BAYESIAN BELIEF NETWORKS

Scott Wooldridge

Australian Institute of Marine Science

Prepared for CSIRO Centre for Complex Systems Science

Introduction

Bayesian Belief Networks (also knows as Belief Networks, Causal Probabilistic Networks, Causal Nets, Graphical Probability Networks, and Probabilistic Cause-Effect Models) are an emerging modelling approach of artificial intelligence (AI) research that aim to provide a decision-support framework for problems involving uncertainty, complexity and probabilistic reasoning. The approach is based on conceptualising a model domain (or system) of interest as a graph (i.e. network) of connected nodes and linkages. In the graph, nodes represent important domain variables, and a link from one node to another represents a dependency relationship between the corresponding variables. To provide quantitative description of the dependency links, Bayesian Belief Networks (BBNs) utilise probabilistic relations, rather than deterministic expressions.

Absolutely anything can be modelled by a BBN. The model might be of your house, or your car, your body, your community, an ecosystem, a stock-market, etc. All the possible states of the nodes in the network represent all the possible 'worlds' that can exist, that is, all the possible ways that the parts or states can be configured. The car engine can be running normally or giving trouble. Its tires can be inflated or flat. Your body can be sick or healthy, and so on.

The main use of BBNs is in situations that require statistical *inference* – in addition to statements about the probabilities (i.e. *likelihood*) of events, the user knows some *evidence*, that is, some events that have actually been observed, and wishes to update his/her *belief* in the likelihood of other events, which have not as yet been observed. Given the node-link structure for the model domain, BBNs use probability calculus and Bayes theorem to efficiently *propagate* the evidence throughout the network, thereby updating the strength of belief in the occurrence of the unobserved events. BBNs can use both 'forward' and 'backward' inference.

Although the probability and Bayesian theory that forms the basis of BBNs has been around for a long time, it is only in the last few years that efficient algorithms and software tools to implement them, have been developed to enable evidence propagation in networks with a reasonable number of variables. The recent explosion of interest in BBNs is due to these developments, since for the first time they can be used to solve realistic size problems.

Why the method is useful

The best way to understand Bayesian Belief Networks is to imagine trying to model a situation in which dependency between variables is known to play a role but where our understanding of what is going on (or has gone on) is incomplete, so we need to describe things probabilistically. The probabilities aim to reflect the fact that some states in our model domain will tend to occur more frequently when other states are also present (i.e. *conditional probabilities*). For example, if you are sick, the chances of a runny nose are higher. If it is cloudy, the chances of rain are higher, and so on.

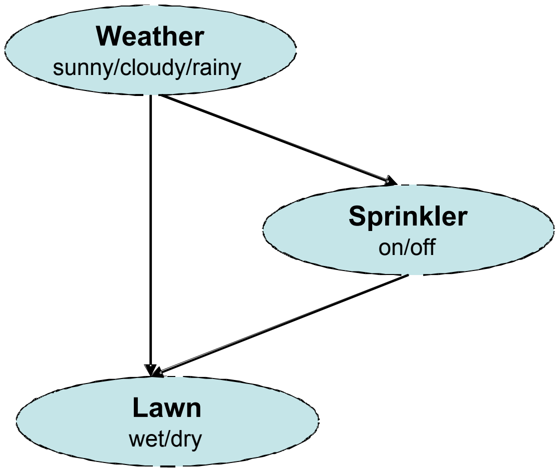


Figure 0. Simple BBN to describe weather, lawn, and sprinkler-use-behaviour

Figure 1 is a simple BBN that illustrates these concepts. In this simple world, let us say the weather can have three states: sunny, cloudy, or rainy, also that the lawn can be wet or dry, and that the sprinkler can be on or off. Now there are some dependence (causal) links in this world. If it is rainy, then it will make the lawn wet directly. But if it is sunny for a long time, that too can make the lawn wet, indirectly, by causing us to turn on the sprinkler.

When actual probabilities are entered into this BBN that reflect the reality of real weather, lawn, and sprinkler-use-behaviour, such a net can be made to answer a number of useful questions, like, "if the lawn is wet, what are the chances it was caused by rain or by the sprinkler", and "if the chance of rain increases, how does that affect me having to budget time for watering the lawn".

