Information processing

VISTA is a system used by NASA when launching space shuttles. Its purpose is to filter and display information on the propulsion system (Horvitz & Barry 1995).

Bruza & van der Gaag (1993) developed a language for constructing Bayesian networks for information retrieval, and Fung & Favero (1995) describe another system for information retrieval.

Medicine

Child helps in diagnosing congenital heart diseases (Franklin et al. 1989, Lauritzen et al. 1994). The system is described in Section 3.5.

MUNIN is a system for obtaining a preliminary diagnosis of neuromuscular diseases on the basis of electromyografic findings (Andreassen et al. 1989).

Painulim diagnoses neuromuscular diseases (Xiang et al. 1993).

Pathfinder is of assistance to community pathologists with the diagnosis of lymph-node pathology (Heckerman et al. 1992, Heckerman & Nathwani 1992a,b). The system is described in Section 5.6. Pathfinder has been integrated with videodiscs to the commercial system *Intellipath* (Nathwani et al. 1990).

SWAN is a system for insulin dose adjustment of diabetes patients (Andreassen et al. 1991, Hejlesen et al. 1993).

Miscellaneous

Hailfinder was developed for forecasting severe weather in the plane of northeastern Colorado (Abramson et al. 1996).

FRAIL is an automatic Bayesian network construction system (Goldman & Charniak 1993). It has been developed for building Bayesian networks for interpretation of written prose (Charniak & Goldman 1991).

Chapter 2

Causal and Bayesian networks

This chapter introduces *causal networks* as graphical representations of causal relations in a domain. Through several examples, basic rules for chained reasoning about certainty are introduced. These rules are formalized in the concept of *d-separation.*

In Section 2.3 we present the probability calculus used in this book, and we define the concept of a *Bayesian network.* In Section 2.4 the introductory examples are modelled as Bayesian networks and the reasoning is performed through probability calculations.

Finally we describe the BOBLO system.

2.1 Examples

In this section we give three examples. They illustrate crucial points to consider when reasoning about certainty has to be formalized.

2.1.1 Icy roads

Police Inspector Smith is impatiently awaiting the arrival of Mr Holmes and Dr Watson; they are late and Inspector Smith has another important appointment (lunch). Looking out of the window he wonders whether the roads are icy. Both are notoriously bad drivers, so if the roads are icy they may crash.

His secretary enters and tells him that Dr Watson has had a car accident. "Watson? OK. It could be worse ... icy roads! Then Holmes has most probably crashed too. I'll go for lunch now."

"Icy roads?", the secretary replies, "It is far from being that cold, and furthermore all the roads are salted." Inspector Smith is relieved. "Bad luck for Watson. Let us give Holmes ten minutes more."

To formalize the story, let the events be represented by variables with two states, *yes* and *no*. Suppose also that to each event is associated a *certainty*, which is a real number. So, we have the three variables: *icy roads (I), Holmes crashes (H)* and *Watson crashes (W). I* has the effect of increasing the certainty of both *H* and

cause to the certainty of the effect. The situation is illustrated in Figure 2.1.

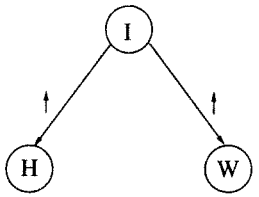


Figure 2.1 A network model of *icy roads*. The arrows on the links model the causal impact, and the small arrows attached to the links indicate the direction of the impact on the certainty.

When Inspector Smith is told that Watson has had a car accident, he is doing a reasoning in the opposite direction to the causal arrows. Since the impact function pointing at W is increasing, the inverse function is also increasing. Hence, he gets an increased certainty of I. The increased certainty of I in turn creates a new expectation, namely an increased certainty of H.

Next, when his secretary tells him that the roads cannot possibly be icy, the fact that Watson has crashed cannot change his expectation concerning road conditions and, consequently, Watson's crash has no influence on H.

This is an example of how dependence/independence changes with the information at hand. When nothing is known about the condition of the roads, then H and W are *dependent*: information on either event affects the certainty of the other. However, when the condition of the roads is known for certain, then they are *independent*: information on W has no effect on the certainty of H and vice versa. This phenomenon is called *conditional independence.*

2.1.2 Wet grass

Mr Holmes now lives in Los Angeles. One morning when Holmes leaves his house, he realizes that his grass is wet. Is it due to rain (R), or has he forgotten to turn off the sprinkler (S)? His belief in both events increases.

Next he notices that the grass of his neighbour, Dr Watson, is also wet. Elementary: Holmes is almost certain that it has been raining.

A formalization of the situation is shown in Figure 2.2.

When Holmes notices his own wet grass, he is doing a reasoning in the opposite direction to the causal arrows. Since both impact functions pointing at H are increasing, his certainty of both R and S increases. The increased certainty of R in turn creates an increased certainty of W.

Therefore Holmes checks Watson's grass, and when he discovers that it is also wet, he immediately increases the certainty of R drastically.

The next step in the reasoning is hard for machines, but natural for human beings, namely *explaining away*: Holmes' wet grass has been explained and thus there is

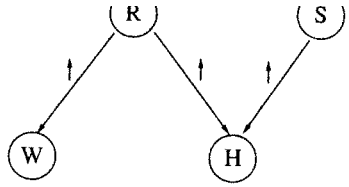


Figure 2.2 A network model for the *wet grass* example. *Rain* and *sprinkler* are causes of *Holmes'* grass being wet. Only *rain* can cause *Watson's* grass to be wet.

no longer any reason to believe that the sprinkler has been on. Hence, the certainty of S is reduced to its initial size.

Explaining away is another example of dependence changing with the information available. In the initial state, when nothing is known, R and S are independent. However, when we have information on Holmes' grass, then R and S become dependent.

2.1.3 Causation and reasoning

A possible source of confusion should be sorted out at this point. The graphs in Figures 2.1 and 2.2 were presented as models for impacts between events, but the reasoning based on the graphs is concerned with how our certainty of the various events is affected by new certainty of other events.

Actually, the models are guidelines for ways of reasoning about unknown events. When reasoning in the direction of the links, the statement in the model is:

The event A causes with certainty x the event B.

From this we reason:

If we know that A has taken place, then B has taken place with certainty x.

Reasoning in the opposite direction to the links is more delicate. So far we have only said that the certainty of the cause A increases when the consequence B has taken place. If you want to get a quantitative statement, your certainty calculus must have a way of inverting the causal statements. In Section 2.4 we show that for probability calculus, Bayes' rule is used for the inversion.

Some scientists take the point of view that the networks are not causal models, but models for how information may propagate between events. This is, from a foundational point of view, perfectly valid as long as you do not model interfering actions in your network. We shall expand on this in Chapter 6.

2.1.4 Earthquake or burglary

Mr Holmes is working at his office when he receives a telephone call from Watson, who tells him that Holmes' burglar alarm (A) has gone off. Convinced that a burglar

his way he listens to the radio (R), and in the news it is reported that there has been a small earthquake (E) in the area. Knowing that earthquakes have a tendency to turn the burglar alarm on, he returns to his work leaving his neighbours the pleasure of the noise. Figure 2.3 gives a model for the reasoning.

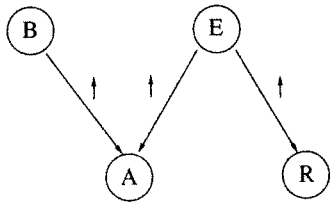


Figure 2.3 A model for the *earthquake* example. Notice that the structure is similar to Figure 2.2.

2.1.5 Prior certainties

It has been typical of the reasoning in the examples of this section that if some event is known, then the certainty of other events must be changed. If, in a certainty calculus, the actual certainty of a specific event has to be calculated, then knowledge of certainties prior to any information is also needed. In particular, prior certainties are required for the events which are not effects of causes in the network.

Take for instance the *wet grass* example. Given that Holmes' grass is wet, the certainty of R is still dependent on whether rain at night is a rare event (as in Los Angeles) or very common (as in London).

The same goes for the earthquake in Section 2.1.4. Though E may have a stronger effect on A than B has, and therefore information on A will increase the certainty of earthquake more than on burglary, the resulting certainty on E should still be lower than the certainty on B. To be able to do this reasoning, prior certainties on E and B are required.

2.2 Causal networks and d-separation

The models in Section 2.1 are examples of *causal networks*. A causal network consists of a set of *variables* and a set of *directed links* between variables. Mathematically the structure is called a directed graph. When talking about the relations in a directed graph we use the wording of family relations: if there is a link from A to B we say that B is a *child* of A, and A is a *parent* of B.

The variables represent events (propositions). In Section 2.1, each variable had the states *yes* and *no* reflecting whether a certain event had taken place or not. In general, a variable can have any number of states. A variable may, for example, be the colour of a car (states $blue, green, red, brown$), the number of children in a family (states $0, 1, 2, 3, 4, 5, 6, > 6$), or a disease (states *bronchitis, tuberculosis,*

book we solely consider variables with a finite number of states.

In a causal network a variable represents a set of possible states of affairs. A variable is in exactly one of its states; which one may be unknown to us.

Reasoning about uncertainty also has a quantitative part, namely calculation and combination of certainty numbers. The considerations in this section are independent of the particular uncertainty calculus. Whatever calculus is used, it must obey the rules illustrated in Section 2.1 that we formalize in this section.

Serial connections

Consider the situation in Figure 2.4. A has an influence on B which in turn has influence on C. Obviously, evidence on A will influence the certainty of B which then influences the certainty of C. Similarly, evidence on C will influence the certainty on A through B. On the other hand, if the state of B is known, then the channel is blocked, and A and C become independent. We say that A and C are *d-separated given* B, and when the state of a variable is known we say that it is *instantiated.*

We conclude that evidence may be transmitted through a serial connection unless the state of the variable in the connection is known.

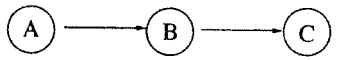


Figure 2.4 Serial connection. When B is instantiated it blocks communication between A and C.

Diverging connections

The situation in Figure 2.5 is a generalization of the *icy roads* example. Influence can pass between all the children of A unless the state of A is known. We say that $B, C, \ldots, E$ are d-separated given A.

So, evidence may be transmitted through a diverging connection unless it is instantiated.

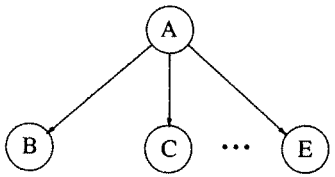


Figure 2.5 Diverging connection. If A is instantiated, it blocks communication between its children.

A description of the situation in Figure 2.6 requires a little more care. If nothing is known about A except what may be inferred from knowledge of its parents $B, \ldots, E$, then the parents are independent: evidence on one of them has no influence on the certainty of the others.

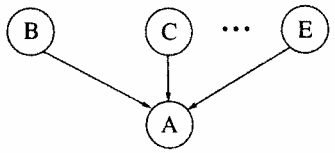


Figure 2.6 Converging connection. If A changes certainty, it opens communication between its parents.

Now, if any other kind of evidence influences the certainty of A, then the parents become dependent due to the principle of explaining away. The evidence may be direct evidence on A, or it may be evidence from a child. This phenomenon is called *conditional dependence*. In Figure 2.7 some illustrating examples are listed.

The conclusion is that evidence may only be transmitted through a converging connection if either the variable in the connection or one of its descendants has received evidence.

Remark. Evidence on a variable is a statement of the certainties of its states. If the statement gives the exact state of the variable we call it *hard* evidence, otherwise it is called *soft*. Hard evidence is also called *instantiation*. Blocking in the case of serial and diverging connections requires hard evidence, while opening in the case of converging connections holds for all kinds of evidence.

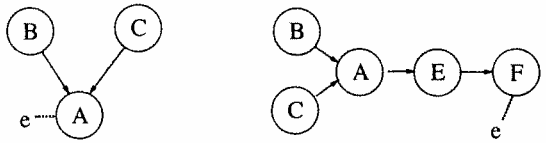


Figure 2.7 Examples where the parents of A are dependent. The dotted lines indicate insertion of evidence.

2.2.1 d-separation

The three cases given above cover all the ways in which evidence may be transmitted through a variable, and following the rules it is possible to decide for any pair of variables in a causal network whether they are dependent given the evidence entered into the network. The rules are formulated in the following.

Definition (d-separation). Two variables A and B in a causal network are *d-separated* if for all paths between A and B there is an intermediate variable V such that either

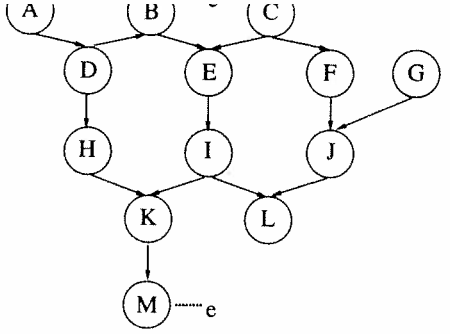


Figure 2.8 A causal network with B and M instantiated. A is d-separated from G only.

– the connection is serial or diverging and the state of V is known

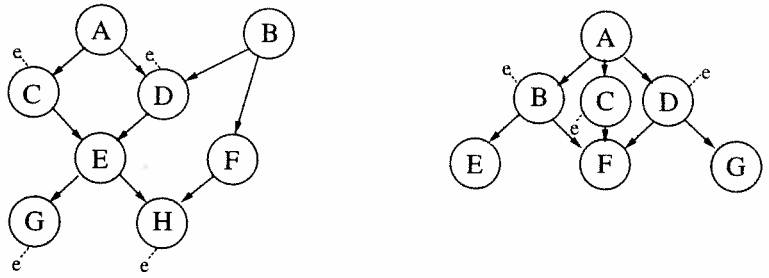
or

– the connection is converging and neither V nor any of Vs descendants have received evidence.

If A and B are not d-separated we call them *d-connected.*

Figure 2.8 gives an example of a larger network. The evidence entered at B and M represents instantiation. If evidence is entered at A it may be transmitted to D. The variable B is blocked, so the evidence cannot pass through B to E. However, it may be passed to H and K. Since the child M of K has received evidence, evidence from H may pass to I and further to E, C, F, J and L. So, the path $A - D - H - K - I - E - C - F - J - L$ is a d-connecting path.

Figure 2.9 gives two illustrating examples.



(a) (b)

Figure 2.9 Causal networks with hard evidence entered (the variables are instantiated). (a) Although all neighbours of E are instantiated it is d-connected to F, B and A. (b) F is d-separated from the remaining un-instantiated variables.

change the belief in B.

You may wonder why we have introduced d-separation as a definition rather than as a theorem. A theorem should be as follows.

Claim. If A and B are d-separated, then changes in the certainty of A have no impact on the certainty on B.

However, the claim cannot be established as a theorem without a more precise description of the concept of "certainty". You can take d-separation as a property of human reasoning and require that any certainty calculus must comply with the claim.

2.3 Bayesian networks

So far nothing has been said about the quantitative part of certainty assessment. Various certainty calculi exist, but in this book we only treat the so called Bayesian calculus, which is *classical probability calculus*.

2.3.1 Basic axioms

The probability $P(A)$ of an event A is a number in the unit interval $[0, 1]$. Probabilities obey the following basic axioms.

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

(ii) If A and B are mutually exclusive, then

$$P(A \vee B) = P(A) + P(B).$$

2.3.2 Conditional probabilities

The basic concept in the Bayesian treatment of certainties in causal networks is *conditional probability*. Whenever a statement of the probability, $P(A)$, of an event A is given, then it is given conditioned by other known factors. A statement like "The probability of the die turning up 6 is $\frac{1}{6}$" usually has the unsaid prerequisite that it is a fair die – or rather, as long as I know nothing of it, I assume it to be a fair die. This means that the statement should be "Given that it is a fair die, the probability ...". In this way, any statement on probabilities is a statement conditioned on what else is known.

A conditional probability statement is of the following kind:

Given the event B, the probability of the event A is x.

The notation for the statement above is $P(A \mid B) = x$.

It should be stressed that $P(A \mid B) = x$ does not mean that whenever B is true then the probability for A is x. It means that if B is true, and *everything else known is irrelevant for* A, then $P(A) = x$.

$$P(A \mid B)P(B) = P(A, B), \tag{2.1}$$

where $P(A, B)$ is the probability of the joint event $A \wedge B$. Remembering that probabilities should always be conditioned by a context C, the formula should read

$$P(A \mid B, C)P(B \mid C) = P(A, B \mid C). \tag{2.2}$$

From 2.1 it follows that $P(A \mid B)P(B) = P(B \mid A)P(A)$ and this yields the well known *Bayes' rule*:

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A)}. \tag{2.3}$$

Bayes' rule conditioned on C reads

$$P(B \mid A, C) = \frac{P(A \mid B, C)P(B \mid C)}{P(A \mid C)}. \tag{2.4}$$

Formula (2.2) should be considered an axiom for probability calculus rather than a theorem. A justification for the formula can be found by counting frequencies: suppose we have m cats (C) of which n are brown (B), and i of the brown cats are Abyssinians (A). Then the frequency of As given B among the cats, $f(A \mid B, C)$, is $\frac{i}{n}$, the frequency of Bs, $f(B \mid C)$, is $\frac{n}{m}$, and the frequency of brown Abyssinian cats, $f(A, B \mid C)$ is $\frac{i}{m}$. Hence,

$$f(A \mid B, C)f(B \mid C) = f(A, B \mid C).$$

Likelihood

Sometimes $P(A \mid B)$ is called the *likelihood of B given A*, and it is denoted $L(B \mid A)$.

The reason for this is the following. Assume $B_1, \ldots, B_n$ are possible scenarios wilth an effect on the event A, and we know A. Then $P(A \mid B_i)$ is a measure of how likely it is that B_i is the cause. In particular, if all B_is have the same prior probability, Bayes' rule yields

$$P(B_i \mid A) = \frac{P(A \mid B_i)P(B_i)}{P(A)} = kP(A \mid B_i),$$

where k is independent of i.

2.3.3 Subjective probabilities

The justification in the previous section for the fundamental rule was based on frequencies. This does not mean that we only consider probabilities based on frequencies. Probabilities may also be completely subjective estimates of the certainty of an event.

A subjective probability may, for example, be my personal assessment of the chances of selling more than 2,000 copies of this book in 1997.

A way to assess this probability could be the following. I am given the choice between two gambles:

(2) I will by the end of 1997 be allowed to draw a ball from an urn with n red balls and $100 - n$ white balls. If my ball is red I will get \$100.

Now, if all balls in the urn are red I will prefer (2), and if all balls are white I will prefer (1). There is a number n for which the two gambles are equally attractive, and for this n, $\frac{n}{100}$ is my estimate of the probability of selling more than 2,000 copies of this book in 1997 (I shall not disclose the n to the reader).

For subjective probabilities defined through such ball drawing gambles the fundamental rule can also be proved.

2.3.4 Probability calculus for variables

As stated in Section 2.2, the nodes in a causal network are *variables* with a *finite number of mutually exclusive states*.

If A is a variable with states $a_1, \ldots, a_n$, then $P(A)$ is a probability distribution over these states:

$$P(A) = (x_1, \ldots, x_n) \qquad x_i \geq 0 \qquad \sum_{i=1}^{n} x_i = 1,$$

where x_i is the probability of A being in state a_i.

Notation. The probability of A being in state a_i is denoted $P(A = a_i)$ and denoted $P(a_i)$ if the variable is obvious from the context.

If the variable B has states $b_1, \ldots, b_m$, then $P(A \mid B)$ is an $n \times m$ table containing numbers $P(a_i \mid b_j)$ (see Table 2.1).

$P(A, B)$, the joint probability for the variables A and B, is also an $n \times m$ table. It consists of a probability for each configuration (a_i, b_j) (see Table 2.2).

When the fundamental rule (2.1) is used on variables A and B, then the procedure is to apply the rule to the $n \cdot m$ configurations (a_i, b_j):

$$P(a_i \mid b_j)P(b_j) = P(a_i, b_j).$$

This means that in the table $P(A \mid B)$, for each j the column for b_j is multiplied by $P(b_j)$ to obtain the table $P(A, B)$. If $P(B) = (0.4, 0.4, 0.2)$ then Table 2.2 is the result of using the fundamental rule on Table 2.1. When applied to variables, we use the same notation for the fundamental rule:

$$P(A \mid B)P(B) = P(A, B).$$

From a table $P(A, B)$ the probability distribution $P(A)$ can be calculated. Let a_i be a state of A. There are exactly m different events for which A is in state a_i, namely the mutually exclusive events $(a_i, b_1), \ldots, (a_i, b_m)$. Therefore, by axiom (ii)

$$P(a_i) = \sum_{j=1}^{m} P(a_i, b_j).$$

Note that the columns sum to one.

	b_1	b_2	b_3
a_1	0.4	0.3	0.6
a_2	0.6	0.7	0.4

Table 2.2 An example of $P(A, B)$. Note that the sum of all entries is one.

	b_1	b_2	b_3
a_1	0.16	0.12	0.12
a_2	0.24	0.28	0.08

This calculation is called *marginalization* and we say that the variable B is marginalized out of $P(A, B)$ (resulting in $P(A)$). The notation is

$$P(A) = \sum_{B} P(A, B).$$

By marginalizing B out of Table 2.2 we get $P(A) = (0.4, 0.6)$.

The division in Bayes' rule (2.3) is treated in the same way as the multiplication in the fundamental rule (see Table 2.3).

2.3.5 Conditional independence

The blocking of transmission of evidence as described in Section 2.2.1 is, in Bayesian calculus, reflected in the concept of *conditional independence*. The variables A and C are *independent given the variable* B if

$$P(A \mid B) = P(A \mid B, C).$$

This means that if the state of B is known then no knowledge of C will alter probability of A.

Table 2.3 $P(B \mid A)$ as a result of applying Bayes' rule to Table 2.1 and $P(B) = (0.4, 0.4, 0.2)$.

	a_1	a_2
b_1	0.4	0.4
b_2	0.3	0.47
b_3	0.3	0.13

Remark. If condition B is empty, we simply say that ...

Conditional independence appears in the cases of serial and diverging connections (see Figure 2.10).

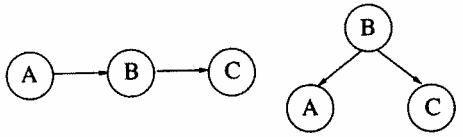


Figure 2.10 Examples where A and C are conditionally independent given B.

Definition (2.6) may look asymmetric; however, if (2.6) holds, then – by the conditioned Bayes' rule (2.4) – we get

$$P(C \mid B, A) = \frac{P(A \mid C, B)P(C \mid B)}{P(A \mid B)} = \frac{P(A \mid B)P(C \mid B)}{P(A \mid B)} = P(C \mid B).$$

The proof requires that $P(A \mid B) > 0$. That is, for states a, b with $P(A = a \mid B = b) = 0$ the calculation is not valid. However, for our considerations it does not matter; if B is in state b then the evidence $A = a$ is impossible and will not appear. So, why bother with the transmission of it?

2.3.6 Definition of Bayesian networks

Causal relations also have a quantitative side, namely their *strength*. This is expressed by attaching numbers to the links.

Let A be a parent of B. Using probability calculus it would be natural to let $P(B \mid A)$ be the strength of the link. However, if C is also a parent of B, then the two conditional probabilities $P(B \mid A)$ and $P(B \mid C)$ alone do not give any clue on how the impacts from A and B interact. They may co-operate or counteract in various ways. So, we need a specification of $P(B \mid A, C)$.

It may happen that the domain to be modelled contains feed-back cycles (see Fig. 2.11).

Feed-back cycles are difficult to model quantitatively (this is, for example, what differential equations are all about); for causal networks no calculus has been developed that can cope with feed-back cycles Therefore we require that the network does not contain cycles.

A Bayesian network consists of the following.

A set of *variables* and a set of *directed edges* between variables.

Each variable has a finite set of mutually exclusive states.

The variables together with the directed edges form a *directed acyclic graph* (DAG). (A directed graph is *acyclic* if there is no directed path $A_1 \to \cdots \to A_n$ such that $A_1 = A_n$.)

To each variable A with parents $B_1, \ldots, B_n$ there is attached a conditional probability table $P(A \mid B_1, \ldots, B_n)$.

