INTRODUCTION TO INFERENCE FOR BAYESIAN NETWORKS

ROBERT COWELL
City University, London.
The School of Mathematics, Actuarial Science and Statistics, City University, Northampton Square, London EC1E 0HT

1. Introduction

The field of Bayesian networks, and graphical models in general, has grown enormously over the last few years, with theoretical and computational developments in many areas. As a consequence there is now a fairly large set of theoretical concepts and results for newcomers to the field to learn. This tutorial aims to give an overview of some of these topics, which hopefully will provide such newcomers a conceptual framework for following the more detailed and advanced work. It begins with revision of some of the basic axioms of probability theory.

2. Basic axioms of probability

Probability theory, also known as inductive logic, is a system of reasoning under uncertainty, that is under the absence of certainty. Within the Bayesian framework, probability is interpreted as a numerical measure of the degree of consistent belief in a proposition, consistency being with the data at hand.

Early expert systems used deductive, or Boolean, logic, encapsulated by sets of production rules. Attempts were made to cope with uncertainty using probability theory, but the calculations became prohibitive, and the use of probability theory for inference in expert systems was abandoned. It is with the recent development of efficient computational algorithms that probability theory has had a revival within the AI community.

Let us begin with some basic axioms of probability theory. The probability of an event A, denoted by $P(A)$, is a number in the interval [0,1], which obeys the following axioms:

1 $P(A) = 1$ if and only if A is certain.

2 If A and B are mutually exclusive, then $P(A \text{ or } B) = P(A) + P(B)$.

We will be dealing exclusively with discrete random variables and their probability distributions. Capital letters will denote a variable, or perhaps a set of variables, lower case letter will denote values of variables. Thus suppose A is a random variable having a finite number of *mutually exclusive states* $(a_1, \ldots, a_n)$. Then $P(A)$ will be represented by a vector of non-negative real numbers $P(A) = (x_1, \ldots, x_n)$ where $P(A = a_i) = x_i$ is a scalar, and $\sum_i x_i = 1$.

A basic concept is that of conditional probability, a statement of which takes the form: *Given the event $B = b$ the probability of the event $A = a$ is x, written $P(A = a \mid B = b) = x$.* It is important to understand that this is not saying: "If $B = b$ is true then the probability of $A = a$ is x". Instead it says: "If $B = b$ is true, and any other information to hand is irrelevant to A, then $P(A = a) = x$". (To see this, consider what the probabilities would be if the state of A was part of the extra information).

Conditional probabilities are important for building Bayesian networks, as we shall see. But Bayesian networks are also built to facilitate the calculation of conditional probabilities, namely the conditional probabilities for variables of interest given the data (also called evidence) at hand.

The fundamental rule for probability calculus is the product rule[1]

$$P(A \text{ and } B) = P(A \mid B)P(B). \tag{1}$$

This equation tells us how to combine conditional probabilities for individual variables to define joint probabilities for sets of variables.

3. Bayes' theorem

The simplest form of Bayes' theorem relates the joint probability $P(A \text{ and } B)$ – written as $P(A, B)$ – of two events or hypotheses A and B in terms of marginal and conditional probabilities:

$$P(A, B) = P(A \mid B)P(B) = P(B \mid A)P(A). \tag{2}$$

By rearrangement we easily obtain

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}, \tag{3}$$

which is Bayes' theorem.

This can be interpreted as follows. We are interested in A, and we begin with a *prior* probability $P(A)$ for our belief about A, and then we observe

[1] Or more generally $P(A \text{ and } B \mid C) \equiv P(A \mid B, C)P(B \mid C)$.

B. Then Bayes' theorem, (3), tells us that our revised belief for A, the *posterior* probability $P(A \mid B)$ is obtained by multiplying the prior $P(A)$ by the ratio $P(B \mid A)/P(B)$. The quantity $P(B \mid A)$, as a function of varying A for fixed B, is called the *likelihood* of A. We can express this relationship in the form:

$$\begin{array}{rcl} \textbf{posterior} & \propto & \textbf{prior} \times \textbf{likelihood} \\ P(A \mid B) & \propto & P(A)P(B \mid A). \end{array}$$

Figure 1 illustrates this prior-to-posterior inference process. Each diagram

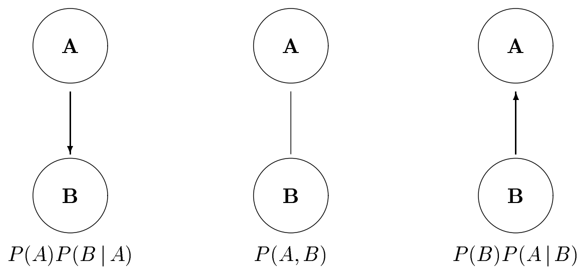


Figure 1. Bayesian inference as reversing the arrows

represents in different ways the joint distribution $P(A, B)$, the first represents the prior beliefs while the third represents the posterior beliefs. Often, we will think of A as a possible "cause" of the "effect" B, the downward arrow represents such a causal interpretation. The "inferential" upwards arrow then represents an "argument against the causal flow", from the observed effect to the inferred cause. (We will not go into a definition of "causality" here.)

