COMP4702/7703 – Machine Learning

The following lecture notes were developed by Prof. Tom Downs over a number of years in the School of ITEE and its predecessor Department of Electrical and Computer Engineering at The University of Queensland. They are provided unedited as a primary resource for this course.

Lecture 10 Bayesian Belief Networks

In the previous lecture, I introduced the basic ideas underlying the Bayesian approach to learning systems and we went on to look at a simplified version of the Bayesian method, namely the Naïve-Bayes classifier. In this lecture we will see how the Naïve-Bayes concept can be incorporated into network structures to allow decision-making in complex, uncertain situations, by drawing upon whatever data are available and using expert opinion where data are lacking. Although the Naïve-Bayes classifier often gives good results (especially for smallish data sets) it is clear that the assumption that all features are independent is a sweeping one. In Bayesian belief networks the assumption of independence is only applied to subsets of the features, giving a method that is both computationally tractable and generally more accurate than Naïve-Bayes.

10.1 Historical development

It has been known for a long time that the Bayes approach can be used in situations where little or no data exist. In such situations, you can set your prior probabilities using expert opinion and then use Bayes' rule to update these probabilities (by computing the posterior) whenever any new data become available. Probabilities based upon expert opinion are called *subjective* probabilities and have been the main source of disagreement between classical statisticians and Bayesians with the classicists being difficult to convince that subjective probabilities are probabilities at all. Nevertheless, as we shall see, decision-making that makes use of subjective probabilities has been quite widespread in industry and this, of course, is an indication that such methods are both very effective and very useful.

It was recognized as early as the 1950's that Bayes' rule provided a basis for the development of automated systems for diagnosis, ie establishing the cause of malfunctions in complex systems. These systems might be human beings (in which case we are talking about medical diagnosis) or they might be mechanical, electrical or some other complex engineering system. Whatever the system type, we will have a set O of observations o_i and a set H of hypotheses, h_j. In the medical case, the h_j are possible illnesses and the o_i are symptoms and for simplicity here, we will assume that all o_i and h_j are binary. What this means, in the case of symptoms, is that they are either present or not. We are ignoring such things a mild or partial symptoms, so the patient either has the thing or not, ie true or false. The same applies to illnesses. I'll stick with the medical diagnosis theme for a while so that I'm not talking in the abstract all the time.

A diagnosis D_k is a subset of the set H of hypotheses. There may be more than one hypothesis in the diagnosis because patients may have more than one thing wrong with them. To use Bayes' rule, the medical practitioner would need to supply a prior probability $P(D_k)$ over all diagnoses based upon the knowledge they have accrued over the years and an assessment of the patient. The medical practitioner may know, or suspect, that the patient has some particularly naughty habits and this can be factored into the prior distribution. As a relatively mild example, the doctor may know that the patient is a heavy drinker so that a liver problem may be the prime cause of the patient's difficulties. Relevant liver complaints would then be given suitably high probabilities in $P(D_k)$. Note that the distribution of the D_k is discrete, and so is specified by a particular probability for each k.

Now suppose that the patient has been sent off to have blood and other tests done and the doctor receives the reports on these. This gives the doctor new data that can be used to update the prior distribution. Let us call the new data *evidence* and give it the symbol E. The long history of medical practice has provided a

great deal of data over the years and in more recent times it has begun to be properly recorded. So, it is possible to conceive of having available distributions of the form $P(E \mid D_k)$, ie distributions showing the probability that the tests yield evidence E given that the true diagnosis is D_k. If the doctor had such distributions available for each possible diagnosis, they could be used to update the prior distribution using Bayes' rule in the form:

$$P(D_k \mid E) = \frac{P(E \mid D_k)P(D_k)}{\sum_k P(E \mid D_k)P(D_k)} \qquad (10.1)$$

Given the material in the previous lecture, this should be a familiar equation. But, familiar or not, it is generally difficult to compute for the kind of problem we are considering, given the way we have formulated it. The main difficulty comes from the fact that a patient may have more than one disease out of, say, n possible diseases and, in such a case, the number of possible diagnoses (ie disease combinations) is 2^n. This is the number of disease combinations if we allow for the possibility that the patient has the full set of n, reminding us of the old joke "what do you give the man who has everything?" If we allow for the possibility that the patient has the full set of diseases, this means that the number of probabilities that have to be specified for the complete prior distribution $P(D_k)$ is 2^n-1 (for n=1, you have to specify one probability $P(D_1)$, for n=2, you have to specify $P(D_1)$, $P(D_2)$ and $P(D_{12})$, etc.).

Now, the tests that the patient was sent away for will yield evidence E, and if E involves measurements of m different things, then E could occur in $2^m - 1$ different combinations. Given the number of combinations that D_k can occur in, this means that there are $(2^m-1)(2^n-1)$ different $P(E \mid D_k)$, each of which needs to be estimated in order to make use of (10.1) in all possible circumstances. Clearly, for increasing values of m and n, the number of combinations rapidly becomes enormous, indicating that it's impractical to use (10.1) and that some sort of simplification needs to be made. This was certainly true back in the early 1960's when people were first contemplating this sort of technique - the fastest mainframe computer back then would be thrashed by today's hand-calculators.

So, back in the early days, some fairly sweeping simplifications had to be made. The two that were most often made were to assume that

(i) the possible illnesses h_j are mutually exclusive, ie that no patient has more than one disease, and

(ii) the test measurements (which provide the observations o_i making up the evidence E) are conditionally independent, meaning that

$$P(E \mid h_j) = \prod_j P(o_j \mid h_j) \qquad (10.2)$$

which you should recognize as the Naïve-Bayes assumption.

With these assumptions, the number of conditional probabilities that have to be estimated reduces to $m \times n$ and the number of prior probabilities needed reduces to $n-1$.

<u>Question.</u> Why is the number of priors needed $n-1$ and not n?

Several systems for medical diagnosis were built using the above simplifying assumptions and, quite surprisingly, they performed extremely well. For instance, a system for diagnosing abdominal pain [1] averaged 90% correct diagnoses, where "expert" medical practitioners were able to average only 65% - 80%. Similarly, a diagnostic aid for chest pain (described in [2]) averaged 80% accuracy where our esteemed members of the medical profession were averaging only 51%. It may well be that part of the explanation for these performance levels lies in the fact that the simple mathematical model, unlike the human diagnostician, is not subject to whims, carelessness or misguided inspiration but be that as it may, outstanding performance was insufficient for general acceptance by the medical profession.