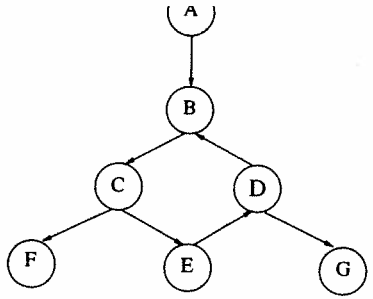


Figure 2.11 A directed graph with a feed-back cycle. This is not allowed in Bayesian networks.

Note that if A has no parents then the table reduces to unconditional probabilitie $P(A)$. For the DAG in Figure 2.12 the prior probabilitiess $P(A)$ and $P(B)$ must b specified. It has been claimed that prior probabilities are an unwanted introductio of bias to the model, and calculi have been invented in order to avoid it. However, a discussed in Section 2.1.5, prior probabilities are necessary – not for mathematica reasons – but because prior certainty assessments are an integral part of huma reasoning about certainty.

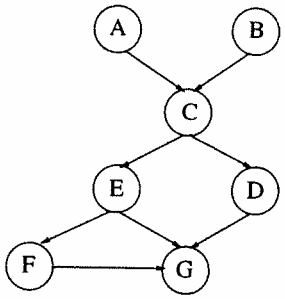


Figure 2.12 A directed acyclic graph (DAG). The probabilities to specify are $P(A)$, $P(B)$, $P(C \mid A, B)$, $P(E \mid C)$, $P(D \mid C)$, $P(F \mid E)$ and $P(G \mid D, E, F)$.

One of the advantages of Bayesian networks is that they *admit d-separation*. if A and B are d-separated in a Bayesian network with evidence e entered, ther $P(A \mid B, e) = P(A \mid e)$. This means that you can use d-separation to read-off conditional independencies. We will use this fact without proof.

2.3.7 The chain rule

Let $U = (A_1, \ldots, A_n)$ be a universe of variables. If we have access to the joint probability table $P(U) = P(A_1, \ldots, A_n)$, then we can also calculate $P(A_i)$ as well

nentially with the number of variables, and U need not be very large before the table becomes intractably large. Therefore, we look for a more compact *representation* of $P(U)$: a way of storing information from which $P(U)$ can be calculated if needed.

A Bayesian network over U is such a representation. If the conditional independencies in the Bayesian network hold for U, then $P(U)$ can be calculated from the conditional probabilities specified in the network.

Theorem 2.1 (The chain rule.) *Let* BN *be a Bayesian network over*

$$U = \{A_1, \ldots, A_m\}.$$

Then the joint probability distribution $P(U)$ is the product of all conditional probabilities specified in BN*:*

$$P(U) = \prod_i P(A_i \mid pa(A_i))$$

where $pa(A_i)$ is the parent set of A_i.

Proof. (Induction in the number of variables in the universe U.)

If U consists of one variable then the theorem is trivial.

Assume the chain rule to be true for all networks consisting of $n-1$ variables, and let U be the universe for a DAG with n variables. Since the network is acyclic there is at least one variable A without children. Consider the DAG with A removed.

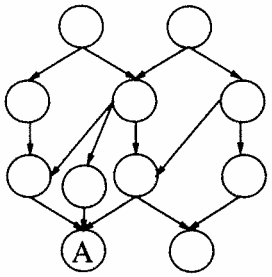


Figure 2.13 A DAG with n variables. If the variable A is removed, the induction hypothesis can be applied.

From the induction hypothesis we have that $P(U \setminus \{A\})$ is the product of all specified probabilities – except $P(A \mid pa(A))$.

By the fundamental rule we have

$$P(U) = P(A \mid U \setminus \{A\})P(U \setminus \{A\}).$$

Since A is independent of $U \setminus (\{A\} \cup pa(A))$ given $pa(A)$ (see Fig. 2.13), we get

$$P(U) = P(A \mid U \setminus \{A\})P(U \setminus \{A\}) = P(A \mid pa(A))P(U \setminus \{A\}).$$

The righthand side above is the product of all specified probabilities.

	$I=y$	$I=n$
$H=y$	0.8	0.1
$H=n$	0.2	0.9

$P(H \mid I)$

	$I=y$	$I=n$
$W=y$	0.8	0.1
$W=n$	0.2	0.9

$P(W \mid I)$

Table 2.5 Joint probability table for $P(W, I)$ and $P(H, I)$.

	$I=y$	$I=n$
y	0.56	0.03
n	0.14	0.27

2.4 The examples revisited

In this section we apply the rules of probability calculus on the introductory examples. This is done to illustrate that probability calculus can be used to perform the reasoning in the examples – in particular explaining away. In Chapter 4 we give a general algorithm for probability updating in Bayesian networks. This algorithm makes the calculations considerably easier than those in this section.

2.4.1 Icy roads

(See Fig. 2.1.) For the quantitative modelling we need three probability assessments: $P(H \mid I)$, $P(W \mid I)$ and $P(I)$. The model in Figure 2.1 reflects that only knowledge of icy roads is relevant for H and W. We should then attach a certainty to I based on whatever knowledge may be available. In this case the police inspector has been looking out of the window and wondering whether the roads were icy. We let the probability for icy roads be 0.7.

Since both Holmes and Watson are bad drivers, we put the probability of a crash in the case of icy roads to 0.8, and the probability of a crash if the roads are not icy we put to 0.1 (they *are* bad drivers). An overview of the conditional probabilities is given in Table 2.4.

To calculate the initial probabilities for H and W we first use the fundamental rule (2.1) to calculate $P(W, I)$ and $P(H, I)$:

$$P(W = y, I = y) = P(W = y \mid I = y)P(I = y) = 0.8 \cdot 0.7 = 0.56.$$

Table 2.5 gives all four probabilities.

In order to get the probabilities for W and H we marginalize I out of Table 2.5 and get

$$P(W) = P(H) = (0.59, 0.41).$$

The information that Watson has crashed is now used to update the probability of

$$
\begin{aligned}
P(I \mid W = y) &= \frac{P(W = y \mid I)P(I)}{P(W = y)} \\
&= \frac{1}{0.59}(0.8 \cdot 0.7, 0.1 \cdot 0.3) \\
&= (0.95, 0.05).
\end{aligned}
$$

To update the probability of H, first we use the fundamental rule (2.1) to calculate $P(H, I)$ as shown in Table 2.6.

Table 2.6 Tables showing the calculation of $P(H, I)$.

	$I = y$	$I = n$
$H = y$	$0.8 \cdot 0.95$	$0.1 \cdot 0.05$
$H = n$	$0.2 \cdot 0.95$	$0.9 \cdot 0.05$

$=$

	$I = y$	$I = n$
$H = y$	0.76	0.005
$H = n$	0.19	0.045

Finally, calculate $P(H)$ by marginalizing I out of $P(H, I)$. The result is

$$P(H) = (0.765, 0.235).$$

This is the quantitative effect of the information that Watson has crashed.

At last, when Inspector Smith is convinced that the roads are not icy, then $P(H \mid I = n) = (0.1, 0.9)$.

The calculation can be considered in a different way. First we calculate $P(H, I)$ and $P(W, I)$ (Table 2.5), and we have two joint probability tables with the variable I in common.

If evidence on W now arrives in the form of $P^*(W) = (0, 1)$, then

$$P^*(W, I) = P(I \mid W)P^*(W) = \frac{P(W, I)}{P(W)}P^*(W).$$

This means that the joint probability table for W and I is updated by multiplying by the new distribution and dividing by the old one. The multiplication consists of annihilating all entries with $W = n$. The division by $P(W)$ only has an effect on entries with $W = y$, so therefore the division is by $P(W = y)$.

Next, calculate $P^*(I)$ from $P^*(W, I)$ by marginalization, and use $P^*(I)$ to update $P(H, I)$

$$P^*(H, I) = \frac{P(H, I)}{P(I)} \cdot P^*(I)$$

and finally $P^*(H)$ is calculated by marginalizing $P^*(H, I)$.

2.4.2 Wet grass

(See Fig. 2.2.) Let the prior probabilities for R and S be $P(R) = (0.2, 0.8)$ and $P(S) = (0.1, 0.9)$. The remaining probabilities are listed in Table 2.7. First, calculate the prior probabilities for W and H by formulae (2.1) and (2.5). That is, first

Table 2.7 The probabilities for the *wet grass* example. The vectors (α, β) in the righthand table represent $(H = y, H = n)$.

	$R = y$	$R = n$
$W = y$	1	0.2
$W = n$	0	0.8

$P(W \mid R)$

	$R = y$	$R = n$
$S = y$	(1, 0)	(0.9, 0.1)
$S = n$	(1, 0)	(0, 1)

$P(H \mid R, S)$

The calculation of $P(H, R, S)$ follows the same scheme, only the product is

$$P(H, R, S) = P(H \mid R, S)P(R, S).$$

Since R and S are independent (see Fig. 2.2) we have (see Exercise 2.9)

$$P(H, R, S) = P(H \mid R, S)P(R)P(S).$$

The result is given in Table 2.8. Marginalizing R and S out of $P(H, R, S)$ yields $P(H) = (0.272, 0.728)$. We shall use the approach outlined at the end of Section 2.4.1. We have established joint probability tables for two of the clusters, (W, R) and (H, R, S), with the variable R in common.

Table 2.8 The prior probability table for $P(H, R, S)$. The vectors (α, β) in the table represent $(H = y, H = n)$.

	$R = y$	$R = n$
$S = y$	(0.02, 0)	(0.072, 0.008)
$S = n$	(0.18, 0)	(0, 0.72)

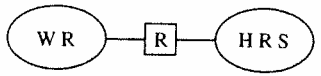


Figure 2.14 The clusters for the *wet grass* example. They communicate through the variable R.

The evidence $H = y$ is used to update $P(H, R, S)$ by annihilating all entries with $H = n$ and dividing by $P(H = y)$. Since the result shall be a probability table with all entries summing to one we need not calculate $P(H)$. After all entries with $H = n$ have been annihilated (Table 2.9), we simply normalize the table by dividing by the sum of the remaining entries (see Table 2.10).

The distributions $P^*(R)$ and $P^*(S)$ are calculated through marginalization of

$$P^*(H, R, S).$$

with $H = n$ annihilated.

	$R = y$	$R = n$
$S = y$	(0.02, 0)	(0.072, 0)
$S = n$	(0.18, 0)	(0, 0)

Table 2.10 The calculation of $P^*(H, R, S) = P(H, R, S \mid H = y)$.

	$R = y$	$R = n$		$R = y$	$R = n$
$S = y$	$\frac{1}{0.272}(0.02, 0)$	$\frac{1}{0.272}(0.072, 0)$		(0.074, 0)	(0.264, 0)
$S = n$	$\frac{1}{0.272}(0.18, 0)$	$\frac{1}{0.272}(0, 0)$	$\approx$	(0.662, 0)	(0, 0)

We get $P^*(R = y) = 0.736$ and $P^*(S = y) = 0.339$.

Use $P^*(R)$ to update $P(W, R)$ (see Table 2.11):

$$P^*(W, R) = P(W \mid R)P^*(R) = P(W, R)\frac{P^*(R)}{P(R)}.$$

Table 2.11 Calculation of $P^*(W, R) = P(W, R)\frac{P^*(R)}{P(R)}$.

	$R = y$	$R = n$		$R = y$	$R = n$
$W = y$	$0.2 \cdot \frac{0.736}{0.2}$	$0.16 \cdot \frac{0.264}{0.8}$		0.736	0.0528
$W = n$	0	$0.64 \cdot \frac{0.264}{0.8}$	=	0	0.2112

Now use $W = y$ to update the distribution for (W, R) (see Table 2.12). We get $P^{**}(R = y) = 0.93$.

We still have to calculate $P^{**}(S) = P(S \mid W = y, H = y)$. The result must reflect the explaining away effect; since the wet grass is explained by rain, the probability for $S = y$ should decrease to its initial value.

The calculation follows the same pattern. A message on $P^{**}(R)$ is sent from (W, R) to (H, R, S) (see Fig. 2.14),

$$P^{**}(H, R, S) = P^*(H, R, S)\frac{P^{**}(R)}{P^*(R)}.$$

By marginalizing we get $P^{**}(S = y) = 0.161$.

$P(W, R \mid W = y, H = y)$.

	$R = y$	$R = n$
$W = y$	$\frac{0.736}{0.7888}$	$\frac{0.0528}{0.7888}$
$W = n$	0	0

Table 2.13 $P^{**}(R, S) = P(R, S \mid H = y, W = y)$.

	$R = y$	$R = n$
$S = y$	0.094	0.067
$S = n$	0.839	0

The reason why the probability for sprinkler does not drop to the prior probabilit of 0.1 is that Dr Watson is a forgetful fellow who may have forgotten his sprinkle and an explanation may be that both sprinklers have been forgotten. This is reflecte in the probability $P(W = y \mid R = n) = 0.2$.

2.5 BOBLO

BOBLO is a system which helps in the verification of parentage for Jersey cattl through blood-type identification. The introduction of embryo transplantation tech nology and the increasing trade of semen and embryos have stressed the importanc of proper pedigree registration, and therefore there is a need for sophisticated meth ods for individual identification and parentage control of cattle.

Heredity is determined by *genes* which are placed in chromosomes (see Fig. 2.15)

Figure 2.15 A pair of chromosomes. The pearls in the strings are loci.

Except for the sex chromosomes, chromosomes go in structurally identical pairs - one chromosome inherited from each parent. A chromosome may be considered as a string of genes. The places where the genes are positioned are called *loci*. Each gene has a particular locus of position and genes which can be placed at a particular locus are called *allels*. The pair of allels at a locus (one from each chromosome) is called a *genotype*, and the property determined by a genotype is called the *phenotype*.

For the blood group determination of cattle, ten different independent blood-group systems are used. These systems control 52 different blood-group *factors* which can

nation is relatively simple (controlling from one to four blood-group factors only). However, the systems B- and C- are rather complicated, controlling respectively 26 and 10 of the above-mentioned 52 blood-group factors.

Heredity of blood type follows the normal genetic rules, however, the blood groups are attached to sets of loci rather than to single loci, and instead of allels the term *phenogroup* is used. So, for each blood group, a Bayesian network for inheritance will be as in Figure 2.16.

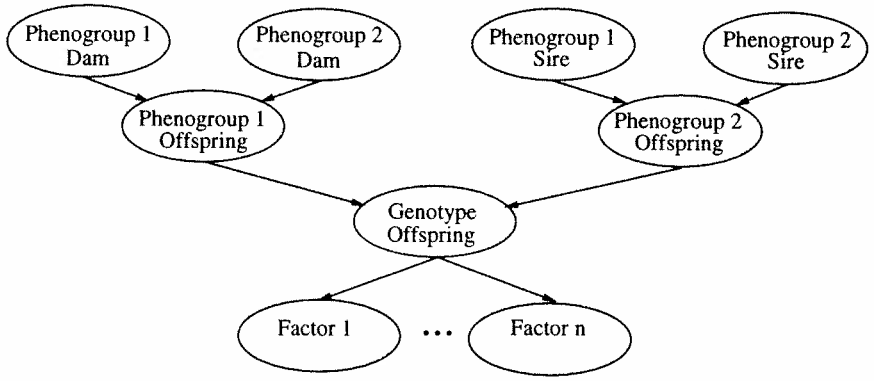


Figure 2.16 Heredity of blood type. From each parent one out of two phenotypes are chosen. This constitutes the genotype of the offspring, and the genotype determines a set of factors measurable in a laboratory (the phenotype).

If nothing is known of the phenogroups of the parents they are given a prior probability equal to the frequencies of the various phenogroups. Let us, for the example, suppose that there are three phenogroups f_1, f_2, f_3 with frequencies (0.58, 0.1, 0.32) (this is the situation for the so-called *F-system*).

When a calf is registered, the parents are stated and their phenogroups are already registered. If the stated parents are the true parents we have no problems, but what if they are not so? Then we will say that the phenogroups of the true parents are distributed as the prior probabilities, that is (0.58, 0.1, 0.32).

So, for modelling the part concerning possible parental errors, we can introduce a node *parental error* with states *both, sire, dam* and *no*, and with prior probabilities to be the frequency of parental errors. This leads to the Bayesian network in Figure 2.17.

The network model in BOBLO also has a part that models the risks of mistakes in the laboratory procedures (see Exercise 3.6). For now, assume that evidence on factors are entered directly to the nodes *factor*. It is assumed that the stated parents are so well known that their genotypes are known, and therefore the state of the variables *phenogroup stated d/s* is known.

Note how the impact of evidence flows from the *factor* nodes to the node *parental error*: it first flows to *phenogroup true d/s* (serial connections). Since evidence has

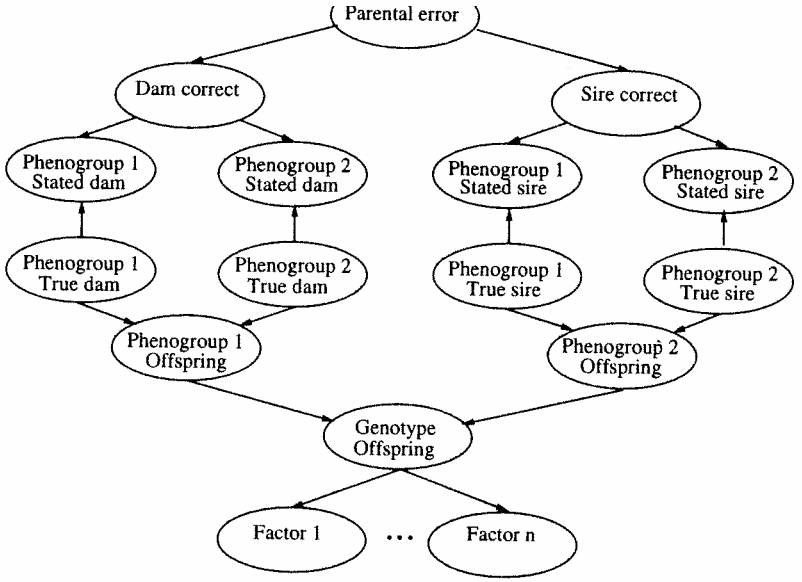


Figure 2.17 The part of BOBLO modelling parental error. Evidence is entered into the variables *factor* and *phenogroup stated d/s*. Evidence from *factor* is transmitted to *parental error* because *phenogroup stated* has received evidence.

been entered to *phenogroup stated d/s* the evidence is transmitted further to *dan correct* and *sire correct* (converging connections) to end in *parental error*.

BOBLO is an acronym for BOvine BLOod typing, and it has been in use at th Danish Blood Type Laboratory improving the accuracy of detecting parental error (tests quantifying the improvement have not been finished).

2.6 Summary

d-separation in causal networks

Two variables A and B in a causal network are d-separated if for all paths betwee A and B there is an intermediate variable V such that either

- the connection is serial or diverging and the state of V is known or
- the connection is converging, and neither V nor any of Vs descendants have received evidence.

The fundamental rule for probability calculus

$$P(A \mid B, C)P(B \mid C) = P(A, B \mid C)$$

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

Marginalization

$$P(A) = \sum_i P(A, b_i) = P(A, b_1) + \cdots + P(A, b_n)$$

Conditional independence

A and C are independent given B if $P(A \mid B) = P(A \mid B, C)$.

Definition of Bayesian networks

A Bayesian network consists of the following.

A set of *variables* and a set of *directed edges* between variables.

Each variable has a finite set of states.

The variables together with the directed edges form a *directed acyclic graph* (DAG).

To each variable A with parents $B_1, \ldots, B_n$ there is attached a conditional probability table $P(A \mid B_1, \ldots, B_n)$.

Admittance of d-separation in Bayesian networks

If A and B are d-separated in a Bayesian network with evidence e entered, then $P(A \mid B, e) = P(A \mid e)$.

The chain rule

Let BN be a Bayesian network over $U = \{A_1, \ldots, A_m\}$. Then the joint probability distribution $P(U)$ is the product of all conditional probabilities specified in BN:

$$P(U) = \prod_i P(A_i \mid pa(A_i)),$$

where $pa(A_i)$ is the parent set of A_i.

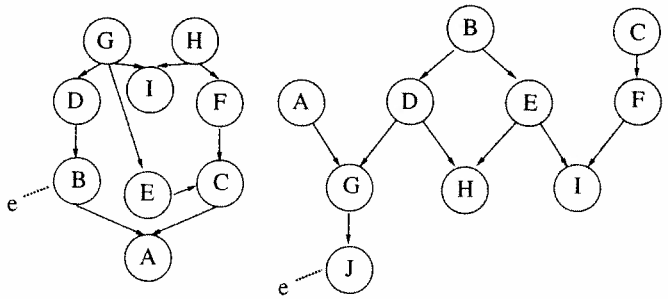
The two Examples 2.1.2 and 2.1.4 are inspired by Pearl (1988). The concepts of causal network, d-connection, and the definition in Section 2.2.1 are due to Pearl (1986b) and Verma (1987). A proof that Bayesian networks admit d-separation can be found in Pearl (1988) or in Lauritzen (1996). Bayesian networks have a long history in statistics, and in the first half of the 1980s they were introduced to the field of expert systems through work by Pearl (1982) and Spiegelhalter & knill-Jones (1984). BOBLO is documented in Rasmussen (1995a,b).

Exercises

Exercise 2.1 Show that d-connectedness is *symmetric* (if A is d-connected to B, then B is d-connected to A).

Give an example proving that d-connectedness is not *transitive* (A d-connected to B and B d-connected to C, but A and C are not d-connected).

Exercise 2.2 In the graphs below determine which variables are d-connected to A.



(a) (b)

Figure for Exercise 2.2

Exercise 2.3 Let A be a variable in a DAG. Assume that the following variables are instantiated: the parents of A, the children of A, the spouses of A (variables that share a child with A).

Show that A is d-separated from the remaining uninstantiated variables.

Exercise 2.4 Let D_1 and D_2 be DAGs over the same variables. D_1 is an *I-submap* of D_2 if all d-separation properties of D_1 also hold for D_2. If, also, D_2 is an I-submap of D_2, they are sid to be *I-equivalent.*

Which of the four DAGs in the figure below are I-equivalent?

	b_1	b_2	b_3
a_1	0.05	0.10	0.05
a_2	0.15	0.00	0.25
a_3	0.10	0.20	0.10

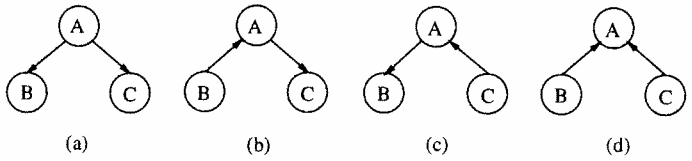


Figure for Exercise 2.4.

Exercise 2.5 Calculate $P(A)$, $P(B)$, $P(A \mid B)$, and $P(B \mid A)$ from Table 2.14.

Table 2.15 $P(A, B, C)$ for Exercise 2.6.

	b_1	b_2
a_1	(0.006, 0.054)	(0.048, 0.432)
a_2	(0.014, 0.126)	(0.032, 0.288)

Table 2.16 Conditional probability tables for Exercise 2.7.

	a_1	a_2
b_1	0.2	0.3
b_2	0.8	0.7

$P(B \mid A)$

	a_1	a_2
c_1	0.5	0.6
c_2	0.5	0.4

$P(C \mid A)$

Exercise 2.6 In Table 2.15, a joint probability table for the binary variables A, B, and C is given.

(i) Calculate $P(B, C)$ and $P(B)$.

(ii) Are A and C independent given B?

Exercise 2.7 The DAG (a) in Exercise 2.4 has $P(A) = (0.1, 0.9)$ and the conditional probability given in Table 2.16.

Calculate $P(A, B, C)$.

Exercise 2.8 Perform a Bayesian calculation of the reasoning in Section 2.1.4 (earthquake or burglary). Use the probabilities in Table 2.17 and $P(B) = (0.01, 0.99)$, $P(E) = (0.001, 0.999)$.

and alarm.

	$E = y$	$E = n$
$R = y$	0.95	0.01
$R = n$	0.05	0.99

$P(R \mid E)$

	$B = y$	$B = n$
$E = y$	(0.98, 0.02)	(0.95, 0.05)
$E = n$	(0.95, 0.05)	(0.03, 0.97)

$P(A \mid B, E)$

Exercise 2.9 Let $P(c_i \mid b_j) \neq 0$ for all i, j. Prove that A and C are independent given B if and only if $P(A, C \mid B) = P(A \mid B)P(C \mid B)$.

