than j; and v_j is not found in any earlier set, i.e., $v_j \notin C_l$ for all $l < j$. These sets are called the *elimination sets* induced by the numbering, and they may be used to form a tree structure called an *elimination tree* (Cowell 1994); this can be useful as an intermediate step to forming a junction tree of cliques and as the basis for the propagation algorithms (see Chapters 6 and 8).

Algorithm 4.14 [ELIMINATION TREE CONSTRUCTION]

1. Associate a node of the tree with each set C_i.

2. For $i = 1, \ldots, k$, if C_i contains more than one vertex, add an edge between C_i and C_j where j is the largest index of a vertex in $C_i \setminus \{v_i\}$.

□

It is a simple matter to see that the sequence $(C_1, \ldots, C_k)$ has the running intersection property and that the elimination tree is therefore a junction tree of sets. However, the elimination tree is generally not a junction tree of cliques of the triangulated graph $\mathcal{G}'$, because although the sequence $(C_1, \ldots, C_k)$ will contain the cliques of $\mathcal{G}'$, it will also contain some subsets of the cliques. Figure 4.10 shows the elimination tree, using the elimination ordering obtained by ordering the nodes numerically, derived from the graph of Figure 4.9. The notation 5:23 reflects that when node 5 is eliminated its boundary, bd(5), is 23.

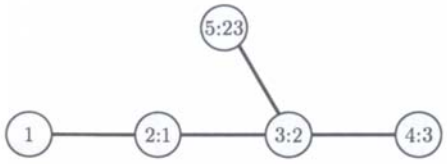


FIGURE 4.10. Elimination tree obtained from the chordal graph of Figure 4.9 using the given numbering.

Theorem 4.15 *The cliques of the triangulated graph $\mathcal{G}'$ are contained in the set of elimination sets $(C_1, \ldots, C_k)$.*

Proof. Let C be any clique of the triangulated graph $\mathcal{G}'$ formed by the triangulation algorithm. Then it possesses a greatest-numbered vertex v. At the stage when v was eliminated, it must have been a neighbour of all the other vertices in the clique; for if there is a vertex w for which this is not true, at no stage subsequent to eliminating v could the edge $v \sim w$ be added because the algorithm only adds edges between unnumbered vertices, and at all later stages v remains numbered. Hence, we may identify C with the elimination set formed by eliminating v because the algorithm adds sufficient edges to ensure all pairs of unnumbered neighbours of v are joined. □

One can thus find the cliques of $\mathcal{G}'$ simply by deleting redundant elimination sets, i.e., sets that are contained in other sets. However, this will in general destroy the running intersection property of the sequence. For example, deleting the elimination set $\{2,3\}$ of the graph in Figure 4.9 (with elimination tree displayed in Figure 4.10) will destroy the running intersection property of the sequence $(\{1\},\{1,2\},\{2,3\},\{3,4\},\{2,3,5\})$.

However, Lemma 4.16 below, due to Leimer (1989) (see also Lemma 2.13 of Lauritzen (1996)) shows how to modify the ordering when a redundant subset is to be deleted. We omit the proof.

Lemma 4.16 *Let $C_1,\ldots,C_k$ be a sequence of sets having the running intersection property. Assume that $C_t \subseteq C_p$ for some $t \neq p$ and that p is minimal with this property for fixed t. Then:*

(i) *If $t > p$, then $C_1,\ldots,C_{t-1},C_{t+1},\ldots,C_k$ has the running intersection property.*

(ii) *If $t < p$, then $C_1,\ldots,C_{t-1},C_p,C_{t+1},\ldots,C_{p-1},C_{p+1},\ldots,C_k$ has the running intersection property.*

Note that the condition $t < p$ is always true for the sequence of elimination sets generated by a vertex numbering because C_p contains v_p by construction, and no lower numbered elimination set can contain v_p because v_p remains numbered when these sets are formed.

We can remove those sets in $(C_1,\ldots,C_k)$ that are proper subsets of others until there is no redundancy, by repeatedly applying Lemma 4.16. The result is an ordering of the cliques of $\mathcal{G}'$ having the running intersection property. These can now be joined up to form a junction tree using Algorithm 4.8. Alternatively, knowing that $\mathcal{G}'$ is now chordal, maximum cardinality search combined with Algorithm 4.11 can be used to find the cliques and order them with the running intersection property, enabling the junction tree of cliques to be built.

4.5 Background references and further reading

There are many general textbooks on graph theory: Harary (1972) and Berge (1973) are standard references.

Chordal graphs are well-studied objects which appear under a variety of names, including triangulated and decomposable graphs, and also *rigid circuit graphs* (Dirac 1961). They are extensively dealt with in Golumbic (1980). Chain graphs were introduced by Lauritzen and Wermuth (1984); see also Lauritzen and Wermuth (1989).

The notion of a graph decomposition has deep connections to many areas of mathematics (Lauritzen et al. 1984; Diestel 1987, 1990), including the four-colour problem (Wagner 1937), measure theory (Kellerer 1964a, 1964b; Vorob'ev 1962, 1963), the solution of systems of linear equations (Parter 1961; Rose 1970, 1973), game theory (Vorob'ev 1967), and relational databases (Beeri et al. 1981, 1983).

The notion of a junction tree has appeared under an abundance of names. The first explicit identification seems to be in relational databases, where it has been known as a *join tree* (Maier 1983); the terms *k–tree* (Arnborg et al. 1987), *Markov tree* and *hypertree* (Shenoy and Shafer 1990), or simply *clique tree* have also been used.

There is an extensive literature concerned with algorithms for manipulating decomposable graphs in an efficient way. It includes other algorithms for checking decomposability of a graph and finding their cliques (Rose et al. 1976; Gavril 1972), for constructing optimal junction trees for given decomposable graphs (Jensen and Jensen 1994), and for constructing optimal decompositions of a non-chordal graph into its indecomposable components (Tarjan 1985; Leimer 1993). There is also some efficiency to be gained by constructing junction trees of special types (Almond 1995; Shenoy 1997). For recent results on triangulation algorithms, see Becker and Geiger (1996), Larrañaga et al. (1997), and Meilă and Jordan (1997).

5
Markov Properties on Graphs

The Markov properties of graphs provide a theoretical foundation of localized computation for inference in probabilistic networks. General chain graphs and their specializations — directed and undirected graphs — each have different types of Markov properties. A common theoretical tool to understanding these properties is the notion of conditional independence.

5.1 Conditional independence

In Chapter 2 we saw simple examples of probabilistic networks, in which the factorization of joint distributions is expressed by directed graphs, and inference consists of reversing arrows. Our aim is to develop tools for manipulating the probability distributions of variables in models that factorize over graphical structures having more complicated topologies, and to enable efficient inference to be performed for such models.