BBNs are particularly useful for making probabilistic inference about model domains that are characterised by inherent complexity and uncertainty. This uncertainty may be due to imperfect understanding of the domain, incomplete knowledge of the state of the domain at the time where a given task is to be performed, randomness in the mechanisms governing the behaviour of the domain, or a combination of these. Once developed and parameterised, BBNs provide a rational framework to infer for a modelled domain "whether information on some event should influence our *belief* in other events".

In addition to being able to deal with problems whose complexity cannot be feasibly modelled by other approaches, BBNs offer many advantages over other methods for dealing with uncertainty, and limited data.

Merging different types of information: Due to their Bayesian Probability formalism, BBNs provide a rational technique to combine both subjective (e.g. expert opinion) and quantitative (e.g. monitoring data, modelling results etc) information. The flexible nature of BBNs also means that new information can easily be incorporated as it becomes available. Only the conditional probabilities of the affected variables require re-determination.

Formal structuring of our understanding: BBNs are helpful for challenging experts to articulate what they know about the model domain, and to knit those influences into dependency networks. The graphical (visual) nature of BBNs therefore facilitates the easy transfer of understanding about key linkages. In addition, because subjective expert opinions (hypotheses) are made explicit in the formal structure of the network, they can be challenged and revised, and can also be directly evaluated (potentially with process-based models) to determine whether results are robust.

Modular design: Given their network structuring, BBNs successfully capture the notion of modularity i.e. a complex system is built by combining simpler parts. You can start them off small, with limited knowledge about a domain, and grow them (add additional variables) as you acquire new knowledge.

Informed decision-making before scientific knowledge is complete: Formalization of a model domain through the use of a BBN means that you don't need complete knowledge about the instance of the world you are applying it to. Because uncertainty in particular linkages can be acknowledged in the probabilistic dependency relationships, the models are not necessarily limited by the mechanistic detail of existing information or understanding. As such BBNs can facilitate informed decision-making before scientific understanding is complete.

Predictions are amenable to risk analysis: BBNs express predicted outcomes as likelihood's, which can form the basis for risk analysis. Such risk estimates provide a sound basis for adopting rational decisions based on a precautionary (risk-averse) attitude.

Future scenario testing: BBNs provide an ideal framework to test the most 'likely' consequence of future events or scenarios. This contributes to 'future memory', and understanding of *what will happen when* …….. The ability of BBNs to perform bi-directional reasoning also provides an excellent diagnostic tool for troubleshooting the most likely causes of system failures.

The theory behind the technique

BBNs are a modern inclusion to a family of techniques known as *expert systems*. A common definition of an expert system is a software system that emulates the problem solving behaviour of a human expert over some restricted domain. Other popular examples of expert systems include rule-based systems, fuzzy logic algorithms, and neural networks.

Early expert systems (known as Rule-based systems) where based on deterministic rules of the form: "IF X_1 & … & X_n THEN Y" where $X_1,\ldots,X_n$ are conditions, and Y is some evaluation or action that can be inferred if $X_1,\ldots,X_n$ are true. For instance Fig.2 provides a set of rules for classifying tree types. A rule-based system consists of a library of such rules. These rules reflect essential relationships within the domain, or rather: they reflect ways to reason about the domain.

If Stem is Woody, Position is Upright, There is one Main Trunk, Leaves are Not Broad and Flat, Leaf Shape is Scale-like **Then** Plant is of type: *tree* Class: *gymnosperm* Family: *cypress*	**If** Stem is Woody, Position is Upright, There is one Main Trunk, Leaves are Not Broad and Flat, Leaf Shape is Scale-like Leaf Pattern is Needlelike **Then** Plant is of type: *tree* Class: *gymnosperm* Family: *pine*

Figure 0. A rule-based expert system for classifying tree types

A major problem with rule-based systems is that often the connections reflected by the rules are not absolutely certain (i.e. deterministic), and similarly the gathered information is often subject to uncertainty. To overcome this problem, a certainty measure can be added to the premises as well as the conclusions in the rules of the system. In such a system the local rule becomes a function that describes how much a change in the certainty of the premise will change the certainty of the conclusion. In its simplest form, this looks like: "If A (with certainty x) then B (with certainty f(x))". Although the individual (i.e. local) production rules are relatively simple and self documented, for complex model domains their interactions within the larger set of rules can often lead to erroneous inference. This is because the combination of uncertainty is not a local phenomenon, but it is strongly dependent on the entire situation (in principle a global matter).