I think there is little doubt that the medical profession felt threatened by these developments. I can remember my own GP back in the 1970s asking me about computer courses because he felt that he needed to know more about computer systems and their capabilities. However, the medicos seem somehow to have

convinced everyone that the formal mathematics on which the Bayesian approach is based is so different from the qualitative approach of human reasoning that it wasn't possible to explain its results in a way that users (ie the medicos themselves) could understand and trust them. In recent times the same sort of argument has been trotted out to prevent neural networks gaining wide acceptance in medical practice.

So, in spite of excellent performance, systems of naïve-Bayes type fell from favour and their place was taken by *expert systems*, which are based upon *production rules*, ie rules of the IF . . . THEN . . . ELSE type. The rules are usually put together by means of lengthy question-answer sessions with one or more experts in the area for which the expert system is to be developed. As a very simple example, let's consider an expert system that has been developed for trouble-shooting in motor cars. A user feeds his problem into the computer which guides him/her through a test sequence by means of an interactive dialogue that might run something as follows:

Computer: What's the problem?

User: The air-conditioning system doesn't work.

(The computer turns to a list of rules associated with air-conditioning failure.)

Computer: Does the car have a thermal limiter?

User: No.

Computer: Connect a test light to the clutch wire. What is the status of the test light?

User: Off.

Computer: The fuse is blown.

The list of rules that the computer has drawn upon here will contain something like the following:

Rule 002: IF car has a thermal limiter AND thermal limiter does not have power, THEN fuse is blown.

Rule 008: IF clutch does not have power AND car does not have thermal limiter THEN fuse is blown.

Rule 017: IF test light connected to clutch wire AND test light is off THEN clutch does not have power.

Pretty simple stuff. But some relatively sophisticated and quite widely-used expert systems were developed in the late 1970s and early 1980s. Two of the best-known are *MYCIN* [3], which was produced to assist medicos in the diagnosis and treatment of bacterial infection, and *PROSPECTOR* [4], which was developed as an aid for the identification of mineral deposits. For applications such as these, it isn't possible to use iron-clad rules like the ones in the above example. Instead, you have to incorporate probabilities in some way to account for uncertainties in relation to predictions made by the system. MYCIN and PROSPECTOR both employed probabilities which were associated with predictions and which could be updated whenever additional data became available. Thus, for instance, for a given patient, MYCIN might, on the basis of test results, predict a variety of different diseases, but each with a particular probability. And then, if additional test results become available, the probabilities would be adjusted. Although they achieved some prominence, these methods eventually came in for criticism and fell into disuse because the way they handled probabilities was shown to contain deficiencies and inconsistencies – see, for instance [5].

What was needed was a method that allowed for richer knowledge representations that employed probabilities in a principled and more acceptable way.

10.2 Bayesian Belief Networks

In the previous lecture, I described the naïve-Bayes method and explained the underlying, and sweeping, assumption it makes regarding independence of features. We saw that, in conjunction with the method of boosting, some excellent pattern classification results could be achieved using naïve-Bayes and we

have just seen in the previous section that in the 1970s, methods making the same sweeping assumptions were able to achieve high-quality performance even without the use of boosting. But, as I pointed out, these methods fell into disuse and one of the reasons for this was suspicion about the independence assumption. Expert systems then appeared on the scene but did not stay long, largely because they failed to handle probabilities in a satisfactory way. Bayesian belief networks were the next significant development and they can be considered a generalization of both the expert system and the naïve-Bayes technique.

For some reason, the writers of elementary introductions to Bayesian networks like to use little story lines involving Holmes and Watson – that's Sherlock Holmes for the uninitiated. To save time for myself, I'll just draw on one or two of their stories to begin with. So here goes. The first example is taken from [6].

Police Inspector Smith is impatiently awaiting the arrival of Mr Holmes and Dr Watson. They are late and Inspector Smith has another important engagement (lunch). Looking out of the window he wonders if 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 Watson has had a car accident. . . .

We'd better formalize things before I get carried away and start trying to do a Conan Doyle. Let the events in this story be represented by variables that have the two states *yes* and *no*. Let these events and variables be *icy roads* (*I*), *Holmes crashes* (*H*) and *Watson crashes* (*W*). Let each event have an associated probability. The event *I* will have the effect of increasing the probability of both events *H* and *W*. The influence of *I* on *H* and *W* can be represented in a diagram, as depicted in Fig. 10.1.

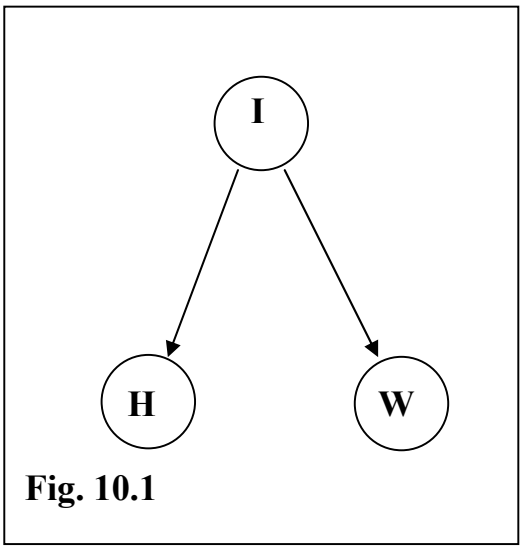

Fig. 10.1

The graph reflects the way that the event I is an independent variable which has influence on *H* and *W*. The probabilities associated with the three nodes are P(*I*), P(*H* | *I*) and P(*W* | *I*). This is a very elementary Bayesian network but it does introduce a few of the basic concepts. Notice that there is no connection from H to W which indicates that the two have no influence on one another. But they are not completely independent because both Holmes and Watson are more likely to crash when it is icy (so their probabilities of crashing are correlated in this way). But they *are* independent once the state of node I is known. And that is what graphs like the one in Fig. 10.1 represent. It is called *conditional independence*. The probabilities of Holmes and Watson crashing are independent once the state of node I is known.

The police inspector has been looking out of the window and wondering whether the roads are icy. We will set the probability of icy roads at 0.7. Now, both Holmes and Watson are well-known to be bad drivers, so we'll put the probability of a crash in the case of icy roads at 0.8 (for both of them). And for the case where the roads are not icy, we'll put the probability of a crash at 0.1 for both (they really are bad drivers).

Let us calculate the probability that Holmes or Watson crashes before the secretary comes in and announces that Watson has had an accident. The probability will be the same for both Holmes and Watson because they have been assigned the same probabilities. So, we'll work out the information we want for Watson only. The probability that the roads are icy *and* Watson crashes is given by

$$\mathrm{P}(WI) = \mathrm{P}(W \mid I)\mathrm{P}(I) = 0.8 \times 0.7 = 0.56.$$

Other combinations are calculated similarly. The outcome is summarized in Table 10.1.

