

Bayesian Networks*

Peter Lucas

Institute for Computing and Information Sciences
University of Nijmegen
Email: peterl@cs.kun.nl

Linda van der Gaag

Institute of Information and Computing Sciences
Utrecht University
Email: linda@cs.uu.nl

Contents

1	Introduction	1
2	Knowledge representation in a Bayesian network	2
3	Evidence propagation in a Bayesian network	5
4	The reasoning method of Kim and Pearl	6
5	The reasoning method of Lauritzen and Spiegelhalter	10

1 Introduction

In the mid-1980s a new trend in probabilistic reasoning with uncertainty in knowledge-based systems became discernable taking a graphical representation of knowledge as a point of departure. We use the phrase *network models* to denote this type of model. In the preceding sections, we have concentrated primarily on models for plausible reasoning that were developed especially for knowledge-based systems using production rules for knowledge representation. In contrast, the network models depart from another knowledge-representation formalism: the so-called *Bayesian network*. Common synonyms for the formalism are: belief network, probabilistic network, Bayesian belief network, and causal probabilistic network. Informally speaking, a Bayesian network is a graphical representation of a problem domain consisting of the statistical variables discerned in the domain and their probabilistic interrelationships. The relationships between the statistical variables are quantified by means of ‘local’ probabilities together defining a total probability function on the variables. This section presents a brief introduction to network models. In Section 2 we shall discuss the way knowledge is

*This report is an adaptation of: “Peter Lucas and Linda van der Gaag. *Principles of Expert Systems*. Addison-Wesley, Wokingham, 1991 (Chapter 5)”.

represented in a Bayesian network. The Sections 4 and 5 discuss two approaches to reasoning with such a network.

2 Knowledge representation in a Bayesian network

We have mentioned before that Bayesian networks provide a formalism for representing a problem domain. A Bayesian network comprises two parts: a *qualitative representation* of the problem domain and an associated *quantitative representation*. The qualitative part takes the form of an acyclic directed graph $G = (V(G), A(G))$ where $V(G) = \{V_1, \dots, V_n\}$, $n \geq 1$, is a finite set of *vertices* and $A(G)$ is a finite set of arcs (V_i, V_j) , $V_i, V_j \in V(G)$. Each vertex V_i in $V(G)$ represents a statistical variable which in general can take one of a set of values. In the sequel, however, we shall assume for simplicity's sake that the statistical variables can take only one of the truth values *true* and *false*. We take an arc $(V_i, V_j) \in A(G)$ to represent a direct 'influential' or 'causal' relationship between the variables V_i and V_j : the arc (V_i, V_j) is interpreted as stating that ' V_i directly influences V_j '. Absence of an arc between two vertices means that the corresponding variables do not influence each other directly. In general, such a directed graph has to be configured by a domain expert from human judgment; hence the phrase *belief network*. We give an example of such a qualitative representation of a problem domain.

Example 1 Consider the following qualitative medical information:

Shortness-of-breath (V_7) may be due to tuberculosis (V_2), lung cancer (V_4) or bronchitis (V_5), or more than one of them. A recent visit to Asia (V_1) increases the chances of tuberculosis, while smoking (V_3) is known to be a risk factor for both lung cancer and bronchitis. The results of a single chest X-ray (V_8) do not discriminate between lung cancer and tuberculosis (V_6), as neither does the presence or absence of shortness-of-breath.

In this information, we may discern several statistical variables; with each variable we have associated a name V_i . The information has been represented in the acyclic directed graph G shown in Figure 1. Each vertex in G represents one of the statistical variables, and the arcs in G represent the causal relationships between the variables. The arc between the vertices V_3 and V_4 for example represents the information that smoking may cause lung cancer. Note that although the graph only depicts direct causal relationships, we can read indirect influences from it. For example, the graph shows that V_3 influences V_7 indirectly through V_4 , V_5 and V_6 : smoking may cause lung cancer and bronchitis, and these may in turn cause shortness-of-breath. However, as soon as V_4 , V_5 and V_6 are known, V_3 itself does not provide any further information concerning V_7 . \square

The qualitative representation of the problem domain now is interpreted as the representation of all probabilistic dependency and independency relationships between the statistical variables discerned. With the graph, a domain expert associates a numerical assessment of the 'strengths' of the represented relationships in terms of a probability function P on the sample space defined by the statistical variables. Before discussing this in further detail, we introduce the notions of predecessor and successor.

Definition 1 Let $G = (V(G), A(G))$ be a directed graph. Vertex $V_j \in V(G)$ is called a successor of vertex $V_i \in V(G)$ if there is an arc $(V_i, V_j) \in A(G)$; alternatively, vertex V_i is

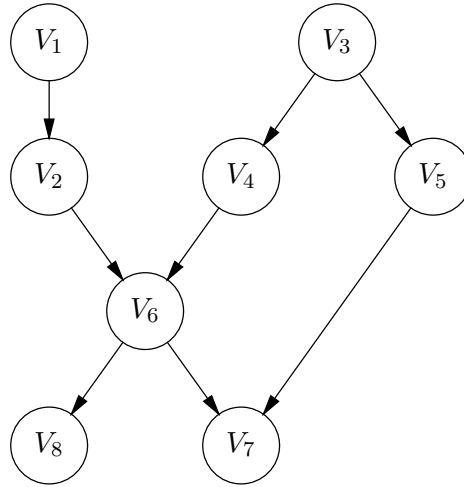


Figure 1: The acyclic directed graph of a Bayesian network.

called a predecessor of vertex V_j . A vertex V_k is a neighbour of V_i if V_k is either a successor or a predecessor of V_i .

Now, for each vertex in the graphical part of a Bayesian network, a set of (conditional) probabilities describing the influence of the values of the predecessors of the vertex on the values of the vertex itself, is specified. We shall illustrate the idea with the help of our example shortly.