Chapter 3

Building models

Bayesian networks create a very efficient language for building models of domains with inherent uncertainty. However, as can be seen from the calculations in Section 2.4, it is a tedious job to perform evidence transmission even for very simple Bayesian networks. Fortunately, software tools which can do the calculation job for us are available. Several commercial products exist containing both an editor for Bayesian networks and a runtime module which takes care of evidence transmission. In the rest of this book we assume that the reader has access to the HUGIN system provided by the diskette attached to the book, or to any other Bayesian network programming environment.

Therefore we can start by concentrating on how to use Bayesian networks in model building and defer a presentation of the methods for probability updating to Chapter 4.

In Section 3.1 we examine, through three examples, the considerations when determining the structure of a Bayesian network model. Section 3.2 gives examples of estimation of the conditional probabilities. The examples cover theoretically well-founded probabilities as well as probabilities taken from data bases and purely subjective estimates. Section 3.3 gives several modelling tricks to use when the amount of numbers to acquire is overwhelming. In Section 3.4 we touch upon methods for learning structure from a data base and for adapting the conditional probabilities to incoming cases.

Finally we describe the system *Child*.

3.1 Catching the structure

3.1.1 Family out?

> When I go home at night, I want to know if my family is home before I try the doors. (Perhaps the most convenient door to enter is double locked when nobody is home.) Now, often when my wife leaves the house she turns on an outdoor light. However, she sometimes turns on this light if she is expecting a guest. Also, we have a dog. When

the dog has bowel trouble. Finally, if the dog is in the back yard, I will probably hear her barking, but sometimes I can be confused by other dogs barking.

The first thing to have in mind when organizing a Bayesian model for a decision support system is that its purpose is to give estimates of certainties for events which are *not observable* (or only observable at an unacceptable cost). So, the primary task in model building is to identify these events. We call them *hypothesis events*.

Here we have two hypothesis events, namely *family at home* and *family out*.

Now, the hypothesis events have to be organized into a set of variables. A variable incorporates an exhaustive set of mutually exclusive events. That is, for each variable precisely one of its events is true.

Here it is very easy to organize the events into one variable *F-out?* with states *y* and *n*.

The next thing to have in mind is that in order to come up with a certainty estimate, we should provide some *information channels*. So, the task is to identify the types of achievable information which may reveal something about the state of some hypothesis variable. This is also done by establishing certain variables, *information variables*, such that a piece of information corresponds to a statement about the state of an information variable. Typically, the information will be a statement that a particular information variable is in a particular state; but also more soft statements are allowed.

Here, the information variables are *L-on?* (light on) with states *y* and *n* and *H-bark?* (hear bark) also with states *y* and *n*.

Now it is time to consider the causal structure between the variables. At this stage we need not worry about how information is transmitted through the network. The only thing to worry about is which events have a direct causal impact on other events.

In this example it is clear that *F-out?* has an impact on *L-on?* as well as on *H-bark?*, and that there is no causal relation between *H-bark?* and *L-on?*.

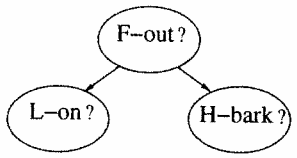


Figure 3.1 A causal structure for *family-out?*

We may stop with the model in Figure 3.1 and start specifying the probabilities $P(F\text{-}out)$, $P(H\text{-}bark? \mid F\text{-}out?)$ and $P(L\text{-}on? \mid F\text{-}out?)$. We will defer the remaining treatment of this example to the section on specification of the probabilities (Section 3.2.4).

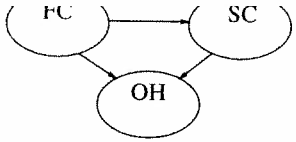


Figure 3.2 An oversimplified structure for the poker game. The variables are *FC* (first change), *SC* (second change), and *OH* (opponent's hand).

3.1.2 A simplified poker game

In this poker game each player receives three cards and is allowed two rounds of changing cards. In the first round you may discard any number of cards from your hand and get replacements from the pack of cards. In the second round you may discard at most two cards. After the two rounds of card changing, I am interested in an estimate of my opponent's hand.

The hypothesis events are the various types of hands in the game. They may be classified in the following way (in increasing rank): nothing special, 1 ace, 2 of the same value, 2 aces, flush (3 of a suit), straight (3 of consecutive value), 3 of the same value, straight flush. Ambiguities are resolved according to rank. This is of course a simplification, but you often have to do so when modelling. The hypothesis events are collected into one hypothesis variable *OH* (opponent's hand) with the classes given above as states.

The only information to acquire is the number of cards the player discards in the two rounds. (By saying so, we again are making an approximation. The information on the cards you have seen is relevant for your opponent's hand. If, for example, you have seen three aces then he cannot have two aces.)

So, the information variables are *FC* (first change) with states *0, 1, 2, 3* and *SC* (second change) with states *0, 1, 2*.

A causal structure for the information variables and the hypothesis variable could be as in Figure 3.2.

However, this structure will leave us with no clue as to how to specify the probabilities.

What we need are variables describing the opponent's hands in the process: the initial hand *OH0* and the hand *OH1* after the first change of cards. The causal structure will then be as in Figure 3.3.

To determine the states of *OH0* and *OH1* we have to produce a classification which is relevant for the determination of the states of the children (*FC* and *OH1*, say). We may let *OH0* and *OH1* have the following states: *nothing special, 1 ace, 2 of consecutive value, 2 of a suit, 2 of the same value, 2 of a suit and 2 of consecutive value, 2 of a suit and 2 of the same value, 2 of consecutive value and 2 of the same value, flush, straight, 3 of the same value, straight flush.*

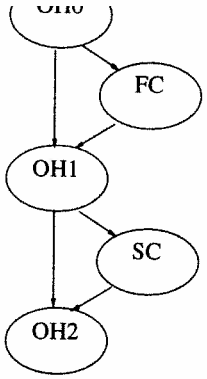


Figure 3.3 A structure for the poker game. The two mediating variables *OH0* and *OH1* are introduced. *OH2* is the variable for my opponent's final hand.

We defer further discussion of the classification to the section on specifying the probabilities (Section 3.2.2).

Variables in a model which are neither hypothesis variables nor information variables are called *mediating variables*. The decision on how to incorporate mediating variables is mainly a question of convenience. Usually mediating variables will ease the acquisition of conditional probabilities and thereby also increase the precision of the model. On the other hand there is a risk of increasing the complexity to a level which may jeopardize performance.

Another point is that it may happen that two variables – A and B – are dependent, but this dependence does not factor through any of the other variables. On the other hand, there is no obvious causal direction on the dependence. This should be taken as an indication that a mediating variable should be introduced as a parent of A and B. The next example illustrates this point.

3.1.3 Insemination

> Six weeks after insemination of a cow there are three tests for the result: blood test (BT), urine test (UT) and scanning (Sc). The results of the blood test and the urine test are mediated through the hormonal state (Ho) which is affected by a possible pregnancy (Pr). (This is a constructed example.)

A model will be like the one shown in Figure 3.4.

For both the blood test and the urine test there is a risk that a pregnancy does not show after six weeks. This is due to the fact that the change in the hormonal state may be too weak. Therefore, given pregnancy, the variables BT and UT are dependent.

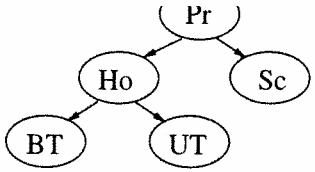


Figure 3.4 A model for test of pregnancy (Pr). Both the blood test (BT) and the urine test (UT) measure the hormonal state (Ho).

If we did not include the mediating variable, the model would be the one shown in Figure 3.5.

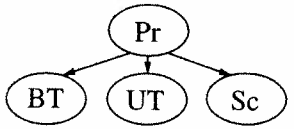


Figure 3.5 The pregnancy model without the *hormonal state* variable.

This model assumes the two tests to be independent given Pr.

If the model in Figure 3.5 is used for diagnosing a possible pregnancy, a negative outcome of both the blood test and the urine test will be counted as two independent pieces of evidence and therefore overestimate the probability for the insemination to have failed. (See Exercise 3.1.)

3.1.4 Simple Bayes models

The first Bayesian diagnostic systems were constructed through the following procedure.

- Let the possible diseases be collected into one hypothesis variable H with prior probability $P(H)$.
- For all information variables I, acquire the conditional probability $P(I \mid H)$ (the likelihood of H given I).
- For any set of findings $f_1, \ldots, f_n$ on the variables $I_1, \ldots, I_n$ calculate the product $L(H \mid f_1, \ldots, f_n) = P(f_1 \mid H)P(f_2 \mid H) \cdots P(f_n \mid H)$. This product is called the *likelihood* for H given $f_1, \ldots, f_n$. The posterior probability for H is calculated as $\mu P(H)L(H \mid f_1, \ldots, f_n)$, where μ is a normalization constant.

The calculations above reflect the simple model shown in Figure 3.6. (See Exercise 3.2.)

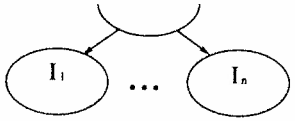


Figure 3.6 A simple Bayes model.

The model assumes that the information variables are independent given the hypothesis variable. As can be seen from the insemination example, the assumption need not hold, and if the model is used anyway, the conclusions may be misleading.

3.1.5 Causality

In the examples presented in the previous section there was no problem in establishing the links and their direction. However, you cannot expect this part of the modelling to always go smoothly.

First of all, causal relations are not always obvious – recall the debate on whether or not smoking causes lung cancer, or whether a person's sex has an impact on their abilities in the technical sciences. Furthermore, causality is not a well understood concept: is a causal relation a property of the real world, or, rather, is it a concept in our minds helping us to organize our perception of the world? We shall, however, not go into the scientific debate on causality and how to discover causal relations.

One point only. Causality has to do with actions where the state of the world is changed: you may, for example, find yourself confronted with two correlated variables A and B, but you cannot determine a direction. If you observe the state of A you will change your belief of B, and vice versa. A good test is then to imagine that some outside agent *fixes* the state of A. If this does not make you change the belief of B, then A is not a cause of B.

On the other hand, if this imagined test indicates a causal arrow in both directions, then you should look for an event which has a causal impact on both A and B. If C is such a candidate, then check whether A and B become independent given C.

3.2 Determining the conditional probabilities

The basis for the conditional probabilities in a Bayesian network can have different epistemological status ranging from well-founded theory over frequencies in a data base to subjective estimates. We shall give examples of each type.

3.2.1 Stud farm

The stallion Brian has sired Dorothy with the mare Ann and sired Eric with the mare Cecily. Dorothy and Fred are the parents of Henry, and Eric has sired Irene with Gwenn. Ann is the mother of both Fred and Gwenn, but their fathers are in no way related. The colt John with the parents Henry and Irene has been born recently; unfortunately, it turns out that John suffers from a life threatening hereditary disease carried by a recessive gene. The disease is so serious that John is displaced instantly, and as the stud farm wants the gene out of production, Henry and Irene are taken out of breeding. What are the probabilities for the remaining horses to be carriers of the unwanted gene?

heritance. The numbers (α, β, γ) are the child's probabilities for *(aa, aA, AA)*.

	aa	*aA*	*AA*
aa	(1, 0, 0)	(0.5, 0.5, 0)	(0, 1, 0)
aA	(0.5, 0.5, 0)	(0.25, 0.5, 0.25)	(0, 0.5, 0.5)
AA	(0, 1, 0)	(0, 0.5, 0.5)	(0, 0, 1)

The geneological structure for the horses is given in Figure 3.7.

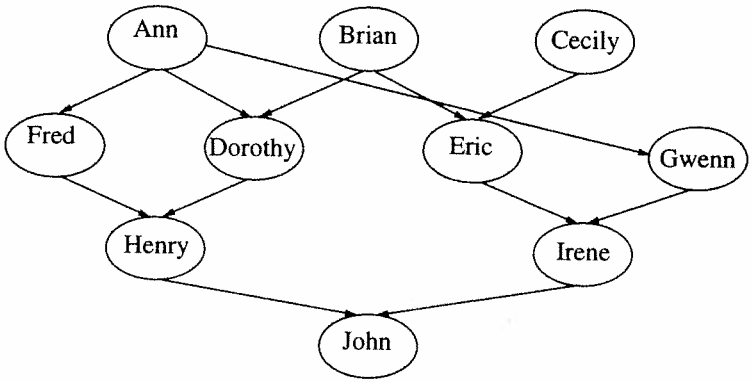


Figure 3.7 Geneological structure for the horses in the stud farm.

The only information variable is John. Before the information on John is acquired he may have three genotypes: he may be sick (aa), a carrier (aA), or he may be pure (AA). The hypothesis events are the genotypes of all other horses in the stud farm.

The conditional probabilities for inheritance are both empirically and theoretically well studied, and the probabilities are as shown in Table 3.1. The inheritance tables could be as Table 3.1. However, for all horses except John we have additional knowledge. Since they are in production they cannot be of type aa. A way to incorporate this would be to build a Bayesian network where all inheritance is modelled in the same way and afterwards enter the findings that all horses but John are not aa. It is also possible to calculate the conditional probabilities directly. If

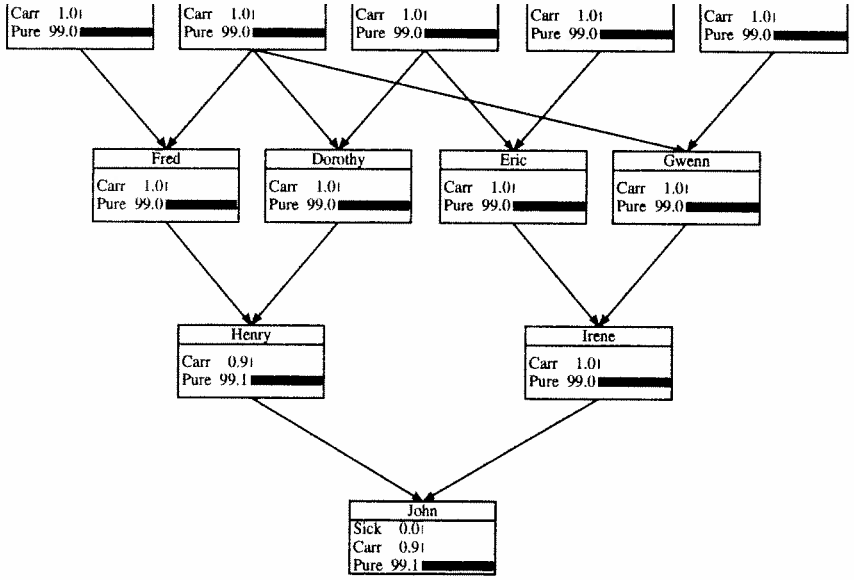


Figure 3.8 The *stud farm* model with initial probabilities. (HUGIN dump.)

we first consider inheritance from parents which may only be of genotype aA or AA, we get Table 3.2.

Table 3.2 $P(\text{child} \mid \text{father, mother})$ when the parents are not sick.

	aA	AA
aA	(0.25, 0.5, 0.25)	(0, 0.5, 0.5)
AA	(0, 0.5, 0.5)	(0, 0, 1)

The table for John is the same as in Table 3.2. For the other horses we know that aa is impossible. This is taken care of by removing the state aa from the distribution and normalizing the remaining distribution. For example $P(\text{child} \mid aA, aA) = (0.25, 0.5, 0.25)$, but since aa is impossible we get the distribution $(0, 0.5, 0.25)$ which is normalized to $(0, 0.67, 0.33)$. The final result is shown in Table 3.3.

In order to deal with Fred and Gwenn we introduce the two unknown fathers, I and K, as mediating variables and assume that they are not sick. For the horses at the top of the network we shall specify prior probabilities. This will be an estimate of the frequency of the unwanted gene, and there is no theoretical way to come up with it. Let us assume that the frequency is so that the prior belief of a horse being a carrier is 0.01.

P(child | father, mother) with aa removed.

	aA	AA
aA	(0.67, 0.33)	(0.5, 0.5)
AA	(0.5, 0.5)	(0, 1)

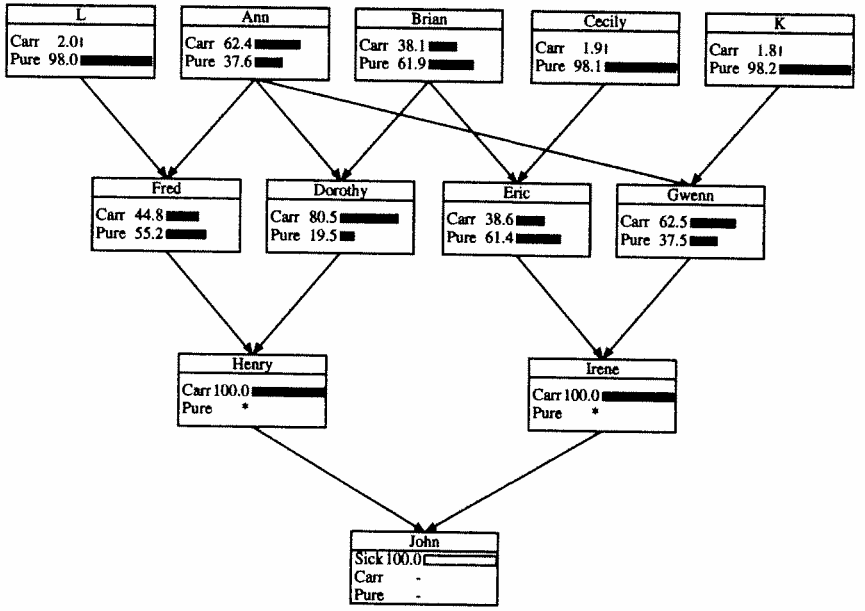


Figure 3.9 *Stud farm* probabilities given that *John* is sick. (HUGIN dump.)

In Figure 3.8 the final model with initial probabilities is shown, Figure 3.9 gives the posterior probabilities given John is aa, and in Figure 3.10 you can see the posterior probabilities with the prior beliefs at the top changed to 0.0001. Note that the sensitivity to the prior beliefs is very small for the horses where the posterior probability for *carrier* is well beyond zero, e.g. Ann and Brian.

3.2.2 Conditional probabilities for the poker game

In the *stud farm* example the conditional probabilities were mainly established through theoretical considerations. This should also be attempted for the model of the poker game developed in Section 3.1.2, but it cannot be carried through entirely.

Consider, for example, $P(FC \mid OH0)$. It is not possible to give probabilities which are valid for any opponent. It is heavily dependent on the opponent's insight, psychology and game strategies. We shall assume the following strategy.

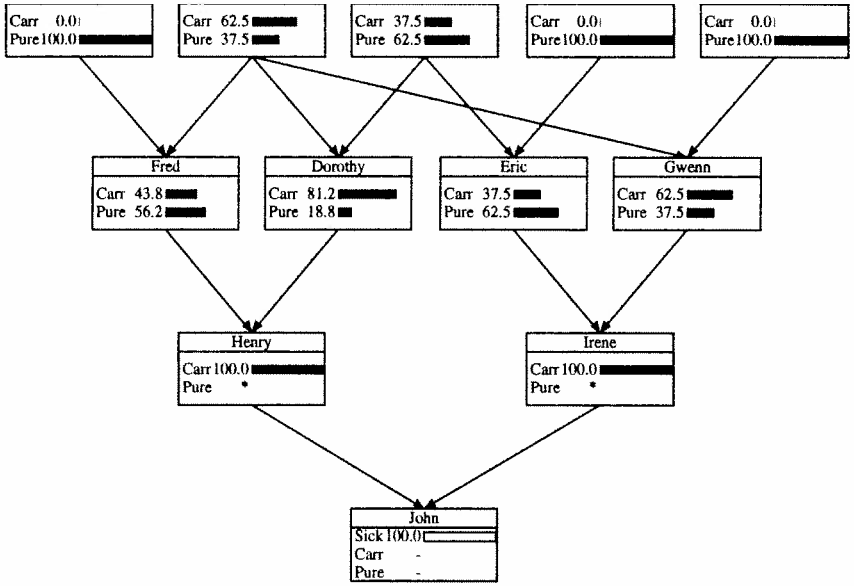


Figure 3.10 *Stud farm* probabilities with prior probabilities for top variables changed to (0.0001, 0.9999). (HUGIN dump.)

If nothing special (*no*), then change 3.

If 1 ace (*1 a*), then keep the ace.

If 2 of consecutive value (*2 cons*) or 2 of a suit (*2 s*) or 2 of the same value (*2 v*) then discard the third card.

If 2 of a suit and 2 of consecutive value, then keep 2 of a suit. (This strategy could be substituted by a random strategy for either keeping 2 of a suit or 2 of consecutive value.)

If 2 of a suit and 2 of the same value or 2 of consecutive value and 2 of the same value, then keep the 2 of the same value,.

If flush (*fl*), straight (*st*), 3 of the same value (*3 v*) or straight flush (*sfl*), then keep it.

Based on the strategy above, a logical link between *FC* and *OH0* is established. Note that the strategy makes the states for combined hands redundant. They play no role, and therefore we remove them.

The strategy for $P(SC \mid OH1)$ is the same except that in the case of *no*, only 2 cards are discarded.

The remaining probabilities to specify are $P(OH0), P(OH1 \mid OH0, FC)$ and $P(OH2 \mid OH1, SC)$.

		(OH0, FC)				
		(no, 3)	*(1 a, 2)*	*(2 cons, 1)*	*(2 s, 1)*	*(2 v, 1)*
	no	0.1583	0	0	0	0
	1 a	0.0534	0.1814	0	0	0
	2 cons	0.0635	0.0681	0.3470	0	0
	2 s	0.4659	0.4796	0.3674	0.6224	0
OH1	*2 v*	0.1694	0.1738	0.1224	0.1224	0.9592
	fl	0.0494	0.0536	0	0.2143	0
	st	0.0353	0.0383	0.1632	0.0307	0
	3 v	0.0024	0.0026	0	0	0.0408
	sfl	0.0024	0.0026	0	0.0102	0

P(*OH0*). The states are (*no, 1 a, 2 cons, 2 s, 2 v, fl, st, 3 v, sfl*).

Through various (approximated) combinatorial calculations the prior probability distribution is found to be

$$P(OH0) = (0.1672, 0.0445, 0.0635, 0.4659, 0.1694, 0.0494, 0.0353, 0.0024, 0.0024)$$

P(*OH1* | *OH0*, *FC*). Due to the logical links between *OH0* and *FC* it is sufficient to consider only nine out of the possible 36 parent configurations, namely *(no, 3), (1 a, 2), (2 cons, 1), (2 s, 1), (2 v, 1), (fl, 0), (st, 0), (3 v, 0), (sfl, 0)*. The last four are obvious. In Table 3.4 the results of approximate combinatorial calculations are given.

The probabilities for the remaining parent configurations may be whatever convenient. So, put, for example, $P(OH1 \mid 3\ v,\ 1) = (1, 0, 0, 0, 0, 0, 0, 0, 0)$.

P(*OH2* | *OH1, SC*). First a table $P(OH2' \mid OH1, SC)$ similar (but not identical in the numbers) to Table 3.4 can be calculated. However, the states of OH2′ are not the ones we are interested in. We are interested in the *value* of the hand and a state like *2 cons* is of no value unless one of them is an ace. Therefore, the probabilities for the states of *OH2′* are transformed to probabilities for *OH2*. For the transformation, the following rules are used:

$$1\ a = 1\ a + \frac{1}{6}(2\ cons + 2\ s)$$

$$no = no + \frac{5}{6}(2\ cons + 2\ s).$$

The probabilities of *2 a* are calculated specifically. The resulting probabilities are given in Table 3.5.