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

Table 10.1 Joint probabilities, P(*WI*) and P(*HI*)

We now want the probability of a crash by either driver regardless of whether the roads are icy. This is a *marginal* probability that we obtain by summing the probabilities of a crash on either icy or non-icy roads. This marginal probability is therefore equal to 0.56 + 0.03 = 0.59.

The information that Watson has crashed can now be used by Inspector Smith to update his belief that the roads are icy. To do this, the Inspector will need to know Bayes' rule (but that's a prerequisite for promotion to Inspector, isn't it?). Whatever, the update takes the form

$$P(I|W) = \frac{P(W|I)P(I)}{P(W)} = \frac{1}{0.59}0.8\times 0.7 = 0.95.$$

Having updated our belief that the roads are icy, we can now update our belief that Holmes will have crashed, too. We will use the standard formula P(*AB*) = P(*A* | *B*)P(*B*) in the form

$$\mathrm{P}(HI \mid W) = \mathrm{P}(H \mid I,W)\mathrm{P}(I \mid W) = 0.8\times 0.95 = 0.76$$

And, as before we compute the other combinations as well to give Table 10.2.

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

Table 10.2 Calculation of P(*HI* | *W*)

Then, finally, we can obtain P(H) by marginalizing, ie summing the probabilities relating to H = y. So we sum the first row to get P(H) = 0.765. The information that Watson crashed has allowed us to update the probability that Holmes will crash, too, and this has increased from 0.59 (calculated just after Table 10.1) to 0.765. Before we could update the H node, we had to first update the I node (the probability that the roads are icy). This is the way Bayesian networks operate – as new information comes in, relating to any node in the network, that information can be used to update prior beliefs, first in nodes neighbouring that node, then in nodes neighbouring those nodes and so on, until every node in the network has been updated.

We used *marginalization* to obtain P(*H*), ie we summed over the probabilities relating to *H* = *y*. In this way, we obtained the probability of Holmes crashing regardless of the value of *I*. Summing in this way is the discrete-variable equivalent of integrating out a variable in the continuous case. The term marginalization comes from the fact that before computers and hand-calculators, most probability calculations used to be done with pencil and paper and if you were dealing with variables that had several possible values (rather than the binary ones in Table 10.2) you would add up all the values in a row and put the result in the margin. I hasten to add that I am not speaking from experience here, they *did* have computers when I started working.

In doing these calculations, we have just been using Bayes' rule and a few other elementary probability operations so you might well be wondering where the network structure comes into it. Well, we were able to deal with this problem using the obvious formulae because it was so simple, but for more complex problems, the network structure helps guide and simplify the computations. The important thing in the example we've just done is that it demonstrates how you can revise your beliefs about different propositions whenever new information becomes available and this is where Bayesian networks become very useful. They are often used to assist with planning for the future where educated guesses have to be made for most of the probabilities. As an example, when I was in England in 1999, I visited the research headquarters of one of the major banks and they were using Bayesian networks to identify the most suitable branches to shut down in order to reduce their bank branch numbers by 20%. They needed probability estimates for such things as likelihood of loss of custom –

what were the probabilities that this would be small, medium or large – the probability of bad publicity – this would normally be greater if a branch in an affluent area were shut down, and so on. They then had an enormous Bayesian network to cover all their branches and they could use the network to observe the effects of different branches being closed down. On the basis of these effects, they were able to identify the best set of branches to close.

It might seem a little outlandish to make these sort of decisions on the basis of educated guesses for probabilities, but in this case, the probabilities were arrived at by pooling the guesses of a good number of banking experts. There is a wide-held view that the accuracy of educated guesses tends to increase as more and more guesses are included. This is the basis of the so-called *Delphi method* (named after the oracle at Delphi through which the ancient Greeks sought the advice of Apollo) and we might try it out when we do this lecture.

Example 2 (adapted from [7])

This example leads to a slightly more complex Bayesian network which requires a little more from the probability toolkit.

Mr Holmes receives a telephone call from his neighbour Dr Watson who tells him that he can hear a burglar alarm from the direction of Mr Holmes's house. Preparing to rush home, Mr Holmes recalls that Dr Watson is known to be a tasteless practical joker and he decides that he will first call his other neighbour, Mrs Gibbons who, despite occasional drinking problems, is far more reliable.

We must now construct a network to represent this situation. Clearly the event "burglary" (B) has direct influence on whether or not the alarm (A) sounds, so we should have an arrow directed from B to A. And what the two individuals have to say depends on whether an alarm is sounding and not on whether there has been a burglary. So the Bayesian network has the form shown in Fig. 10.2 where G represents Gibbons' testimony and W represents Watson's.

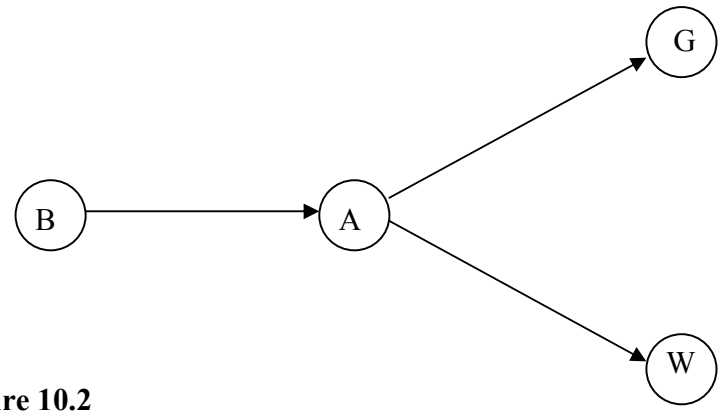

Figure 10.2

Local crime statistics will provide a probability for $P(B)$ the probability of a burglary taking place at a house in his neighbourhood on any given day. We can assume that Holmes would know P(B), but he now Holmes needs to update it in the light of the testimonies of Mrs Gibbons and Dr Watson. To do this he will need to assign probabilities to the accuracy of those testimonies given Watson's predilection for practical joking and Gibbons' weakness for the demon drink. Once he has assigned these probabilities, he will need to apply Bayes' rule to update his beliefs about event B. (And, of course, we can be confident that Holmes would be familiar with Bayes' rule.) The event B has two states – burglary (B = b1) and no burglary (B = b2) so Holmes would apply Bayes' rule in the form

$$P(B = b_1 | G,W) = \frac{P(G,W | B = b_1)P(B = b_1)}{\sum_{j=1}^{2} P(G,W | B = b_j)} \tag{10.3}$$

and since the denominator in (10.3) is only there to make sure that probabilities add to unity, (10.3) can be written

$$P(B \mid G, W) = \alpha P(G, W \mid B)P(B) \qquad (10.4)$$

where α is a constant. As we shall see below, the value of α in a given problem can be determined directly from the probabilities in that problem.