As a precursor to this we need to introduce the notion of *conditional independence*, which will allow us to justify the local computations developed for our inference process later in the book. For simplicity, we mainly confine ourselves to distributions of discrete variables, each having a finite number of states. We will let V denote the index set of a collection of variables $(X_v), v \in V$ taking values in probability spaces $\mathcal{X}_v, v \in V$. The probability spaces can be quite general, just sufficiently well-behaved to ensure the existence of regular conditional probabilities. For A being a typical subset of

V, we let $\mathcal{X}_A = \times_{v\in A}\mathcal{X}_v$ and further $\mathcal{X} = \mathcal{X}_V$. Typical elements of $\mathcal{X}_A$ are denoted by $x_A = (x_v)_{v\in A}$ and so on.

Thus, let $X, Y, Z, \ldots$ denote random variables with a joint distribution P, having density p with respect to a product measure. Note that each variable may itself be a random vector. We can consider, for any possible value y of Y, the conditional distribution of X given $Y = y$, denoted by $D(X \mid Y = y)$. This is defined if $p(y) > 0$, in which case we may call y a *possible value* of Y. Thus, if A denotes a set of possible values for X, then $D(X \mid Y = y)$ attaches to A the conditional probability value $P(X \in A \mid Y = y)$.

Definition 5.1 [CONDITIONAL INDEPENDENCE]
We say X is *conditionally independent of Y given Z*, and write $X \perp\!\!\!\perp Y \mid Z$, if, for any possible pair of values (y, z) for (Y, Z), we have $D(X \mid Y = y, Z = z) = D(X \mid Z = z)$, i.e., for any A, $P(X \in A \mid Y, Z) = P(X \mid Z)$. □

As a special case we note that the expression $X \perp\!\!\!\perp Y$ means that we have $D(X \mid Y = y) = D(X)$ (the marginal distribution of X), for any possible value y of Y; we say that X and Y are (marginally) *independent*.

Suppose for simplicity that all variables are discrete. Similar properties hold for continuous quantities. Let $p(x, y \mid z)$ denote $P(X = x, Y = y \mid Z = z)$, and let $a(x, z)$, for example, denote unspecified functions of x, z, etc. Then $X \perp\!\!\!\perp Y \mid Z$ if and only if any of the following equivalent conditions holds:

C1a :	$p(x \mid y, z) \equiv p(x \mid z)$	if $p(y, z) > 0$
C1b :	$p(x \mid y, z)$ has the form $a(x, z)$	if $p(y, z) > 0$
C2a :	$p(x, y \mid z) \equiv p(x \mid z)p(y \mid z)$	if $p(z) > 0$
C2b :	$p(x, y \mid z)$ has the form $a(x, z)b(y, z)$	if $p(z) > 0$
C3a :	$p(x, y, z) \equiv p(x \mid z)p(y \mid z)p(z)$	
C3b :	$p(x, y, z) \equiv p(x, z)p(y, z)/p(z)$	if $p(z) > 0$
C3c :	$p(x, y, z)$ has the form $a(x, z)b(y, z)$	

The ternary relation $X \perp\!\!\!\perp Y \mid Z$ has the following properties:

P1 : If	$X \perp\!\!\!\perp Y \mid Z$		then	$Y \perp\!\!\!\perp X \mid Z$
P2 : If	$X \perp\!\!\!\perp Y \mid Z$ and	U is a function of X	then	$U \perp\!\!\!\perp Y \mid Z$
P3 : If	$X \perp\!\!\!\perp Y \mid Z$ and	U is a function of X	then	$X \perp\!\!\!\perp Y \mid (Z, U)$
P4 : If	$X \perp\!\!\!\perp Y \mid Z$ and	$X \perp\!\!\!\perp W \mid (Y, Z)$	then	$X \perp\!\!\!\perp (W, Y) \mid Z$

Another property sometimes holds, viz.:

P5 : If $X \perp\!\!\!\perp Y \mid (Z, W)$ and $X \perp\!\!\!\perp Z \mid (Y, W)$ then $X \perp\!\!\!\perp (Y, Z) \mid W$,

but only under additional assumptions, essentially that there be no non-trivial logical relationships between Y and Z. This is true when the density

p is strictly positive. For if $p(x, y, z, w) > 0$ and both $X \perp\!\!\!\perp Y \mid (Z, W)$ and $X \perp\!\!\!\perp Z \mid (Y, W)$ hold, then by C3c

$$p(x, y, z, w) = g(x, y, w)\, h(y, z, w) = k(x, z, w)\, l(y, z, w)$$

for suitable strictly positive functions g, h, k, l. Thus, for all z we must have

$$g(x, y, w) = \frac{k(x, z, w)\, l(y, z, w)}{h(y, z, w)}.$$

Hence, choosing a fixed $z = z_0$ we have $g(x, y, w) = \pi(x, w)\rho(y, w)$, where $\pi(x, w) = k(x, z_0, w)$ and $\rho(y, w) = l(y, z_0, w)/h(y, z_0, w)$. Thus, $p(x, y, z, w) = \pi(x, w)\rho(y, w)h(y, z, w)$, and hence $X \perp\!\!\!\perp (Y, Z) \mid W$.

If properties (P1) to (P4) are regarded as axioms with "is a function of" replaced by a suitable partial order, then it is possible to develop an *abstract calculus of conditional independence* which applies to other mathematical systems than probabilistic conditional independence. Any such model of these abstract axioms has been termed a *semi-graphoid* by Pearl (1988) or, if (P5) is also satisfied, a *graphoid*. A range of examples is described by Dawid (1998). Important statistical applications include *meta* conditional independence, which generalizes the concept of a *cut* in a parametric statistical family (Barndorff-Nielsen 1978); and *hyper* conditional independence, which imposes, in addition, corresponding independence properties on a prior distribution over the parameters. A detailed description and discussion may be found in Dawid and Lauritzen (1993). Further application areas of interest in artificial intelligence include concepts of conditional independence for belief functions (Shafer 1976) and various purely logical structures such as, e.g., embedded multi-valued dependencies (Sagiv and Walecka 1982) and natural conditional functions (Spohn 1988; Studený 1995). Purely mathematical examples include orthogonality of linear spaces and the various separation properties in graphs that form the basis of this chapter. The last properties form the reason for Pearl's nomenclature. Virtually all the general results on conditional independence which can be shown to hold for probability distributions can be reinterpreted in these alternative models, and remain valid.

Now let $(X_v), v \in V$ be a collection of random variables, and let $\mathcal{B}$ be a collection of subsets of V. For $B \in \mathcal{B}$, let $a_B(x)$ denote a non-negative function of x depending only on $x_B = (x_v)_{v \in B}$.

Definition 5.2 [HIERARCHICAL DISTRIBUTION]
A joint distribution P for X is $\mathcal{B}$-hierarchical if its probability density p factorizes as

$$p(x) = \prod_{B \in \mathcal{B}} a_B(x).$$

□

Note that if all of the functions a are strictly positive, then P is $\mathcal{B}$-hierarchical if and only if it satisfies the restrictions of a hierarchical log–linear model with generating class $\mathcal{B}^*$ (Christensen 1990), where $\mathcal{B}^*$ is obtained from $\mathcal{B}$ by removing sets that are subsets of other sets in $\mathcal{B}$.