Using a model like the one in Figure 3.3 and with the conditional probability tables specified in this section, we have established a model for assisting a (novice) poker player. However, if my opponent knows that I use the system he may choose

		(OH1, Sc)				
		(no, 2)	(1 a, 2)	(2 cons, 1)	(2 s, 1)	(2 v, 1)
	no	0.5613	0	0.5903	0.5121	0
	1 a	0.1570	0.7183	0.1181	0.1024	0
	2 v	0.1757	0.0667	0.1154	0.1154	0.8838
OH2	2 a	0.0055	0.1145	0.0096	0.0096	0.0736
	fl	0.0559	0.0559	0	0.2188	0
	st	0.0392	0.0392	0.1666	0.0313	0
	3 v	0.0027	0.0027	0	0	0.0426
	sfl	0.0027	0.0027	0	0.0104	0

to change his strategies. His goal is to win rather than to obtain good hands, and he therefore may choose a strategy that makes me overestimate his hand. For instance, it seems a good strategy to discard two cards instead of three in the case of *no*. I will be convinced that he has an ace, and his chances for a good hand are not substantially reduced. We will return to this point in Chapter 6 on decision making.

3.2.3 Transmission of symbol strings

A language L over 2 symbols (**a**, **b**) is transmitted through a channel. Each word is surrounded by the delimiter symbol c. In the transmission some characters may be corrupted by noise and be confused with others.

A five-letter word is transmitted. Give a model which can determine the probabilities for the transmitted symbols given the received symbols.

There are five hypothesis variables $T_1, \ldots, T_5$ with states a and b and five information variables $R_1, \ldots, R_5$ with states a, b, c. Besides, mediating variables for the delimiters before and after the word may be considered. There is a causal relation from T_i to R_i. Furthermore, there may also be a relation from T_i to $T_{i+1} (i = 1, \ldots, 4)$. You could also consider more involved relations from pairs of symbols to symbols, but for now we refrain from that. The structure is given in Figure 3.11.

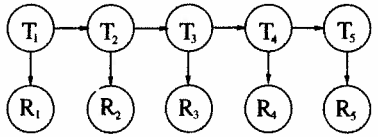


Figure 3.11 A model for symbol transmission. T_i are the symbols transmitted, R_i are the symbols received.

The conditional probabilities can be established through experience. The probabilities $P(R_i \mid T_i)$ will be based on statistics describing the frequencies of confusion. Let Table 3.6 be the result.

transmission.

	$T = a$	$T = b$
$R = a$	0.80	0.15
$R = b$	0.10	0.80
$R = c$	0.10	0.05

Table 3.7 Frequencies of five-letter words in L. The word **abaab** for example has frequency 0.040.

	Last 3 letters							
First 2 letters	**aaa**	**aab**	**aba**	**abb**	**baa**	**bab**	**bba**	**bbb**
aa	0.017	0.021	0.019	0.019	0.045	0.068	0.045	0.068
ab	0.033	0.040	0.037	0.038	0.011	0.016	0.010	0.015
ba	0.011	0.014	0.010	0.010	0.031	0.046	0.031	0.045
bb	0.050	0.060	0.056	0.057	0.016	0.023	0.015	0.023

You may obtain the probabilities $P(T_{i+1} \mid T_i)$ by investigating the five-letter words in L. What is the frequency of the first letter? What is the frequency of the second letter given that the first letter is **a**, etc. You can refine this frequency analysis by also taking the frequency of the words into consideration. Let Table 3.7 be the result of a frequency analysis.

You can calculate the required probabilities from Table 3.7. The prior probabilities for T_1 are (0.5, 0.5). Table 3.8 gives two conditional probabilities.

An alternative model would be to have a hypothesis variable, *Word*, with 32 states and with Table 3.7 as prior probabilities (see Fig. 3.12).

This is manageable because of the small amount of five-letter words over {**a**, **b**}; but if the alphabet had 24 symbols and six-letter words were considered the number of states in *Word* would become intractably large. On the other hand, the model of Figure 3.11 may be too simple to catch the dependencies in Table 3.7. So, the task really is to analyze the table in order to find the simplest structure describing

Table 3.8 Two conditional probabilities for five-letter words in L.

	a	b
a	0.6	0.4
b	0.4	0.6

$P(T_2 \mid T_1)$

	a	b
a	0.4	0.74
b	0.6	0.26

$P(T_3 \mid T_2)$

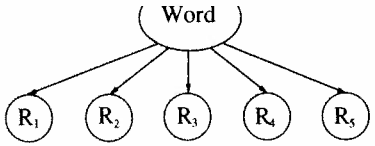


Figure 3.12 An alternative model for symbol transmission. *Word* is the possible transmitted words.

it. There are methods for doing this, and we shall revert to these in Section 3.4.

3.2.4 Family out?

The estimation of the conditional probabilities for the example introduced in Section 3.1.1 is a rather subjective task. Therefore, the model in Figure 3.1 may be changed in order to get better founded probabilities.

P(*F-out?*). I should give an estimate of how often my family is out when I return from work. Out of the five working days a week it only happens once, so I put $P(F\text{-}out?) = (0.2, 0.8)$.

P(*L-on?* | *F-out?*). As a rule the light is on when the family is out. Though I do not recall, they may have forgotten it now and then. Consequently, I put $P(L\text{-}on? \mid F\text{-}out? = y) = (0.99, 0.01)$.

As a rule, the light is off when the family is at home. However, when we expect guests the light is on, and this happens three times a month. We can either include this in the probability or we can add a mediating variable *Exp-g?* to make it explicit. Which one to use is partly a matter of taste and partly a question of how easy the required probabilities would be to estimate. Here, we hide the exception to the rule in the probabilities and put $P(L\text{-}on? \mid F\text{-}out? = n) = (0.1, 0.9)$.

There are several reasons why I can hear barking in the case of my family being at home as well as being out. In order to sort this out we introduce the mediating variable *D-out?* (see Fig. 3.13).

Again, to estimate $P(D\text{-}out? \mid F\text{-}out?)$ several factors are involved. Sometimes the dog is out due to *F-out?*, sometimes due to bowel problems and sometimes just because she wants to be. We introduce a mediating variable *BP?* (bowel problems) with prior probability (0.05, 0.95). Basically I would say that the dog is outside 20% of the time "just because"; so $P(D\text{-}out? \mid F\text{-}out? = n, BP? = n) = (0.2, 0.8)$. I also estimate that in 15% of the cases where my family is out they forget to let the dog out, and in 95% of the cases where the dog has bowel problems they let her out. What if the family is out and the dog has bowel problems? We can say that there is a "background" probability of 0.2 that the dog is out. Out of the remaining 80%, 85% of the time the dog is out due to the family being out, and in the remaining 12% , 95% of the time the dog is out due to bowel problems

Table 3.9 $P(D\text{-}out? = y \mid F\text{-}out?, BP?)$.

	$BP? = y$	$BP? = n$
$F\text{-}out? = y$	0.994	0.880
$F\text{-}out? = n$	0.960	0.200

(see Table 3.9). We shall revert to this line of reasoning in Section 3.3.2 on "noisy or". To estimate $P(H\text{-}bark? \mid D\text{-}out?)$ we may also perform a detailed analysis introducing my neighbour's annoying dog who is always out howling incessantly; but we may also make him implicit in the table such that $H\text{-}bark? = y$ means that I hear something which I interpret as my dog's barking.

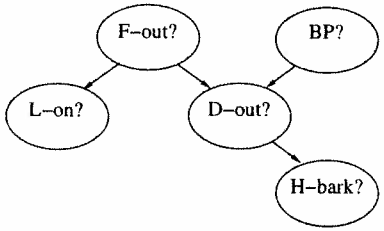


Figure 3.13 The final model for *Family out?*. We have introduced the mediating variable *D-out?* with the additional parent *BP?*.

3.3 Modelling tricks

Much scepticism to Bayesian networks stems from the question "Where do the numbers come from?". As shown in the previous section, they come from many different sources. If you are building a model over a domain where experts actually *do* take decisions based on estimates, why shouldn't you be able to make your Bayesian network estimate at least as well as the experts? You can, for example, use the technique described in Section 2.3.3 to acquire the probabilities from the experts. The acquisition of numbers is of course not without problems, and in this section we give some methods which can help you in this job.

3.3.1 Undirected relations

It may happen that the model must contain dependence relations between variables A, B, C, say; but it is neither desirable nor possible to attach directions on them. (In that case the model is called a *chain graph*. A chain graph is an acyclic graph with both directed and non-directed links, where *acyclic* means that all cycles consist of only non-directed links.) The relation may, for example, be a description of possible

as described in Section 2.2.1 (converging influence).

Let $R(A, B, C)$ describe the relation in numbers from [0,1]. Add a new variable D with two states y and n, and let A, B, C be parents of D (see Fig. 3.14).

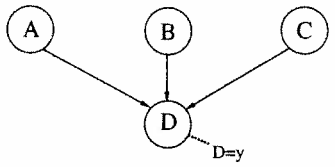


Figure 3.14 A way to introduce undirected relations between A, B and C.

Let $P(D = y \mid A, B, C) = R(A, B, C)$, let $P(D = n \mid A, B, C) = 1 - R(A, B, C)$ and enter the evidence $D = y$.

Example. I have washed two pairs of socks in the washing machine. The washing has been rather hard on them, so they are now difficult to distinguish. However, it is important for me to pair them correctly. To classify the socks I have pattern and colour. A classification model may be like the one in Figure 3.15. The variables S_i have states t_1 and t_2 for the two types, the variables P_i have two pattern types, and the variables C_i have two colour types. The constraint that there are exactly two socks of each type is described in Table 3.10.

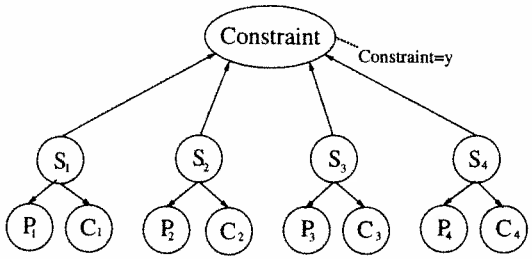


Figure 3.15 A model for classifying pairs of socks.

The situation is more subtle if the relation $R(A, B, C)$ is of a probabilistic nature. If A, B and C have no parents, $R(A, B, C)$ can be a joint probability table. On the other hand, if A has a parent then it is not obvious what $R(A, B, C)$ represents. We shall not deal with this problem but refer the reader to the literature on chain graphs.

3.3.2 Noisy or

When a variable A has several parents you must specify $P(A \mid c^*)$ for each configuration c^* of the parents. If you take the distributions from a data base, the number of cases for each configuration may become too small. Also, the configurations

are the two states of S_1, S_2, S_3, S_4.

S_1	t_1	t_1	t_1	t_1	t_1	t_1	t_1	t_1	t_2	t_2	t_2	t_2	t_2	t_2	t_2	t_2
S_2	t_1	t_1	t_1	t_1	t_2	t_2	t_2	t_2	t_1	t_1	t_1	t_1	t_2	t_2	t_2	t_2
S_3	t_1	t_1	t_2	t_2	t_1	t_1	t_2	t_2	t_1	t_1	t_2	t_2	t_1	t_1	t_2	t_2
S_4	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2
P	0	0	0	1	0	1	1	0	0	1	1	0	1	0	0	0

may be too specific for any expert. You may also be in the situation that you have reasonable estimates of $P(A \mid B)$ and $P(A \mid C)$, but you require $P(A \mid B, C)$. Then you should look for assumptions which reduce the amount of distributions to specify.

Table 3.11 Calculation of $P(D\text{-}out? = y \mid F\text{-}out?, BP?)$.

	$BP? = y$	$BP? = n$
$F\text{-}out? = y$	$1 - 0.8 \cdot 0.05 \cdot 0.15$	$1 - 0.8 \cdot 0.15$
$F\text{-}out? = n$	$1 - 0.8 \cdot 0.05$	$1 - 0.8$

Consider in "Family out?" (Section 3.2.4) the conditional probability

$$P(D\text{-}out? \mid F\text{-}out?, BP?).$$

It was possible to get estimates of $P(D\text{-}out? \mid F\text{-}out?)$ and $P(D\text{-}out? \mid BP?)$, but is there a general way to describe how they combine into $P(D\text{-}out? \mid F\text{-}out?, BP?)$? The following is a way of describing it.

There are three events causing the dog to be outside:

- the "background event" that in 20% of the time the dog is outside "just because";
- *F-out?* which causes the dog to be outside with probability 0.85;
- *BP?* which causes the dog to be outside with probability 0.95.

The above uncertainty can be interpreted in the following way. If any of the causes are present then the dog is outside, unless something has prevented it. In other words, if the family is out then the dog is outside unless they have forgotten to let it out, and there is a 15% chance that they will forget. In the same way there is a 5% chance that some inhibitor prevents the dog from being let out when it has bowel problems and the background event is prevented with probability 0.8.

Now, if we assume that the preventing factors are independent, then the combined probabilities are easy to calculate as one minus the product of the appropriate probabilities for the inhibitors (note that the background event is always a fact). The probabilities are given in Table 3.11.

noisy or.

Let $A_1, \ldots, A_n$ be binary variables listing all the causes of the binary variable B. Each event $A_i = y$ causes $B = y$ unless an inhibitor prevents it, and the probability for that is q_i (see Fig. 3.16).

That is, $P(B = n \mid A_i = y) = q_i$. We assume that *all inhibitors are independent.* Then $P(B = n \mid A_1, A_2, \ldots, A_n) = \prod_{j \in Y} q_j$ where Y is the set of indices for variables in the state y. For example

$$\begin{aligned} P(B = y \mid A_1 = y, A_2 = y, A_3 = \cdots = A_n = n) \\ &= 1 - P(B = n \mid A_1 = y, A_2 = y, A_3 = \cdots = A_n = n) \\ &= 1 - q_1 \cdot q_2. \end{aligned}$$

By assuming "noisy or", the number of probabilities to estimate grows linearly with the number of parents.

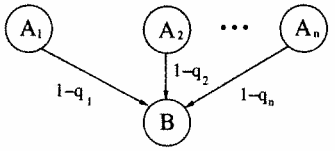


Figure 3.16 The general situation for noisy or. q_i is the probability that the impact of A_i is inhibited.

Note 1. We require $P(B = y \mid A_1 = \cdots = A_n = n)$ to be zero. This may seem to restrict the applicability of the approach. However, as in the example above, if $P(B = y) > 0$ when none of the causing events in the model are on, then introduce a background event which is always on.

Note 2. The complementary construction to noisy or is called *noisy and*. A set of causes shall all be on in order to have an effect. However, the causes have random inhibitors which are mutually independent.

Note 3. The noisy or-gate can be modelled directly without performing the calculations (see Fig. 3.17). This highlights the assumptions behind the noisy or-gate. If a cause is on, then its effect may be prevented by an inhibitor, and the probabilities for the inhibitors to be present are independent.

3.3.3 Causal independence

Let $C_1, \ldots, C_n$ be a list of variables all of which are causes of A. If you wish to specify $P(A \mid C_1, \ldots, C_n)$ you might have a very large knowledge acquisition task ahead of you. Usually it will only be possible to obtain partial specifications like $P(A \mid C_i)$, and in the noisy or case, for example, you have to add some assumptions on how the various impacts on A combine. Often you would have some kind of feeling that the causes act independently on A, but this is not a particularly well defined term.

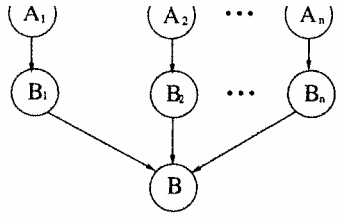


Figure 3.17 Direct modelling of a noisy or-gate. $P(B_i \mid A_i)$ is the original $P(B \mid A_i)$, and $P(B \mid B_1, \ldots, B_n)$ is logical or.

Headache. Headache *(Ha)* may be caused by fever *(Fe)*, hangover *(Ho)*, fibrositis *(Fb)*, brain tumor *(Bt)* and other causes *(Ot)*, and you may choose to soothe it with aspirin *(As)*. (We ignore the effect aspirin has on fever.) Let *Ha* have the states *no, mild, moderate, severe*. The various causes support each other in the effect. If for example $Ho = y$ or $Fb = y$ are present then they may yield a *mild Ha*, but if they are both present then the *Ha* would be *moderate*. Furthermore, if also $As = y$ then *Ha* may drop to *no* or *mild*. Although the various parents of *Ha* combine in a rather involved manner we still have the feeling that the causes' impacts are independent. This kind of independence can be described as follows: if the headache is at level l and we add an extra cause for headache then the result is a headache at level q independently of how the initial state has been caused.

More formally it means the following.

Let $C_1, \ldots, C_n$ be the parents of A. $C_1, \ldots, C_n$ are causally independent if the following holds for each parent configuration $(c_1, \ldots, c_n)$ and all i: if at some time A is in state a and the state of C_i is changed to c_i' then the probability distribution of A afterwards is a function of a, c_i and c_i' alone.

If we add auxiliary variables C_i' and A_i' the condition above is reflected in the conditional independences which can be seen in Figure 3.18.

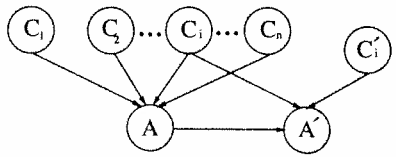


Figure 3.18 A' is independent of $C_1, \ldots, C_{i-1}, C_{i+1}, \ldots, C_n$ given A and C_i.

It can now be seen that aspirin is not causally independent of the causes for headache according to this definition. Suppose that aspirin has been taken without any reason and then a fever arises. Due to the aspirin, the headache will not reach the level it would have reached otherwise. On the other hand, when all present causes for headache are taken into account then the effect of aspirin is only a function of the state of *Ha*.

probabilities of the kind $P(Ha - Fe \mid Ha, Fe)$, which shall be estimated for the following rule: if Ha is in state h and Fe changes from *no* to f then the probability distribution for headache is $P(Ha - Fe \mid h, f)$. We can then combine these probabilities as shown in Figure 3.19.

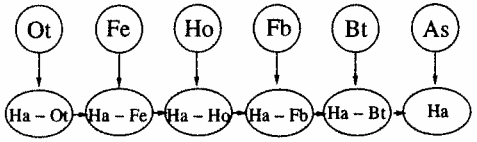


Figure 3.19 A network modelling the multiple cause relation for headache (Ha). Ot,Other causes; Fe, Fever; Ho, Hangover; Fb, Fibrositis; Bt, Brain tumor; As, Aspirin.

Note that the chain is started by the background variable Ot and the aspirin variable finishes the chain. The order of the intermediate variables is unimportant.

3.3.4 Divorcing

Noisy or, as well as causal independence represent simplifying assumptions to use when the space of parent configurations is too large. Both methods can be seen as special cases of a more general technique called *divorcing*.

The set of parents $A_1, \ldots, A_i$ for B is divorced from the parents $A_{i+1}, \ldots, A_n$ for B by introducing a mediating variable C, making C a child of $A_1, \ldots, A_i$ and a parent of B (see Fig. 3.20).

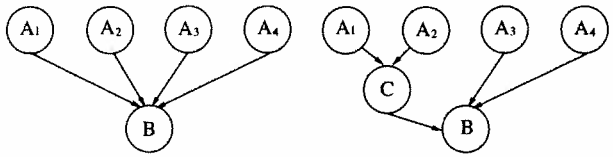


Figure 3.20 A_1 and A_2 are divorced from A_3 and A_4 by introducing the variable C.

If all variables in Figure 3.20 are ternary, you will have to specify 81 distributions before divorcing and only 36 distributions after divorcing. Even if C turns out to require five states, the saving is considerable.

The assumption behind divorcing is the following (with reference to Fig. 3.20).

The set of configurations of (A_1, A_2) can be partitioned into the sets $c_1, \ldots, c_m$ such that whenever two configurations (a_1, a_2) and (a_1', a_2') are elements in the same c_i, then $P(B \mid a_1, a_2, A_3, A_4) = P(B \mid a_1', a_2', A_3, A_4)$. The divorcing variable then has $c_1, \ldots, c_m$ as states.

Example. To help the bank decide when a customer applies for a mortgage on a house, the customer is asked to fill in a form giving information on various economic

will get their money back.

The information is the following: type of job, yearly income, other financial commitments, number and type of cars in the family, size and age of the house, price of the house, number of previous addresses during the last five years, number of children in the family, number of divorces, and number of children not living in the family.

In principle each slot in the form represents a variable with a causal impact on the variable *money back?*. However, the information can be partitioned into variables describing the economic potentials of the applicant, variables describing the stability of the applicant, and variables describing the security of the mortgage. So, the many parents can be divorced by three variables.

3.4 Learning

By *learning* we understand semi-automatic methods using experience gained to construct or modify a model. Learning can be divided into *qualitative* and *quantitative* learning. Qualitative learning concerns the structure of the model (i.e. the network) and quantitative learning is the specification of conditional probabilities.

There are two types of learning situations. *Batch learning* is the situation where a database of cases is used to establish a model. *Adaptation* is the process of modifying a model successively when new cases are gathered.

3.4.1 Batch learning

In *transmission of symbol strings* (Section 3.2.3) a model was used for the relations between the letters in the words transmitted (see Fig. 3.21), and the table of frequencies (Table 3.7) was used to determine the probabilities for the model. We shall call this model M_{Simp}. However, there may be other models, and we need some means of evaluating these possible models. There are two matters to consider.

- How well can the original table be reconstructed from the model?
- How much space does the model require?

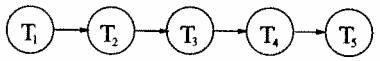


Figure 3.21 The Bayesian network, M_{Simp}, used in Section 3.2.3 for modelling the relation between the letters in the words transmitted

The chain rule (Theorem 2.1) applied to M_{Simp} yields the joint probability table

$$P^*(T_1, T_2, T_3, T_4, T_5) = P(T_1)P(T_2 \mid T_1)P(T_3 \mid T_2)P(T_4 \mid T_3)P(T_5 \mid T_4).$$

The result is shown in Table 3.7. A direct comparison of Table 3.7 and Table 3.7

First 2 letters	Last 3 letters							
	aaa	**aab**	**aba**	**abb**	**baa**	**bab**	**bba**	**bbb**
aa	0.016	0.023	0.018	0.021	0.044	0.067	0.050	0.061
ab	0.030	0.044	0.033	0.041	0.011	0.015	0.012	0.014
ba	0.010	0.016	0.012	0.014	0.029	0.045	0.033	0.041
bb	0.044	0.067	0.059	0.061	0.016	0.023	0.017	0.021

shows some differences, and we may or may not say that M_{Simp} is a sufficiently accurate representation of Table 3.7.

3.4.2 Distance measures

To compare a "true" distribution with an approximation we need a measure of distance between distributions. Let P be a "true" distribution, and let P^* be another distribution over *Word*. Two distance measures are often used.

The *euclidean distance* is

$$\text{Dist}_Q(P, P^*) = \sum_{w \in Word} (P(w) - P^*(w))^2.$$

Cross entropy is

$$\text{Dist}_L(P, P^*) = -\sum_{w \in Word} P(w) \log \frac{P(w)}{P^*(w)}.$$

Note that cross entropy is not symmetric in P and P^*.

The two distance measures have a theoretical foundation, namely *scoring for predictions*: you predict the next word to be transmitted in the form of a probability distribution. When the next word is known, you are penalized by a score such that the penalty is minimal if you had predicted the actual word with probability one, and it is maximal if you assigned probability zero to the actual word.

Let $\text{Act}_w(Word)$ denote the table consisting of zeros except at the place for the actual word, w (where the value is one), and let $P^*(Word)$ be the predicted probabilities. The *quadratic scoring rule* (also called the *Brier scoring rule*) is

$$\text{QS}(w, P^*) = \sum_{x \in Word} (\text{Act}_w(x) - P^*(x))^2 = 1 - 2P^*(w) + \sum_{x \in Word} P^*(x)^2.$$

If you predict a large number of times according to the distribution P^* while the true distribution is P, then the average score will be

$$\begin{aligned} \text{AvQS}(P, P^*) &= \sum_{w \in Word} P(w)\text{QS}(w, P^*) \\ &= 1 - 2 \sum_{w \in Word} P(w)P^*(w) + \sum_{x \in Word} P^*(x)^2. \end{aligned}$$

ing a minimal score is the true one (see Exercise 3.17). A strictly proper scoring rule has as a consequence that you are punished if you do not forecast according to your belief (a tempting behaviour if you work with politics). The *distance* between the true distribution P and the estimated distribution P^* is then defined as the difference between the score for the two distributions:

$$\text{Dist}_Q(P, P^*) = \text{AvQS}(P, P^*) - \text{AvQS}(P, P) = \sum_{w \in Word} (P(w) - P^*(w))^2.$$

In our example the true distribution is Table 3.7, and the one used for forecasting is Table 3.7. We get

$$\text{Dist}_Q(P, P^*) = 0.000337.$$

It is more or less up to you to decide how large a distance to accept and fix a threshold. For our tables we would say that the difference between the probabilities should not be larger than 0.004. So, a reasonable threshold would be $32(0.004)^2 = 0.000512$, yielding M_{Simp} acceptable.