But (10.4) doesn't involve the event A and surely this has to play a role. Equation (10.4) seeks to update P(B) by passing evidence from nodes G and W directly through node A. In a situation like this, where you are passing evidence through a node in a direction opposite to the arrows, you have to take account of all possible states that the middle node can be in. In this case, the alarm can be either on (A = a1) or off (A = a2). To incorporate the effect of this into (10.4) we must condition the probabilities on the right-hand side so we can sum over all states that A can be in (effectively so that we can sum out the effects of A). Thus (10.4) becomes

$$P(B \mid G, W) = \alpha P(B)\sum_j P(G, W \mid B, a_j)P(a_j \mid B) \qquad (10.5)$$

Now, look again at Fig. 10.2. It tells us that B influences A and that A influences both G and W. But how much influence does B have on G and W? Well, in statistical terms it has none at all because if we know the state of A (eg we know the alarm is on) the testimonies of Gibbons and Watson will depend on that fact alone. The fact that the house may have been burgled has nothing to do with their views on whether the alarm is sounding. What this means is

$$P(G, W \mid B, a_j) = P(G, W \mid a_j)$$

so (10.5) can be written

$$P(B \mid G, W) = \alpha P(B)\sum_j P(G, W \mid a_j)P(a_j \mid B) \qquad (10.6)$$

And we can simplify further because the fact that Gibbons and Watson are connected to node A in the manner shown in Fig. 10.2 means that they are conditionally independent given the value of A. That is

$$P(G, W \mid B, a_j) = P(G \mid a_j)\, P(W \mid a_j) \qquad (10.7)$$

and substituting in (10.6) gives

$$P(B \mid G, W) = \alpha P(B)\sum_j P(G \mid a_j)P(W \mid a_j)P(a_j \mid B) \qquad (10.8)$$

Let's put some numbers in to get a better idea of how to use this scheme. How do we obtain the numbers? Some are obtainable by collecting statistics over a period of time and others have to be based upon expert opinion. In this case, the ones that can be obtained by collecting statistics are P(B) and P(aj | B). The others have to be based on expert opinion, but since the opinions are coming from Sherlock Holmes, we can feel fairly comfortable that they will be close to the mark.

The probability that a given house will be burgled on any given day can be estimated from local crime statistics and ought to be fairly low, otherwise the occupants would probably move somewhere else. So let's assume the statistics indicate a 1 in 10,000 chance of Holmes's house being burgled on any particular day, which means he can expect to be burgled about once in every 27 years, so that makes him reasonably secure. Thus, if B = b1 when a burglary occurs, we set $P(B = b_1) = 10^{-4}$ so that the for the other state of B (no burglary, B = b2) we have $P(B = b_2) = 0.9999$.

The probability that the alarm will be triggered during a burglary (ie P(a1 | b1)) and the probability that it won't be (ie P(a2 | b1)) can be obtained from the alarm supplier. Of course, sales people have a tendency to oversell, but for this example let's assume we can obtain accurate figures and set P(a1 | b1) = 0.95 and P(a2 | b1) = 0.05.

As we shall see, although it is not explicitly required by (10.8), we also need the probability of the alarm being triggered when there is no burglary, ie $P(a_1 \mid b_2)$. This is needed so that we can correctly normalize the final outcome using the multiplier α. Let's assume that the probability of getting a false alarm is 0.01 so that $P(a_1 \mid b_2) = 0.01$.

The other probabilities have to be estimated on the basis of Holmes's judgement. For $P(G \mid a_1)$ he has to go by the conversation he has with Mrs Gibbons and, unfortunately, when he calls her, he soon realizes that she is somewhat the worse for drink. She goes on and on about an operation she just had and then starts revealing some personal details he'd rather not hear. When he finally manages to hang up and thinks about the few pertinent remarks she made, he estimates that there is an 80% chance that Mrs Gibbons could hear an alarm from her window. Thus he sets $P(G \mid a_1) = 0.8$, so we also have $P(G \mid a_2) = 0.2$.

As regards the phone call from Dr Watson, Sherlock Holmes feels that, in spite of his reputation as a hoaxter, Watson is unlikely to make a call of this kind to his personal friend and assigns a probability of 0.9 to $P(W \mid a_1)$. This means $P(W \mid a_2) = 0.1$.

Substituting in (10.8), we obtain

$$P(b_1 \mid G,W) = \alpha 10^{-4}[(0.8)(0.9)(0.95) + (0.2)(0.1)(0.05)] = 6{\cdot}85 \times 10^{-4}\alpha \qquad (10.9)$$

And

$$P(b_2 \mid G,W) = \alpha(0.9999)[(0.8)(0.9)(0.01) + (0.2)(0.1)(0.99)] = 0.027\alpha. \qquad (10.10)$$

We now determine α from the fact that these two probabilities should add to unity, so

$$\alpha = (6{\cdot}85 \times 10^{-4} + 0.027)^{-1} = 36.94.$$

Substituting for α in (10.9) and (10.10) gives

$$P(b_1 \mid G,W) = 0.00253, \quad P(b_2 \mid G,W) = 0.9974.$$

So, on the basis of the slightly dubious testimony of both Mrs Gibbons and Dr Watson, Holmes is able to conclude that the probability that his house is being burgled has increased by a factor of a little over 25, from the daily average (when there is no alarm) of 0.0001 to 0.00253.

Comments

The above examples are not in any sense practical problems, but they are able to illustrate some of the basic mechanisms involved in the application of Bayesian networks. Most importantly, they demonstrate how the Bayes approach allows "real" probabilities (obtained from statistics) to be mixed with subjective probabilities (which are based upon human judgement). Classical statistical methods simply do not allow for this.

They have also introduced the notion of conditional independence. The nodes W and H, in Fig. (10.1) are independent given the state of node I and the nodes G and W in Fig. 10.2 are independent once we know the state of node A.