Example 5.3 Let $V = \{A, B, C\}$ and $\mathcal{B} = \{\{A, B\}, \{B, C\}\}$; then the density factorizes as $p(x_A, x_B, x_C) = a(x_A, x_B)b(x_B, x_C)$. □

Example 5.4 Now let $V = \{A, B, C\}$ and $\mathcal{B} = \{\{A, B\}, \{B, C\}, \{A, C\}\}$; then the density factorizes as $p(x_A, x_B, x_C) = a(x_A, x_B)b(x_B, x_C)c(x_A, x_C)$. □

In Example 5.3 the factorization is equivalent to having $X_A \perp\!\!\!\perp X_C \mid X_B$, but Example 5.4 shows that not all factorizations have representations in terms of conditional independence. We can ask what conditional independence properties are implicit in a hierarchical distribution? To answer this we form an undirected graph $\mathcal{G}$ with node set V, in which we join nodes u and v if and only if they occur together within any subset in $\mathcal{B}$. Figure 5.1 shows the graphs obtained in this way for the Examples 5.3 and 5.4.

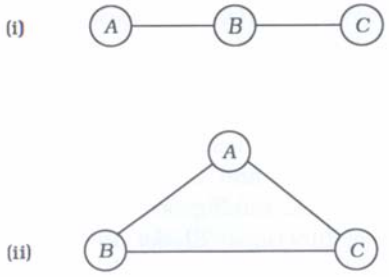


FIGURE 5.1. Undirected graphs formed from the hierarchical distributions of (i) Example 5.3 and (ii) Example 5.4.

Clearly any subset in $\mathcal{B}$ is a complete subset of $\mathcal{G}$. However, in the process of forming $\mathcal{G}$, other complete sets not belonging to $\mathcal{B}$ may be introduced, for example, $\{A, B, C\}$ in Example 5.4. Now let $\mathcal{C}$ denote the collection of cliques of $\mathcal{G}$. Since any subset in $\mathcal{B}$ is contained within some clique in $\mathcal{C}$, it follows that every $\mathcal{B}$-hierarchical distribution is also $\mathcal{C}$-hierarchical. Thus, in discussing the conditional independence properties of hierarchical distributions, we are led to consider the cliques of an underlying graph and, as we shall see, separation properties in such graphs.

5.2 Markov fields over undirected graphs

Let $\mathcal{G} = (V, E)$ be an undirected graph, and consider a collection of random variables $(X_v), v \in V$. Let P be probability measure on $\mathcal{X}_v$ which *factorizes*

according to $\mathcal{G}$, i.e., there exist non-negative functions ϕ_A defined on $\mathcal{X}_A$ for only complete subsets A, and a product measure $\mu = \otimes_{v \in V} \mu_v$ on $\mathcal{X}$, such that the density p of P with respect to μ factorizes in the form

$$p(x) = \prod_A \phi_A(x_A).$$

In other words, P factorizes if and only if P is $\mathcal{A}$-hierarchical, for $\mathcal{A}$ the class of complete subsets of $\mathcal{G}$. The functions ϕ_A are referred to as *factor potentials* of P. They are not uniquely determined because there is arbitrariness in the choice of the measure μ, and also groups of potentials can be multiplied together or split up in different ways. One can without loss of generality assume that only the cliques of $\mathcal{G}$ appear in the sets $\mathcal{A}$, i.e., that

$$p(x) = \prod_{C \in \mathcal{C}} \psi_C(x_C), \tag{5.1}$$

where $\mathcal{C}$ is the set of cliques of $\mathcal{G}$, or, in other words, that P is $\mathcal{C}$-hierarchical. If P factorizes as above, we also say that P has property (F).

Associated with the graph $\mathcal{G}$ is a range of Markov properties, different in general. Write $A \perp\!\!\!\perp B \mid C$ if $X_A \perp\!\!\!\perp X_B \mid X_C$ under P. A probability measure P on $\mathcal{X}$ is said to obey:

(P) *the pairwise Markov property*, relative to $\mathcal{G}$, if for any pair (α, β) of non-adjacent vertices,

$$\alpha \perp\!\!\!\perp \beta \mid V \setminus \{\alpha, \beta\};$$

(L) *the local Markov property*, relative to $\mathcal{G}$, if for any vertex $\alpha \in V$,

$$\alpha \perp\!\!\!\perp V \setminus \mathrm{cl}(\alpha) \mid \mathrm{bd}(\alpha);$$

(G) *the global Markov property*, relative to $\mathcal{G}$, if for any triple (A, B, S) of disjoint subsets of V such that S separates A from B in $\mathcal{G}$,

$$A \perp\!\!\!\perp B \mid S.$$

Note that, if we write $A \perp\!\!\!\perp_{\mathcal{G}} B \mid S$ to denote that S separates A from B in $\mathcal{G}$, replace "function" by "subset" in (P2) and (P3), and similarly (Z, U) by $Z \cup U$, etc., then the subsets of V constitute a graphoid under $\perp\!\!\!\perp_{\mathcal{G}}$, and the various Markov definitions relate properties of probabilistic conditional independence $\perp\!\!\!\perp$ to corresponding obvious properties of graph separation $\perp\!\!\!\perp_{\mathcal{G}}$.

In the terminology defined above we have that

$$(\mathrm{F}) \Rightarrow (\mathrm{G}) \Rightarrow (\mathrm{L}) \Rightarrow (\mathrm{P}), \tag{5.2}$$

but in general the properties are different. Note that (5.2) only depends on the properties (P1) to (P4) of conditional independence. If P admits a strictly positive density p with respect to μ, (P5) can also be used and then all the properties are equivalent. This is a consequence of the theorem below, due to Pearl and Paz (1987) (see also Pearl (1988)).

Theorem 5.5 (Pearl and Paz) *If a probability distribution on $\mathcal{X}$ is such that* (P5) *holds for all pairwise disjoint subsets, then*

$$(\mathrm{G}) \iff (\mathrm{L}) \iff (\mathrm{P}).$$

Proof. We need to show that (P) implies (G), so assume that S separates A from B in $\mathcal{G}$ and that (P) as well as (P5) hold. The proof is then reverse induction on the number of vertices $n = |S|$ in S. If $n = |V| - 2$, then both A and B consist of one vertex, and the required conditional independence follows from (P).

So assume $|S| = n < |V| - 2$, and that the independence $A \perp\!\!\!\perp B \mid S$ holds for all S with more than n elements. We first assume that $A \cup B \cup S = V$, implying that at least one of A and B has more than one element, A, say. If $\alpha \in A$ then $S \cup \{\alpha\}$ separates B from $A \setminus \{\alpha\}$, and also $S \cup A \setminus \{\alpha\}$ separates B from α. Thus, by the inductive hypothesis

$$B \perp\!\!\!\perp A \setminus \{\alpha\} \mid S \cup \{\alpha\} \text{ and } B \perp\!\!\!\perp \alpha \mid S \cup A \setminus \{\alpha\}.$$

Now (P5) gives $A \perp\!\!\!\perp B \mid S$.