Bayesian networks are generally more complicated than the ones in Figure 1, but the general principles are the same in the following sense. A Bayesian network provides a model representation for the joint distribution of a set of variables in terms of conditional and prior probabilities, in which the orientations of the arrows represent influence, usually though not always of a causal nature, such that these conditional probabilities for these particular orientations are relatively straightforward to specify (from data or eliciting from an expert). When data are observed, then typically an inference procedure is required. This involves calculating marginal probabilities conditional on the observed data using Bayes' theorem, which is diagrammatically equivalent to reversing one or more of the Bayesian network arrows. The algorithms which have been developed in recent years allows these calculations to be performed in an efficient and straightforward manner.

4. Simple inference problems

Let us now consider some simple examples of inference. The first is simply Bayes' theorem with evidence included on a simple two node network; the remaining examples treat a simple three node problem.

4.1. PROBLEM I

Suppose we have the simple model $X \to Y$, and are given: $P(X)$, $P(Y \mid X)$ and $Y = y$. The problem is to calculate $P(X \mid Y = y)$.

Now from $P(X)$, $P(Y \mid X)$ we can calculate the marginal distribution $P(Y)$ and hence $P(Y = y)$. Applying Bayes' theorem we obtain

$$P(X \mid Y = y) = \frac{P(Y = y \mid X)P(X)}{P(Y = y)}. \tag{4}$$

4.2. PROBLEM II

Suppose now we have a more complicated model in which X is a parent of both Y and Z: $Z \leftarrow X \to Y$ with specified probabilities $P(X)$, $P(Y \mid X)$ and $P(Z \mid X)$, and we observe $Y = y$. The problem is to calculate $P(Z \mid Y = y)$. Note that the joint distribution is given by $P(X, Y, Z) = P(Y \mid X)P(Z \mid X)P(X)$. A 'brute force' method is to calculate:

1. The joint distribution $P(X, Y, Z)$.
2. The marginal distribution $P(Y)$ and thence $P(Y = y)$.
3. The marginal distribution $P(Z, Y)$ and thence $P(Z, Y = y)$.
4. $P(Z \mid Y = y) = P(Z, Y = y)/P(Y = y)$.

An alternative method is to exploit the given factorization:

1. Calculate $P(X \mid Y = y) = P(Y = y \mid X)P(X)/P(Y = y)$ using Bayes' theorem, where $P(Y = y) = \sum_X P(Y = y \mid X)P(X)$.
2. Find $P(Z \mid Y = y) = \sum_X P(Z \mid X)P(X \mid Y = y)$.

Note that the first step essentially reverses the arrow between X and Y. Although the two methods give the same answer, the second is generally more efficient. For example, suppose that all three variables have 10 states. Then the first method in explicitly calculating $P(X, Y, Z)$ requires a table of 1000 states. In contrast the largest table required for the second method has size 100. This gain in computational efficiency by exploiting the given factorizations is the basis of the arc-reversal method for solving influence diagrams, and of the junction-tree propagation algorithms. The following example shows the same calculation using propagation on a junction tree.

4.3. PROBLEM. III

Suppose now that we are given the *undirected* structure $ZX - X - XY$, and probabilities $P(Z, X)$, $P(X)$ and $P(Y, X)$. Again the problem is to calculate $P(Z \mid Y = y)$. Note that:

$$\begin{aligned} P(Z, X) &= P(Z \mid X) P(X) \\ P(Y, X) &= P(Y \mid X) P(X) \\ P(X, Y, Z) &= P(Z, X) P(Y, X) / P(X). \end{aligned}$$

The calculational steps now proceeds using a 'message' in step 1 which is 'sent' in step 2:

1. Calculate $\bar{P}(X) \equiv \sum_Y P(X, Y = y)$.
2. Find $\bar{P}(Z, X) \equiv P(Z, X) \bar{P}(X) / P(X)$.
3. Find $P(Z, Y = y) = \sum_X \bar{P}(Z, X)$.
4. Find $P(Z \mid Y = y) = P(Z, Y = y) / \sum_Z P(Z, Y = y)$