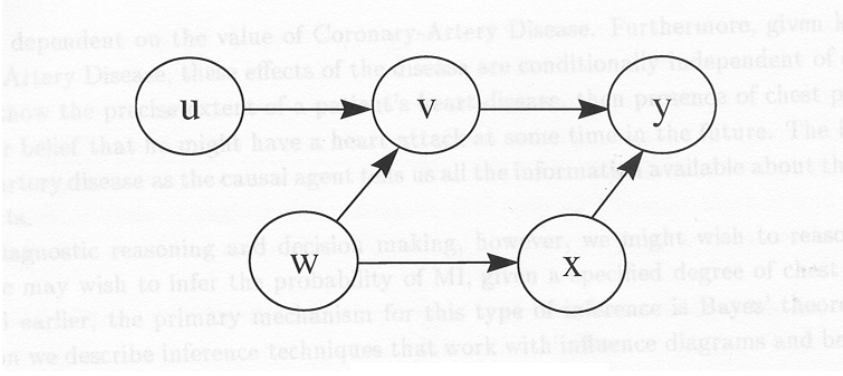

Figure 10.3

When it comes to practical problems, conditional independencies are quite easily and naturally identified and, importantly, they lead to a considerable simplification of the probabilistic representation of a given problem. For a five-variable problem like the one depicted in Fig. 10.3, if none of the variables were independent, we would have to somehow estimate the parameters of a five-variable distribution to describe the problem fully. In the case of binary variables, this would require estimation of $2^5 - 1 = 31$ different probabilities. But the independences indicated in the figure tell us that the overall distribution can be written

$$p(u,v,w,x,y) = p(y|v,x)p(v|u,w)p(x|w)p(w)p(u) \qquad (10.11)$$

and for this, the probabilities in the product are much easier to estimate because they contain less variables.

When it comes to updating probabilities in a Bayesian network, each node can be treated as a processor which maintains a set of probabilities relating to the variable that the node represents. In addition, the node manages the communication links that connect it to other nodes in the network. The communication links are assumed to be open at all times so a processor can pass messages to neighbouring nodes whenever its probabilities are updated.

What kinds of messages do the various nodes send to one another? Clearly love poems wouldn't be much use. But we can gain some appreciation of how a message-passing procedure might operate by looking back at example 2. When we wish to update P(B) based upon the new evidence from G and W, we need to implement (10.8). Now, each node stores its own probability plus the conditional probabilities relating to the nodes it is influenced by. Thus, for instance, node A stores its own probability P(A) (ie P(A = a1), P(A = a2)) plus the conditional probabilities P(A | B), ie P(A=a1 | B = b1), etc.

Node B can update its own probability if it is sent the summation term in (10.8). When probabilities are assigned to nodes G and W these can be sent, in the form P(G | aj) and P(W | aj) to node A which can use its own stored values for P(aj | B) to form the summation in (10.8) and send it on to node B. Node B can then implement (10.8) to update its probability.

It is by this type of message passing that updates can be spread through the network. We'll now illustrate message-passing in a further example. This one is also quite contrived, but practical examples tend to be too complicated for illustrating the method. At least this one doesn't involve Holmes and Watson.

Example 3 (Adapted from [8])

Suppose it is Saturday and one of our older research assistants (unemployable in the real world) is working in his windowless office as usual. He is looking forward to going to a baseball game later (yes, ref [8] was written by an American), but he has some concern about the likelihood of rain, which could lead to the game being cancelled. One of the research assistant's kids is at the beach and the research assistant is wondering if the child might get sunburn. This seems like a bit of confused thinking (will the game be rained off and will my child get sunburn?) but I told you it was written by an American. Anyway the situation can be represented by the belief network shown in Fig. 10.4.