If $A \cup B \cup S \subset V$ we choose $\alpha \in V \setminus (A \cup B \cup S)$. Then $S \cup \{\alpha\}$ separates A and B, implying $A \perp\!\!\!\perp B \mid S \cup \{\alpha\}$. Further, either $A \cup S$ separates B from $\{\alpha\}$ or $B \cup S$ separates A from $\{\alpha\}$. Assuming the former gives $B \perp\!\!\!\perp \{\alpha\} \mid A \cup S$. Using (P5) we derive $A \perp\!\!\!\perp B \mid S$. The latter case is similar. □

The global Markov property (G) is important because it gives a general criterion for deciding when two groups of variables A and B are conditionally independent given a third group of variables S.

In the case where all state spaces are discrete and P has a positive density, we can show that (P) implies (F), and thus that all Markov properties are equivalent. More precisely, we have the classical result:

Theorem 5.6 *A probability distribution P on a discrete sample space with strictly positive density satisfies the pairwise Markov property if and only if it factorizes.*

Proof. See Lauritzen (1996). □

In general, without positivity assumptions on the density, the global Markov property (G) may not imply the factorization property (F). An example was given by Moussouris (1974) for the graph being a four-cycle (see also Lauritzen (1996)).

When we use the term *Markov probability distribution* on an undirected graph $\mathcal{G}$ without further qualification, we shall always mean one that factorizes, hence satisfies all of the properties. The set of such probability distributions is denoted by $M(\mathcal{G})$. When (A, B, S) forms a decomposition of $\mathcal{G}$ the Markov property is decomposed accordingly:

Proposition 5.7 *Assume that (A, B, S) decomposes $\mathcal{G} = (V, E)$. Then P factorizes with respect to $\mathcal{G}$ if and only if both $P_{A\cup S}$ and $P_{B\cup S}$ factorize with respect to $\mathcal{G}_{A\cup S}$ and $\mathcal{G}_{B\cup S}$ respectively and the density p satisfies*

$$p(x) = \frac{p_{A\cup S}(x_{A\cup S})p_{B\cup S}(x_{B\cup S})}{p_S(x_S)}. \tag{5.3}$$

Proof. Suppose that P factorizes with respect to $\mathcal{G}$ such that

$$p(x) = \prod_{C\in\mathcal{C}} \psi_C(x).$$

Since (A, B, S) decomposes $\mathcal{G}$, all cliques are subsets of either $A \cup S$ or of $B \cup S$, so that

$$p(x) = \prod_{C\in\mathcal{A}} \psi_C(x) \prod_{C\in\mathcal{B}} \psi_C(x) = h(x_{A\cup S})k(x_{B\cup S}).$$

By direct integration we find

$$p(x_{A\cup S}) = h(x_{A\cup S})\bar{k}(x_S)$$

where

$$\bar{k}(x_S) = \int k(x_{B\cup S})\mu_B(dx_B),$$

and similarly with the other marginals. This gives (5.3) as well as the factorizations of both marginal densities. The converse is trivial. □

In the case of discrete sample spaces we further have, if we take $0/0 = 0$:

Proposition 5.8 *Assume that (A, B, S) decomposes $\mathcal{G} = (V, E)$ and the sample space is discrete. Then P is globally Markov with respect to $\mathcal{G}$ if and only if both $P_{A\cup S}$ and $P_{B\cup S}$ are globally Markov with respect to $\mathcal{G}_{A\cup S}$ and $\mathcal{G}_{B\cup S}$ respectively, and*

$$p(x) = \frac{p(x_{A\cup S})p(x_{B\cup S})}{p(x_S)}. \tag{5.4}$$

Proof. See Lauritzen (1996). □

When $\mathcal{G}$ is decomposable, recursive application of (5.3) shows that a distribution P is Markov with respect to $\mathcal{G}$ if and only if it factorizes as

$$p(x) = \frac{\prod_{C\in\mathcal{C}} p(x_C)}{\prod_{S\in\mathcal{S}} p(x_S)},$$

where $\mathcal{C}, \mathcal{S}$ are, respectively, the sets of cliques and separators of $\mathcal{G}$. The clique-marginals $\{P_C\}$ can be assigned arbitrarily, subject only to implying identical marginals over any common separators. Markov properties of decomposable graphs are studied by Dawid and Lauritzen (1993).

5.3 Markov properties on directed acyclic graphs

Before we proceed to the case of a general chain graph we consider the same set-up as in the previous section, except that now the graph $\mathcal{D}$ is assumed to be directed and acyclic.

We say that a probability distribution P admits a *recursive factorization* according to $\mathcal{D}$ if there exist (σ-finite) measures μ_v over $\mathcal{X}$ and non-negative functions $k^v(\cdot,\cdot), v \in V$, henceforth referred to as *kernels*, defined on $\mathcal{X}_v \times \mathcal{X}_{\mathrm{pa}(v)}$ such that

$$\int k^v(y_v, x_{\mathrm{pa}(v)})\mu_v(dy_v) = 1,$$

and P has density p with respect to the product measure $\mu = \otimes_{v\in V}\mu_v$ given by

$$p(x) = \prod_{v\in V} k^v(x_v, x_{\mathrm{pa}(v)}).$$

We then also say that P *has property* (DF). It is easy to show that if P admits a recursive factorization as above, then the kernels $k^v(\cdot, x_{\mathrm{pa}(v)})$ are in fact densities for the conditional distribution of X_v, given $X_{\mathrm{pa}(v)} = x_{\mathrm{pa}(v)}$, and thus

$$p(x) = \prod_{v\in V} p(x_v \mid x_{\mathrm{pa}(v)}). \tag{5.5}$$

Also it is immediate that if we form the (undirected) moral graph $\mathcal{D}^m$ (see Section 4.1) we have the following:

Lemma 5.9 *If P admits a recursive factorization according to the directed acyclic graph $\mathcal{D}$, it factorizes according to the moral graph $\mathcal{D}^m$ and therefore obeys the global Markov property relative to $\mathcal{D}^m$.*

Proof. The factorization follows from the fact that, by construction, the sets $\{v\} \cup \mathrm{pa}(v)$ are complete in $\mathcal{D}^m$ and we can therefore let $\psi_{\{v\}\cup\mathrm{pa}(v)} = k^v$. The remaining part of the statement follows from (5.2). □