We introduce some new notions and notational conventions. From now on, the variable V_i taking the truth value *true* will be denoted by v_i ; the probability that the variable V_i has the value *true* will then be denoted by $P(v_i)$. We use $\neg v_i$ to denote that $V_i = \text{false}$; the probability that $V_i = \text{false}$ then is denoted by $P(\neg v_i)$. Now, let $V(G) = \{V_1, \dots, V_n\}$, $n \geq 1$, again be the set of all statistical variables discerned in the problem domain. We consider a subset $V \subseteq V(G)$ with $m \geq 1$ elements. A conjunction of length m in which for each $V_i \in V$ either v_i or $\neg v_i$ occurs, is called a *configuration* of V . The conjunction $v_1 \wedge \neg v_2 \wedge v_3$ is an example of a configuration of the set $V = \{V_1, V_2, V_3\}$. The conjunction of length m in which each $V_i \in V$ is named only, that is, specified without its value, is called the *configuration template* of V . For example, the configuration template of $V = \{V_1, V_2, V_3\}$ is $V_1 \wedge V_2 \wedge V_3$. Note that we can obtain the configuration $v_1 \wedge \neg v_2 \wedge v_3$ from the template $V_1 \wedge V_2 \wedge V_3$ by filling in v_1 , $\neg v_2$, and v_3 for the variables V_1 , V_2 , and V_3 , respectively. In fact, every possible configuration of a set V can be obtained from its template by filling in proper values for the variables occurring in the template.

We return to the quantitative part of a Bayesian network. With each variable, that is, with each vertex $V_i \in V(G)$ in the qualitative part of the belief network, a domain expert associates conditional probabilities $P(v_i | c)$ for all configurations c of the set of predecessors of V_i in the graph. Note that for a vertex with m incoming arcs, 2^m probabilities have to be assessed; for a vertex V_i with zero predecessors, only one probability has to be specified, namely the prior probability $P(v_i)$.

Example 2 Consider the medical information from the previous example and its graphical representation in Figure 1 once more. For example, with the vertex V_3 the domain expert associates the prior probability that a patient smokes. For the vertex V_4 two conditional

probabilities have to be specified: the probability that a patient has lung cancer given the information that he smokes, that is, the probability $P(v_4 | v_3)$, and the probability that a non-smoker gets lung cancer, that is, the probability $P(v_4 | \neg v_3)$. Corresponding with the graph, a domain expert therefore has to assess the following eighteen probabilities:

$$\begin{aligned}
&P(v_1) \\
&P(v_2 | v_1) \text{ and } P(v_2 | \neg v_1) \\
&P(v_3) \\
&P(v_4 | v_3) \text{ and } P(v_4 | \neg v_3) \\
&P(v_5 | v_3) \text{ and } P(v_5 | \neg v_3) \\
&P(v_6 | v_2 \wedge v_4), P(v_6 | v_2 \wedge \neg v_4), P(v_6 | \neg v_2 \wedge v_4), \text{ and } P(v_6 | \neg v_2 \wedge \neg v_4) \\
&P(v_7 | v_5 \wedge v_6), P(v_7 | v_5 \wedge \neg v_6), P(v_7 | \neg v_5 \wedge v_6), \text{ and } P(v_7 | \neg v_5 \wedge \neg v_6) \\
&P(v_8 | v_6) \text{ and } P(v_8 | \neg v_6)
\end{aligned}$$

Note that from these probabilities we can uniquely compute the ‘complementary’ probabilities; for example, we have that $P(\neg v_7 | v_5 \wedge v_6) = 1 - P(v_7 | v_5 \wedge v_6)$. \square We observe that a probability function P on a sample space defined by n statistical variables V_1, \dots, V_n , $n \geq 1$, is completely described by the probabilities $P(c)$ for all configurations c of $V(G) = \{V_1, \dots, V_n\}$. The reader can easily verify that from these probabilities any other probability may be computed using the axioms of probability theory. In the sequel, therefore, we will frequently use the template $P(V_1 \wedge \dots \wedge V_n)$ to denote a probability function: note that from this template we can obtain the probabilities $P(c)$ for all configurations c of $V(G)$, from which we can compute any probability of interest. Since there are 2^n different configurations c of $V(G)$, in theory 2^n probabilities $P(c)$ are necessary for defining a probability function. In a belief network, however, often far less probabilities suffice for doing so: an important property is that under the assumption that the graphical part of a Bayesian network represents *all* independency relationships between the statistical variables discerned, the probabilities associated with the graph provide enough information to define a unique probability function on the domain of concern. To be more precise, we have

$$P(V_1 \wedge \dots \wedge V_n) = \prod_{i=1}^n P(V_i | C_{\rho(V_i)})$$

where $C_{\rho(V_i)}$ is the configuration template of the set $\rho(V_i)$ of predecessors of V_i . Note that the probability of any configuration of $V(G)$ can be obtained by filling in proper values for the statistical variables V_1 up to V_n inclusive and then computing the resulting product on the right-hand side from the initially assessed probabilities. We look again at our example.

Example 3 Consider the previous examples once more. We have that

$$\begin{aligned}
P(V_1 \wedge \dots \wedge V_8) &= P(V_8 | V_6) \cdot P(V_7 | V_5 \wedge V_6) \cdot P(V_6 | V_2 \wedge V_4) \cdot \dots \\
&\quad P(V_5 | V_3) \cdot \dots \\
&\quad P(V_4 | V_3) \cdot P(V_3) \cdot P(V_2 | V_1) \cdot P(V_1)
\end{aligned}$$

Note that in this example only eighteen probabilities suffice for specifying a probability function on our problem domain. \square In a Bayesian network, the quantitative representation of the problem domain only comprises probabilities that involve a vertex and its predecessors in the qualitative part of the network. Note that the representation of uncertainty in such local factors closely resembles the approach followed in the quasi-probabilistic models in which uncertainty is represented in factors that are local to the production rules constituting the qualitative representation of the domain.

3 Evidence propagation in a Bayesian network

In the preceding section we have introduced the notion of a Bayesian network as a means for representing a problem domain. Such a Bayesian network may be used for reasoning with uncertainty, for example for interpreting pieces of evidence that become available during a consultation. For making probabilistic statements concerning the statistical variables discerned in the problem domain, we have to associate with a Bayesian network two methods:

- A method for computing probabilities of interest from the Bayesian network.
- A method for processing evidence, that is, a method for entering evidence into the network and subsequently computing the conditional probability function given this evidence. This process is generally called *evidence propagation*.