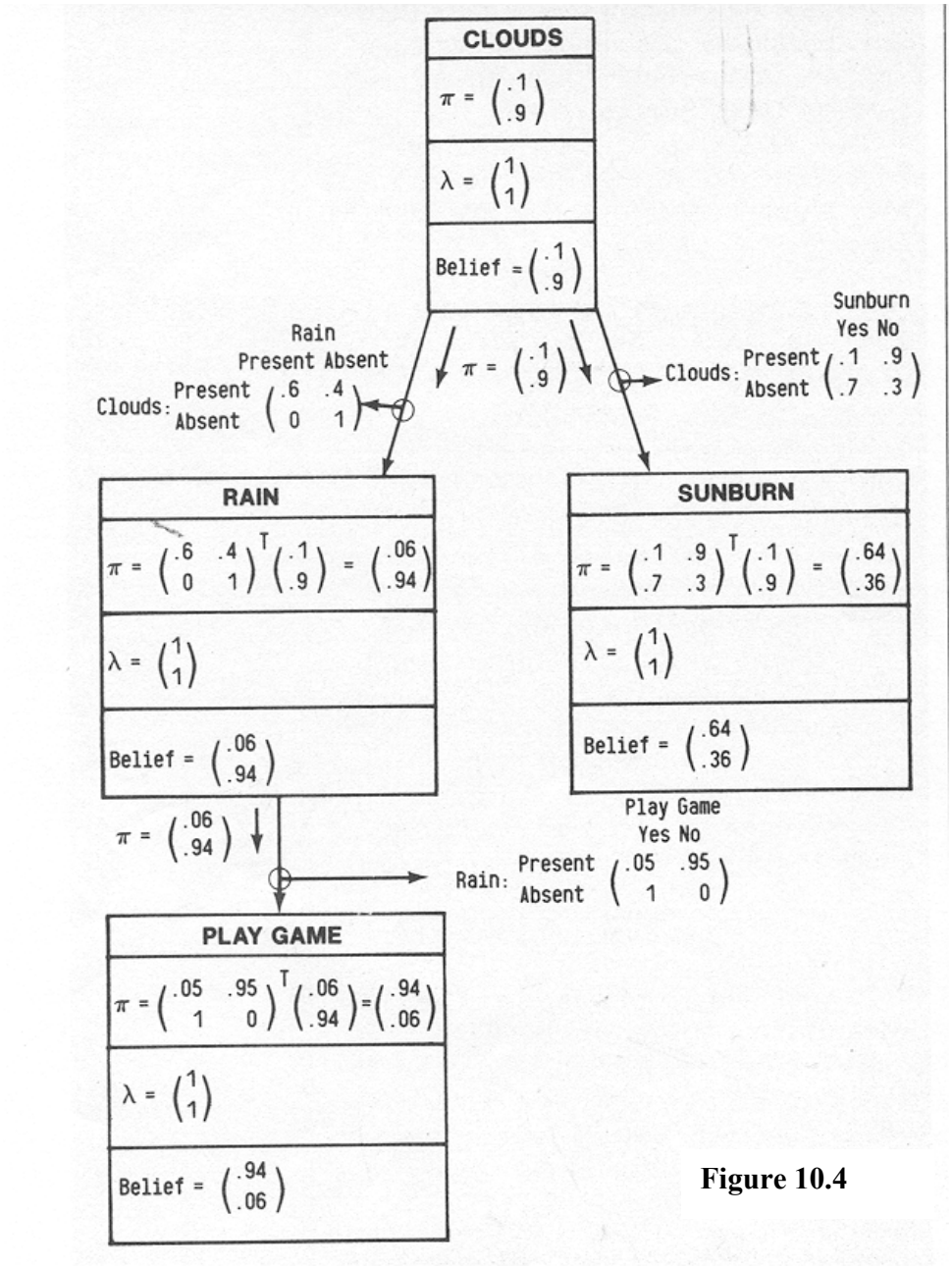

Figure 10.4

The root node contains the prior probability of cloud in the area where the research assistant lives and works. The weather forecast has indicated that it can be expected to be cloudy 10% of the time so, if $C=c_1$ represents cloud and $C=c_2$ represents no cloud, we have $P(c_1) = 0.1$ and $P(c_2) = 0.9$ stored in the root node. To initiate computations, this probability vector is passed as a message to the nodes for rain and sunburn. Messages that are sent in the direction of the arrows are distinguished from those that are sent in the opposite direction to the arrows. Thus, the probability vector sent from the cloud node to the rain node is called a π-message (as are any messages sent in the direction of the arrows) and any message sent in the opposite direction to the arrows is called a λ-message. This is standard notation for this method.

The rain node stores the conditional probabilities for rain (r_1) and no rain (r_2) given the presence of cloud (c_1) and the absence of cloud (c_2). These conditional probabilities are fixed and simply determined by average weather behaviour. The rain node makes use of the probability vector sent to it by the cloud node to determine the probability of rain. The computations take the following form:

$$P(r_1 \mid c_1)P(c_1) + P(r_1 \mid c_2)P(c_2) \;=\; \mathrm{P}(r_1c_1) + P(r_1c_2) \;=\; P(r_1)$$

$$P(r_2 \mid c_1)P(c_1) + P(r_2 \mid c_2)P(c_2) \;=\; \mathrm{P}(r_2c_1) + P(r_2c_2) \;=\; P(r_2)$$

The computation can therefore be represented as a matrix of conditional probabilities multiplying the vector of probabilities sent from the cloud node. In Fig. 10.4, the matrix is shown alongside the link between the cloud node and the rain node and the outcome of the matrix multiplication is shown inside the rain node. This gives us a new probability vector that the rain node can pass as a π-message to its daughter (the "play game" node) to allow computation of beliefs regarding whether the game will be played. The meaning of the matrix of conditional probabilities that is used (and stored in the "play game" node) is shown to the right of the link entering the node. The sunburn node computes the probability of the child getting sunburn using the same approach.

Now suppose that the research assistant has a rain alarm and that it goes off during the afternoon. Like most alarms, it won't be entirely reliable, so let's suppose P(alarm | rain) = $P(a_1 \mid r_1)$ = 0.8 and P(alarm | no rain) =$P(a_1 \mid r_2)$ = 0.04. This requires another node in the Bayes network and this is shown as the circle to the right of the rain node in Fig. 10.5. The rain alarm is obviously influenced by the rain node so the arrow between the two is directed toward the alarm node. Thus, when the alarm node sends its probabilities to the rain node, the message goes against the direction of the arrow and so constitutes a λ-message.

Bayes' theorem then gives, using our beliefs about rain contained in the rain node of Fig. 10.5:

$$P(R \mid A) = \alpha P(A \mid R)P(R) \qquad (10.12)$$

where A represents the state of the alarm node. In particular this gives

$$P(r_k \mid a_1) = \alpha P(a_1 \mid r_k)P(r_1)$$

from which $P(r_1 \mid a_1)$ = 0.56 and $P(r_2 \mid a_2)$ = 0.44. The new evidence has changed our belief about rain and the new belief has been inserted in the rain node in Fig. 10.5.

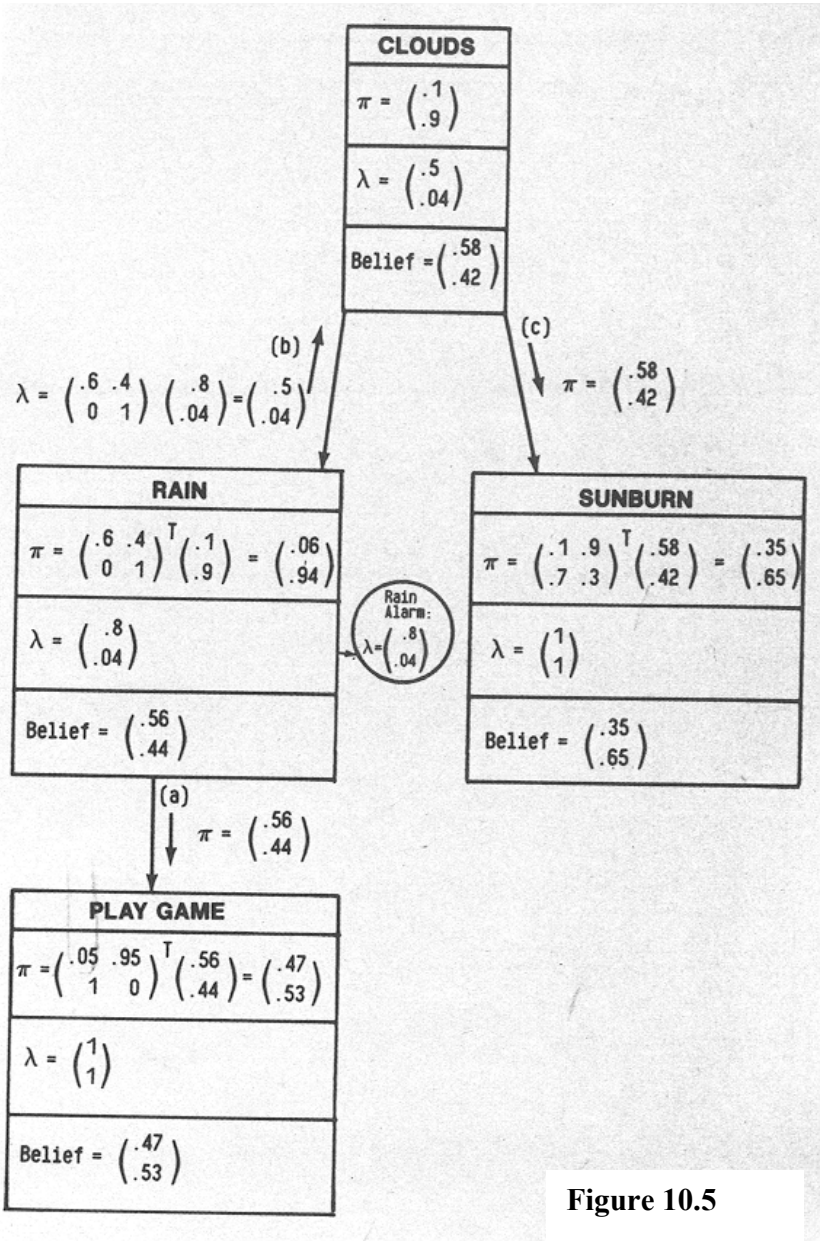

Figure 10.5

Equation (10.12) is an illustration of a very important property of this method. Note that $P(R)$ is the π-message sent from the cloud node to the rain node and that $P(A \mid R)$ is the sent from the alarm node to the rain node. Thus (10.12) can be written