Another example of a scoring rule is *logarithmic score:*

$$\text{LS}(w, P^*) = -\log P^*(w).$$

Note that the LS score is extreme if you have assessed a possible event to be impossible. The logarithmic scoring rule is also strictly proper, and the distance measure derived from it is cross entropy.

3.4.3 Search for possible structures

We look for Bayesian networks which represent a joint probability table within an acceptable distance from $P(Word)$; this is not sufficient, though. It may happen that several models are acceptable, and bearing in mind that we look for simple models we also have to take the size of the model into account.

Let M be a Bayesian network with variables U. For each variable A with parents $\text{pa}(A)$ we define $\text{Sp}(A)$ to be the number of entries in $P(A \mid \text{pa}(A))$, and the size is

$$\text{Size}(M) = \sum_{A \in U} \text{Sp}(A).$$

For example, $\text{Size}(M_{\text{Simp}}) = 18$.

To take care of the trade-off between size and distance we define an *acceptance measure:*

$$\text{Acc}(P, M^*) = \text{Size}(M^*) + k\text{Dist}(P, P^*),$$

where P^* is the joint probability table for U determined by M^*, and k is a positive real number.

In the problem with transmission of symbol strings we use $\text{Dist}_Q(P, P^*)$ and choose $k = 10\,000$, and we will work with a distance threshold of 0.0005. The task is to determine an acceptable Bayesian network which minimizes Acc.

In principle, we shall investigate all possible DAGs over the variables T_1, T_2, T_3, T_4, T_5. However, there are too many of them; we therefore add some structure

link from T_i to T_j is only allowed if $i < j$.

We start with the largest model M_{Max} meeting the structure constraint. It is shown in Figure 3.22.

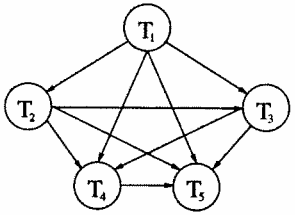


Figure 3.22 The Model M_{Max}, the largest model meeting the constraint that a link from T_i to T_j is only allowed if $i < j$.

Let $P(Word \mid M_{\text{Max}})$ denote the distribution determined by M_{Max}. From the calculation below we see that $P(Word \mid M_{\text{Max}}) = P(Word$:

$$\begin{aligned} P(Word) &= P(T_1, T_2, T_3, T_4, T_5) \\ &= P(T_5 \mid T_1, T_2, T_3, T_4)P(T_1, T_2, T_3, T_4) \\ &= P(T_5 \mid T_1, T_2, T_3, T_4)P(T_4 \mid T_1, T_2, T_3)P(T_1, T_2, T_3) \\ &= P(T_5 \mid T_1, T_2, T_3, T_4)P(T_4 \mid T_1, T_2, T_3)P(T_3 \mid T_1, T_2)P(T_2 \mid T_1) \\ &\quad P(T_1) \\ &= P(Word \mid M_{\text{Max}}). \end{aligned}$$

Therefore M_{Max} is below the threshold and $\text{Acc}(P, M_{\text{Max}})$ is calculated to be 62.

There are 2^{10} DAGs to investigate. This is a heavy task, but if we follow a procedure where we – starting with M_{Max} – successively delete links, we need not remove further when a model is rejected. The result of this search is that the model M_{Min} in Figure 3.23 is the best one.

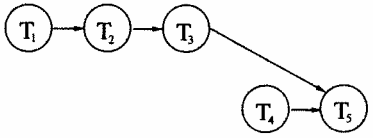


Figure 3.23 M_{Min}. The best model for $P(Word)$.

The tables for M_{Min} are given in Table 13, and the joint probability table determined by M_{Min} is shown in Table 3.14.

We have $\text{Acc}(P, M_{\text{Min}}) = 20.14$ and $\text{Acc}(P, M_{\text{Simp}}) = 21.37$.

	T_1	
T_2	a	b
a	0.6	0.4
b	0.4	0.6
$P(T_2 \mid T_1)$		

	T_2	
T_3	a	b
a	0.25	0.75
b	0.75	0.25
$P(T_3 \mid T_2)$		

	T_3	
T_4	a	b
a	(0.45, 0.55)	(0.4, 0.6)
b	(0.5, 0.5)	(0.4, 0.6)
$P(T_5 \mid T_3, T_4)$		

3.4.4 Statistical methods

In the considerations so far we assumed the distribution from the database to be the true one, and the task was to find a compact representation approximating it. Usually it is too bold an assumption to consider the database as a true distribution. It is more correct to consider a database as a *sample* from an unknown true distribution. Table 3.7 may, for example, be based on 1,000 words. This means that although M_{Min} is closer to P, then it may still be that P is sampled from M_{Simp}.

Let DB be a database of cases, and let $\mathbb{M}$ be a set of models. What you would try to do is to maximize $P(M \mid \text{DB})$ for $M \in \mathbb{M}$. $P(M \mid \text{DB})$ is a rather awkward probability to calculate. However, Bayes' rule (2.3) can help us:

$$P(M \mid \text{DB}) = \frac{P(\text{DB} \mid M)P(M)}{P(\text{DB})}.$$

Since $P(\text{DB})$ is independent of M it plays no role when determining the maximum. If there is no prior knowledge of the probabilities of the models, we will assume them to be equally likely. Hence the likelihood of M, $P(\text{DB} \mid M)$, can play the same role as distance in the search for a good model. Either you can look for a model of maximal likelihood or you can balance the likelihood with size.

Still, there are many problems. First of all, although there is only a finite number of possible structures, each structure has a continuum of possible tables, and among them we have to find the ones maximizing the probability of DB. Also, very often the database is not a list of cases for which the states of all variables are known. You may have many missing values. It may also happen that you have several databases over different overlapping sets of variables and with different numbers of cases. To make a long story short, the calculation of likelihoods is not a trivial task, and we shall not go into it. In Section 3.7 some references are given.

3.4.5 Adaptation

When a system is at work you repeatedly get new cases, and you would like to learn from these cases. The situation may be that you are pretty certain on the structure of the network, however, the conditional probabilities are dependent on a context which varies from place to place, and you want to build a system which automatically adapts to the particular context in which it is placed. The situation may also be that you have consulted several experts during the construction of the system, and they have not agreed upon the quantitative part of the network. So, the

First 2 letters	Last 3 letters: aaa	aab	aba	abb	baa	bab	bba	bbb
aa	0.017	0.021	0.019	0.019	0.045	0.068	0.045	0.068
ab	0.034	0.040	0.037	0.038	0.010	0.015	0.010	0.015
ba	0.011	0.014	0.010	0.010	0.031	0.045	0.030	0.045
bb	0.051	0.062	0.057	0.057	0.015	0.023	0.015	0.023

conditional probabilities are uncertain. This type of uncertainty is called *second-order uncertainty*. Second-order uncertainty calls for an automatic way of adapting the conditional probabilities to the real world as it presents itself through the cases.

In Figure 3.24 the variable A is directly influenced by B and C, and the strength is modelled by $P(A \mid B, C)$. The uncertainty in $P(A \mid B, C)$ may be modelled explicitly by introducing an extra parent, T, for A (Fig. 3.24(b)). The variable T can be considered as a type variable, for example types of context or different experts' assessments. To reflect credibility of the experts or frequencies of the context types, a prior distribution $P(T)$ is given.

When a case is entered to the network, the propagation will yield a new distribution $P^*(T)$, and we may say that the change of the distribution for T reflects what has been learnt from the case. $P^*(T)$ can now be used as a new prior distribution. All variables whose table is dependent on the context shall be children of T.

If the uncertainty of the conditional probabilities cannot be modelled explicitly as above, statistical methods can be used. Each entry in a table for a network is a parameter of the model, and the statistical task is to modify the estimates of the parameters gradually with the cases entered. This is an intractable task unless some assumptions on the dependencies between the parameters are added. For the situation above, the dependence is modelled through the T-variable.

Two simplifying assumptions are often used. *Global independence* says that the second-order uncertainty for the various variables is independent.

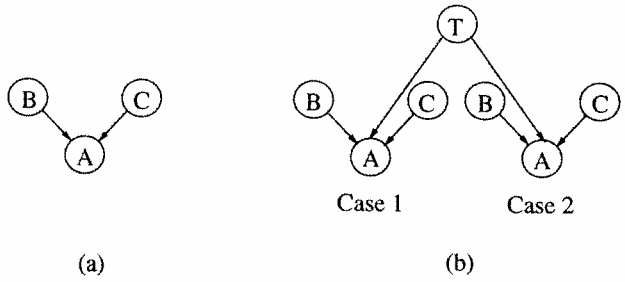


Figure 3.24 Adaptation through a type variable T. The distribution of T is updated by *Case 1* and used in the next case.

Local independence says that the uncertainty of the distributions for different parent configurations are independent. To be more precise, let (a_i, c_j) and (a_i', c_j') be different configurations, then the (second-order) uncertainty on $P(A \mid a_i, c_j)$ is independent of the (second-order) uncertainty on $P(A \mid a_i', c_j')$.

Consider $P(A \mid B, C)$, and let all variables be ternary. Under the assumption of global and local independence we may now think of $P(A \mid b_i, c_j) = (x_1, x_2, x_3)$ as a distribution established through a number of past cases where (B, C) were in state (b_i, c_j). We can then express our certainty of the distribution by a fictitious sample size s, the larger the sample size the smaller the second-order uncertainty. So, behind the distribution we have a table $(n_1, n_2, n_3) = (sx_1, sx_2, sx_3)$.

When a new case arrives with (B, C) in state (b_i, c_j) and with A, for example, in state a_1, then n_1 and s are counted up by one, yielding a new distribution:

$$(x_1^*, x_2^*, x_3^*) = \left(\frac{n_1 + 1}{s + 1}, \frac{n_2}{s + 1}, \frac{n_3}{s + 1}\right).$$

This scheme only works if the state of A as well as the states of its parents are known. In general we may anticipate that the provided evidence e may leave uncertainty on both the state of A and of its parents.

Let $P(b_i, c_j \mid e) = x$ and $P(A \mid b_i, c_j, e) = (y_1, y_2, y_3)$. A simple approach is to distribute the probability mass x over $P(A \mid b_i, c_j)$ according to the current distribution (y_1, y_2, y_3). Since

$$(y_1 x, y_2 x, y_3 x) = P(A \mid b_i, c_j, e) P(b_i, c_j \mid e) = P(A, b_i, c_j \mid e)$$

we have

$$(n_1^*, n_2^*, n_3^*) = (n_1 + P(a_1, b_i, c_j \mid e), n_2 + P(a_2, b_i, c_j \mid e), n_3 + P(a_3, b_i, c_j \mid e)).$$

Note that the sample size is counted up by $P(b_i, c_j \mid e)$.

This scheme is known as *fractional updating*. Unfortunately, the scheme has a serious drawback, namely that it tends to overestimate the count up of s, thereby overestimating our certainty of the distribution. Assume, for example, that $e = \{B = b_i, C = c_j\}$. Then the case tells us nothing about $P(A \mid b_i, c_j)$, but nevertheless fractional updating will add a count of one to s and take it as a confirmation of the present distribution.

A statistically correct updating is intractable. However, an approximation of it can be performed. It does not have the same drawback as mentioned for fractional updating. Essentially, the distribution is updated as in fractional updating, however, the sample size is modified in a different way. We shall not go into that, but in Section 3.7 some references are given.

3.5 Child

The Great Ormond Street Hospital in London (GOS) specializes in child diseases, and it acts as a regional center for the South-East of England. Whenever a "blue baby" is born in the region, the paediatrician calls the 24 hour telephone service at

on information provided by the calling paediatrician. The clinician then decides whether or not to transfer the baby to GOS.

To help the clinician, a Bayesian network has been constructed. In fact two networks are constructed, a *subjective* network (*SN*) and a *batch learned* network (*BN*).

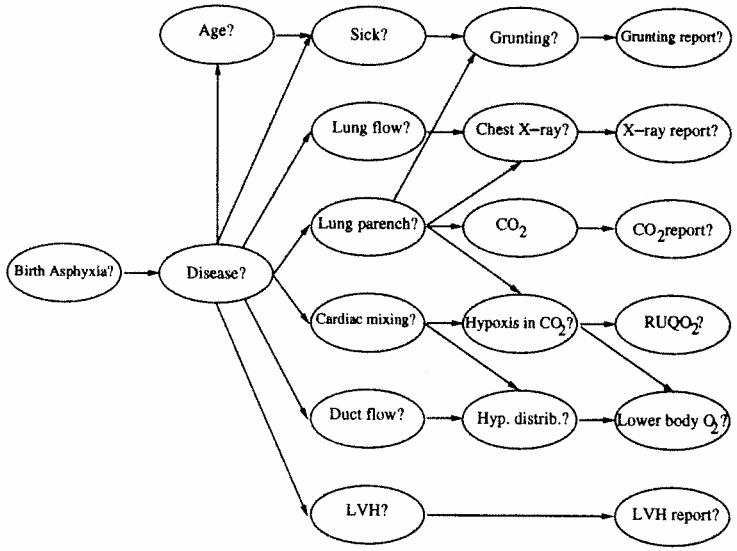


Figure 3.25 The Bayesian network *SN* for congenital heart disease. It is the experts' subjective model. *Disease?* is the only hypothesis variable; *Age?, Birth Asphyxia?, Sick?*, and the rightmost variables are information variables.

SN, which is shown in Figure 3.25, was established through intensive dialogue with the GOS specialists on congenital heart diseases. They provided the model and the conditional probabilities. These probabilities were given together with an estimate of their second-order uncertainty. A set of past cases with known diagnosis was used to adapt the probabilities.

BN was based on 151 cases with established disease, and the resulting model is shown in Figure 3.26. Due to the way batch learning is performed, *BN* can only contain variables from the data base. That is, *BN* does not contain mediating variables.

The two models have been tested in 87 cases (different from the 151 learning cases). The performance of the models was measured through the quadratic scoring rule as well as the logarithmic scoring rule. The conclusion was that both models performed at a level similar to that of the clinicians at GOS, and *SN* performed slightly better than *BN*.

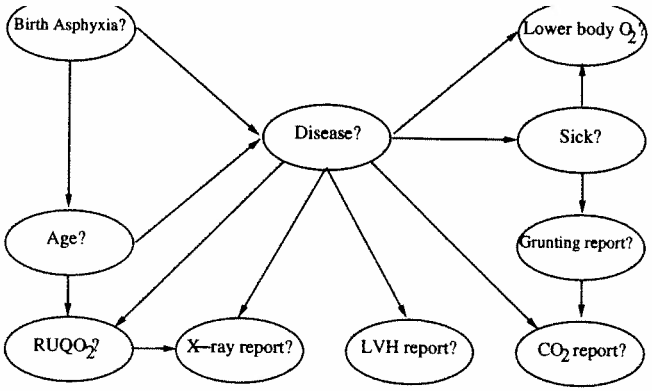


Figure 3.26 The batch learned model for congenital heart disease. The model has no mediating variables.

3.6 Summary

Types of variables when building a Bayesian network model

Hypothesis variables. Variables with a state that is asked for. They are, however, either impossible or too costly to observe.

Information variables. Variables which can be observed.

Mediating variables. Variables introduced for a special purpose. It may be to reflect properly the independence properties in the domain, it may be to facilitate the acquisition of conditional probabilities, it may be to reduce the amount of distributions to acquire for the network, or it may be for other purposes.

Warning: it is tempting to introduce mediating variables in order to have a more refined model of the domain, however, if they do not serve any other purpose you should get rid of them. They jeopardize performance.

Acquiring conditional probabilities

Theoretically well-founded probabilities as well as frequencies and purely subjective estimates can be used for the same network.

If the amount of distributions is too large for a reasonable estimation, simplifying assumptions can reduce it.

Noisy or: Let B have the parents $A_1, \ldots, A_n$ (all variables binary). Suppose that $A_i = y$ causes $B = y$ unless it is inhibited by an inhibitor Q_i which is active with probability q_i. Assume that the inhibitors are independent. Then

$$P(B = n \mid a_1, \ldots, a_n) = \prod_{j \in Y} q_j,$$

where Y is the set of indices for the states y.

Causal independence. Let B have the parents $A_1, \ldots, A_n$. $A_1, \ldots, A_n$ are causally independent with respect to B if the following holds for each configuration $(a_1, \ldots, a_n)$

and all i. If at some time, B is in state b and the state of A_i is changed from a_i to a_i', then the probability distribution of B afterwards is a function of b, a_i and a_i' alone.

Divorcing. Let B have the parents $A_1, \ldots, A_n$. Assume that the set of configurations of $(A_1, \ldots, A_i)$ can be partitioned into the sets $c_1, \ldots, c_m$ such that whenever two configurations a^* and a_1^* of $(A_1, \ldots, A_n)$ are elements in the same c_i, then

$$P(B \mid a^*, A_{i+1}, \ldots, A_n) = P(B \mid a_1^*, A_{i+1}, \ldots, A_n).$$

Then $A_1, \ldots, A_j$ can be divorced from $A_{i+1}, \ldots, A_n$ by introducing a mediating variable C with states $c_1, \ldots, c_m$, making C a child of $A_1, \ldots, A_i$ and a parent of B.

Batch learning

Let U denote the set of configurations over a universe of variables. Let P be a "true" distribution over U taken from a data base of cases, let M^* be a candidate Bayesian network for P, and let P^* be the distribution determined by M^*.

The euclidean distance.

$$\mathrm{Dist}_Q(P, P^*) = \sum_{x \in U} (P(x) - P^*(x)).$$

The cross entropy distance.

$$\mathrm{Dist}_L(P, P^*) = -\sum_{x \in U} P(x) \log \frac{P(x)}{P^*(x)}.$$

Both distance measures are based on strictly proper scoring rules.

Size measure.

$$\mathrm{Size}(M^*) = \sum_i \mathrm{Sp}(A_i),$$

where $\mathrm{Sp}(A)$ is the number of entries in $P(A \mid \mathrm{pa}(A))$.

Acceptance measure.

$$\mathrm{Acc}(P, M^*) = \mathrm{Size}(M^*) + k\mathrm{Dist}(P, P^*).$$

General search method. Choose a threshold t for $\mathrm{Dist}(P,P^*)$, and a k for $\mathrm{Acc}(P,M^*)$. Among the models with distance to P less than t choose one of minimal $\mathrm{Acc}(P, M^*)$.

Heuristics for searching through the models.

Partition the variables into sets $\boldsymbol{Z_1}, \ldots, \boldsymbol{Z_m}$, and consider only models with links from elements in $\boldsymbol{Z_i}$ to elements in $\boldsymbol{Z_j}$, with $i \leq j$. Start with the maximal model and delete one link at a time (breadth-first search). Whenever a model M^* with a distance beyond the threshold t is reached, we need not consider submodels of M^*.

Adaptation

Let A be a variable with parents $\mathrm{pa}(A)$. Second-order uncertainty is uncertainty on the conditional probability table $P(A \mid \mathrm{pa}(A))$. Adaptation consists of using the incoming cases to reduce second-order uncertainty.

Adaptation through type variables. The second-order uncertainty can be characterized as uncertainty on which table out of $\boldsymbol{t_1}, \ldots, \boldsymbol{t_m}$ is the correct one for $P(A \mid \mathrm{pa}(A))$.

Add a type variable T with states $t_1, \ldots, t_m$ and with A as child. The prior probability $P(t_1, \ldots, t_m)$ reflects your belief in the various tables. Let $P(A \mid \mathrm{pa}(A), t_i) = \mathbf{t_i}$.

Whenever a case e has been processed, the probability $P(t_1, \ldots, t_m \mid e)$ is achieved, and it can be used as a prior probability distribution for the next case.

Fractional updating. Assume that the second-order uncertainty obeys both the *global* and the *local independence* requirement. Global independence means that the second-order uncertainty for the various variables is independent. Local independence means that the second-order uncertainty of $P(A \mid a^*)$ and $P(A \mid a_1^*)$ are independent for different configurations a^* and a_1^* of $\mathrm{pa}(A)$.

For each parent configuration a^*, choose a fictitious sample size n, expressing the present certainty of $P(A \mid a^*)$. This yields a fictitious sample size $n_a = nP(a \mid a^*)$ for the configuration (a, a^*).

When a case e has been processed it yields $P(a, a^* \mid e)$. Add $P(a, a^* \mid e)$ to n_a. Thereby the sample size is increased by $P(a^* \mid e)$.

Warning: fractional updating reduces the second-order uncertainty too fast.

3.7 Bibliographical notes

The *Family out?* example is borrowed from Charniak (1991). Simple Bayes was used by de dombal et al. (1972). Noisy or was first described by Pearl (1986b); the modelling of causal independence presented here is suggested by Heckerman (1993); divorcing was used in MUNIN Andreassen et al. (1989). Exercise 3.16 is based on Cooper (1990). Chain graphs are treated in depth by Lauritzen (1996).

Learning has a long history in statistics. Edwards & Havranek (1985) introduce a model selection procedure well suited for Bayesian networks. The relation to Bayesian networks was established through work in the early 1990s (Fung & Crawford 1990, Spiegelhalter & Lauritzen 1990a, Dawid & Lauritzen 1993, Cooper & Herskovits 1992). The learning method presented in this chapter is a simplification of the method in BIFROST (Højsgaard & Thiesson 1995). An improved version of fractional updating is presented in Spiegelhalter & Lauritzen (1990), and a further analysis with experiments can be found in Spiegelhalter & Cowell (1992). Buntine (1994) gives an overview of the state of the art of learning with graphical models. Also Heckerman et al. (1994) is a good paper for further studies.

The various versions of Child are documented in Franklin et al. (1989), Franklin et al. (1991), and in Lauritzen et al. (1994).

	$Pr = y$	$Pr = n$
$Ho = y$	0.9	0.01
$Ho = n$	0.1	0.99

	$Ho = y$	$Ho = n$
$BT = y$	0.7	0.1
$BT = n$	0.3	0.9

	$Ho = y$	$Ho = n$
$UT = y$	0.8	0.1
$UT = n$	0.2	0.9

	$Pr = y$	$Pr = n$
$Sc = y$	0.9	0.01
$Sc = n$	0.1	0.99

Exercises

Exercise 3.1 ^{H} Consider the insemination example from Section 3.1.3. Let the probabilities be as Table 3.15. ($Ho = y$ means that hormonal changes have taken place) $P(Pr) = (0.87, 0.13)$.)

(i) What is $P(Pr \mid BT = n, UT = n)$?

(ii) Calculate $P(BT \mid Pr)$ and $P(UT \mid Pr)$ and use them for a simple Bayes model. What is $P(Pr \mid BT = n, UT = n)$ in this model?

Exercise 3.2 Show that the procedure described in Section 3.1.4 is equivalent to updating in the model in Figure 3.6.

Exercise 3.3 Consider the stud farm example in Section 3.2.1 and let the prior probability for aA be 0.005.

(i) Add to the model the frequency 0.001 for mutation of the gene from A to a.

(ii) Construct a model for the situation in part (i), but for a recessive gene borne by the female sex chromosome. (Note that horses with the disease are taken out of production.)

(iii) ^{H} Copy *Stud farm* from the HUGIN diskette and modify the model according to your answers for (i) and (ii).

Exercise 3.4 ^{H} Consider the transmission example from Section 3.2.3.

(i) From Table 3.7 calculate the remaining conditional probabilities for the model in Figure 3.11.

(ii) Implement the model in HUGIN.

(iii) The sequence *baaca* is received. What is the most probable symbols transmitted according to the model in Figure 3.11? What is the most probable word?

(iv) What is the most probable word according to the model in Figure 3.12.

Exercise 3.5 ^{H} Consider the simplified poker game in Section 3.1.2.

(i) Implement the system in HUGIN.

(ii) Extend the system with a facility giving the chances that your hand is better than your opponent's hand.