In the relevant literature, the emphasis lies on methods for evidence propagation; in this chapter we do so likewise.

Now recall that the probabilities associated with the graphical part of a Bayesian network uniquely define a probability function on the sample space defined by the statistical variables discerned in the problem domain. The impact of a value of a specific variable becoming known on each of the other variables, that is, the conditional probability function given the evidence, can therefore be computed from these initially assessed local probabilities. The resulting conditional probability function is often called the *updated probability function*. Calculation of a conditional probability from the initially given probabilities in a straightforward manner will generally not be restricted to computations which are local in terms of the graphical part of the Bayesian network. Furthermore, the computational complexity of such an approach is exponential in the number of variables: the method will become prohibitive for larger networks. In the literature, therefore, several less naive schemes for updating a probability function as evidence becomes available have been proposed. Although all methods build on the same notion of a Bayesian network, they differ considerably in concept and in computational complexity. All schemes proposed for evidence propagation however have two important characteristics in common:

- For propagating evidence, the graphical part of a Bayesian network is exploited more or less directly as a computational architecture.
- After a piece of evidence has been processed, again a Bayesian network results. Note that this property renders the notion of a Bayesian network invariant under evidence propagation and therefore allows for recursive application of the method for processing evidence.

In the following two sections, we shall discuss different methods for evidence propagation. In Section 4, we shall discuss the method presented by J.H. Kim and J. Pearl. In this method, computing the updated probability function after a piece of evidence has become available essentially entails each statistical variable (that is, each vertex in the graphical part of the Bayesian network) updating the probability function locally from messages it receives from its neighbours in the graph, that is, from its predecessors as well as its successors, and then in turn sending new, updated messages to them. S.L. Lauritzen and D.J. Spiegelhalter have presented another, elegant method for evidence propagation. They have observed that calculating the updated probability function after a piece of evidence has become available will

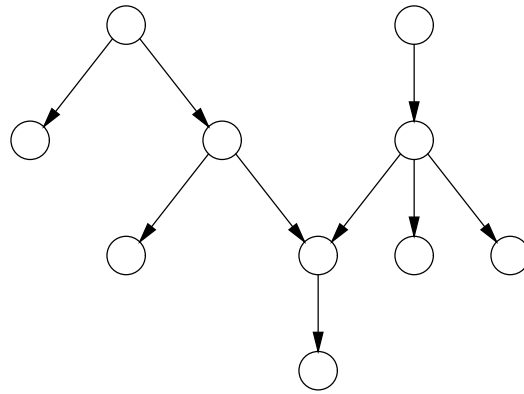


Figure 2: A causal polytree.

generally entail going against the initially assessed ‘directed’ conditional probabilities. They concluded that the directed graphical representation of a Bayesian network is not suitable as an architecture for propagating evidence directly. This observation, amongst other ones, motivated an initial transformation of the Bayesian network into an undirected graphical and probabilistic representation of the problem domain. We shall see in Section 5 where this method will be discussed in some detail, that this new representation allows for an efficient method for evidence propagation in which the computations to be performed are local to small sets of variables.

4 The reasoning method of Kim and Pearl

One of the earliest methods for reasoning with a Bayesian network was proposed by J.H. Kim and J. Pearl. Their method is defined for a restricted type of Bayesian network only. It therefore is not as general as the method of Lauritzen and Spiegelhalter which will be discussed in the following section.

The method of Kim and Pearl is applicable to Bayesian networks in which the graphical part is a so-called causal polytree. A *causal polytree* is an acyclic directed graph in which between any two vertices at most one path exists. Figure 2 shows such a causal polytree; note that the graph shown in figure 5.8 is not a causal polytree since there exist two different paths from the vertex V_3 to the vertex V_7 . For evidence propagation in their restricted type of Bayesian network, Kim and Pearl exploit the mentioned topological property of a causal polytree. Observe that from this property we have that by deleting an arbitrary arc from a causal polytree, it falls apart into two separate components. In a causal polytree G , therefore, we can identify for a vertex V_i with m neighbours, m subgraphs of G each containing a neighbour of V_i such that after removal of V_i from G there does not exist a path from one such subgraph to another one. The subgraphs corresponding with the predecessors of the vertex will be called the *upper graphs* of V_i ; the subgraphs corresponding with the successors of V_i will be called the *lower graphs* of V_i . The following example illustrates the idea. From now on, we shall restrict the discussion to this example; the reader may verify, however, that it can easily be extended to apply to more general causal polytrees.

Example 4 Figure 3 shows a part of a causal polytree G . The vertex V_0 has the four neighbours V_1 , V_2 , V_3 , and V_4 . V_0 has two predecessors and therefore two upper graphs, which are

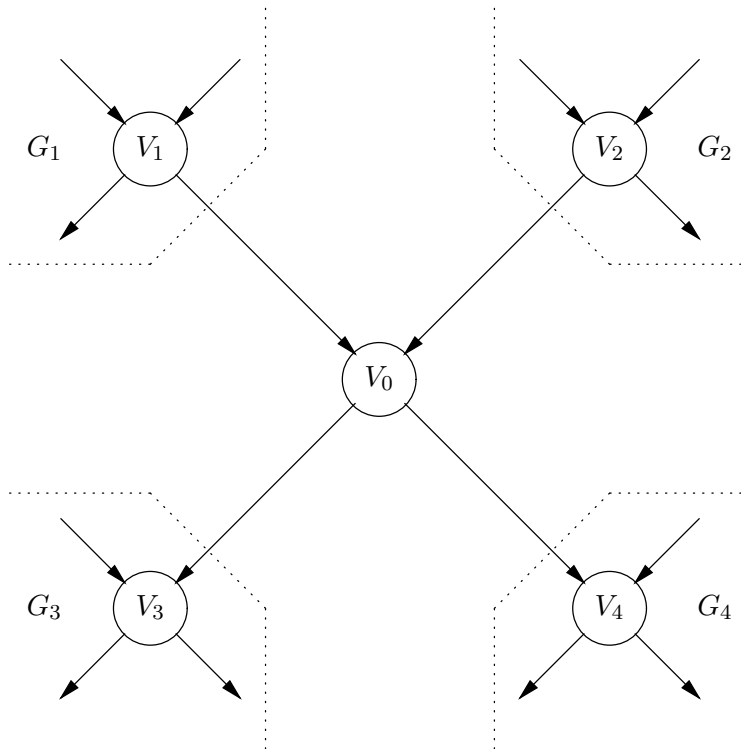


Figure 3: A part of a causal polytree.