5. Conditional independence

In the last example we had that

$$P(X, Y, Z) = P(Y \mid X) P(Z \mid X) P(X),$$

from which we get

$$\begin{aligned} P(Y \mid Z, X) &= \frac{P(X, Y, Z)}{P(Z, X)} \\ &= \frac{P(Y \mid X) P(Z \mid X) P(X)}{P(Z, X)} \\ &= P(Y \mid X) \end{aligned}$$

and likewise for $P(Z \mid Y, X) = P(Z \mid X)$. Hence given $X = x$ say, we obtain $P(Y \mid Z, X = x) = P(Y \mid X = x)$ and $P(Z \mid Y, X = x) = P(Z \mid X = x)$. This is an example of conditional independence (Dawid(1979)). We associated the graph $Z \leftarrow X \rightarrow Y$ with this distribution, though this is not unique. In fact the joint probability can be factorized according to three distinct directed graphs:

$Z \leftarrow X \rightarrow Y : P(X, Y, Z) = P(X) P(Y \mid X) P(Z \mid X)$.
$Z \rightarrow X \rightarrow Y : P(X, Y, Z) = P(Y \mid X) P(X \mid Z) P(Z)$.
$Z \leftarrow X \leftarrow Y : P(X, Y, Z) = P(X \mid Y) P(Z \mid X) P(Y)$.

Each of these factorizations follows from the conditional independence properties which each graph expresses, viz $Z \perp\!\!\!\perp Y \mid X$, (which is to be read as "Z is conditionally independent of Y given X") and by using the general factorization property:

$$\begin{aligned} P(X_1, \dots X_n) &= P(X_1 \mid X_2, \dots, X_n) P(X_2, \dots, X_n) \\ &= P(X_1 \mid X_2, \dots, X_n) P(X_2 \mid X_3, \dots, X_n) P(X_3, \dots, X_n) \\ &= \vdots \\ &= P(X_1 \mid X_2, \dots, X_n) \dots P(X_{n-1} \mid X_n) P(X_n). \end{aligned}$$

Thus for the third example

$$P(X, Y, Z) = P(Z \mid X, Y) P(X \mid Y) P(Y) = P(Z \mid X) P(X \mid Y) P(Y).$$

Note that the graph $Z \rightarrow X \leftarrow Y$ does not obey the conditional independence property $Z \perp\!\!\!\perp Y \mid X$ and is thus excluded from the list; it factorizes as $P(X, Y, Z) = P(X \mid Y, Z) P(Z) P(Y)$.

This example shows several features of general Bayesian networks. Firstly, the use of the conditional independence properties can be used to simplify the general factorization formula for the joint probability. Secondly, that the result is a factorization that can be expressed by the use of directed acyclic graphs (DAGs).

6. General specification in DAGs

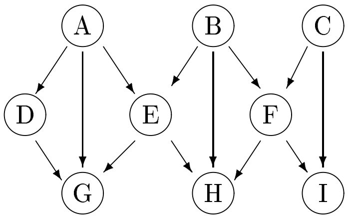
It is these features which work together nicely for the general specification of Bayesian networks. Thus a Bayesian network is a directed acyclic graph, whose structure defines a set of conditional independence properties. These properties can be found using graphical manipulations, eg *d-separation* (see eg Pearl(1988)). To each node is associated a conditional probability distribution, conditioning being on the parents of the node: $P(X \mid \mathrm{pa}(X))$. The joint density over the set of all variables U is then given by the product of such terms over all nodes:

$$P(U) = \prod_X P(X \mid \mathrm{pa}(X)).$$

This is called a *recursive factorization* according to the DAG; we also talk of the distribution being graphical over the DAG. This factorization is equivalent to the general factorization but takes into account the conditional independence properties of the DAG in simplifying individual terms in the product of the general factorization. Only if the DAG is complete will this formula and the general factorization coincide, (but even then only for one ordering of the random variables in the factorization).

6.1. EXAMPLE

Consider the graph of Figure 2.



$$\begin{aligned}
&P(A,B,C,D,E,F,G,H,I)\\
&=P(A)P(B)P(C)\\
&\quad P(D \mid A)P(E \mid A,B)P(F \mid B,C)\\
&\quad P(G \mid A,D,E)P(H \mid B,E,F)P(I \mid C,F).
\end{aligned}$$

Figure 2. Nine node example.