Exercise 3.6 ^{H} For the BOBLO network in Figure 2.17 the following quantitative relations hold:

- there are four blood-group factors, *F1*, *F2*, *F3*, *F4*, and each factor is either present or absent;
- *F1* is only present if some phenogroup is *f1*;
- *F2* is only absent if both phenogroups are *f2*;
- *F3* is only present if some phenogroup is *f2*;
- *F4* is only absent if both phenogroups are *f3*;
- the prior probabilities for *parental error* are 0.0045 for both being incorrect, 0.0125 for sire incorrect, and 0.0018 for dam incorrect.

(i) Use the information in Section 2.5 to construct a BOBLO network.

(ii) Suppose for each factor that the risk of mistakes by the laboratory is 1 out of 1000. Extend the BOBLO network to incorporate laboratory mistakes.

(iii) For a calf, the stated dam has genotype (*f1*, *f2*), and the stated sire has (*f1*, *f3*). The laboratory reports factor 3 to be present and the other factors absent. What are the probabilities for *parental error*?

Exercise 3.7 A new family with two children has moved into the neighbourhood. I notice that one of them is a girl.

(i) Show that the probability that the other child is a boy is $\frac{2}{3}$.

(ii) I now imagine that I ask the girl whether she is the oldest. If she is, then the probability for the second child to be a boy is 0.5. The same holds if she is the youngest child. So I need not ask the question at all. The probability is 0.5. What is wrong?

Exercise 3.8 You are confronted with three doors A, B, and C. Behind exactly one of the doors there is \$10 000. The money is yours if you choose the correct door. After you have made your first choice of door but still not opened it, an official opens another one with nothing behind, and you are allowed to alter your choice. Should you do that?

Exercise 3.9 Extend the model in Figure 3.15 to incorporate constraints on colour and pattern for the same sock.

Exercise 3.10 The *drive* in golf is the first shot when playing a hole. If you drive with a *spoon* (a particular type of golf club) there is a 2% risk of a miss (a bad drive) and out of the good drives $\frac{1}{4}$ have a length of 180 m, $\frac{1}{2}$ are 200 m and $\frac{1}{4}$ have a length of 220 m. You may also use a *driver* (another type of golf club). This will on average increase the length by 10%, but you will also have three times as high a risk of a miss. Now, both wind and the slope of the hole may affect the result of the drive. Wind doubles the risk of a miss, and the length is affected by 10% (longer if the wind is from behind and shorter otherwise). A downhill slope yields 10% longer drives and uphill decreases the length by 10%.

Estimate the probabilities for a miss and length of drive given the various factors.

	No	*Mild*	*Moderate*	*Severe*
No	0.1	0.0	0.0	0.0
Mild	0.8	0.3	0.0	0.0
Moderate	0.1	0.6	0.8	0.0
Severe	0.0	0.1	0.2	1.0

Table (a).

	No	*Mild*	*Moderate*	*Severe*
No	0.3	0.0	0.0	0.0
Mild	0.2	0.4	0.0	0.0
Moderate	0.2	0.2	0.3	0.0
Severe	0.3	0.4	0.7	1.0

Table (b).

	No	*Mild*	*Moderate*	*Severe*
No	1.0	0.7	0.1	0.0
Mild	0.0	0.3	0.7	0.1
Moderate	0.0	0.0	0.2	0.7
Severe	0.0	0.0	0.0	0.2

Table (c).

Exercise 3.11 The *putt* is the (hopefully) last shot on a golf hole. My ball is lying 1 m away from the hole and under normal circumstances I will miss 1 out of 10. However, when it rains I miss 1 out of 7, if it is windy I miss 1 out of 4, if the green is curved I miss 1 out of 3, and if I am putting for a birdie (one under par) I miss 1 out of 2.

Estimate the probabilities for success and failure given the various factors.

Exercise 3.12 ^{H} Consider the headache example in Section 3.3.3. Let $P(Ha) = (0.93, 0.04, 0.02, 0.01)$ and let the tables below describe the effects on headache. Table 3.16(a) describes the effect of fever, hangover and fibrositis. Table 3.16(b) describes the effect of brain tumor, and Table 3.16(c) shows the effect of aspirin.

Use the technique from Figure 3.19 to establish the probabilities for headache given the causes and aspirin.

Exercise 3.13 Show that the model in Figure 3.17 corresponds to the one in Figure 3.16.

Exercise 3.14 ^{H} Show that noisy or may be modelled as described in Figure 3.19 and in Figure 3.20. Apply this to the putting problem of Exercise 3.11, and compare the amount of numbers to specify.

Exercise 3.15 ^{H} Consider the following example of scene interpretation. The image shows a breakfast table for one person, and the task is to determine whether it is a continental or a British breakfast table.

British breakfast is usually composed of tea, bacon and eggs, and toast with marmelade, while continental breakfast consists of coffee, boiled eggs and rolls with jam.

Possible objects. Plate (big or small), cup (tea or coffee), pot (tea or coffee), jar (red or orange contents), cutlery (knife, spoon, fork).

Big plates are confused with small plates with probability 0.1 (and vice versa). Tea cups may be taken for coffee cups with probability 0.3 and coffee cups for tea cups with probability 0.2. Tea pots and coffee pots are confused with probability 0.4, and the colour of the contents of a jar is determined correctly in 95 % of all cases. Knives are taken for spoons with probability 0.05 and for forks with probability 0.1. Spoons are never taken for knives, but for forks with probability 0.25. A fork is recognized as a spoon with probability 0.2, and as a knife with probability 0.1.

Cutlery never come in identical pairs, and if there is a fork, then there is also a knife on the table.

Six objects are identified on the table: a pot, a jar, a plate, a cup, two pieces of cutlery.

Construct a model for interpretation of the scene. It may, for example, be an idea to interpret "usual" as 99 out of 100.

Exercise 3.16 The following relations hold for the Boolean variables A, B, C, D, E and F:

$(A \vee \neg B \vee C) \wedge (B \vee C \vee \neg D) \wedge (\neg C \vee E \vee \neg F) \wedge (\neg A \vee D \vee F) \wedge (A \vee B \vee \neg C) \wedge (\neg B \vee \neg C \vee D) \wedge (C \vee \neg E \vee \neg F) \wedge (A \vee \neg D \vee F)$.

(i) What are the possible configurations?

(ii) We recieve the evidence that A is false and B is true. What are the possible configurations now?

(iii) The *satisfiability problem* for propositional calculus is: given a Boolean expression $\mathbb{E}$ (over n Boolean variables), is there a truth value assignment to the variables which makes $\mathbb{E}$ true?

Show that a method for calculation of probabilities in Bayesian networks yields a method for solving the satisfiability problem for propositional calculus. (Hint: assume that $\mathbb{E}$ is in conjunctive normal form and represent $\mathbb{E}$ as a Bayesian network.)

(iv) Show that a probability calculation in Bayesian networks is NP-hard.

Exercise 3.17 (A proof that the quadratic scoring rule is strictly proper.)

(i) Show that $\text{AvQS}(P, P^*)$ is minimal if $P = P^*$. (Hint: use the fact that the function $f(x) = x^2 - 2ax$ is minimal for $x = a$.)

(ii) Show that if $\text{AvQS}(P, P^*) = \text{AvQS}(P, P)$ then $P = P^*$. (Hint: show that $\text{AvQS}(P, P^*) - \text{AvQS}(P, P) = \sum_{w \in Word}(P(w) - P^*(w))^2$.)

Chapter 4

Propagation in Bayesian networks

This chapter presents the algorithm used in HUGIN for probability updating in Bayesian networks. The algorithm does not work directly on the Bayesian network, but on a so-called *junction tree* which is a tree of clusters of variables. The clusters are also called *cliques*, because they are cliques in a *triangulated graph*, which is a special graph constructed over the network. Each clique holds a table over the configurations of its variables, and HUGIN propagation consists of a series of operations on these tables. The subjects in this chapter are rather mathematical, and the reader interested in the results rather than in the reasoning behind them can jump directly to the summary in Section 4.7, which should give sufficient background for the reading of Chapters 5 and 6.

In Section 4.1 we define the multiplication and division of tables to be used in the algorithm. Section 4.2 gives methods for entering evidence and updating probabilities provided the full joint probability table is available, and in Section 4.3 we give the architecture of the algorithm when the cluster tree is available. Section 4.4 defines the concept junction tree, and we prove the correctness of the algorithm when applied on a junction tree. Section 4.5 is devoted to the construction of a junction tree from the Bayesian network.

The HUGIN algorithm yields the exact updated probabilities, but if you are unlucky, the algorithm will require so much space or time that the task is intractable. In Section 4.6 we present a technique, *stochastic simulation*, which can be used to get approximate probabilities when this happens.

4.1 An algebra of belief tables

Before we treat probability updating, we will introduce more formally the multiplication of belief tables, which we have used implicitly already.

	a_1	a_2	a_3
b_1	x_1	x_2	x_3
b_2	y_1	y_2	y_3
b_3	z_1	z_2	z_3

$\mathbf{t}$

	a_1	a_2	a_3
b_1	x_1'	x_2'	x_3'
b_2	y_1'	y_2'	y_3'
b_3	z_1'	z_2'	z_3'

$\mathbf{t}'$

	a_1	a_2	a_3
b_1	x_1x_1'	x_2x_2'	x_3x_3'
b_2	y_1y_1'	y_2y_2'	y_3y_3'
b_3	z_1z_1'	z_2z_2'	z_3z_3'

$\mathbf{t} \cdot \mathbf{t}'$

4.1.1 Multiplication and division

Let $\mathbf{t}$ and $\mathbf{t}'$ be two tables over the same variables. Then the product $\mathbf{t} \cdot \mathbf{t}'(c^*) = \mathbf{t}(c^*) \cdot \mathbf{t}'(c^*)$ for all configurations c^*.

Table 4.1 gives an example.

If the two tables are over different sets of variables we can also perform a multiplication.

Let $\mathbf{t}_{AB}$ be a table over $\{A, B\}$, and let $\mathbf{t}_{AC}$ be a table over $\{A, C\}$. Then $\mathbf{t}_{AB}$ and $\mathbf{t}_{AC}$ are multiplied by constructing a table $\mathbf{t}_{ABC}$ over $\{A, B, C\}$, and letting $\mathbf{t}_{AB} \cdot \mathbf{t}_{AC}(a, b, c) = \mathbf{t}_{AB}(a, b) \cdot \mathbf{t}_{AC}(a, c)$ for all configurations (a, b, c).

See Table 4.2 for an example.

Table 4.2 Multiplication of $\mathbf{t}_{AB}$ with $\mathbf{t}_{AC}$.

	a_1	a_2
b_1	x_1	x_2
b_2	x_3	x_4

$\mathbf{t}_{AB}$

	a_1	a_2
c_1	y_1	y_2
c_2	y_3	y_4

$\mathbf{t}_{AC}$

	a_1	a_2
b_1	(x_1y_1, x_1y_3)	(x_2y_2, x_2y_4)
b_2	(x_3y_1, x_3y_3)	(x_4y_2, x_4y_4)

$\mathbf{t}_{AB} \cdot \mathbf{t}_{AC}$

Division can be performed in the same way. Only, we have to be careful with zeros. If the denominator table has zero-entries, then the numerator table must have zero at the same places. In that case we put $\frac{0}{0} = 0$.

4.1.2 Marginalization

Let $\mathbf{t}_V$ be a table over V, and let W be a subset of V. A table $\mathbf{t}_W$ over W can be constructed by *marginalization*. For each configuration w^* let $\mathbf{t}_W(w^*)$ be the sum of all $\mathbf{t}_V(v^*)$, where v^* is a configuration of V coinciding with w^*. The notation is

$$\mathbf{t}_W = \sum_{V \setminus W} \mathbf{t}_V.$$

We shall use the following proposition later.

Proposition 4.1 *Let W and V be disjoint sets of variables, and let $\mathbf{t}_W$ and $\mathbf{t}_V$ be tables over W and V. Then*

$$\sum_V (\mathbf{t}_W \cdot \mathbf{t}_V) = \mathbf{t}_W \cdot \sum_V \mathbf{t}_V.$$

taken out of marginalization. See Table 4.3 for an example.

Table 4.3 An example that $\sum_A \mathbf{t}_B \cdot \mathbf{t}_A = \mathbf{t}_B \sum_A \mathbf{t}_A$.

y_1		x_1	
y_2		x_2	
y_3		x_3	

$\mathbf{t}_B \quad \mathbf{t}_A$

	a_1	a_2	a_3
b_1	y_1x_1	y_1x_2	y_1x_3
b_2	y_2x_1	y_2x_2	y_2x_3
b_3	y_3x_1	y_3x_2	y_3x_3

$\mathbf{t}_B\mathbf{t}_A$

b_1	$y_1x_1 + y_1x_2 + y_1x_3$	y_1	
b_1	$y_2x_1 + y_2x_2 + y_2x_3$	y_2	$(x_1 + x_2 + x_3)$
b_1	$y_3x_1 + y_3x_2 + y_3x_3$	y_3	

$\sum_A \mathbf{t}_B\mathbf{t}_A \qquad \mathbf{t}_B \sum_A \mathbf{t}_A$

4.2 Probability updating in joint probability tables

Let A be a variable with $P(A) = (x_1, \ldots x_n)$. Assume we get the information e that A can only be in states i and j. This statement says that all states except i and j are impossible, and we have the belief $P(A, e) = (0, \ldots, 0, x_i, 0, \ldots, x_j, 0, \ldots, 0)$. Note that $P(e)$, the prior probability of e, is $x_i + x_j$, the sum of the probabilities of the possible states. To calculate $P(A \mid e)$ we use the fundamental rule:

$$P(A \mid e) = \frac{P(A, e)}{P(e)} = \frac{P(A, e)}{\sum_A P(A, e)}.$$

The way that e is entered can be interpreted as a multiplication of $P(A)$ with the table $\underline{e} = (0, \ldots, 0, 1, 0, \ldots, 0, 1, 0, \ldots, 0)$ resulting in $P(A, e)$.

Definition. Let A be a variable with n states. A *finding* on A is an n-dimensional table of zeros and ones.

Semantically, a finding is a statement that certain states of A are impossible.

Now, let U be a universe of variables, and assume that we have easy access to $P(U)$, the joint probability table. Then, $P(B)$ for any variable B in U is easy to calculate:

$$P(B) = \sum_{U \setminus \{B\}} P(U).$$

Suppose we wish to enter the above finding. Then $P(U, e)$ is the table resulting from $P(U)$ by giving all entries with A in state i or j the value zero and leaving the other entries unchanged. Again, $P(e)$ is the sum of all entries in $P(U, e)$ and

$$P(U \mid e) = \frac{P(U, e)}{P(e)} = \frac{P(U, e)}{\sum_U P(U, e)}.$$

findings $\{f_1, \ldots, f_m\}$ each finding can be entered separately, and $P(U, e)$ is the product of $P(U)$ and the findings $\underline{f_i}$. We can express the considerations above in the following theorem.

Theorem 4.1 *Let U be a universe of variables and let $e = \{f_1, \ldots, f_m\}$. Then*

$$P(U, e) = P(U) \cdot \underline{f_1} \cdot \cdots \cdot \underline{f_m} \text{ and } P(U, e) = \frac{P(U \mid e)}{P(e)},$$

where

$$P(e) = \sum_U P(U, e).$$

Theorem 4.1 says that if we have access to $P(U)$, then we can enter evidence and perform probability updating. However, even for small sets of variables, the table $P(U)$ is intractably large, and we have to find a smaller representation.

4.3 Cluster trees

As shown in Section 2.3.7 (the chain rule), a Bayesian network over U is a representation of $P(U)$. This means that we can, in principle, calculate $P(U)$ as the product of all conditional probabilities from the network. The question then, is whether we can enter evidence and perform probability updating in Bayesian networks without being forced to calculate $P(U)$. It has turned out to be rather difficult.

Instead we can work with another representation called *cluster trees.*

Definition. A *cluster tree over U* is a tree of clusters of variables from U. The nodes are subsets of U, and the union of all nodes is U. (A tree is an undirected graph without cycles.)

The links are labelled with *separators* which consist of the intersection of the adjacent nodes.

Each node and separator holds a real numbered table over the configurations of its variable set.

In Figure 4.1 we give a cluster tree for the network M_{min}

Figure 4.1 The Bayesian network M_{min} and a corresponding cluster tree. Separators are in square boxes.

Now, let *BN* be a Bayesian network over U. A *Cluster tree corresponding to BN* is constructed in the following way:

- there is a family of nodes such that for each variable A with parent set $pa(A)$ there is at least one node V such that $pa(A) \cup \{A\} \subseteq V$;
- organize the nodes as a tree with separators (so far there is no restriction on how you organize the tree);
- give all nodes and separators a table of ones.
- for each variable A choose exactly one node V containing $pa(A) \cup \{A\}$ and multiply $P(A \mid pa(A))$ on Vs table.

Then the product of all node tables in the cluster tree is the product of all conditional probability tables in *BN*, and therefore we have the following theorem.

Theorem 4.2 *Let* BN *be a Bayesian network over U. Then any cluster tree corresponding to* BN *is a representation of $P(U)$, and $P(U)$ is the product of all cluster tables divided by the product of all separator tables.*

Remark. In Theorem 4.2 we divide the product of all cluster tables by the product of all separator tables. This does not do any harm, because the separator tables consist of ones, but the reader may wonder why. The reason is that, when we now start to move the information around in the cluster tree, then the product of all cluster tables divided by all separator tables is invariant, and thereby the tree remains a representation of $P(U)$.

It is easy to insert findings into a cluster tree. Let e be a finding on A. Multiply $\underline{e}$ on the table of any node containing A. Then, by the chain rule and Theorem 4.1 the product of all node tables is $P(U) \cdot \underline{e} = P(U, e)$.

To calculate $P(B, e)$ for an arbitrary variable B is not as easy, and the coming sections are devoted to this problem.

4.3.1 Absorption in cluster trees

We introduce an operation in cluster trees. It has the effect of re-arranging the information stored in the tables.

Definition. Let V and W be neighbours in a cluster tree, let S be their separator, and let $\mathbf{t}_V$, $\mathbf{t}_W$ and $\mathbf{t}_S$ be their tables. The operation *absorption* is the result of the following procedure:

- calculate $\mathbf{t}_S^* = \sum_{V \setminus S} \mathbf{t}_V$;
- give S the table $\mathbf{t}_S^*$;
- give W the table $\mathbf{t}_W^* = \mathbf{t}_W \frac{\mathbf{t}_S^*}{\mathbf{t}_S}$.

We then say that W has *absorbed* from V or that W calibrates to V.

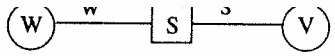


Figure 4.2 W absorps from V. $\mathbf{t}_S^* = \sum_{V \setminus S} \mathbf{t}_V$; $\mathbf{t}_W^* = \mathbf{t}_W \cdot \frac{\mathbf{t}_S^*}{\mathbf{t}_S}$.

Remarks.

(1) The idea behind absorption is that the information which V and W can have in common is the information on S, and this is what W receives from V. If W, V and S hold the same information on S, that is if

$$\sum_{W \setminus S} \mathbf{t}_W = \mathbf{t}_S = \sum_{V \setminus S} \mathbf{t}_V,$$

then absorption does not change anything. We then say that the link is *consistent.* If all links in the cluster tree are consistent we say that the tree is consistent. If a tree is consistent, then absorption does not have any effect at all.

Assume that the link is consistent, but now some evidence changes $\mathbf{t}_V$ to $\mathbf{t}_V^*$. Then after W has absorbed from V, the three tables all hold Vs new information on S:

$$\sum_{W \setminus S} \mathbf{t}_W^* = \sum_{W \setminus S} \mathbf{t}_W \frac{\mathbf{t}_S^*}{\mathbf{t}_S} = \frac{\mathbf{t}_S^*}{\mathbf{t}_S} \sum_{W \setminus S} \mathbf{t}_W = \frac{\mathbf{t}_S^*}{\mathbf{t}_S} \mathbf{t}_S = \mathbf{t}_S^* = \sum_{V \setminus S} \mathbf{t}_V^*.$$

(2) W can only absorb from V through S if $\mathbf{t}_W$ has zeros in the entries corresponding to the zero-entries in $\mathbf{t}_S$. We say that a link in a cluster tree is *supportive* if it allows absorption in both directions, and a cluster tree is supportive if all its links are supportive. Note that the cluster trees constructed in Section 4.2 are supportive since the separator tables have no zero-entries.

Lemma 4.1 *Supportiveness is preserved under absorption.*

Proof. Let W absorb from V through the separator S. Then

$$\mathbf{t}_W^* = \mathbf{t}_W \cdot \frac{\mathbf{t}_S^*}{\mathbf{t}_S},$$

where

$$\mathbf{t}_S^* = \sum_{V \setminus S} \mathbf{t}_V.$$

Then any zero-entry in $\mathbf{t}_S^*$ is also a zero-entry in $\mathbf{t}_W^*$. This clearly also holds for $\mathbf{t}_V$. □

Theorem 4.3 *Let T be a supportive cluster tree. Then the product of all cluster tables divided by the product of all separator tables is invariant under absorption.*

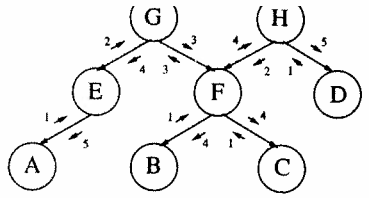


Figure 4.3 Certainty updating through message passing in a cluster tree. The numbers on the links indicate the order in which the messages are passed and in which direction.

Proof. When W absorbs from V through the separator S, only the tables of W and S are changed. Therefore it is enough to prove that the fraction of Ws and Ss table is unchanged. We have

$$\frac{\mathbf{t}_W^*}{\mathbf{t}_S^*} = \frac{\mathbf{t}_W \cdot \frac{\mathbf{t}_S^*}{\mathbf{t}_S}}{\mathbf{t}_S^*} = \frac{\mathbf{t}_W}{\mathbf{t}_S}.$$

Theorem 4.3 ensures that if we start with a Bayesian network over U, construct a corresponding cluster tree T, and then perform a series of absorptions, then T remains a representation of $P(U)$, and $P(U)$ can be calculated as the product of all cluster tables divided by the product of all separator tables.

4.3.2 Message passing in cluster trees

The next question is how many absorptions can we perform, and can they help us in transforming the tables in a cluster tree into a form where it is easy to calculate $P(A)$ for single variables?

We can think of absorptions as messages passed between the nodes in the tree. That is, a node V *sends a message* to its neighbour W when W absorbs from V.

Message passing scheme. A node V can send exactly one message to a neighbour W, and it may only be sent when V has received a message from each of its other neighbours.

Consider, for example, the cluster tree in Figure 4.3. The leaves of the tree (the nodes A, B, C, D) can send to their single neighbour (1). Then E can send to G, and H can send to F (2). Next, G can send to F, and F can send to G (3), F can send to H, B and C, and G can send to E (4). Finally E can send to A and H to D (5). Now each node has sent to all of its neighbours.

As can be seen, the message passing algorithm is not sequential, and a good way of thinking of it is that each variable is busy waiting, eager to send messages. Each time it receives a message it updates its own table and sends a message to the eligible neighbours (if any).

Theorem 4.4 *Let T be a supportive cluster tree, and suppose that messages are passed according to the message passing scheme. Then:*

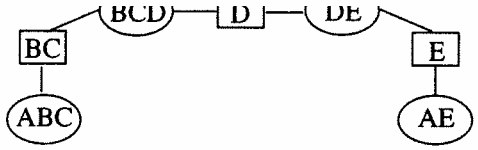


Figure 4.4 A cluster tree over binary variables. All variables except A are in state y. In the node (A, B, C) A is in state y, and in the node (A, E) A is in state n. Though the cluster tree is consistent, the table for $\mathbf{t}_A$ marginalized from $\mathbf{t}_{ABC}$ is different from the marginal taken from $\mathbf{t}_{AE}$.

(i) message passing can continue until a message has been passed in both directions of each link;

(ii) when a message has been passed in both directions of each link then T is consistent.

Proof. (i) Exercise 4.3.

(ii) If T consists of only one node then the theorem is obviously true.