denoted by G_1 and G_2 , respectively; V_0 has also two lower graphs, denoted by G_3 and G_4 . Note that there do not exist any paths between these subgraphs G_1 , G_2 , G_3 , and G_4 other than through V_0 . \square

So far, we have only looked at the graphical part of a Bayesian network. Recall that associated with the causal polytree we have a quantitative representation of the problem domain concerned: for each vertex, a set of local probabilities has been specified.

Let us suppose that evidence has become available that one of the statistical variables in the problem domain has adopted a specific value. This piece of evidence has to be entered into the Bayesian network in some way, and subsequently its effect on all other variables has to be computed to arrive at the updated probability function. The method for propagating evidence associated with this type of Bayesian network will be discussed shortly. First, however, we consider how probabilities of interest may be computed from the network. In doing so, we use an object-oriented style of discussion and view the causal polytree of the Bayesian network as a *computational architecture*. The vertices of the polytree are viewed as *autonomous objects* which hold some *private data* and are able to perform some computations. Recall that with each vertex is associated a set of local probabilities; these probabilities constitute the private data the object holds. The arcs of the causal polytree are taken as *communication channels*: the vertices are only able to communicate with their direct neighbours.

Now suppose that we are interested in the probabilities of the values of the variable V_0 after some evidence has been processed. It will be evident that, in terms of the graphical part of the Bayesian network, these probabilities cannot be computed from the private data

the vertex holds; they are dependent upon the information from its upper and lower graphs as well. We shall see, however, that the neighbours of V_0 are able to provide V_0 with all information necessary for computing the probabilities of its values locally.

We introduce one more notational convention. After several pieces of evidence have been entered into the network and processed, some of the statistical variables have been *instantiated* with a value and some have not. Now, consider the configuration template $C_V(G) = V_1 \wedge \dots \wedge V_n$ of the vertex set $V(G) = \{V_1, \dots, V_n\}$, $n \geq 1$, in such a situation: we have that in the template some variables have been filled in. We shall use the notation $\tilde{c}_V(G)$ to denote the instantiated part of the template. If, for example, we have the configuration template $C = V_1 \wedge V_2 \wedge V_3$ and we know that the variable V_2 has adopted the value *true* and that the variable V_3 has the value *false*, and we do not know as yet the value of V_1 , then $\tilde{c} = v_2 \wedge \neg v_3$.

We return to our example.

Example 5 Consider the causal polytree from Figure 3 once more. We are interested in the probabilities of the values of the variable V_0 . It can easily be proven, using Bayes' theorem and the independency relationships shown in the polytree, that these probabilities may be computed according to the following formula:

$$\begin{aligned} P(V_0 \mid \tilde{c}_V(G)) &= \alpha \cdot P(\tilde{c}_{V(G_3)} \mid V_0) \cdot P(\tilde{c}_{V(G_4)} \mid V_0) \\ &\quad \cdot [P(V_0 \mid v_1 \wedge v_2) \cdot P(v_1 \mid \tilde{c}_{V(G_1)}) \cdot P(v_2 \mid \tilde{c}_{V(G_2)}) \\ &\quad + P(V_0 \mid \neg v_1 \wedge v_2) \cdot P(\neg v_1 \mid \tilde{c}_{V(G_1)}) \cdot P(v_2 \mid \tilde{c}_{V(G_2)}) \\ &\quad + P(V_0 \mid v_1 \wedge \neg v_2) \cdot P(v_1 \mid \tilde{c}_{V(G_1)}) \cdot P(\neg v_2 \mid \tilde{c}_{V(G_2)}) + \\ &\quad P(V_0 \mid \neg v_1 \wedge \neg v_2) \cdot P(\neg v_1 \mid \tilde{c}_{V(G_1)}) \cdot P(\neg v_2 \mid \tilde{c}_{V(G_2)})] \end{aligned}$$

where α is normalization factor chosen so as to guarantee $P(v_0 \mid \tilde{c}_V(G)) = 1 - P(\neg v_0 \mid \tilde{c}_V(G))$. We take a closer look at this formula. Note that the probabilities $P(v_0 \mid v_1 \wedge v_2)$, $P(v_0 \mid \neg v_1 \wedge v_2)$, $P(v_0 \mid v_1 \wedge \neg v_2)$, and $P(v_0 \mid \neg v_1 \wedge \neg v_2)$ necessary for computing the updated probabilities of the values of V_0 have been associated with V_0 initially: V_0 holds these probabilities as private data. So, if V_0 were to obtain the probabilities $P(\tilde{c}_{V(G_i)} \mid v_0)$ and $P(\tilde{c}_{V(G_i)} \mid \neg v_0)$ from its successors V_i , and the probabilities $Pr(v_j \mid \tilde{c}_{V(G_j)})$ and $Pr(\neg v_j \mid \tilde{c}_{V(G_j)})$ from each of its predecessors V_j , then V_0 would be able to locally compute the probabilities of its values. \square

In the previous example we have seen that the vertex V_0 has to receive some specific probabilities from its successors and predecessors before it is able to compute locally the probabilities of its own values. The vertex V_0 has to receive from each of its successors a so-called *diagnostic evidence parameter*: the diagnostic evidence parameter that the successor V_i sends to V_0 is a function λ_{V_i} defined by $\lambda_{V_i}(v_0) = P(\tilde{c}_{V(G_i)} \mid v_0)$ and $\lambda_{V_i}(\neg v_0) = P(\tilde{c}_{V(G_i)} \mid \neg v_0)$. The vertex V_0 furthermore has to receive from each of its predecessors a *causal evidence parameter*: the causal evidence parameter that the predecessor V_j sends to V_0 is a function π_{V_0} defined by $\pi_{V_0}(v_j) = P(v_j \mid \tilde{c}_{V(G_j)})$ and $\pi_{V_0}(\neg v_j) = P(\neg v_j \mid \tilde{c}_{V(G_j)})$. These evidence parameters may be viewed as being associated with the arcs of the causal polytree; Figure 4 shows the parameters associated with the causal polytree from Figure 3. Note that the π and λ parameters may be viewed as *messages* sent between objects.