This simple lemma has very useful consequences when constructing the inference engine in a probabilistic expert system (see Section 3.2.1 for an

example of this). Also, using the local Markov property on the moral graph $\mathcal{D}^m$, we find that

$$v \perp\!\!\!\perp V \setminus v \,|\, \mathrm{bl}(v),$$

where $\mathrm{bl}(v)$ is the so-called *Markov blanket* of v. The Markov blanket is the set of neighbours of v in the moral graph $\mathcal{D}^m$. It can be found directly from the original DAG $\mathcal{D}$ as the set of v's parents, children, and children's parents:

$$\mathrm{bl}(v) = \mathrm{pa}(v) \cup \mathrm{ch}(v) \cup \{w : \mathrm{ch}(w) \cap \mathrm{ch}(v) \neq \emptyset\}. \tag{5.6}$$

The following result is easily shown:

Proposition 5.10 *If P admits a recursive factorization according to the directed acyclic graph $\mathcal{D}$ and A is an ancestral set, then the marginal distribution P_A admits a recursive factorization according to $\mathcal{D}_A$.*

Corollary 5.11 *Let P factorize recursively according to $\mathcal{D}$. Then*

$$A \perp\!\!\!\perp B \,|\, S$$

whenever A and B are separated by S in $(\mathcal{D}_{\mathrm{An}(A \cup B \cup S)})^m$, the moral graph of the smallest ancestral set containing $A \cup B \cup S$.

The property in Corollary 5.11 will be referred to as the *directed global Markov property* (DG), and a distribution satisfying it is a *directed Markov field* over $\mathcal{D}$. If we now reinterpret $\perp\!\!\!\perp_{\mathcal{D}}$ to denote the relation between subsets described in Corollary 5.11, the subsets of V again form a graphoid under $\perp\!\!\!\perp_{\mathcal{D}}$, and the global directed Markov property again relates probabilistic conditional independence $\perp\!\!\!\perp$ with graph separation $\perp\!\!\!\perp_{\mathcal{D}}$.

One can show that the global directed Markov property has the same role as the global Markov property does in the case of an undirected graph, in the sense that it gives the sharpest possible rule for reading conditional independence relations off the directed graph. The procedure is illustrated in the following example:

Example 5.12 Consider a directed Markov field on the first graph in Figure 5.2 and the problem of deciding whether $a \perp\!\!\!\perp b \,|\, S$. The moral graph of the smallest ancestral set containing all the variables involved is shown in the second graph of Figure 5.2. It is immediate that S separates a from b in this moral graph, implying $a \perp\!\!\!\perp b \,|\, S$. □

An alternative formulation of the global directed Markov property was given by Pearl (1986a) with a formal treatment in Verma and Pearl (1990). Recall that a trail in $\mathcal{D}$ is a sequence of vertices that forms a path in the undirected version $\mathcal{D}^{\sim}$ of $\mathcal{D}$, i.e., when the directions of arrows are ignored. A trail π from a to b in a directed acyclic graph $\mathcal{D}$ is said to be *blocked* by S if it contains a vertex $\gamma \in \pi$ such that either

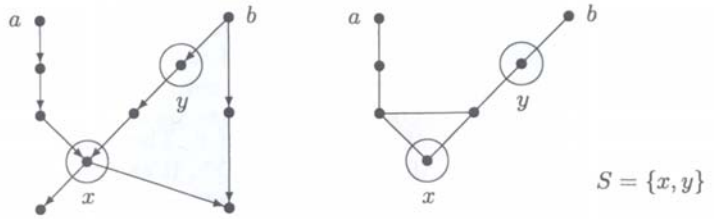


FIGURE 5.2. The directed global Markov property. Is $a \perp\!\!\!\perp b \mid S$? In the moral graph of the smallest ancestral set in the graph containing $\{a\} \cup \{b\} \cup S$, clearly S separates a from b, implying $a \perp\!\!\!\perp b \mid S$.

- $\gamma \in S$ and arrows of π do not meet head-to-head at γ, or
- γ and all its descendants are not in S, and arrows of π meet head-to-head at γ.

A trail that is not blocked by S is said to be *active*. Two subsets A and B are said to be *d-separated* by S if all trails from A to B are blocked by S. We then have the following result:

Proposition 5.13 *Let A, B, and S be disjoint subsets of a directed acyclic graph $\mathcal{D}$. Then S d-separates A from B if and only if S separates A from B in $(\mathcal{D}_{\mathrm{An}(A \cup B \cup S)})^m$.*

Proof. Suppose S does not d-separate A from B. Then there is an active trail from A to B such as, for example, the one indicated in Figure 5.3.

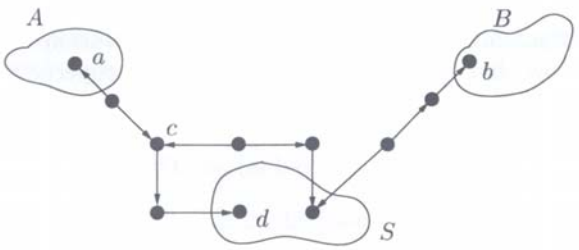


FIGURE 5.3. Example of an active trail from A to B. The path from c to d is not part of the trail, but indicates that c must have descendants in S.

All vertices in this trail must lie within $\mathrm{An}(A \cup B \cup S)$. Because if the arrows meet head-to-head at some vertex γ, either $\gamma \in S$ or γ has descendants in S. And if not, either of the subpaths away from γ either meets another arrow, in which case γ has descendants in S, or leads all the way to A or B. Each of these head-to-head meetings will give rise to a marriage in the moral graph, such as illustrated in Figure 5.4, thereby creating a trail from A to B in $(\mathcal{D}_{\mathrm{An}(A \cup B \cup S)})^m$, circumventing S.

Suppose conversely that A is not separated from B in $(\mathcal{D}_{\mathrm{An}(A\cup B\cup S)})^m$. Then there is a trail in this graph that circumvents S. The trail has pieces that correspond to edges in the original graph and pieces that correspond to marriages. Each marriage is a consequence of a meeting of arrows head-to-head at some vertex γ. If γ is in S or it has descendants in S, the meeting does not block the trail. If not, γ must have descendants in A or B since the ancestral set was smallest. In the latter case, a new trail can be created with one less head-to-head meeting, using the line of descent, such as illustrated in Figure 5.5.

Continuing this substitution process eventually leads to an active trail from A to B, and the proof is complete. □

We illustrate the concept of d-separation by applying it to the query of Example 5.12. As Figure 5.6 indicates, all trails between a and b are blocked by S, whereby the global Markov property gives that $a \perp\!\!\!\perp b \,|\, S$.

Geiger and Pearl (1990) show that the criterion of d-separation cannot be improved, in the sense that, for any given directed acyclic graph $\mathcal{D}$, one can find state spaces $\mathcal{X}_\alpha, \alpha \in V$ and a probability P such that

$$A \perp\!\!\!\perp B \,|\, S \iff S \text{ } d\text{-separates } A \text{ from } B. \tag{5.7}$$

Indeed, we can take each state space to be the real plane, with the overall distribution Gaussian. Meek (1995) proved a similar result for the case where the state spaces are all binary.