It is useful to note that marginalising over a childless node is equivalent to simply removing it and any edges to it from its parents. Thus for example, marginalising over the variable H in the above gives:

$$\begin{aligned}
P(A,B,C,D,E,F,G,I) &= \sum_H P(A,B,C,D,E,F,G,H,I)\\
&= \sum_H P(A)P(B)P(C)P(D \mid A)P(E \mid A,B)P(F \mid B,C)\\
&\qquad P(G \mid A,D,E)P(H \mid B,E,F)P(I \mid C,F)\\
&= P(A)P(B)P(C)P(D \mid A)P(E \mid A,B)P(F \mid B,C)\\
&\qquad P(G \mid A,D,E)P(I \mid C,F)\sum_H P(H \mid B,E,F)\\
&= P(A)P(B)P(C)P(D \mid A)P(E \mid A,B)P(F \mid B,C)\\
&\qquad P(G \mid A,D,E)P(I \mid C,F),
\end{aligned}$$

which can be represented by Figure 2 with H and its incident edges removed.

Directed acyclic graphs can always have their nodes linearly ordered so that for each node X all of its parents $\mathrm{pa}(X)$ precedes it in the ordering. Such and ordering is called a *topological ordering* of the nodes. Thus for example (A,B,C,D,E,F,G,H,I) and (B,A,E,D,G,C,F,I,H) are two of the many topological orderings of the nodes of Figure 2.

A simple algorithm to find a topological ordering is as follows: Start with the graph and an empty list. Then successively delete from the graph any node which does not have any parents, and add it to the end of the list. Note that if the graph is not acyclic, then at some stage a graph will be obtained in which no node has no parent nodes, hence this algorithm can be used as an efficient way of checking that the graph is acyclic.

Another equivalent way is to start with the graph and an empty list, and successively delete nodes which have no children and add them to the beginning of the list (cf. marginalisation of childless nodes.)

6.2. DIRECTED MARKOV PROPERTY

An important property is the *directed Markov property*. This is a conditional independence property which states that a variable is conditionally independent of its non-descendents given it parents:

$$X \perp\!\!\!\perp \mathrm{nd}(X) \mid \mathrm{pa}(X).$$

Now recall that the conditional probability $P(X \mid \mathrm{pa}(X))$ did not necessarily mean that if $\mathrm{pa}(X) = \pi^*$ say, then $P(X = x) = P(x \mid \pi^*)$, but included the caveat that any other information is irrelevant to X for this to hold. For the DAGs this 'other information' means, from the directed Markov property, knowledge about the node itself or any of its descendents. For if all of the parents of X are observed, but additionally observed are one or more descendents D_X of X, then because X influences D_X, knowing D_X and $\mathrm{pa}(X)$ is more informative than simply knowing about $\mathrm{pa}(X)$ alone. However having information about a non-descendent does not tell us anything more about X, because either it cannot influence or be influenced by X either directly or indirectly, or if it can influence X indirectly, then only through influencing the parents which are all known anyway.

For example, consider again Figure 2. Using the previous second topological ordering we may write the general factorization as:

$$\begin{aligned} P(A,B,C,D,E,F,G,I,H) =& P(B) \\ &* P(A \mid B) \\ &* P(E \mid B,A) \\ &* P(D \mid B,A,E) \\ &* P(G \mid B,A,E,D) \\ &* P(C \mid B,A,E,D,G) \\ &* P(F \mid B,A,E,D,G,C) \\ &* P(I \mid B,A,E,D,G,C,F) \\ &* P(H \mid B,A,E,D,G,C,F,I) \end{aligned} \tag{5}$$

but now we can use $A \perp\!\!\!\perp B$ from the directed Markov property to simplify $P(A \mid B) \to P(A)$, and similarly for the other factors in (5) etc, to obtain the factorization in Figure 2. We can write the general pseudo-algorithm of what we have just done for this example as

Topological ordering +
General factorization +
Directed Markov property
$\Longrightarrow$ Recursive factorization.

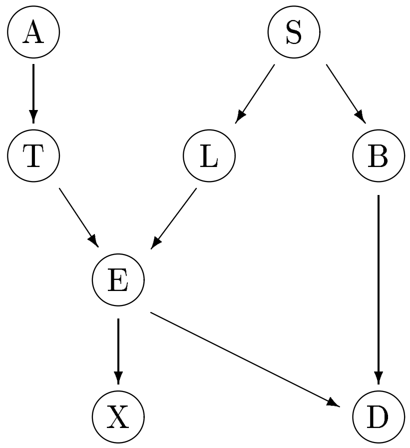
7. Making the inference engine

We shall now move on to building the so called "inference engine" to introduce new concepts and to show how they relate to the conditional independence/recursive factorization ideas that have already been touched upon. Detailed justification of the results will be omitted, the aim here is to give an overview, using the use the fictional ASIA example of Lauritzen and Spiegelhalter.