Until now we have not addressed the question how a vertex computes the evidence parameters to be sent to its neighbours. We therefore turn our attention to evidence propagation. Suppose that evidence becomes available that a certain variable $V_i \in V(G)$ has adopted a

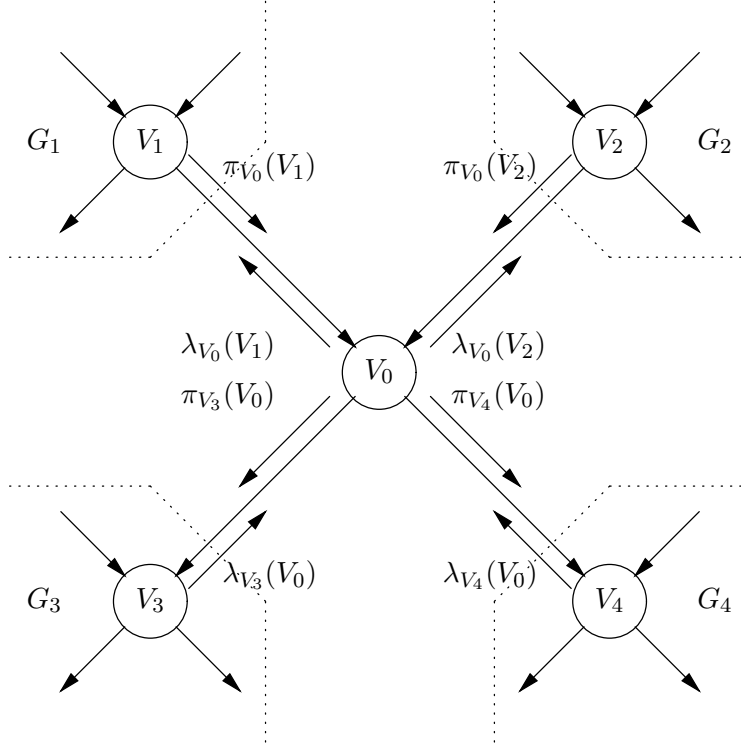


Figure 4: The π and λ parameters associated with the causal polytree.

certain value, say *true*. Informally speaking, the following happens. This evidence forces that variable V_i to update his private data: it will be evident that the updated probabilities for the values of V_i are $P(v_i) = 1$ and $P(\neg v_i) = 0$, respectively. From its local knowledge about the updated probability function, V_i then computes the proper π and λ parameters to be sent to its neighbours. V_i 's neighbours subsequently are forced to update their local knowledge about the probability function and to send new parameters to their neighbours in turn. This way evidence, once entered, is spread through the Bayesian network.

Example 6 Consider the causal polytree from Example 5.11 once more. The vertex V_0 computes the following causal evidence parameter to be sent to its successor V_3 :

$$\begin{aligned} \pi_{V_3}(V_0) = & \alpha \cdot \lambda_{V_4}(V_0) \cdot [P(V_0 | v_1 \wedge v_2) \cdot \pi_{V_0}(v_1) \cdot \pi_{V_0}(v_2) \\ & + P(V_0 | \neg v_1 \wedge v_2) \cdot \pi_{V_0}(\neg v_1) \cdot \pi_{V_0}(v_2) \\ & + P(V_0 | v_1 \wedge \neg v_2) \cdot \pi_{V_0}(v_1) \cdot \pi_{V_0}(\neg v_2) \\ & + P(V_0 | \neg v_1 \wedge \neg v_2) \cdot \pi_{V_0}(\neg v_1) \cdot \pi_{V_0}(\neg v_2)] \end{aligned}$$

where α again is a normalization factor. In computing this causal evidence parameter, V_0 uses its private data and the information it obtains from its neighbours V_1, V_2 , and V_4 . Note that, if due to some new evidence for example the information $\lambda_{V_4}(V_0)$ has changed, then this change is propagated from V_4 through V_0 to V_3 .

The vertex V_0 furthermore computes the following diagnostic evidence parameter to be sent to its predecessor V_1 :

$$\lambda_{V_0}(V_1) = \alpha \cdot \lambda_{V_3}(v_0) \cdot \lambda_{V_4}(v_0) \cdot [P(v_0 | V_1 \wedge v_2) \cdot \pi_{V_0}(v_2)$$

$$\begin{aligned}
& + P(v_0 \mid V_1 \wedge \neg v_2) \cdot \pi_{V_0}(\neg v_2)] \\
& + \alpha \cdot \lambda_{V_3}(\neg v_0) \cdot \lambda_{V_4}(\neg v_0) \cdot [P(\neg v_0 \mid V_1 \wedge v_2) \cdot \pi_{V_0}(v_2) \\
& + P(\neg v_0 \mid V_1 \wedge \neg v_2) \cdot \pi_{V_0}(\neg v_2)]
\end{aligned}$$

where α once more is a normalization factor. \square

We add to this example that the vertices V_i having no predecessors send a causal evidence parameter defined by $\pi_{V_j}(V_i) = P(V_i)$ to their successors V_j ; furthermore, the vertices V_i having no successors initially send a diagnostic evidence parameter defined by $\lambda_{V_i}(V_j) = 1$ to their successors V_j .