A variant on the d-separation criterion, well-suited to computation of separating sets, is the "Bayes-ball" algorithm of Shachter (1998).

To complete this section we say that P obeys the *local directed Markov property* (DL) if any variable is conditionally independent of its non-descendants, given its parents

$$v \perp\!\!\!\perp \mathrm{nd}(v) \,|\, \mathrm{pa}(v).$$

A seemingly weaker requirement, the *ordered directed Markov property* (DO), replaces all non-descendants of v in the above condition by the predecessors $\mathrm{pr}(v)$ of v in some given well-ordering of the nodes:

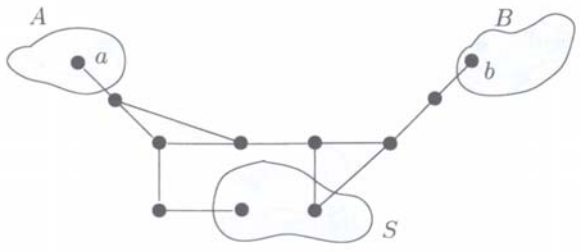


FIGURE 5.4. The moral graph corresponding to the active trail in $\mathcal{D}$.

$$v \perp\!\!\!\perp \mathrm{pr}(v) \mid \mathrm{pa}(v).$$

In contrast with the undirected case we have that all the four properties (DF), (DL), (DG), and (DO) are equivalent just assuming existence of the density p. This is stated formally as:

Theorem 5.14 *Let $\mathcal{D}$ be a directed acyclic graph. For a probability distribution P on $\mathcal{X}$ which has density with respect to a product measure μ, the following conditions are equivalent:*

(DF) *P admits a recursive factorization according to $\mathcal{D}$;*
(DG) *P obeys the global directed Markov property, relative to $\mathcal{D}$;*
(DL) *P obeys the local directed Markov property, relative to $\mathcal{D}$;*
(DO) *P obeys the ordered directed Markov property, relative to $\mathcal{D}$.*

Proof. That (DF) implies (DG) is Corollary 5.11. That (DG) implies (DL) follows by observing that $\{v\} \cup \mathrm{nd}(v)$ is an ancestral set and that $\mathrm{pa}(v)$ obviously separates $\{v\}$ from $\mathrm{nd}(v) \setminus \mathrm{pa}(v)$ in $(\mathcal{D}_{\{v\}\cup\mathrm{nd}(v)})^m$. It is trivial that (DL) implies (DO), since $\mathrm{pr}(v) \subseteq \mathrm{nd}(v)$. The final implication is shown by induction on the number of vertices $|V|$ of $\mathcal{D}$. Let v_0 be the last vertex of $\mathcal{D}$. Then we can let k^{v_0} be the conditional density of X_{v_0}, given $X_{V\setminus\{v_0\}}$, which by (DO) can be chosen to depend on $x_{\mathrm{pa}(v_0)}$ only. The marginal distribution of $X_{V\setminus\{v_0\}}$ trivially obeys the ordered directed Markov property and admits a factorization by the inductive assumption. Combining this factorization with k^{v_0} yields the factorization for P. This completes the proof. □

Since the four conditions in Theorem 5.14 are equivalent, it makes sense to speak of a *directed Markov field* as one where any of the conditions is satisfied. The set of such distributions for a directed acyclic graph $\mathcal{D}$ is denoted by $M(\mathcal{D})$.

In the particular case when the directed acyclic graph $\mathcal{D}$ is perfect (see Section 4.2) the directed Markov property on $\mathcal{D}$ and the factorization prop-

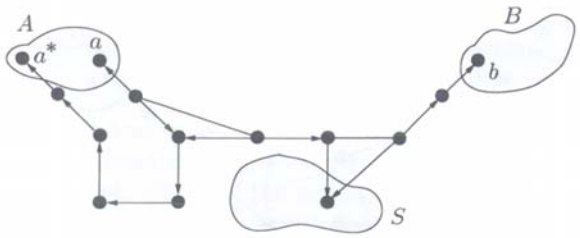


FIGURE 5.5. The trail in the graph $(\mathcal{D}_{\mathrm{An}(A\cup B\cup S)})^m$ makes it possible to construct an active trail in $\mathcal{D}$ from A to B.

erty on its undirected version $\mathcal{D}^\sim$ coincide (note that in this case $\mathcal{D}^\sim$ is decomposable):

Proposition 5.15 *Let $\mathcal{D}$ be a perfect directed acyclic graph and $\mathcal{D}^\sim$ its undirected version. Then a distribution P is directed Markov with respect to $\mathcal{D}$ if and only if it factorizes according to $\mathcal{D}^\sim$.*

Proof. That the graph is perfect means that $\mathrm{pa}(\alpha)$ is complete for all $\alpha \in V$. Hence, $\mathcal{D}^m = \mathcal{D}^\sim$. From Lemma 5.9 it then follows that any $P \in M(\mathcal{D})$ also factorizes with respect to $\mathcal{D}^\sim$.

The reverse inclusion is established by induction on the number of vertices $|V|$ of $\mathcal{D}$. For $|V| = 1$ there is nothing to show. For $|V| = n + 1$ let $P \in M(\mathcal{D}^\sim)$ and find a terminal vertex $\alpha \in V$. This vertex has $\mathrm{pa}_\mathcal{D}(\alpha) = \mathrm{bd}_{\mathcal{D}^\sim}(\alpha)$ and, since $\mathcal{D}$ is perfect, this set is complete in both graphs as well. Hence, $(V \setminus \{\alpha\}, \{\alpha\}, \mathrm{bd}(\alpha))$ is a decomposition of $\mathcal{D}^\sim$ and Proposition 5.7 gives the factorization

$$p(x) = p(x_{V\setminus\{\alpha\}})p(x_{\mathrm{cl}(\alpha)})/p(x_{\mathrm{bd}(\alpha)}) = p(x_{V\setminus\{\alpha\}})k^\alpha(x_\alpha, x_{\mathrm{pa}(\alpha)}),$$

say, where $\int k^\alpha(y_\alpha, x_{\mathrm{pa}(\alpha)})\mu_\alpha(dy_\alpha) = 1$, and the first factor factorizes according to $\mathcal{D}^\sim_{V\setminus\{\alpha\}}$. Using the inductive assumption on this factor gives the full recursive factorization of P. □

5.4 Markov properties on chain graphs

In this section we deal with general chain graphs $\mathcal{K} = (V, E)$. We further assume that all probability measures have positive densities, implying that all five of the basic properties of conditional independence (P1) to (P5) hold. Again there is a pairwise, a local, and a global Markov property. More precisely we say that a probability P satisfies:

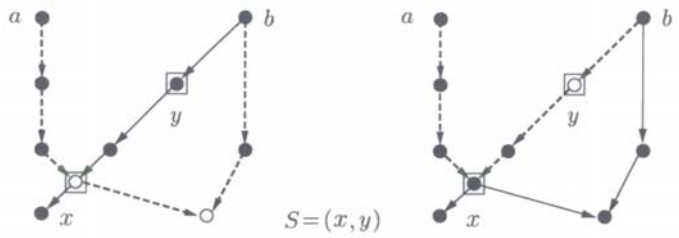


FIGURE 5.6. Illustration of Pearl's d-separation criterion. There are two trails from a to b, drawn with broken lines. Both are blocked, but different vertices γ, indicated with open circles, play the role of blocking vertices.