$$P(R \mid A) = \alpha\lambda_A(\mathrm{R})\pi_C(\mathrm{R}) \qquad (10.13)$$

where $\lambda_A(\mathrm{R})$ is the λ-message sent from node A to node R and $\pi_C(\mathrm{R})$ is the π-message sent from node C to node R. Equation (10.13) always holds at a node (in this case node A) when the information in a λ-message and a π-message are combined to update the probabilities at the node. The operation (10.13) is used very frequently in the updating process and this is why λ-messages are distinguished from π-messages.

This also explains why the λ-vectors in the nodes for the initial network (Fig. 10.5) are vectors of 1s. When they are vectors of 1s they will not change anything when (10.13) is implemented. They are only given other values when a node has information to convey.

The rain alarm has had the effect of changing the probabilities in the rain node so this node now prepares a λ-message to send to its parent, the cloud node. The λ-vector that it sends (whose calculation is shown in the figure) comes from the fact that we are here passing information from the alarm node through the rain node to the cloud node. So, as in example 2, we have to take account of the possible states of the rain node and sum their effects out. Thus

$$P(C|A) = \alpha P(C)\sum_j P(A|r_j)P(r_j|C) \qquad (10.14)$$

Node *C* stores *P*(*C*) so the summation term is what node *R* must send it. Thus, the λ-vector is given by

$$\lambda_1 = P(r_1 \mid c_1)P(a_1 \mid r_1) + P(r_2 \mid c_1)P(a_1 \mid r_2)$$

$$\lambda_2 = P(r_1 \mid c_2)P(a_1 \mid r_1) + P(r_2 \mid c_2)P(a_1 \mid r_2)$$

and this is what is calculated for the λ-message in the figure.

The rain node also sends the new belief to the "play game" node as a π-message and the "play game" node updates its beliefs as before. This is shown in Fig. 10.5.

The author of [8] extends the example to include additional events, but we will terminate the discussion here. My main purpose was to illustrate how message passing is employed in these kinds of networks.

Updating with merging branches

In the examples we looked at in the handout that illustrated the use of Bayesian networks [9], there were branches that joined together at nodes, as in Fig. 10.6, and we haven't looked yet at how updates are carried out in this situation.

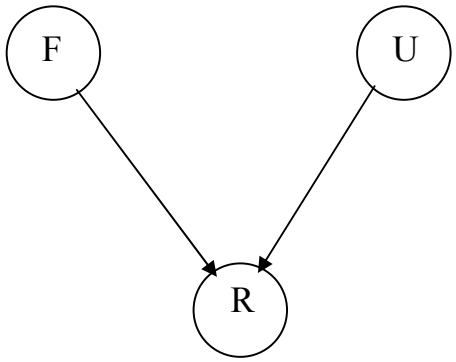

Figure 10.6

This figure represents a situation where the reliability (R) of a software subsystem depends upon the number of faults (F) in the software and the amount of usage (U) that the subsystem is subjected to.

Since *R* is conditional on *F* and *U*, the multidimensional distribution for the three variables can be written

$$P(R,F,U) = P(R \mid F,U)P(F,U)$$

F and U are not connected by arrows so they are independent and the above can be written

$$P(R,F,U) = P(R \mid F,U)P(F)\mathrm{P}(U) \qquad (10.15)$$

This is formula is used in calculating probabilities such as the ones in the reliability node in Fig. 10.7. (Fig. 10.7 is of the same kind as those in [9] but without the histograms.) The probabilities in the reliability node are calculated from the probabilities in the two other nodes. Note that we are here dealing with nodes that can have three states, in contrast to the earlier 2-state nodes. But this is not a big deal – it just means that we have to determine, for instance, $P(R=r_1)$, $P(R=r_2)$ and $P(R=r_3)$ where previously we only had to determine the first two terms.

The earlier examples have shown how nodes store and make use of sets of conditional probabilities. Remember from [9] that each node stores a probability table. In this case, the reliability node R stores a set of conditional probabilities of the form $P(R \mid F,U)$ which have the values shown in Table 10.3.

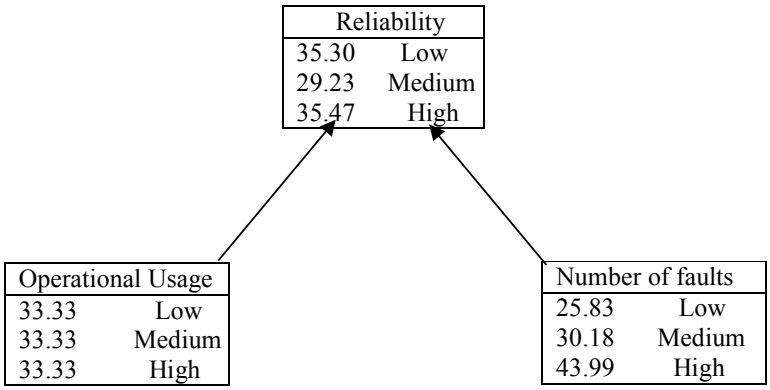

Figure 10.6

Operational Usage		Low			Medium			High		
Number of faults		Low	Med.	High	Low	Med.	High	Low	Med.	High
	Low	0.10	0.20	0.33	0.20	0.33	0.50	0.20	0.33	0.70
Reliability	Medium	0.20	0.30	0.33	0.30	0.33	0.30	0.30	0.33	0.20
	High	0.70	0.50	0.33	0.50	0.33	0.20	0.50	0.33	0.10

Table 10.3

So, the probabilities in the reliability node in Fig. 10.6 are calculated from the probabilities in the other two nodes (which are sent to the reliability node as π-messages) and the probabilities in Table 10.3, which are stored in the reliability node. These calculations are carried out using equation (10.15).