7.1. ASIA: SPECIFICATION

Lauritzen and Spiegelhalter describe their fictional problem domain as follows:

> Shortness-of-breath (**D**yspnoea) may be due to **T**uberculosis, **L**ung cancer or **B**ronchitis, or none of them, or more than one of them. A recent visit to **A**sia increases the chances of **T**uberculosis, while **S**moking is known to be a risk factor for both **L**ung cancer and **B**ronchitis. The results of a single **X**-ray do not discriminate between **L**ung cancer and **T**uberculosis, as neither does the presence or absence of **D**yspnoea.



$$\begin{aligned} P(U) = & P(A)P(S) \\ & P(T \mid A)P(L \mid S) \\ & P(B \mid S)P(E \mid L,T) \\ & P(D \mid B,E)P(X \mid E) \end{aligned}$$

Figure 3. ASIA

The network for this fictional example is shown in Figure 3. Each variable is a binary with the states ("yes", "no"). The **E** node is a logical node taking value "yes" if either of its parents take a "yes" value, and "no" otherwise; its introduction facilitates modelling the relationship of **X**-ray to **L**ung cancer and **T**uberculosis.

Having specified the relevant variables, and defined their dependence with the graph, we must now assign (conditional) probabilities to the nodes. In real life examples such probabilities may be elicited either from some large database (if one is available) as frequency ratios, or subjectively from the expert from whom the structure has been elicited (eg using a fictitious gambling scenario or probability wheel), or a combination of both. However as this is a fictional example we can follow the third route and use made-up values. (Specific values will be omitted here.)

7.2. CONSTRUCTING THE INFERENCE ENGINE

With our specified graphical model we have a representation of the joint density in terms of a factorization:

$$P(U) = \prod_V P(V \mid \mathrm{pa}(V)) \tag{6}$$

$$= P(A) \dots P(X \mid E). \tag{7}$$

Recall that our motivation is to use the model specified by the joint distribution to calculate marginal distributions conditional on some observation of one or more variables. In general the full distribution will be computationally difficult to use directly to calculate these marginals directly. We will now proceed to outline the various stages that are performed to find a representation of $P(U)$ which makes the calculations more tractable. (The process of constructing the inference engine from the model specification is sometimes called *compiling* the model.)

The manipulations required are almost all graphical. There are five stages in the graphical manipulations. Let us first list them, and then go back and define new terms which are introduced.

1. Add undirected edges to all co-parents which are not currently joined (a process called *marrying parents*).
2. Drop all directions in the graph obtained from Stage 1. The result is the so-called *moral graph*.
3. *Triangulate the moral graph*, that is, add sufficient additional undirected links between nodes such that there are no cycles (ie. closed paths) of length 4 or more distinct nodes without a short-cut.
4. Identify the *cliques* of this triangulated graph.
5. Join the cliques together to form the *junction tree*.

Now let us go through these steps, supplying some justification and defining the new terms just introduced as we go along. Consider first the joint density again. By a change of notation this can be written in the form

$$\begin{aligned} P(U) &= \prod_V a(V, \mathrm{pa}(V)) & (8) \\ &= a(A) \ldots a(X, E). & (9) \end{aligned}$$

where $a(X, \mathrm{pa}(X)) \equiv P(V \mid \mathrm{pa}(V))$. That is, the conditional probability factors for V can be considered as a function of V and its parents. We call such functions *potentials*. Now after steps 1 and 2 we have an undirected graph, in which for each node both it and its set of parents in the original graph form a complete subgraph in the moral graph. (A complete graph is one in which every pair of nodes is joined together by an edge.) Hence, the original factorization of $P(U)$ on the DAG G goes over to an equivalent factorization on these complete subsets in the moral graph G^m. Technically we say that the distribution is graphical on the undirected graph G^m. Figure 4 illustrates the moralisation process for the Asia network. Now let us de-

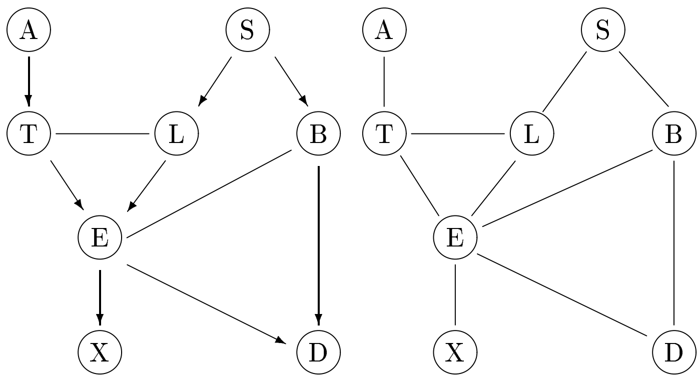


Figure 4. Moralising Asia: Two extra links are required, $A - S$ and $L - B$. Directionality is dropped after all moral edges have been added.