(CP) the *pairwise chain Markov property*, relative to $\mathcal{K}$, if for any pair (α, β) of non-adjacent vertices with $\beta \in \mathrm{nd}(\alpha)$,

$$\alpha \perp\!\!\!\perp \beta \,|\, \mathrm{nd}(\alpha) \setminus \{\alpha, \beta\};$$

(CL) the *local chain Markov property*, relative to $\mathcal{K}$, if for any vertex $\alpha \in V$,

$$\alpha \perp\!\!\!\perp \mathrm{nd}(\alpha) \setminus \mathrm{bd}(\alpha) \,|\, \mathrm{bd}(\alpha);$$

(CG) the *global chain Markov property*, relative to $\mathcal{K}$, if for any triple (A, B, S) of disjoint subsets of V such that S separates A from B in $(\mathcal{K}_{\mathrm{An}(A \cup B \cup S)})^m$, the moral graph of the smallest ancestral set containing $A \cup B \cup S$, we have

$$A \perp\!\!\!\perp B \,|\, S.$$

Studený and Bouckaert (1998) have introduced a definition of *c-separation* of A and B by S in a chain graph, which extends d-separation for directed acyclic graphs and is equivalent to the separation property used in the global chain Markov property above.

Once again, the graph separation property $\perp\!\!\!\perp_{\mathcal{K}}$ described in (CG) engenders a graphoid structure on subsets of V, and the Markov properties relate $\perp\!\!\!\perp$ and $\perp\!\!\!\perp_{\mathcal{K}}$. We note that these Markov properties unify the properties for the directed and undirected cases. For in the undirected case $\mathrm{nd}(\alpha) = V$ and $\mathcal{K} = (\mathcal{K}_{\mathrm{An}(A \cup B \cup S)})^m$, and in the directed case $\mathrm{bd}(\alpha) = \mathrm{pa}(\alpha)$.

When interpreting the conditional independence relationships in a chain graph, it is occasionally more straightforward to use the following approach, an extension of the ordered directed Markov property for directed acyclic graphs: since the graph is a chain graph, the vertex set can be partitioned as $V = V(1) \cup \cdots \cup V(T)$ such that each of the sets $V(t)$ only has undirected edges between its vertices, and any directed edges point from vertices in sets with lower number to those with higher number. Such a partition is called a *dependence chain*. The set of *concurrent* variables of $V(t)$ is defined to be the set $C(t) = V(1) \cup \cdots \cup V(t)$. Then P satisfies the *block-recursive Markov property* (CB) if for any pair (α, β) of non-adjacent vertices we have

$$\alpha \perp\!\!\!\perp \beta \,|\, C(t^*) \setminus \{\alpha, \beta\},$$

where t^* is the smallest t that has $\{\alpha, \beta\} \subseteq C(t)$. It appears that this property depends on the particular partitioning, but it can be shown (Frydenberg 1990) that — if P satisfies (P5) — it is equivalent to any of the above.

Theorem 5.16 *Assume that P is such that* (P5) *holds for subsets of V. Then*

$$(\mathrm{CG}) \iff (\mathrm{CL}) \iff (\mathrm{CP}) \iff (\mathrm{CB}).$$

Proof. See Frydenberg (1990). □

Example 5.17 As an illustration of this, consider the graph in Figure 5.7 and the question of deciding whether $3 \perp\!\!\!\perp 8 \mid \{2,5\}$. The smallest ancestral set containing these variables is the set {1,2,3,4,5,6,7,8}. The moral graph of this adds an edge between 3 and 4, because these both have children in the chain component {5,6}. Thus, the graph in Figure 5.8 appears.

Since there is a path between 3 and 8 circumventing 2 and 5 in this graph, we cannot conclude that $3 \perp\!\!\!\perp 8 \mid \{2,5\}$.

If we instead consider the question whether $3 \perp\!\!\!\perp 8 \mid 2$, the smallest ancestral set becomes {1,2,3,4,7,8}, no edge has to be added between 3 and 4, and Figure 5.9 reveals that $3 \perp\!\!\!\perp 8 \mid 2$. □

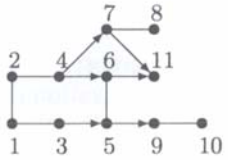


FIGURE 5.7. A chain graph with chain components {1,2,3,4}, {5,6}, {7,8}, {9,10}, {11}. Is $3 \perp\!\!\!\perp 8 \mid \{2,5\}$? Is $3 \perp\!\!\!\perp 8 \mid 2$?

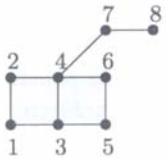


FIGURE 5.8. Moral graph of smallest ancestral set in the graph of Figure 5.7 containing {2,3,5,8}. A connection between 3 and 4 has been introduced since these both have children in the same chain component $\{5,6\}$. We cannot conclude $3 \perp\!\!\!\perp 8 \mid \{2,5\}$.

One way of constructing a distribution that satisfies the chain graph Markov property is through factorization. For example, if $V(1), \ldots, V(T)$ is a dependence chain of $\mathcal{K}$ or the chain components of $\mathcal{K}$, then any distribution P with density p with respect to a product measure μ will factorize as

$$p(x) = \prod_{t=1}^{T} p\left(x_{V(t)} \mid x_{C(t-1)}\right)$$

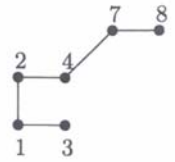


FIGURE 5.9. Moral graph of smallest ancestral set in the graph of Figure 5.7 containing {2,3,8}. We conclude that $3 \perp\!\!\!\perp 2 \mid 8$.

where $C(t) = V(1) \cup \cdots \cup V(t)$ are the concurrent variables, as usual. If $B(t) = \mathrm{pa}(V(t)) = \mathrm{bd}(V(t))$ and p is Markov with respect to $\mathcal{K}$, the factorization reduces to

$$p(x) = \prod_{t=1}^{T} p\left(x_{V(t)} \mid x_{B(t)}\right). \tag{5.8}$$

This factorization is essentially identical to the factorization for directed Markov densities due to the chain graph forming a directed acyclic graph of its chain components. But the factorization does not reveal all conditional independence relationships. To describe the remainder, let $\mathcal{K}^*(t)$ be the undirected graph with vertex set $V(t) \cup B(t)$ and α adjacent to β in $\mathcal{K}^*(t)$ if either $(\alpha, \beta) \in E$ or $(\beta, \alpha) \in E$ or if $\{\alpha, \beta\} \subseteq B(t)$, i.e., $B(t)$ is made complete in $\mathcal{K}^*(t)$ by adding all missing edges between these, and directions on existing edges are ignored. We cannot expect factorization results to be more general for chain graphs than for undirected graphs, since the chain graphs contain these as special cases. But if all variables are discrete, there is a result analogous to Theorem 5.6.