It was during the 1960s that it became apparent that to effectively deal with the global implications of local uncertainties in complex model domains, it was necessary to calculate the probabilities correctly (correctly regarding the axioms of the classical probability theory). For a specific model domain of interest, this meant representing what is called the "joint probability distribution". This is a table of all the probabilities of all the possible combinations of states in that model domain. For modelling domains with more than just a few variables and states, these distributions can become very large, as every possible state combination over every variable must be represented. For example, assuming binary (two-state) variables, a system with 10 variables would require 2^{10} =1024 individual probabilities. This number increases dramatically if the variables can take on more states (which they frequently do)

It was not until the mid 1980s that Pearl (1986) introduced BBNs as a method for making calculations of this type more tractable. Pearl was able to demonstrate that by defining the behaviour of a system in terms of a series of local *conditional probabilities*, BBNs were able to provide the correct global framework to propagate local information and associated uncertainties. Importantly, by using the concept of *conditional independence* (see Section 4), BBNs were also able to derive the information needed from the joint probability distribution using a much smaller number of conditional probabilities.

To understand this computational saving, it helps to understand how BBNs provide a link between probability theory and graph theory. In essence, BBNs use a graphical representation to represent probabilistic structure i.e. there is a direct relationship between a graphical model and a particular form of joint probability distribution. Crucially, this joint probability distribution is far simpler to compute when there are conditionally independent nodes (i.e. nodes that are not connected by links).

Suppose, for example, that we have a network consisting of five variables (nodes) A,B,C,D,E. If we do not specify the dependencies explicitly then we are essentially assuming that all the variables are dependent on (i.e. influence) each other. The *chain rule* from probability theory enables us to calculate the joint distribution p(A,B,C,D,E) as:

$$p(A,B,C,D,E) = p(A|B,C,D,E)*p(B|C,D,E)*p(C|D,E)*p(D|E)*p(E) \qquad 1.$$

However, suppose that the dependencies are explicitly modelled as for the BBN in Fig. 3:

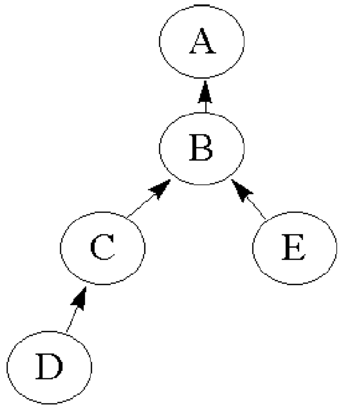


Figure 0. Node-link structure of a hypothetical BBN

Then the joint probability distribution p(A,B,C,D,E) is much simplified:

$$p(A,B,C,D,E) = p(A|B)*p(B|C,E)*p(C|D)*p(D)*p(E) \qquad 2.$$

We can now consider the general case of a joint probability distribution in a BBN. Suppose the set of variables in a BBN is $\{A_1, A_2, \ldots, A_n\}$ and that *parents*(A_i) denotes the set of parents of the node A_i in the BBN. Then the joint probability distribution for $\{A_1, A_2, \ldots, A_n\}$ is:

$$p(A_1,........A_n) = \prod_{i=1}^{n} p(A_i \mid parents(A_i)) \qquad 3.$$

Now it is possible to understand what the links in a BBN mean, and hence what we need to specify to turn the graphical dependence structure of a BBN into a probability distribution. For each node we need the conditional probability of that node taking a certain value given the values of its parents. For discrete networks (nodes taking a fixed number of classes) this amounts to defining a conditional probability table (CPT).

As an example consider node *B* in Fig. 2. It would require a CPT specifying the conditional distribution p(*B*|*C*,*E*). Similarly, the CPTs for nodes *A* and *C* would specify p(*A*|*B*) and p(*C*|*D*) respectively. The nodes *D* and *E* have no parents, so only require prior probability distributions p(*D*) and p(*E*). If we assume that all the variables in Fig. 2 are binary, and can take on the states *true*, *false*, then the CPT for the node *B* would be as shown in Table 1. Note that in a real CPT, the p(……) probability expressions are replaced by probability values between 0 and 1 consistent with the standard axioms of probability theory.