note the set of cliques of the moral graph by C^m. (A clique is a complete subgraph which is not itself a proper subgraph of a complete subgraph, so it is a maximal complete subgraph.) Then each of the complete subgraphs formed from $\{V\} \cup \mathrm{pa}(V)$ is contained within at least one clique. Hence we can form functions a_C such that

$$P(U) = \prod_{c \in C^m} a_C(V_C)$$

where $a_C(V_C)$ is a function of the variables in the clique C. Such a factorization can be constructed as follows: Initially define each factor as unity, i.e.,

$a_C(V_C) = 1$ for all cliques in C^m. Then for each factor $P(V \mid \mathrm{pa}(V))$ find one and only one clique which contains the complete subgraph of $\{V\} \cup \mathrm{pa}(V)$ and multiply this conditional distribution into the function of that clique to obtain a new function. When this is done the result is a potential representation of the joint distributions in terms of functions on the cliques of the moral G^m.

Note that by adding the extra edges in the moralisation process it is not possible to read of all of the conditional independences of the original DAG, though they are still there "buried" in the numerical specification. Those which remain "visible" in the moral graph are used to exploit the efficient local computations which will be described later.

8. Aside: Markov properties on ancestral sets

In fact the moral graph is a powerful construction for elicudating conditional independence. First we require some more definitions. A node A is an ancestor of a node B if either (i) A is a parent of B or (ii) A is an ancestor of (at least) one of the parents of B. The ancestral set of a node is the node itself and the set of its ancestors. The ancestral set of a set of nodes Y is the union of the ancestral sets of the nodes in Y. A set S *separates* the sets A and B if every path between a node $a \in A$ and $b \in B$ passes through some node of S. With these definitions we have:

Lemma 1

Let P factorize recursively according to $\mathcal{G}$. Then

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

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

Lemma 2

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

What these lemmas tell us is that if we want to check conditional independences we can either look at d-separation properties or the smallest ancestral sets of the moral graphs – they are alternative ways of calculation.

To understand why ancestral set come into the picture, let us consider the following simple algorithm for finding them. Suppose that we have the graph G and that we wish to find the ancestral set of a set of nodes $Y \subseteq U$. Then successively delete nodes from G which have no children, provided they are not in the set Y. When it is not possible any longer delete any nodes, the subgraph left is the minimal ancestral set.

Now recall that deleting a childless node is equivalent to marginalising over that node. Hence the marginal distribution of the minimal ancestral set containing $A \perp\!\!\!\perp B \mid S$ factorizes according to the sub-factors of the original joint distribution. So these lemmas are saying that rather than go through the numerical exercise of actually calculating such marginals we can read it off from the graphical structure instead, and use that to test conditional independences. (Note also that the directed Markov property is also lurking behind the scenes here.) The "moral" is that when ancestral sets appear in theorems like this it is likely that such marginals are being considered.

9. Making the junction tree

The remaining three steps of the inference-engine construction algorithm seem more mysterious, but are required to ensure we can formulate a consistent and efficient message passing scheme. Consider first step 3 – adding edges to the moral graph G^m to form a triangulated graph G^t. Note that adding edges to the graph does not stop a clique of the moral graph formed from being a complete subgraph in G^t. Thus for each clique in C^m of the moral graph there is at least one clique in the triangulated graph which contains it. Hence we can form a potential representation of the joint probability in terms of products of functions of the cliques in the triangulated graph:

$$P(U) = \prod_{c \in C^t} a_C(X_C)$$

by analogy with the previous method outline for the moral graph. The point is that after moralisation and triangulation there exists for each a node-parent set at least one clique which contains it, and thus a potential representation can be formed on the cliques of the triangulated graph.

While the moralisation of a graph is unique, there are in general many alternative triangulations of a moral graph. In the extreme, we can always add edges to make the moral graph complete. There is then one large clique. The key to the success of the computational algorithms is to form triangulated graphs which have small cliques, in terms of their state space size.