We now have discussed the way a piece of evidence, once entered, is propagated through the causal polytree. We observe that any change in the joint probability distribution in response to a new piece of evidence spreads through the polytree in a single pass. This statement can readily be verified by observing that any change in the causal evidence parameter π associated with a specific arc of the causal polytree does not affect the diagnostic evidence parameter λ on the same arc (and vice versa), since in computing the diagnostic evidence parameter $\lambda_{V_k}(V_0)$ associated with the arc (V_0, V_k) the causal evidence parameter $\pi_{V_k}(V_0)$ associated with the same arc is not used. So, in a causal polytree a perturbation is absorbed without reflection at the ‘boundary’ vertices, that is, vertices with either one outgoing or one incoming arc.

It remains to be discussed how a piece of evidence may be entered into the network. This is done rather elegantly: if evidence has become available that the variable V_i has the value *true* (or *false*, alternatively), then a dummy successor W of V_i is temporarily added to the polytree sending a diagnostic parameter $\lambda_W(V_i)$ to V_i such that $\lambda_W(v_i) = 1$ and $\lambda_W(\neg v)_i = 0$ (or vice versa if the value *false* has been observed).

5 The reasoning method of Lauritzen and Spiegelhalter

In the previous section we have seen that propagating a piece of evidence concerning a specific statistical variable to the other variables in the graphical part of a Bayesian network will generally involve going against the directions of the arcs. This observation, amongst other ones, motivated S.L. Lauritzen and D.J. Spiegelhalter to transform an initially assessed Bayesian network into an equivalent undirected graphical and probabilistic representation of the problem domain. Their scheme for evidence propagation is defined on this new representation. The scheme has been inspired to a large extent by the existing statistical theory of *graphical models* (probabilistic models that can be represented by an undirected graph). In this theory, the class of so-called decomposable graphs has proven to be an important subclass of graphs. Before we define the notion of a decomposable graph, we introduce several other notions.

Definition 2 Let $G = (V(G), E(G))$ be an undirected graph where $E(G)$ is a finite set of unordered pairs (V_i, V_j) , $V_i, V_j \in V(G)$, called edges. A cycle is a path of length at least one from V_0 to V_0 , $V_0 \in V(G)$. A cycle is elementary if all its vertices are distinct. A chord of an elementary cycle $V_0, V_1, \dots, V_k = V_0$ is an edge (V_i, V_j) , $i \neq (j \pm \text{mod}(k+1))$.

We now are ready to define the notion of a decomposable graph.

Definition 3 An undirected graph is decomposable if all elementary cycles of length $k \geq 4$ have a chord.

It can be shown that a probability function on such a graph may be expressed in terms of local probability functions, called *marginal* probability functions, on small sets of variables. We shall see that a representation of the problem domain in a decomposable graph and an associated representation of the probability function then allows for an efficient scheme for evidence propagation, in which the computations to be performed are local to these small sets of variables.

In order to be able to fully exploit the theory of graphical models, Lauritzen and Spiegelhalter propose a transformation of the initially assessed Bayesian network in which the graphical representation of the Bayesian network is transformed into a decomposable graph, and in which from the probabilistic part of the network a new representation of the probability function in terms of the resulting decomposable graph is obtained. The resulting representation of the problem domain is a new type of Bayesian network, which will henceforth be called a *decomposable Bayesian network*. We shall only describe the transformation of the initially assessed Bayesian network into such a decomposable Bayesian network informally.

The transformation of the original acyclic directed graph G into a decomposable graph involves three steps:

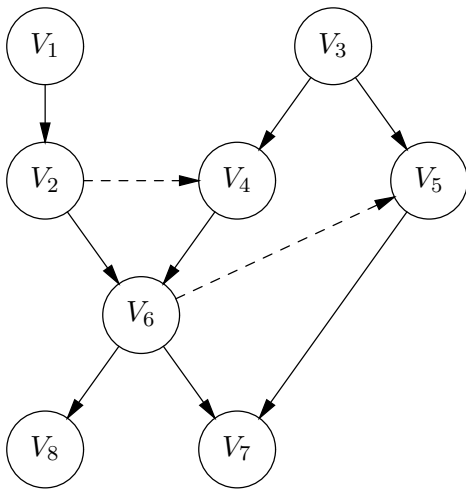
- (1) Add arcs to G in such a way that no vertex in $V(G)$ has non-adjacent predecessors.
- (2) Subsequently, drop the directions of the arcs.
- (3) Finally, cut each elementary cycle of length four or more short by adding a chord.

It will be evident that the resulting graph is decomposable. Note that the result obtained is not unique.

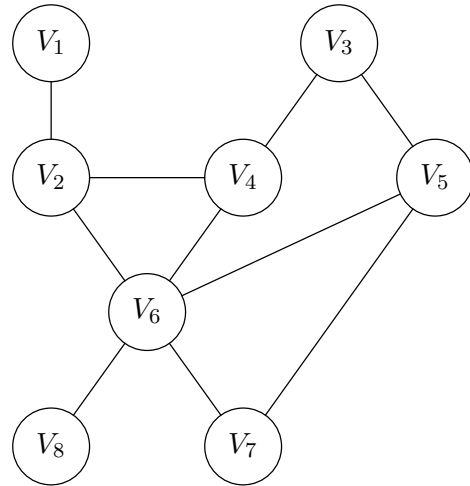
Example 7 Consider the Bayesian network from the example of Section 2 once more. The transformation of the graphical part of this Bayesian network into a decomposable graph is demonstrated in Figure 5. We consider the transformation steps in further detail. First of all, we have to add new arcs to the graph such that no vertex has non-adjacent predecessors. Now observe that in figure 5.8 the vertex V_6 has two predecessors: the vertices V_2 and V_4 . Since there does not exist an arc between V_2 and V_4 , we have that the predecessors of V_6 are nonadjacent. We therefore add an arc between V_2 and V_4 . Note that we also have to add an arc between the vertices V_5 and V_6 . Since we will drop all directions in the second transformation step, the directions of the added arcs are irrelevant. From subsequently dropping the directions of the arcs, we obtain an undirected graph. The resulting graph, however, is still not decomposable, since it has an elementary cycle of length 4 without any shortcut: V_3, V_4, V_6, V_5, V_3 . We cut this cycle short by adding an edge between the vertices V_4 and V_5 . Note that addition of an edge between V_3 and V_6 would have yielded a decomposable graph as well. \square

We now have obtained an undirected graphical representation of the problem domain. With this undirected graph, an ‘undirected’ representation of the probability function is associated. We confine ourselves to a discussion of this new representation, without describing how it is actually obtained from the initially assessed probabilities. It should however be evident that the new representation can be obtained from the original one, since the initial probabilities define a unique probability function.