Theorem 5.18 *A probability distribution on a discrete sample space with strictly positive density p satisfies the pairwise chain graph Markov property with respect to $\mathcal{K}$ if and only if it factorizes as*

$$p(x) = \prod_{t=1}^{T} \frac{p\left(x_{V(t) \cup B(t)}\right)}{p\left(x_{B(t)}\right)}, \tag{5.9}$$

and each of the numerators factorizes on the graph $\mathcal{K}^*(t)$.

Proof. See Lauritzen (1996). □

Corollary 5.19 *If the density p of a probability distribution factorizes as in (5.9), it also factorizes according to the moral graph $\mathcal{K}^m$ and therefore obeys the undirected global Markov property relative to $\mathcal{K}^m$.*

Proof. By construction, sets that are complete in $\mathcal{K}^*(t)$ are also complete in $\mathcal{K}^m$. □

An equivalent formulation of the factorization (5.9) is

$$p(x) = \prod_{t=1}^{T} p\left(x_{V(t)} \,|\, x_{B(t)}\right), \tag{5.10}$$

where each factor further factorizes according to $\mathcal{K}^*(t)$. This is true because $B(t)$ is complete in $\mathcal{K}^*(t)$.

Alternatively, each of the factors in (5.10) must further factorize as

$$p\left(x_{V(t)} \,|\, x_{B(t)}\right) = Z^{-1}\left(x_{B(t)}\right) \prod_{A \in \mathcal{A}(t)} \phi_A(x_A), \tag{5.11}$$

where $\mathcal{A}(t)$ are the complete subsets of $\mathcal{K}^*_{V(t)\cup B(t)}$ and

$$Z\left(x_{B(t)}\right) = \sum_{x_{V(t)}} \prod_{A \in \mathcal{A}(t)} \phi_A(x_A).$$

5.5 Current research directions

There has been much recent research activity developing and extending the links between graphical structures and conditional independence properties which have been the subject of this chapter. Some of this concentrates on logical issues, such as *strong completeness*, i.e., whether a probability distribution (perhaps of a special form, e.g., Gaussian or having a positive density) exists displaying all and only the conditional properties displayed by a given graphical representation (Geiger and Pearl 1990, 1993; Studený and Bouckaert 1998). Studený (1997) and Andersson et al. (1998) give good overviews of recent issues and advances in graphical descriptions of conditional independence. We briefly describe some of the major current research themes below.

5.5.1 Markov equivalence

There may be more than one graph representing the same conditional independence relations. A focus of recent attention has been how to characterize such *Markov equivalence* of graphs and, if possible, nominate a natural representative of any equivalence class. Issues of equivalence of seemingly distinct representations need careful attention when attempting statistical model selection on the basis of data (Heckerman et al. 1995b).

Extending a result of Verma and Pearl (1991) for directed acyclic graphs, Frydenberg (1990) showed that two chain graphs are Markov equivalent if and only if they have the same skeleton (i.e., undirected version), and the same complexes, where a *complex* is a subgraph, induced by a set of nodes $\{v_1, v_2, \ldots, v_k\}$ with $k \geq 3$, whose edge set consists of $v_1 \rightarrow v_2$, $v_{k-1} \leftarrow v_k$,

and $v_i \sim v_{i+1}$ for $2 \leq i \leq k-2$. In any class of Markov equivalent chain graphs there is a unique 'largest' one, having the maximum number of undirected edges; its arrows are present in every other member of the class. If we restrict attention to directed acyclic graphs, there is no natural representative of an equivalence class within the class, but it can be characterized by its *essential graph* (Andersson et al. 1996b), the chain graph (with the same skeleton) in which an edge has an arrow if and only if at least one member of the equivalence class has that arrow, and none has the reverse arrow (see Section 11.5).

5.5.2 Other graphical representations

Alternative graphical representations of conditional independence have been considered. Cox and Wermuth (1996) allow their graph edges to be dashed as well as solid lines and arrows, introducing an appropriately modified semantics to relate the graph structure to marginal and conditional independence properties of Gaussian distributions. This approach is related to that of Andersson et al. (1996a), who use chain graphs, but with a new 'AMP' semantics, based on a graph separation property different from that considered here, which they term 'LWF.' The AMP semantics is related to the interpretation of structural equation models (Bollen 1988). It gives rise to many questions similar to those for the LWF approach: equivalence of different descriptions of graphical separation, Markov equivalence of distinct graphs, etc. However, it does not correspond in a simple fashion to a factorization property of the joint density, an aspect that is crucial for computational efficiency.

Further graphical representations include possibly cyclic directed graphs using essentially the same moralization semantics as in the acyclic case; Markov equivalence and related issues have been addressed by Richardson (1996). An extension to 'reciprocal graphs,' a generalization of chain graphs, has been studied by Koster (1996).

Starting from any graphical Markov criterion, we can also consider the effects of collapsing out over unobservable variables, or conditioning on 'selection variables,' thus broadening still further the range of conditional independence structures that may be represented (Cox and Wermuth 1996). Again, issues of equivalence, etc., need addressing (Spirtes and Richardson 1997).

5.6 Background references and further reading

Dawid (1979, 1980b) proposed the axioms of conditional independence, without any graphical connections, and showed how they could be developed as a unifying concept within probability and statistics. Applications

within theoretical statistics include: sufficiency and ancillarity; nuisance parameters (Dawid 1980a); Simpson's paradox; optional stopping, selection and missing data effects (Dawid 1976; Dawid and Dickey 1977); invariance (Dawid 1985); and model-building (Dawid 1982). An overview is given by Dawid (1998).

Graphical representations of probability models have a long history. Directed models can be traced back to the path analysis of Wright (1921, 1923, 1934), and undirected models to the work of Bartlett (1935) on interactions in contingency tables. The latter was taken up by Darroch et al. (1980) and has led to intensive investigation of graphical models in statistics, well summarized by Whittaker (1990) and Lauritzen (1996). The connections between undirected graphs and conditional independence were first made in the unpublished work of Hammersley and Clifford (1971). Statistical use of directed graphs came into its own with the introduction of influence diagrams (Howard and Matheson 1984), but it was the application by Pearl (1986b) (see also Pearl (1988)) to probability calculations in graphical networks which initiated the recent explosion of interest in directed graphical representations. Their Markov properties were explored by Pearl (1986a) and Verma and Pearl (1990) using d-separation, while Lauritzen et al. (1990) introduced the moralization criterion. A detailed study of chain graph representations can be found in Frydenberg (1990).