Thus after finding the cliques of the triangulated graph – stage 4 – we are left with joining them up to form a junction tree. The important property of the junction tree is the *running intersection property* which means that if variable V is contained in two cliques, then it is contained in every clique along the path connecting those two cliques. The edge joining two cliques is called a separator. This joining up property can always be done, not necessarily uniquely for each triangulated graph. However the choice of tree is immaterial except for computational efficiency considerations. The junction tree captures many, but not necessarily all, of the conditional independence properties of the distribution on the original DAG. It loses some of the conditional independences by the process of adding extra edges to the moral graph. However it does retain conditional independence between (not necessarily neighbouring) cliques given separators between them. It is because of this fact that local computation with message passing becomes possible. The running intersection property ensures *consistence* in the message passing between cliques, and the cliques become the basic unit of the local computation, ie., they define the granularity of the computational algorithms. If the cliques are of manageable size then local computation is possible. Figure 5 shows a triangulated version of Asia and a possible junction tree.

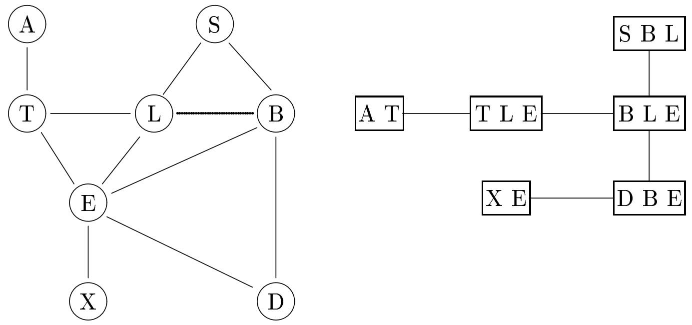


Figure 5. Junction tree for asia

10. Inference on the junction tree

We will summarise some of the basic results of message passing on the junction tree. We have seen that we can form a potential representation of the joint probability using functions defined on the cliques:

$$P(U) = \prod_{C \in C^t} a_C(X_C).$$

This can be generalized to include functions on the separators (the intersections of neighbouring cliques) to form the following so called *generalized potential representation*:

$$P(U) = \frac{\prod_{C \in C^t} a_C(X_C)}{\prod_{S \in S^t} b_S(X_S)}.$$

(for instance by making the separator functions the identity). Now, by sending messages between neighbouring cliques consisting of functions of the separator variables only, which modify the intervening separator and the clique receiving the message, but in such a way that the overall ratio of products remains invariant, we can arrive at the following *marginal representation*:

$$p(U) = \frac{\prod_{C \in \mathcal{C}} p(C)}{\prod_{S \in \mathcal{S}} p(S)}. \tag{10}$$

Marginals for individual variables can be obtained from these clique (or separator) marginals by further marginalisation.

Suppose that we observe "evidence" $\mathcal{E} : X_A = x_A^*$. Define a new function P^* by

$$P^*(x) = \begin{cases} P(x) & \text{if } X_A = x_A^* \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Then $P^*(U) = P(U, \mathcal{E}) = P(\mathcal{E})P(U \mid \mathcal{E})$. We can rewrite (11) as

$$P^*(U) = P(U) \prod_{v \in A} l(v), \tag{12}$$

where $l(v)$ is 1 if $x_v = x_v^*$, 0 otherwise. Thus $l(v)$ is the *likelihood function* based on the partial evidence $X_v = x_v^*$. Clearly this also factorizes on the junction tree, and by message passing we may obtain the following *clique-marginal representation*

$$p(V \mid \mathcal{E}) = \frac{\prod_{C \in \mathcal{C}} p(C \mid \mathcal{E})}{\prod_{S \in \mathcal{S}} p(S \mid \mathcal{E})}. \tag{13}$$

or by omitting the normalization stage,

$$p(V, \mathcal{E}) = \frac{\prod_{C \in \mathcal{C}} p(C, \mathcal{E})}{\prod_{S \in \mathcal{S}} p(S, \mathcal{E})}. \tag{14}$$

Again marginal distributions for individual variables, conditional upon the evidence, can be obtained by further marginalisation of individual clique tables, as can the probability (according to the model) of the evidence, $P(\mathcal{E})$.

11. Why the junction tree?

Given that the moral graph has nice properties, why is it necessary to go on to form the junction tree? This is best illustrated by an example, Figure 6:

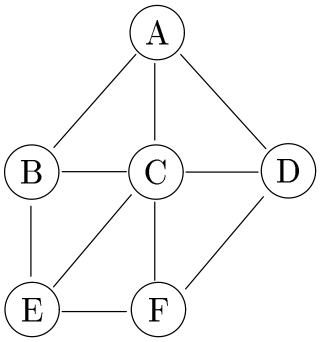


Figure 6. A non-triangulated graph

The cliques are (A, B, C), (A, C, D), (C, D, F), (C, E, F) and (B, C, E) with successive intersections (A, C), (C, D), (C, F), (C, E) and (B, C). Suppose we have clique marginals $P(A, B, C)$ *etc.*. We cannot express $P(A, B, C, D)$ in terms of $P(A, B, C)$ and $P(A, C, D)$ – the graphical structure does *not* imply $B \perp\!\!\!\perp D|(A, C)$. In general there is no closed for expression for the joint distribution of all six variables in terms of its cliques marginals.