Table 1. CPT for Node B in Fig. 2

		B	
C	*E*	true	false
true	true	p(*B*=true | *C*=true, *E*=true)	p(*B*=false | *C* = true, *E*=true)
true	false	p(*B*=true | *C*=true, *E*=false)	p(*B*=false | *C*=true, *E*=false)
false	true	p(*B*=true | *C*=false, *E*=true)	p(*B*=false | *C*=false, *E*=true)
false	false	p(*B*=true | *C*=false, *E*= false)	p(*B*= false | *C*=false, *E*= false)

Comparing Neural Networks and BBNs: It is useful to highlight the key differences between expert systems that are based on BBNs and those based on neural networks. As illustrated in Fig. 4, a neural network consists of several layers of nodes: On top there is a layer of input-nodes, on the bottom a layer of output-nodes and in between these, normally 1-2 hidden layers. All nodes in a layer are in principle connected to all nodes in the layer just below. A node along with the in-going edges belonging to it is called a perceptrone. The fundamental difference between the two types of networks is that a perceptrone in the hidden layers of a neural network does not in itself have an interpretation in the domain of the system, whereas all the nodes of a BBN represent concepts that are well defined with respect to the domain.

In a BBN the meaning of a node and its CPT can be the subject of an external discussion, regardless of their function in the network. This does not make any sense when speaking of neural networks. Perceptrones in the hidden layers only have a meaning in the context of the functionality of the network. As a consequence, in the construction of a neural network the route of inference is fixed. It is decided in advance, about which relations information is gathered, and which relations the system is expected to calculate. BBN are much more flexible in that respect, since they attempt to model actual domain dependencies.

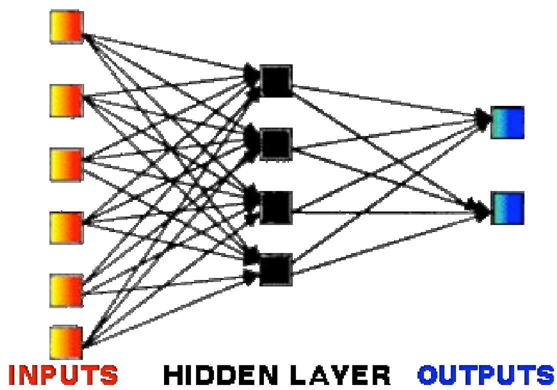


Figure 1. Node-link structure in a typical neural network. Note that the nodes and links in the 'hidden layer' have no physical interpretation in terms of domain variables

Assumptions required

An important concept for the building (as well as the transmission of evidence) in a BBN is the notion of *conditional independence*. Two sets of variables, A and B, are said to be (conditionally) independent given a third set C of variables if when the values of the variables C are known, knowledge about the values of the variables B provides no further information about the values of the variables A. Using probabilistic notation this can be expressed as $p(A \mid B, C) = p(A \mid C)$.

To understand the concept of conditional independence, consider the following three events, for which $p(A \mid B, C) = p(A \mid C)$);

Event A = I take a Panadol tablet;

Event B = I kiss my girlfriend who has the flu;

Event C = I have a fever.

Now notice that me taking a Panadol, and me kissing my sick girlfriend are *not* independent events. However, the event of me taking a Panadol tablet is *conditionally independent* of me kissing my sick girlfriend given that I have fever. Once I have a fever, knowing whether I participated in a high risk activity doesn't change the likelihood of me taking a Panadol. Thus if we were to construct a network model of this space, we could reduce the number of parent nodes from two to one.

As demonstrated in Section 3, the conditional independence assumptions expressed by the graph (i.e. lack of a dependency link) mean that fewer parameters need to be estimated because the probability distribution for each variable depends only on the node's parents. This independence allows us to factorise the network, considering each node and its parents in isolation from the rest of the model.

At this point, it is worth making a distinction between the *causal* and *dependence* view of BBNs. There are a number of ways that the causal view is problematic. The first is that there are many ways of representing a probability distribution as a BBN, and many of those will not relate to causal structures. Further, marginalisation over (that is factoring out the effect of) a variable which two other variables causally depend on will create a dependence between those other variables. However neither one can really be considered to be causally dependent on the other. At best then a causal view of belief network construction is anaemic. The causal view becomes fundamentally wrong when the process is inverted and causal relationships are inferred from a BBN. This is one of the most serious flaws which can be made in statistics. Probabilistic dependence-type relationships do not imply causality, even if the BBN was constructed using prior causal information. To deal with causality is significantly more complicated and involves the consideration, among other things, of counterfactuals. This is well beyond the scope of this tutorial, but readers interested in dealing with causality in this sort of framework should refer to Pearl (2000).