Assume that T has more than one node, and let (V, W) be an arbitrary link with separator S. Let the first message to be passed over (V, W) be from W to V, and let $\mathbf{t}_V$, $\mathbf{t}_S$ and $\mathbf{t}_W$ be the tables before the message is passed.

When the message has been passed we have $\mathbf{t}_S^* = \sum_{W \setminus S} \mathbf{t}_W$. Next, when the message from V and W has to be passed, the tables for S and W have not been changed (W has not received further messages). Let the table for V be $\mathbf{t}_V^{**}$. After message passing we have

$$\mathbf{t}_S^{**} = \sum_{V \setminus S} \mathbf{t}_V^{**} \quad \textit{and} \quad \mathbf{t}_W^{**} = \mathbf{t}_W \frac{\mathbf{t}_S^{**}}{\mathbf{t}_S^*}.$$

Now

$$\sum_{V \setminus S} \mathbf{t}_V^* = \sum_{V \setminus S} \mathbf{t}_V \frac{\mathbf{t}_S^{**}}{\mathbf{t}_S^*} = \frac{\mathbf{t}_S^{**}}{\mathbf{t}_S^*} \sum_{V \setminus S} \mathbf{t}_V = \frac{\mathbf{t}_S^{**}}{\mathbf{t}_S^*} \mathbf{t}_S^* = \mathbf{t}_S^{**} = \sum_{W \setminus S} \mathbf{t}_W^{**}.$$

Therefore the link is consistent. □

4.4 Junction trees

Let T be a cluster tree over U, let A be a variable in U, and suppose that A is an element of the nodes V and W. If T is consistent we would expect $\sum_{V \setminus \{A\}} \mathbf{t}_V = \sum_{W \setminus \{A\}} \mathbf{t}_W$. Certainly this is so if V and W are neighbours, but otherwise there is no guarantee. See Figure 4.4 for an example.

We say that a consistent cluster tree is *globally consistent* if for any nodes V and W with intersection I we have

$$\sum_{V \setminus I} \mathbf{t}_V = \sum_{W \setminus I} \mathbf{t}_W.$$

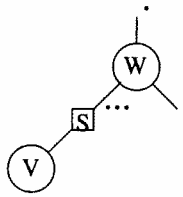


Figure 4.5 V is a leaf of T linked to W and with separator S.

As Figure 4.4 indicates, the reason why consistence does not imply global consistence is that a variable A can be placed in two locations in the tree such that information on A cannot be passed between the two locations. To ensure global consistence we must add a requirement to cluster trees.

Definition. A cluster tree is a *junction tree* if, for each pair of nodes V, W, all nodes on the path between V and W contain the intersection $V \cap W$.

Theorem 4.5 *A consistent junction tree is globally consistent.*

Proof. Exercise 4.7. □

The following theorems will show that if we construct a junction tree corresponding to a Bayesian network, then we have good algorithms for insertion of evidence as well as probability updating. When we construct a cluster tree corresponding to a Bayesian network we have several degrees of freedom, and we shall use them for constructing a junction tree. However, it is not easy. For example, with the clusters in Figure 4.4 it is impossible to construct a tree with the junction tree property. We will leave this problem here, and return to it in Section 4.5.

Theorem 4.6 *Let T be a consistent junction tree over U, and let $\mathbf{t}_U$ be the product of all node tables divided by the product of all separator tables. Let V be a node with table $\mathbf{t}_V$. Then*

$$\mathbf{t}_V = \sum_{U \setminus V} \mathbf{t}_U.$$

Proof. Induction on the number of nodes.

Clearly the theorem holds when T consists of a single node.

Now, assume the theorem to hold for any junction tree with n nodes, and let T be a consistent junction tree with $n + 1$ nodes. Let V be a leaf of T linked to W and with separator S (see Fig. 4.5). Let T' be the junction tree resulting from removing V (and S), and let T' have the universe U'. Then

$$\mathbf{t}_U = \mathbf{t}_{U'} \cdot \frac{\mathbf{t}_V}{\mathbf{t}_S}.$$

where $\mathbf{t}_{U'}$ is the product of all node tables in T' divided by the separator tables in T'. Let D be the set of variables $V \setminus S$, and let H be $W \setminus S$. From the junction tree property we have that $D \cap U' = \emptyset$.

Since T is consistent we have

$$\sum_D \mathbf{t}_V = \mathbf{t}_S = \sum_H \mathbf{t}_W.$$

Now

$$\begin{aligned}\sum_D \mathbf{t}_U &= \sum_D \mathbf{t}_{U'} \cdot \frac{\mathbf{t}_V}{\mathbf{t}_S} \\ &= \mathbf{t}_{U'} \cdot \frac{\sum_D \mathbf{t}_V}{\mathbf{t}_S} \\ &= \mathbf{t}_{U'} \cdot \frac{\mathbf{t}_S}{\mathbf{t}_S} \\ &= \mathbf{t}_{U'}.\end{aligned}$$

Therefore, by the induction hypothesis we have

$$\sum_{U \setminus V_i} \mathbf{t}_U = \mathbf{t}_{V_i}$$

for all V_i in T'.

Furthermore,

$$\begin{aligned}\sum_{U \setminus V} \mathbf{t}_U &= \sum_{U' \setminus S} \mathbf{t}_{U'} \cdot \frac{\mathbf{t}_V}{\mathbf{t}_S} \\ &= \frac{\mathbf{t}_V}{\mathbf{t}_S} \cdot \sum_{U' \setminus S} \mathbf{t}_{U'} \\ &= \frac{\mathbf{t}_V}{\mathbf{t}_S} \cdot \sum_{W \setminus S} \mathbf{t}_W \\ &= \frac{\mathbf{t}_V}{\mathbf{t}_S} \cdot \mathbf{t}_S \\ &= \mathbf{t}_V.\end{aligned}$$

□

The considerations above are summarized in the following theorem.

Theorem 4.7 *Let* BN *be a Bayesian network representing $P(U)$, and let T be a junction tree corresponding to* BN. *After a full round of message passing in T, we have for each node V and each separator S that*

$$\mathbf{t}_V = \sum_{U \setminus V} P(U) = P(V) \text{ and } \mathbf{t}_S = P(S).$$

Proof. By Theorem 4.2, $P(U)$ is the product of the initial node tables divided by the separator tables. Theorems 4.3 and 4.4 give that after a full round of message passing T is consistent, and $P(U)$ is the product of all node tables divided by all separator tables. Theorems 4.5 and 4.6 yield the conclusion. □

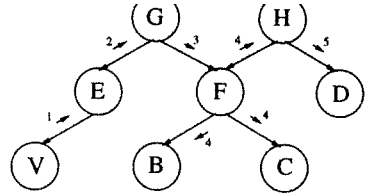


Figure 4.6 The message passing in *DistributeEvidence*(V).

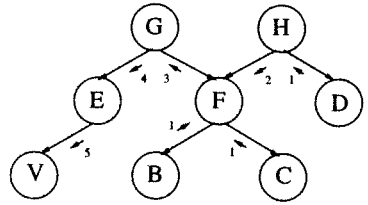


Figure 4.7 The message passing in *CollectEvidence*(V).

Theorem 4.8 *Let* BN *be a Bayesian network representing $P(U)$, and let T be a junction tree corresponding to* BN. *Let $e = \{f_1, \ldots, f_m\}$ be findings on the variables $\{A_1, \ldots, A_m\}$. For each i find a node containing A_i and multiply its table with f_i.*

Then, after a full round of message passing we have for each node V and separator S that

$$\mathbf{t}_V = P(V, e) \qquad \mathbf{t}_S = P(S, e) \qquad P(e) = \sum_V \mathbf{t}_V.$$

Proof. Use Theorem 4.1 and proceed as in the proof of Theorem 4.7. □

4.4.1 HUGIN propagation

Assume that we have a consistent junction tree, and now a single node V receives evidence. Then half of the messages can be avoided: V sends messages to all of its neighbours who recursively send messages to all neighbours except the one from which the message came (see Fig. 4.6). We call this algorithm *DistributeEvidence*.

Now, suppose that we are only interested in the certainty of one node, V. Then half of the certainty updating messages can be avoided: V asks all its neighbours to send it a message, and if they are not allowed to do so, they recursively pass the request to all neighbours except the one from which the request came (see Fig. 4.7). We call this algorithm *CollectEvidence*.

The two algorithms *DistributeEvidence* and *CollectEvidence* can be used for a more organized message passing scheme. No matter the amount of evidence entered, take any variable V. Call *CollectEvidence* from V and after that call *DistributeEvidence* from V. The result is that all messages have been passed, and they were passed when permitted (see Fig. 4.8 and Exercise 4.5).

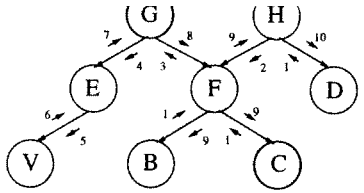


Figure 4.8 Updating through *CollectEvidence*(V) followed by *DistributeEvidence*(V).

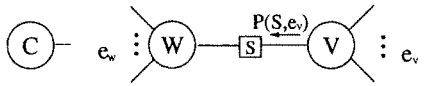


Figure 4.9 Evidence e_V has been entered at the righthand side of S. e_W has been entered at the lefthand side of S. C is used as a root for the propagation.

HUGIN propagation uses corresponding junction trees, and the operations *CollectEvidence* and *DistributeEvidence*. A node Rt in the junction tree is chosen as a root, and whenever a propagation takes place, *CollectEvidence*(Rt) is called followed by a call of *DistributeEvidence*(Rt). When the calls are finished, the tables are *normalized* so that they sum to one.

HUGIN propagation has a nice side effect, namely that it gives access to various probabilities of sets of entered findings.

Let us use Theorem 4.8 to have a closer look at what is actually communicated in the propagation algorithm. The general situation is described in Figure 4.9.

A call of *CollectEvidence*(C) will cause a call of *CollectEvidence*(V), and by Theorem 4.8 this will result in $\mathbf{t}_V^* = P(V, e_V)$. This gives that $P(e_V)$ can be calculated without further propagations. Unfortunately, the situation is not symmetric. In the *DistributeEvidence* phase the message passed from W to S is $P(S, e)$.

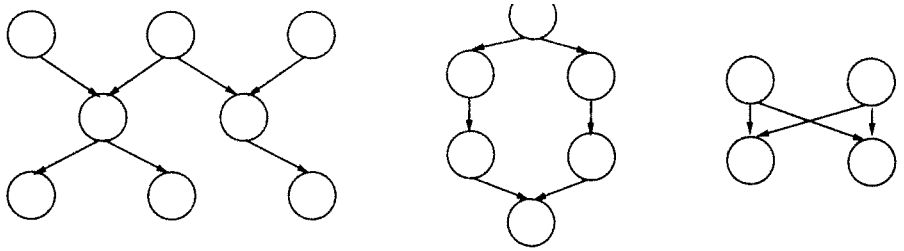
4.5 Construction of junction trees

In this section we shall give a method for constructing junction trees for DAGs.

4.5.1 Singly connected DAGs

A DAG is *singly connected* if the graph you get by dropping the directions of the links is a tree (see Fig. 4.10).

For singly connected DAGs it is easy to construct junction trees. For each variable A with $pa(A) \neq \emptyset$ you form the cluster $pa(A) \cup \{A\}$. Between any two clusters with a non-empty intersection you add a link with the intersection as a separator. The resulting graph is called a *junction graph*. All separators consist of a single variable, and if the junction graph has cycles, then all separators on the cycle contain



Singly connected Multiply connected

Figure 4.10 Examples of singly connected and multiply connected DAGs.

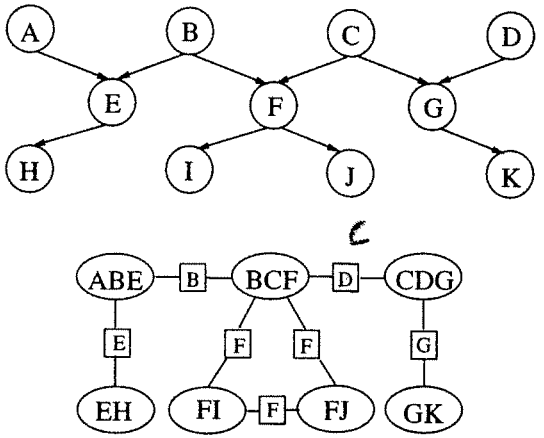
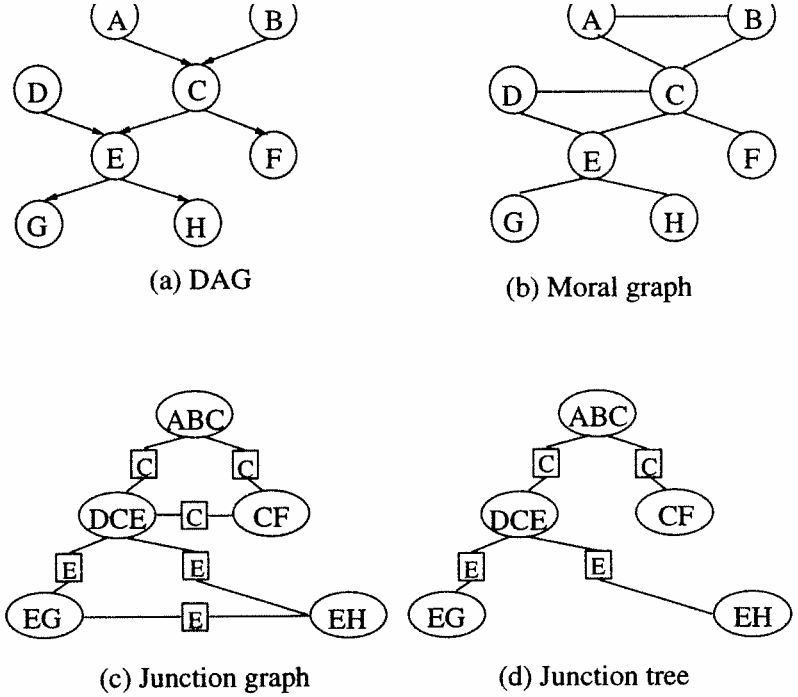


Figure 4.11 A singly connected DAG and its junction graph. By removing any of the links with separator F you get a junction tree.

the same variable. Therefore any of the links can be removed to break the cycle, and by removing links until you have a tree, you get a junction tree (see Fig. 4.11).

We know that when we construct a cluster tree corresponding to a DAG, then for all variables A there must be a cluster V containing $pa(A) \cup \{A\}$. We can illustrate this on a graph by having a link between any pair of variables which must appear in the same cluster. This means that we take the DAG, add a link between any pair of variables with a common child, and drop the directions of the original links. The resulting graph is called the *moral graph*. From the moral graph you can read the clusters to consider, namely the cliques in the graph (maximal sets of variables that are all pairwise linked). In Figure 4.12 we give an example of the construction.



(a) DAG (b) Moral graph

(c) Junction graph (d) Junction tree

Figure 4.12 Construction of a junction tree for a singly connected DAG.

4.5.2 Coping with cycles

Consider the junction graph in Figure 4.13. The intersection of the two clusters of variables is (AB), and a junction tree is easily found.

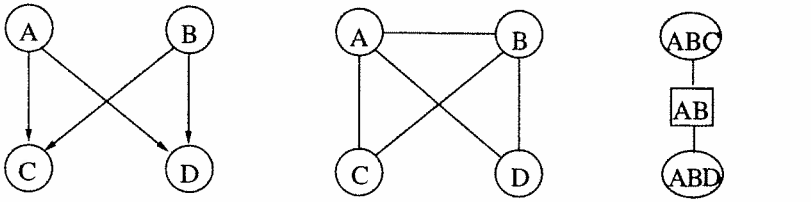
Consider the DAG in Figure 4.14(a) with the moral graph in Figure 4.14(b). Sticking to the approach that the clusters are the cliques in the moral graph, we see that if we join A, B and C, then we get a junction tree.

The DAG in Figure 4.15 is more problematic. The cycle in the junction graph cannot be broken.

The propagation problem is that coupled information (on (DE)) is decoupled but meets again under propagation. This can also be seen from the cycle $D - E - C - A - B - D$ in the moral graph. A way to solve the problem is to add so-called *fill-ins* to the moral graph: add a link between C and D and one between B and C. The result is shown in Figure 4.16 together with the resulting junction tree.

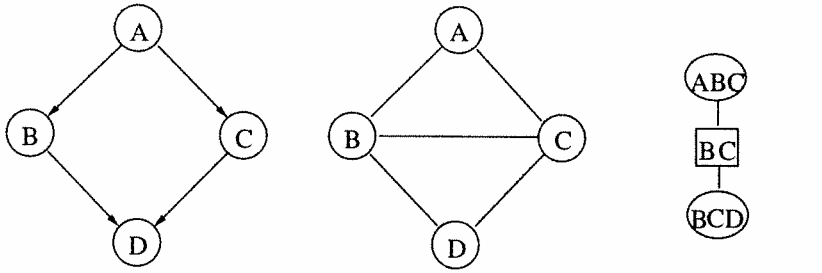
The general rule for filling-in the moral graph is that any cycle with more than three variables shall have a chord. In this case the graph is called *triangulated*.

In Figures 4.17 and 4.18 there is another example of the process from DAG to junction tree. Note that without the fill-in $(B - D)$ the cycle $A - B - F - D - A$ does not have a chord.



(a) DAG (b) Moral graph (c) Junction tree

Figure 4.13 Construction of a junction tree for a simple multiply connected DAG.



(a) DAG (b) Moral graph (c) Junction tree

Figure 4.14 Another simple DAG with a cycle.

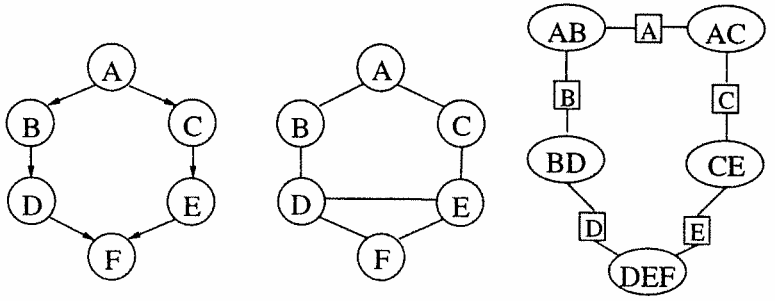


Figure 4.15 A DAG with a large cycle.

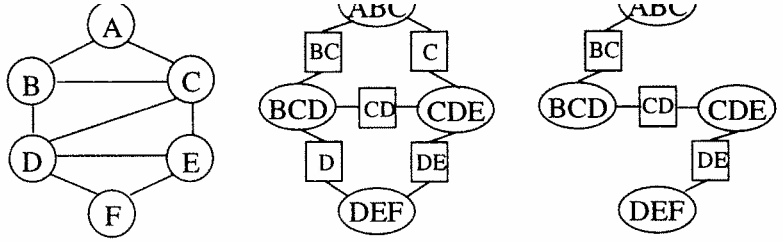


Figure 4.16 The filled-in moral graph from Figure 4.15, the junction graph, and the juction tree resulting from removing the links with separator D and C.

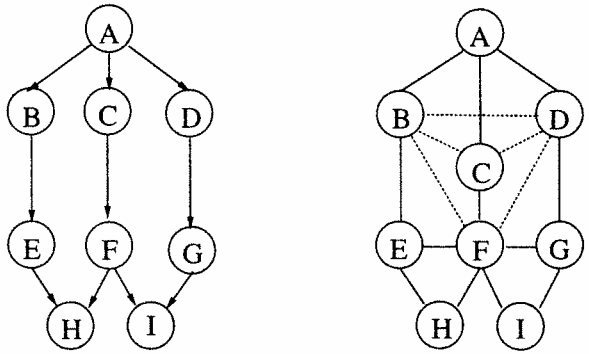


Figure 4.17 A DAG, the moral and triangulated graphs. The fill-ins are indicated by dotted lines.

4.5.3 From DAG to junction tree

In this section we present, without proofs, algorithms for triangulation of graphs and for construction of junction trees from triangulated graphs. Proofs of Theorems 4.9 and 4.10 are given in Appendix A.

Definition. An undirected graph is *triangulated* if any cycle of length > 3 has a chord.

Definition. A node A is *eliminated* by adding links such that all of its neighbours are pairwise linked and then removing A together with its links.

Note that if a node A can be eliminated without adding links, then A cannot be part of a chordless cycle of length > 3.

Theorem 4.9 *A graph is triangulated if and only if all of its nodes can be eliminated one by one without adding any link.*

Theorem 4.9 yields a method for triangulation as well as a test for whether a graph is triangulated. The method consists of eliminating the nodes in some order (adding links, if necessary) and when this is done the resulting graph is triangulated. An example is given in Figure 4.20.

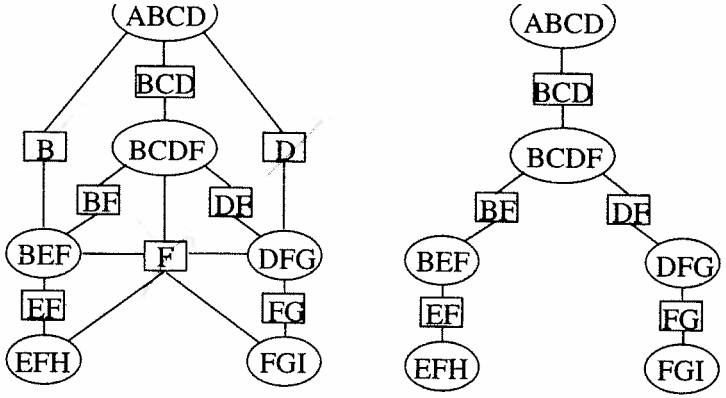


Figure 4.18 The junction graph for the triangulated graph in Figure 4.17 and a junction tree.

Note that there are several triangulations of the graph. Intuitively, triangulations with as few fill-ins as possible are preferred. However, optimality is connected to the resulting junction tree and the computational complexity of the propagation algorithm. We shall return to the question of optimality later.

Definition. A *junction graph* for an undirected graph G is an undirected, labelled graph. The nodes are the cliques in G. Every pair of nodes with a non-empty intersection has a link labelled by the intersection.

There is an easy way of identifying the cliques in a triangulated graph G. Let $A_1, \ldots, A_n$ be an elimination sequence for G, and let C_i be the set of variables containing A_i and all its neighbours at the time of elimination (neighbours with higher numbers). Then every clique of G is a C_i for some i.

The reader may check that the cliques of the graphs in Figure 4.20(a) are C_1, C_2, C_3, C_4, and that the cliques of the graph in Figure 4.20(b) are C_1, C_2, C_3.

The junction tree we are aiming at will be a subgraph of the junction graph. Since message passing will be restricted to links in the junction tree we are not allowed to remove a link from the junction graph if thereby some kind of information cannot be passed. If, for example, the clusters U and V have the variable A in common, they have a link with label A. If this link is removed, there shall be another path in the remaining graph through which information on A can be passed from U to V. So, let us recall the following definition.

Definition. A spanning tree of a junction graph is a *junction tree* if it has the property that for each pair of nodes, U, V, all nodes on the path between U and V contain $U \cap V$. (A subtree of a graph is a spanning tree if all nodes of the graph are nodes in it.)

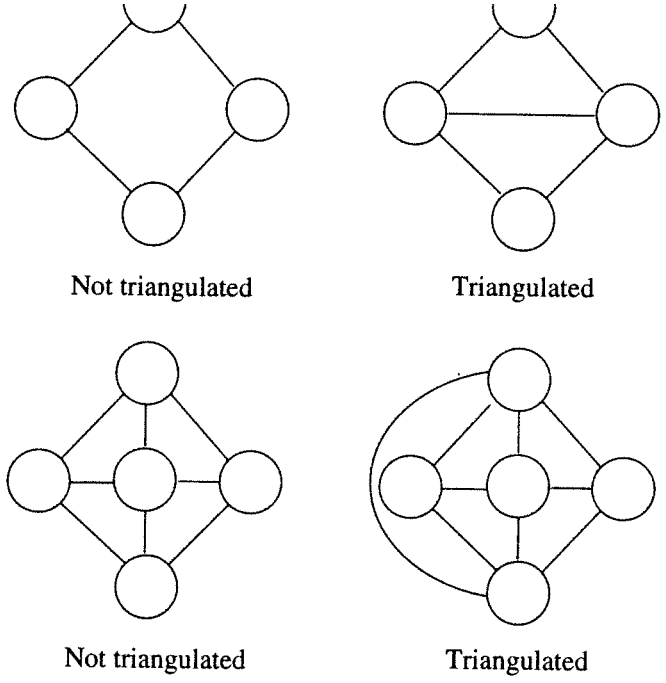


Figure 4.19 Triangulated and not triangulated graphs.