To illustrate this calculation, let the index 1 represent "Low", index 2 represent "Medium" and index 3 represent "High" so, for instance, $P(R=r_1)$ is the probability that the reliability is low and $P(U=u_3)$ is the probability that the usage is high. Then, from (10.15)

$$P(R=\mathrm{r}_1, F, U) = P(R=r_1 \mid F, U)P(F)P(U)$$

so to get the probability that $R=r_1$, we have to sum out all values of F and U:

$$P(R = r_1) = \sum_{i,j} P(R = r_1 | F = f_i, U = u_j) P(f_i) P(u_j)$$

$$= 0.1P(F=f_1)P(U=u_1) + 0.2P(F=f_2)P(U=u_1) + 0.33P(F=f_3)P(U=u_1)$$
$$+ 0.2P(F=f_1)P(U=u_2) + \ldots$$

$$= 0.1(0.2583)(0.3333) + 0.2\,(0.3018)(0.3333) + 0.33(0.4399)(0.3333)$$
$$+0.2(0.2583)(0.3333) + 0.33(0.3018)(0.3333) + 0.5(0.4399)(0.3333)$$
$$+0.2(0.2583)(0.3333) + 0.33(0.3018)(0.3333) + 0.7(0.4399)(0.3333)$$

$$= 0.3549.$$

and similarly,

$$P(R = r_2) = \sum_{i,j} P(R = r_2 | F = f_i, U = u_j) P(f_i) P(u_j)$$
$$= 0.2938$$

and $P(R=r_3) = 0.3566$.

These probabilities are not really probabilities because they add to slightly greater than unity – they add to 1.0053. So we must divide each by 1.0053 and this gives the probabilities in the reliability node in Fig. 10.6.

Concluding remarks

I've shown you some of the techniques that are employed in updating Bayesian networks when new information becomes available. The first three examples related to tree-type networks, but the last item we considered was updating in merging branches, which don't occur in tree networks. The message-passing scheme described in example 3 is only applicable to tree networks and something more elaborate is required for the overall general case. Suitable schemes for the general case were devised in the late 1980s, but they are a bit too involved to be considered in this course.

One additional thing that's worth noting stems from the fact that the Bayesian network is a *directed graph*. That is, its branches are all assigned directions. And there has to be a restriction on the directed branches to ensure consistency in the probability calculations. This restriction is that if you follow the directions of the branches from any starting point, you cannot return to your starting point, or visit any other node twice. What this means is that following the arrows does not take you through any loops. Directed graphs without loops are called *acyclic*. So, a Bayesian network is a directed acyclic graph, or DAG for short. And everywhere in the research literature you will find Bayesian networks referred to as DAGs.

Applications of Bayesian networks are very widespread. The manuscript [9] referred to several applications in the reliability/risk area but you may like to know that Microsoft employed Bayesian networks to develop their Microsoft Office Assistant – that frequently annoying thing that pops up from time to time to try to help you in what you are doing. They employed a team of psychologists and others to sit behind computers that were being used by inexperienced users and their job was to try to work out what they were aiming to do and how to help them. It proved quite difficult, but they eventually developed a Bayesian network to respond to people's needs and that's what underlies the current system.

Given the amount of time we've devoted to pattern classification in this course, you should be interested to know that people are developing methods of using Bayesian networks for pattern classification. But these methods rely upon techniques for training Bayesian networks and, again, this is a topic beyond what we can investigate here.

Finally I point out the Bayes network representation of the naïve Bayes formula because following what we've done in this lecture, I can now state the assumption underlying the naïve Bayes method more correctly. There is an independence assumption that allows the descriminant for class C_i to be written

$$f_i(\boldsymbol{x}) = \Pr(C_i)\prod_{j=1}^{k} \Pr(F_j = x_j \mid C_i) \qquad (9.39)$$

and what the assumption says is that Pr(F1 = x1 | Ci) is independent of Pr (F2 = x2 | Ci) and each is independent of Pr (F3 = x3 | Ci) and so on. That is, the features are independent given the class. We,ve seen this kind of situation several times in this lecture and the Naïve Bayes classifier can be represented in a Bayesian network in the form shown in Fig. 10.7. Once the class is known, the features are independent.

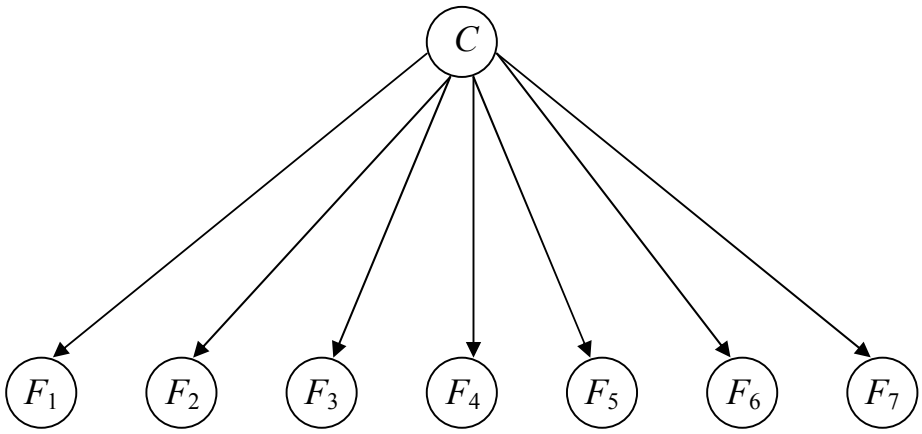

Figure 10.7

References

[1] F T de Dombal et al., "Computer-aided diagnosis of acute abdominal pain", *British Medical Journal*, **2**, 9-13, 1972.

[2] E A Patrick, "Review of pattern recognition in medicine", *IEEE Transactions on Systems, Man and Cybernetics*, **6**, 1977.

[3] B G Buchanan and E H Shortliffe (eds), "Rule-based expert systems: the MYCIN experiments of the Stanford heuristic programming project", Addison-Wesley, 1984.

[4] R Duda, J Gaschnig and P Hart, "Model design in the PROSPECTOR consultant system for mineral exploration", in "Expert systems in the microelectronic age", D Michie (ed), Edinburgh University Press, 153-167, 1979.

[5] E J Horvitz, J D Breese and M Henrion, "Decision theory in expert systems and artificial intelligence", *International Journal of Approximate Reasoning*, **2**, 247-302, 1988.

[6] F V Jensen, "An introduction to Bayesian networks", Springer, 1996.

[7] J Pearl, "Bayesian decision methods", in *Readings in uncertain reasoning*, G Shafer and J Pearl, eds, Morgan Kaufmann, 1990, 345-352.

[8] P Morawski, "Understanding Bayesian networks" *AI Expert*, May, 1989, 44-48.

[9] N Fenton and M Neil, "Making decisions: using Bayesian nets and MCDA", Unpublished manuscript, April, 2000.