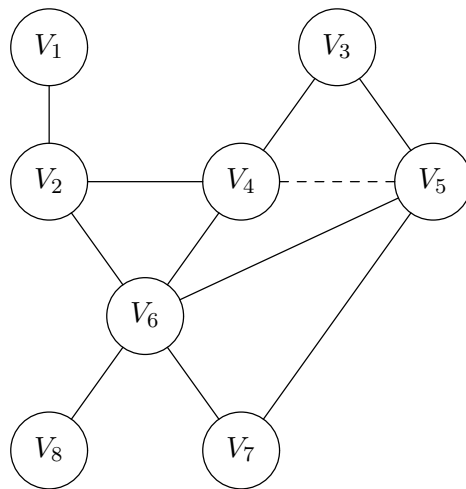
We shall see that the probability function can be expressed in terms of marginal probability functions on the cliques of the decomposable graph. We define the notion of a clique.



(a) Add arcs such that no vertex has non-adjacent predecessors



(b) Drop the directions of the arcs



(c) Cut elementary cycles short

Figure 5: Construction of the decomposable graph.

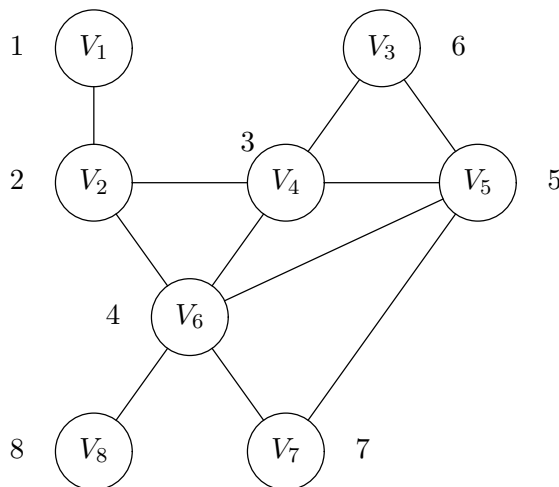


Figure 6: An ordering of the vertices obtained from maximum cardinality search.

Definition 4 Let $G = (V(G), E(G))$ be an undirected graph. A clique of G is a subgraph $H = (V(H), E(H))$ of G such that for any two distinct vertices $V_i, V_j \in V(H)$ we have that $(V_i, V_j) \in E(H)$. H is called a maximal clique of G if there does not exist a clique H' of G differing from H such that H is a subgraph of H' .

In the sequel, we shall take the word clique to mean a maximal clique.

Example 8 Consider the decomposable graph from Figure 5 once more. The reader can easily verify that this graph contains six cliques. \square

To arrive at the new representation of the probability function, we obtain an ordering of the vertices and of the cliques of the decomposable graph. Its vertices are ordered as follows:

- (1) Assign an arbitrary vertex the number 1.
- (2) Subsequently, number the remaining vertices in increasing order such that the next number is assigned to the vertex having the largest set of previously numbered neighbours.

We say that the ordering has been obtained from *maximum cardinality search*. After the vertices of the decomposable graph have been ordered, the cliques of the graph are numbered in the order of their highest numbered vertex.

Example 9 Consider the decomposable graph $G = (V(G), E(G))$ as shown in Figure 5 once more. The vertices of G are ordered using maximum cardinality search. An example of such an ordering is shown in Figure 6. The six cliques of the graph subsequently are numbered in the order of their highest numbered vertex. Let Cl_i be the clique assigned number $i, i = 1, \dots, 6$. Then, we have obtained the following ordering (for ease of exposition we identify a clique with its vertex set):

$$\begin{aligned} Cl_1 &= \{V_1, V_2\} \\ Cl_2 &= \{V_2, V_4, V_6\} \\ Cl_3 &= \{V_4, V_5, V_6\} \end{aligned}$$

$$\begin{aligned} \text{Cl}_4 &= \{V_3, V_4, V_5\} \\ \text{Cl}_5 &= \{V_5, V_6, V_7\} \\ \text{Cl}_6 &= \{V_6, V_8\} \end{aligned}$$

□

We consider the ordering $\text{Cl}_1, \dots, \text{Cl}_m$, $m \geq 1$, of the cliques of a decomposable graph G in further detail. Let $V(\text{Cl}_i)$ denote the vertex set of clique Cl_i , $i = 1, \dots, m$. The ordering now has the following important property: for all $i \geq 2$ there is a $j < i$ such that $V(\text{Cl}_i) \cap (V(\text{Cl}_1) \cup \dots \cup V(\text{Cl}_{i-1})) \subset V(\text{Cl}_j)$. In other words, the vertices a clique has in common with the lower numbered cliques are all contained in one such clique. This property is known as the *running intersection property*. This property now enables us to write the probability function on the decomposable graph as the product of the marginal probability functions on its cliques, divided by a product of the marginal probability functions on the clique intersections:

$$P(C_{V(G)}) = \prod_{i=1}^m \frac{P(C_{V(\text{Cl}_i)})}{P(C_{S_i})}$$

where S_i is the set of vertices Cl_i has in common with the lower numbered cliques.