Theorem 4.10 *An undirected graph is triangulated if and only if its junction graph has a junction tree.*

Definition. The *weight* of a link in a junction graph is the number of variables in the label. The weight of a junction tree is the sum of the weights of the labels.

Theorem 4.11 *(Without proof.) A subtree of the junction graph of a triangulated graph is a junction tree if and only if it is a spanning tree of maximal weight.*

Theorem 4.11 provides an easy way of constructing junction trees, namely Kruskal's algorithm: choose successively a link of maximal weight unless it creates a cycle.

There are other ways of constructing junction trees. In particular, if an elimination sequence for the triangulated graph is known, very efficient algorithms exist (see Exercise 4.8). So, if the graph is triangulated then the construction of a junction tree is rather fast.

The only problematic step in the process from DAG to junction tree is the triangulation. Since any elimination sequence will produce a triangulation it may not seem a problem, but for the propagation algorithm it is. In HUGIN propagation the cliques in the junction graph shall have joint probability tables attached to them. The size of the table is the product of the number of states of the variables. So, the size increases exponentially with the size of the clique. A good triangulation,

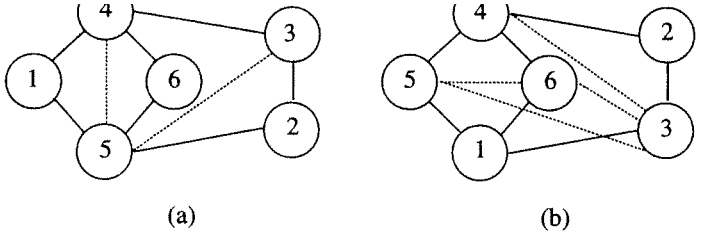


Figure 4.20 Two examples of triangulation through elimination. The numbers on the nodes indicate the elimination order, and the dotted lines are fill-ins.

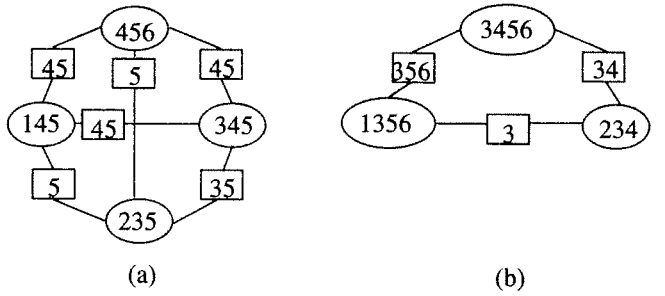


Figure 4.21 Junction graphs for the two triangulated graphs in Figure 4.20.

therefore, is a triangulation yielding small cliques, or to be more precise, yielding small probability tables. The problem of determining an optimal triangulation is NP-complete. However, there is a heuristic algorithm which has proven to give fairly good results. It is a version of the greedy approach: eliminate repeatedly a node not requiring fill-ins and if this is not possible, eliminate a node yielding the smallest table. In Figure 4.23 an example is given.

4.6 Stochastic simulation

The propagation method requires tables for the cliques in the triangulated graph. These cliques may be very large, and it happens that the space requirements cannot be met by the hardware available. In this case an approximate method would be satisfactory.

In this section we shall give a flavour of an approximate method called *stochastic simulation.* The idea behind the simulation is that the causal model is used to simulate the flow of impact. When impact from a set of variables to a variable A is simulated, a random generator is used to decide the state of A.

To illustrate the technique, consider the Bayesian network in Figure 4.24 with the conditional probabilities specified in Table 4.4.

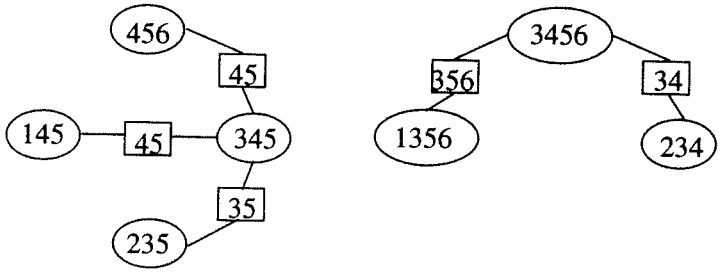


Figure 4.22 Junction trees for the junction graphs in Figure 4.20.

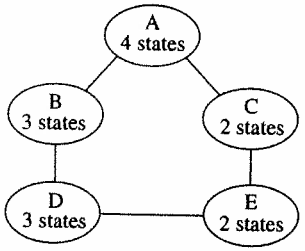


Figure 4.23 A heuristic elimination sequence is E, D (and A, B, C).

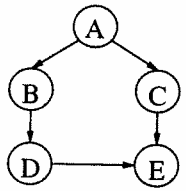


Figure 4.24 An example network. All variables have the states y and n.

network. $P(A) = (0.4, 0.6)$.

	A	
B	y	n
y	0.3	0.8
n	0.7	0.2
	$P(B \mid A)$	

	A	
C	y	n
y	0.7	0.4
n	0.3	0.6
	$P(C \mid A)$	

	B	
D	y	n
y	0.5	0.1
n	0.5	0.9
	$P(D \mid B)$	

	C	
D	y	n
y	(0.9, 0.1)	(0.999, 0.001)
n	(0.999, 0.001)	(0.999, 0.001)
	$P(E \mid C, D)$	

Table 4.5 A set of 100 configurations of (A, B, C, D, E) sampled from the network in Figure 4.24 and Table 4.4

	CDE							
AB	yyy	yyn	yny	ynn	nyy	nyn	nny	nnn
yy	4	0	5	0	1	0	2	0
yn	2	0	16	0	1	0	8	0
ny	9	1	10	0	14	0	16	0
nn	0	0	4	0	0	0	7	0

The idea now is to draw a random configuration of the variables (A, B, C, D, E), and to do this a sufficient number of times.

A random configuration is selected by successively sampling the states of the variables. First the state of A is sampled. A random generator (with even distribution) is asked to give a real number between zero and one. If the number is less than 0.4 the state is y, if not the state is n. Assume that the result is y. From the conditional probability table $P(B \mid A)$ we have that $P(B \mid y) = (0.3, 0.7)$. The random generator is asked again, and if the number is less than 0.3, the state of B is y. This procedure is repeated to get the state of C, D, and E, and a configuration is determined.

The next configuration is sampled through the same procedure, and the procedure is repeated until m configurations are sampled. In Table 4.5 an example set of configurations is given.

The probability distributions for the variables are calculated by counting in the sample set (see Exercise 4.12). For 39 of the samples in Table 4.5 the first state is y, and this gives an estimated probability $P(A) = (0.39, 0.61)$.

The method above, called *forward sampling*, does not require a triangulation of the network, and it is not necessary to store the sampled configurations (like Table 4.5); it is enough to store the counts for each variable. Whenever a sampled configuration has been determined, the counts of all variables are updated, and the sample can

determined in a time linear to the number of variables. The cost is accuracy and time.

So far only the initial probabilities are calculated. When evidence arrives, it can be handled by simply discarding the configurations which do not conform to it. That is, a new series of stochastic simulations are started, and whenever a state of an observed variable is drawn, you stop simulating if the state drawn is not the observed one.

Unfortunately, this method has a serious drawback. Assume in the example above that the observations for the network are $B = n$ and $E = n$. The probability for $(B = n, E = n)$ is 0.00282. This means that in order to get 100 configurations you should for this tiny example, expect to perform more than 35 000 stochastic simulations.

Methods have been constructed for dealing with this problem. A promising method is called *Gibbs sampling*.

In Gibbs sampling you start with some configuration consistent with the evidence (for example determined by forward sampling), and then you randomly change the state of the variables in causal order. In one sweep through the variables you determine a new configuration, and then you use this configuration for a new sweep, etc.

In the example let $B = n$ and $E = n$ be the evidence, and let the starting configuration be *ynyyn*. Now, calculate the probability of A given the other states of that configuration. That is, $P(A \mid B = n, C = y, D = y, E = n)$. From the network we see that it is sufficient to calculate $P(A \mid B = n, C = y)$. It is easily done by Bayes' rule: it is (0.8, 0.2). We draw a number from the random generator, and let us assume that the number is 0.456 resulting in $A = y$. The next free variable is C. We calculate

$$\begin{aligned} P(C \mid A = y, B = n, D = y, E = n) &= P(C \mid A = y, D = y, E = n) \\ &= (0.996, 0.04). \end{aligned}$$

We draw from the random generator, and assume we keep $C = y$.

In general the calculation goes as follows. Let A be a variable in a Bayesian network *BN*, let $B_1, \ldots, B_n$ be the remaining variables, and let $b^* = (b_1, \ldots, b_n)$ be a configuration of $(B_1, \ldots, B_n)$. Then $P(A, b^*)$ is the product of all conditional tables of *BN* with B_i instantiated to b_i. Therefore $P(A, b^*)$ is proportional to the product of the tables involving A, and $P(A \mid b^*)$ is the result of normalizing this product. Note that the calculation of $P(A \mid b^*)$ is a local task.

Back to the example. The next variable is D. We follow the same procedure and assume that the result is $D = y$. Then the configuration from the first sweep is unaltered, i.e. *ynyyn*.

The next sweep follows the same procedure. Assume the result for A is that the state is changed to n. Then we shall calculate $P(C \mid A = n, D = y, E = n)$, and so forth.

In this way a large sample of configurations consistent with the observations are produced. The question is whether the sample is representative for the probability distribution. It is not always so. It may be that the initial configuration is rather improbable, and therefore the first samples, likewise, are out of the mainstream. Therefore you usually discard the first 5–10% of the samples. It is called *burn-in.*

Another problem is that you may be stuck in certain "areas" of the configurations. Perhaps there is a set of very likely configurations, but in order to reach them from the one you are in, a variable should change to a state which is highly improbable given the remaining configuration (see Exercise 4.13).

A third serious problem is that it may be very hard to find a starting configuration. In fact, it is *NP*-hard (see Exercise 4.14).

We shall not deal with these problems, but refer the interested reader to the literature.

4.7 Summary of Sections 4.2–4.5

Junction trees

The nodes of a junction tree are sets of variables, they are called *cliques.* Each link is labelled with a *separator* which is the intersection of the adjacent cliques. Each clique and separator holds a real numbered table over the configurations of its variable set.

The junction tree property. For each pair V, W of cliques, all cliques on the path between V and W contain the intersection $V \cap W$.

A junction tree is said to *represent* the Bayesian network *BN* over the variables U if:

(i) for each variable A, there is a clique containing $pa(A) \cup \{A\}$;

(ii) $P(U)$ is the product of all clique tables divided by all separator tables.

Construction of junction trees

Let *BN* be a Bayesian network over the variables U.

(i) Construct the *moral graph:* the undirected graph with a link between all variables in $pa(A) \cup \{A\}$ for all A.

(ii) *Triangulate* the moral graph: add links until all cycles consisting of more than three links have a chord.

(iii) The nodes of the junction tree are the cliques of the triangulated graph.

(iv) Connect the cliques of the triangulated graph with links such that a junction tree is constructed.

(v) First give all cliques and separators a table consisting of only ones. Then, for each variable A find a clique containing $pa(A) \cup \{A\}$, and multiply $P(A \mid pa(A))$ on its table.

The resulting junction tree represents *BN*.

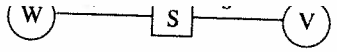


Figure 4.25 W absorps from V. $\mathbf{t}_W^* = \mathbf{t}_W \cdot \frac{\mathbf{t}_S^*}{\mathbf{t}_S}$, $\mathbf{t}_S^* = \sum_{V \setminus S} \mathbf{t}_V$.

Findings

A finding is a statement that some states of a variable are impossible. A finding can be represented as a table of zeros and ones with a zero at the places for impossible states.

A finding on a variable A is entered into a clique V containing A by multiplying Vs table by the table for the finding.

Absorption in junction trees

Definition. Let V and W be neighbours in a junction tree, let S be their separator, and let $\mathbf{t}_V$, $\mathbf{t}_W$ and $\mathbf{t}_S$ be their tables. The operation *absorption* is the result of the following procedure:

- calculate $\mathbf{t}_S^* = \sum_{V \setminus S} \mathbf{t}_V$;
- give S the table $\mathbf{t}_S^*$;
- give W the table $\mathbf{t}_W^* = \mathbf{t}_W \frac{\mathbf{t}_S^*}{\mathbf{t}_S}$.

We then say that W has *absorbed* from V. (See Fig. 4.25.)

HUGIN propagation

An arbitrary clique Rt in the junction tree is chosen as a root. The operation *CollectEvidence* is called in Rt followed by a call of *DistributeEvidence* in Rt.

CollectEvidence(Rt) asks all neighbours to *CollectEvidence* and they proceed down the tree recursively. When all the called neighbours have finished, Rt absorbs from them.

DistributeEvidence(Rt) makes all its neighbours absorb from Rt, and afterwards recursively *DistributeEvidence* to its neighbours (except Rt). See Figure 4.26.

Correctness of HUGIN propagation

Theorem 4.8 Let BN be a Bayesian network representing $P(U)$, and let T be a junction tree corresponding to BN. Let $e = \{f_1, \ldots, f_m\}$ be findings on the variables $\{A_1, \ldots, A_m\}$. For each i find a node containing A_i and multiply its table with f_i.

Then, after a full round of message passing we have for each node V and separator S that

$$\mathbf{t}_V = P(V, e) \qquad \mathbf{t}_S = P(S, e) \qquad P(e) = \sum_V \mathbf{t}_V.$$

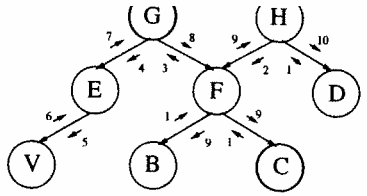


Figure 4.26 Updating through *CollectEvidence*(V) followed by *DistributeEvidence*(V).

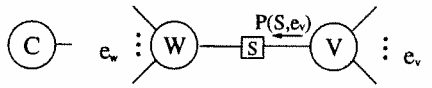


Figure 4.27 Evidence e_V has been entered at the righthand side of S. e_W has been entered at the lefthand side of S. C is used as a root for the propagation.

Side effect of Hugin Propagation

Let Rt be the root for HUGIN propagation, and let W and V be neighbours with separator S. Assume that W is closer to Rt than V. Then S divides the entered evidence in e_V and e_W (see Fig. 4.27).

A call of *CollectEvidence*(Rt) results in the table $P(S, e_V)$ being communicated from V to S. By marginalization you can calculate $P(e_V)$.

4.8 Bibliographical notes

A version of probability updating in singly connected DAGs through message passing was presented by Kim & Pearl (1983). HUGIN propagation was proposed by Jensen et al. (1990). It is a modification of an algorithm proposed by Lauritzen & Spiegelhalter (1988). Similar methods were used for pedigree analysis by Cannings et al. (1978). Shafer & Shenoy (1990) propose a different message-passing method for junction trees. Other propagation methods for multiply connected DAGs exist, e.g. arch reversal proposed by Shachter (1986) or conditioning proposed by Pearl (1986a).

The concepts of triangulated graphs and junction trees have been discovered and rediscovered with various names. In Bertele & Brioschi (1972) they are used for dynamic programming, and Beeri et al. (1983) use them for data base management. A good reference on triangulated graphs is Golumbic (1980). Tarjan & Yannakakis (1984) gives various triangulation methods and very efficient methods for testing whether a graph is triangulated. Jensen & Jensen (1994) contains a proof of Theorem 4.10 together with a method for constructing optimal junction trees from triangulated graphs.

Forward sampling was proposed by Henrion (1988). Gibbs sampling was originally

sampling methods could be Geyer (1992), Fung & Favero (1994), and Jensen et al. (1995). Gilks et al. (1994) have developed a system, BUGS, for Gibbs sampling in Bayesian networks.

Exercises

Exercise 4.1 For Table 4.6, calculate $\mathbf{t}_V\mathbf{t}_W$ and $\frac{\mathbf{t}_W}{\mathbf{t}_V}$.

Table 4.6 Table for Exercise 4.1.

	a_1	a_2	a_3
b_1	1	2	3
b_2	3	2	1

$\mathbf{t}_V$

	c_1	c_2	c_3
b_1	6	12	24
b_2	18	6	12

$\mathbf{t}_W$.

Exercise 4.2 For the universe U over the ternary variables (A, B, C) with the joint probability Table 4.7 we get the findings f_1: "A is in state a_1", and f_2: "C is in state c_1 or c_3".

Table 4.7 Table for Exercise 4.2.

	a_1	a_2	a_3
b_1	(2,4,3)	(1,4,8)	(5,0,7)
b_2	(5,10,4)	(2,3,3)	(1,5,4)
b_3	(1,5,6)	(3,3,3)	(0,6,2)

$P(A, B, C)$ multiplied by ten.

Calculate $P(B \mid f_1, f_2)$, $P(C \mid f_1, f_2)$, $P(f_1)$, $P(f_2)$ and $P(f_1, f_2)$.

Exercise 4.3 Prove that the anarchistic message passing algorithm formulated in Section 4.3.2 never runs into a deadlock: as long as there are unused message channels at least one variable can send a message. (Hint. Induction on the number of nodes and the fact that any sending sequence must start with a leaf sending.)

Exercise 4.4 Let B be independent of C given A, and let $P(A, B)$ and $P(A, C)$ be consistent. What is $P(A, B, C)$?

Exercise 4.5 Prove that a call of *CollectEvidence* in any node followed by a call of *DistributeEvidence* in the same node will result in a full propagation (all messages passed and passed when permitted).

Exercise 4.6 Construct the moral graph and a junction tree for the singly connected DAG below.

Exercise 4.7 Show that a consistent junction tree is globally consistent.

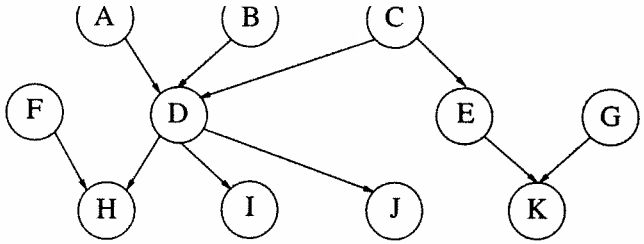


Figure for Exercise 4.6.

Exercise 4.8 (Construction of a junction tree from an elimination sequence.)
G is a triangulated graph over U, and $A_1, \ldots, A_n$ is an elimination sequence of U. C_i is the set of variables containing A_i and all its neighbours at the time of elimination.

(i) Show that each clique of G is a C_i for some i.

(ii) Show that for all $i < n$ there is a $j > i$ such that $C_i \setminus \{A_i\} \subseteq C_j$.

(iii) Assume that C_i and C_j are cliques $(i < j)$ such that $C_i \setminus \{A_i\} \subseteq C_j$. Show that there exists a junction tree for G with the link (C_i, C_j).

(iv) Use (ii) and (iii) to construct a junction tree for the graph in Figure 4.20(a).

Exercise 4.9 (i) Construct a junction tree for the DAG given below, by using the elimination order F, J, D, B, A, I, K, E.

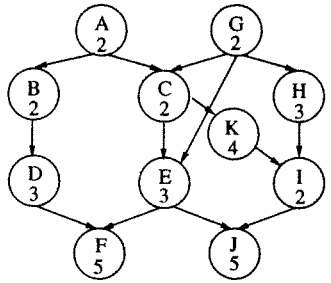


Figure for Exercise 4.9

(ii) The numbers inside the nodes indicate the number of states. Use the procedure from the end of Section 4.5 to construct a junction tree.

Exercise 4.10 (i) For the DAG given below, compute $P(A, B, C)$, when $P(A) = (0.3, 0.7)$ (see Figure and Table 4.8 for Exercise 4.10(i)).
(ii) The DAG is extended as shown in the Figure and Table 4.9 for Exercise 4.10(ii). Calculate $P(B, C, D)$.

	$A = y$	$A = n$
$B = y$	0.2	0.5
$B = n$	0.8	0.5

$P(B \mid A)$

	$A = y$	$A = n$
$C = y$	0.9	0.4
$C = n$	0.1	0.6

$P(C \mid A)$

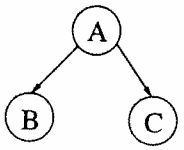


Figure for Exercise 4.10(i).

(iv) We are told that $A = y$ and $D = n$. What is $P(B)$?
(v) Initially, what was $P(A = y, D = n)$?

Exercise 4.11 (Conditioning.) Propagation methods for singly connected DAGs have existed for a long time. A propagation method for multiply connected DAGs consists of reducing a DAG to a set of singly connected DAGs.

(i) Consider the DAG (a) below with $P(A)$, $P(B \mid A)$, $P(C \mid A)$ and $P(D \mid B, C)$ given. Assume that $A = a$. Show that the DAG is reduced to the DAG (b) with $P(B \mid a)$, $P(C \mid a)$, and $P(D \mid B, C)$ given.

(ii) Show that $P(D, a) = P(D \mid b, c)P(B \mid a)P(C \mid a)$.

(iii) Assume that for all states a of A we have a reduced DAG as in (i) . Let evidence e be entered and propagated in all the reduced DAGs, yielding $P(B, e \mid a)$, $P(C, e \mid a)$, $P(D, e \mid a)$ for all a. Calculate $P(B, e)$ and $P(A, e)$.

The procedure above is called *conditioning on A*.

(iv) Reduce the DAG by conditioning on B. Show that the tables are $P(A \mid b)$, $P(C \mid A)$ and $P(D \mid C, b)$.

(v) Show that conditioning on D does not result in a singly connected DAG.

Conditioning over several variables can be performed stepwise.

(vi) Determine a minimal set of conditioning variables for the DAG given below to reduce it to singly connected DAGs.

(vii) The numbers attached to the variables indicate the number of states. Determine a conditioning resulting in a minimal number of singly connected DAGs.

Table 4.9 Table for Exercise 4.10(ii).

	$B = y$	$B = n$
$C = y$	(0, 1)	(0.7, 0.3)
$C = n$	(0.4, 0.6)	(0.5, 0.5)

$P(D \mid B, C)$

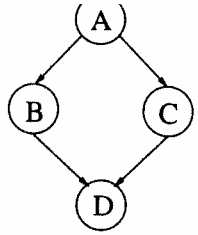


Figure for Exercise 4.10(ii).

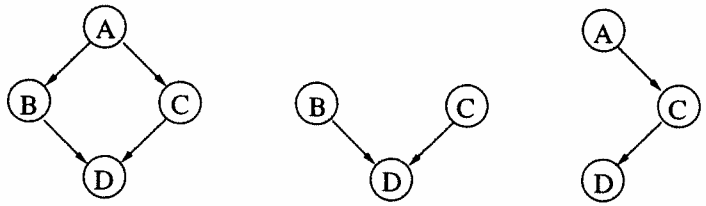


Figure for Exercise 4.11(i)–(v).

Exercise 4.12 Calculate the marginals from the sample in Table 4.5, and compare the result with the exact marginals.

Exercise 4.13 The binary variables A and B are parents of the binary variable C. $P(A) = P(B) = (0.5, 0.5)$, and the conditional probability table is an *exclusive or table*: $C = y$ if and only if exactly one of A and B is in the state y.

Show that Gibbs sampling on this structure will give either $P(C = y) = 1$ or $P(C = n) = 1$.

Exercise 4.14 Given a Bayesian network over U with evidence e entered, show that it is *NP*-hard to find a configuration U^* such that $P(U^*, e) > 0$.
(Hint. Look at Exercise 3.16.)

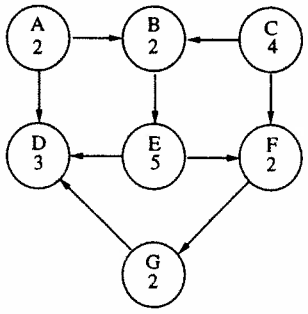


Figure for Exercise 4.11(vi)–(vii).