12. Those extra edges again

Having explained why the cliques of the moral graph are generally not up to being used for local message passing, we will now close by indicating where the extra edges to form a triangulated graph come from.

Our basic message passing algorithm will be one in which marginals of the potentials in the cliques will form the messages on the junction tree. So let us begin with our moral graph with a potential representation in terms of functions on the cliques, and suppose we marginalise a variable Y say,which belongs to more than one clique of the graph, say two cliques, C_1 and C_2, with variables $Y \cup Z_1$ and $Y \cup Z_2$ respectively. They are cliques, but the combined set of variables do not form a single clique, hence there must be at least one pair of variables, one in each clique, which are not joined to each other, u_1 and u_2 say.

Now consider the effect of marginalisation of the variable Y. We will have

$$\sum_Y a_{C_1}(Y \cup Z_1) a_{C_2}(Y \cup Z_2) \equiv f(Z_1 \cup Z_2),$$

a function of the combined variables of the two cliques minus Y. Now this function cannot be accommodated by a clique in the moral graph because the variables u_1 and u_2 are not joined (and there may be others).

Hence we cannot form a potential representation of the joint distribution of $P(U - Y)$ on the moral graph with node Y removed. However, if we fill in the missing edges between the pairs of variables of the two cliques, then this marginal can be accommodated, and we can find a potential representation for $P(U - Y)$ on the reduced moral graph having these extra edges. This is why one adds edges to the moral graph, to be able to accommodate such intermediate marginal expressions. It turns out that one must fill-in sufficiently to form a triangulated graph, and doing so results in being able to set up a consistent message passing scheme.

13. Suggested further reading

Pearl is one of the pioneers who helped Bayesian methods for uncertain reasoning become popular in the artificial intelligence community. His textbook (Pearl, 1988) contains a wealth of material, from introducing probability theory and arguments for its use; axiomatics for graphical models; Markov properties; etc, to propagation in singly connected DAGs (ie prior to the development of making junction trees and propagating with them.) A good collection of papers on uncertain reasoning is Shafer and Pearl(1990), which covers not only probabilistic reasoning but also other formalisms for handling uncertainty. This also contains good overviews by the editors explaining the historical significance of the selected papers. An introductory review for probabilistic expert systems is (Spiegelhalter *et al.*, 1993). Each of these three references contain a large number of references for further reading.

Dawid(1979) introduced the axiomatic basis for treating conditional independence. More recent accounts of conditional independence with emphasis on graphical models and their Markov properties are given by Whittaker(1990) and Lauritzen(1996). (The latter also contains proofs of the lemmas stated in section 8.)

The Asia example was given by Lauritzen and Spiegelhalter(1988), who showed how to do consistent probability calculations in multiply connected DAG's using propagation, (it is also reprinted in (Shafer and Pearl, 1990)). Junction trees arise in other areas and are known by different names (eg join trees in relational databases); see (Lauritzen and Spiegelhalter, 1988) for more on this and also the discussion section of that paper. A recent and general formulation of propagation in junction trees is given by Dawid(1992). A recent introductory textbook on Bayesian networks is (Jensen, 1996).

References

Dawid, A. P. (1979). Conditional independence in statistical theory (with discussion). *Journal of the Royal Statistical Society, Series B*, **41**, pp. 1–31.

Dawid, A. P. (1992). Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, **2**, pp. 25–36.

Jensen, F. V. (1996). *An introduction to Bayesian networks.* UCL Press, London.

Lauritzen, S. L. (1996). *Graphical models.* OUP.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society, Series B*, **50**, pp. 157–224.

Pearl, J. (1988). *Probabilistic inference in intelligent systems.* Morgan Kaufmann, San Mateo.

Shafer, G. R. and Pearl, J. (ed.) (1990). *Readings in uncertain reasoning.* Morgan Kaufmann, San Mateo, California.

Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L., and Cowell, R. G. (1993). Bayesian analysis in expert systems. *Statistical Science*, **8**, pp. 219–47.

Whittaker, J. (1990). *Graphical models in applied multivariate statistics.* John Wiley and Sons, Chichester.