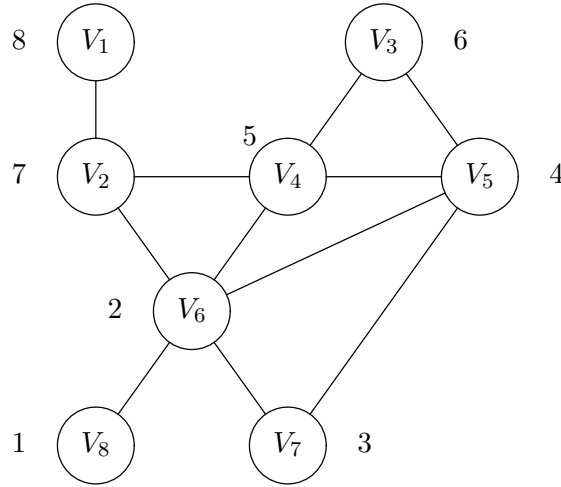
Example 10 Consider the decomposable graph G shown in Figure 6 once more. The probability function on G may be expressed as

$$\begin{aligned} P(V_1 \wedge \dots \wedge V_8) &= P(V_1 \wedge V_2) \cdot \frac{P(V_2 \wedge V_4 \wedge V_6)}{P(V_2)} \cdot \frac{P(V_4 \wedge V_5 \wedge V_6)}{P(V_4 \wedge V_6)} \\ &\quad \cdot \frac{P(V_3 \wedge V_4 \wedge V_5)}{P(V_4 \wedge V_5)} \cdot \frac{P(V_5 \wedge V_6 \wedge V_7)}{P(V_5 \wedge V_6)} \cdot \frac{P(V_6 \wedge V_8)}{P(V_6)} \end{aligned}$$

□ The initially assessed Bayesian network has now been transformed into a decomposable Bayesian network. The scheme for evidence propagation proposed by Spiegelhalter and Lauritzen operates on this decomposable Bayesian network. We emphasize that for a specific problem domain the transformation has to be performed only once: each consultation of the system proceeds from the obtained decomposable Bayesian network.

Recall that for making probabilistic statements concerning the statistical variables discerned in a problem domain we have to associate with a decomposable Bayesian network a method for computing probabilities of interest from it and a method for propagating evidence through it. As far as computing probabilities from a decomposable Bayesian network is concerned, it will be evident that any probability which involves only variables occurring in one and the same clique can simply be computed locally from the marginal probability function on that clique.

The method for evidence propagation is less straightforward. Suppose that evidence becomes available that the statistical variable V has adopted a certain value, say v . For ease of exposition, we assume that the variable V occurs in one clique of the decomposable graph only. Informally speaking, propagation of this evidence amounts to the following. The vertices and the cliques of the decomposable graph are ordered anew, this time starting with the instantiated vertex. The ordering of the cliques then is taken as the order in which the evidence is propagated through the cliques. For each subsequent clique, the updated marginal probability function is computed locally using the computation scheme shown below; we use

Figure 7: An ordering of the vertices starting with V_8 .

P to denote the initially given probability function and P^* to denote the new probability function after updating. For the first clique in the ordering we simply compute:

$$P^*(C_{V(\text{Cl}_1)}) = P(C_{V(\text{Cl}_1)} | v)$$

For the remaining cliques, we compute the updated marginal probability function using:

$$\begin{aligned} P^*(C_{V(\text{Cl}_i)}) &= P(C_{V(\text{Cl}_i)} | v) \\ &= P(C_{V(\text{Cl}_i) \setminus S_i} | C_{S_i} \wedge v) \cdot P(C_{S_i} | v) \\ &= P(C_{V(\text{Cl}_i) \setminus S_i} | C_{S_i}) \cdot P^*(C_{S_i}) \\ &= P(C_{V(\text{Cl}_i)}) \cdot \frac{P^*(C_{S_i})}{P(C_{S_i})} \end{aligned}$$

where S_i once more is the set of vertices Cl_i has in common with the lower numbered cliques. So, an updated marginal probability function is obtained by multiplying the ‘old’ marginal probability function with the quotient of the ‘new’ and the ‘old’ marginal probability function on the appropriate clique-intersection.

We look once more at our example.

Example 11 Consider the decomposable graph from Figure 5 and its associated probability function once more. Suppose that we obtain the evidence that the variable V_8 has the value *true*. Using maximum cardinality search, we renumber the vertices of the graph starting with the vertex V_8 . Figure 7 shows an example of such an ordering. From this new ordering of the vertices we obtain an ordering of the six cliques of the graph (once more, we identify a clique with its vertex set):

$$\begin{aligned} \text{Cl}_1 &= \{V_6, V_8\} \\ \text{Cl}_2 &= \{V_5, V_6, V_7\} \\ \text{Cl}_3 &= \{V_4, V_5, V_6\} \\ \text{Cl}_4 &= \{V_3, V_4, V_5\} \\ \text{Cl}_5 &= \{V_2, V_4, V_6\} \\ \text{Cl}_6 &= \{V_1, V_2\} \end{aligned}$$

The impact of the evidence on the first clique is

$$P^*(V_6) = P(V_6 \mid v_8)$$

For the second clique we find:

$$P^*(V_5 \wedge V_6 \wedge V_7) = P(V_5 \wedge V_6 \wedge V_7) \cdot \frac{P(V_6)}{P(V_6)}$$

For the remaining cliques we obtain similar results. \square

After the marginal probability functions have been updated locally, the instantiated vertex is removed from the graph, and the updated marginal probability functions are taken as the marginal probability functions on the cliques of the remaining graph. The process may now simply be repeated for a new piece of evidence.

Exercises

1. Metastatic cancer is a possible cause of a brain tumor, and is also an explanation for increased total serum calcium. In turn, either of these could explain a patient falling into a coma. Severe headache is also possibly associated with a brain tumour.

Suppose that we use a Bayesian network to represent this information. Give the graphical part of the Bayesian network. Which probabilities have been associated with the graph?

2. Consider the causal polytree from Figure 2 and an associated set of probabilities. Suppose that we apply the method of J.H. Kim and J. Pearl for evidence propagation. Try to find out how evidence spreads through the network if entered in one of the vertices.
3. Consider the Bayesian network obtained in Exercise 10 once more. We transform this Bayesian network into a decomposable Bayesian network as described in Section 5.
 - (a) Give the resulting decomposable graph. Which cliques do you discern?
 - (b) Give the new representation of the originally given probability function.
 - (c) What happens if we obtain the evidence that a specific patient is suffering from severe headaches?