Mathematic calculations involved

The key mathematical calculations undertaken in a BBN application are associated with probabilistic *inference*. This means computing the conditional probability for some variables given information (evidence) on other variables.

Consider the BBN in Fig. 5, which describes the separate likelihood of two different students (*Student_A* and *Student_B*) being late (*true*/*false*) to school given knowledge about the existence of a train strike (*true*/*false*). From Fig. 5 it can be seen that the *hypothesis* variables for the BBN are associated with our belief in *Student_A* or *Student_B* being late; these beliefs are constrained to exist between the two discrete states, *true* or *false*. The *information* variable on which our hypothesis variables have some level of dependence (the strength of which is described by their individual CPTs) is the *Train Strike* variable. In this case, before receiving any additional information, there is a 10% chance of a train strike on any given day.

The key feature of BBNs is that they enable us to model and reason about uncertainty. As described by the CPT for *Student_A*, the existence of a train strike does not imply with certainty that he will be late (he might get a lift in a friends car), but there *is* an increased probability that he will be late. Informally, the particular values in this table tell us that: *Student_A* is very unlikely to be late normally (that is, the probability *Student_A* is late when there is no train strike is 0.1), but if there is a train strike he is very likely to be late (the probability is 0.8). Conversely, the CPT for *Student_B* reflects the fact that he is normally dropped off at school in a car. A train strike can still cause *Student_B* to be late because traffic is heavier in that case. However, the probability table for *Student_B* is very different in content to that of *Student_A*. Informally, *Student_B* is often late, but a train strike only increases the likelihood of his lateness by a small amount. In the event of a train strike *Student_B* is less likely to be late than *Student_A*.

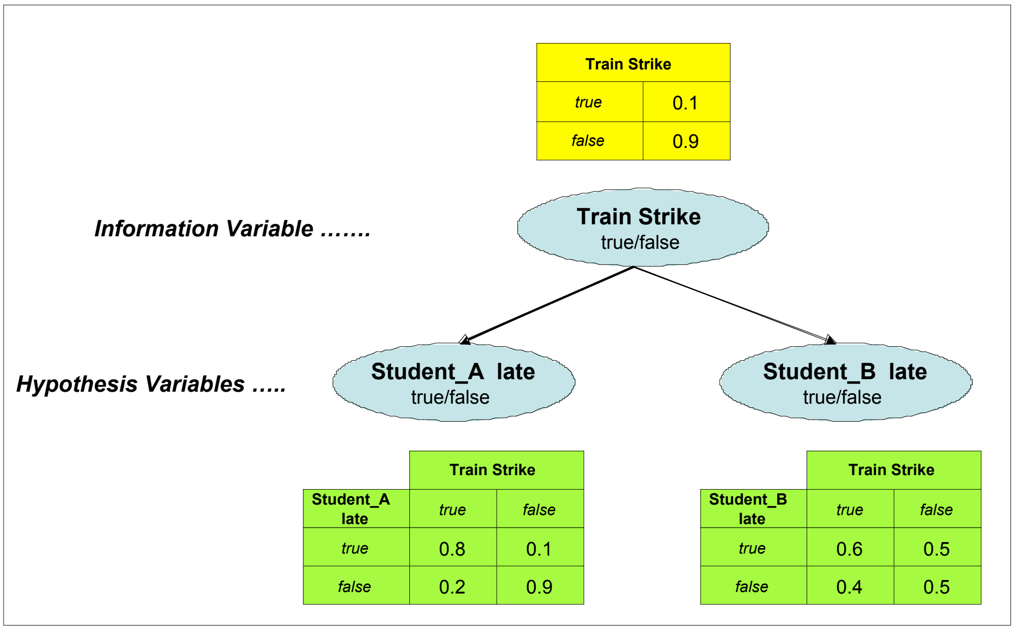


Figure 1. BBN detailing the likely implications of a train strike on the arrival time of two different students (Student_A and Student_B)

<u>Analysing a BBN: entering evidence and propagation</u>

With the dependence structure and associated CPTs identified we can now use Bayesian probability to do various types of analysis. For example, we might want to calculate the (unconditional) probability that *Student_A* is late:

p(*Student_A late*) = p(*Student_A late* | *Train strike*) * p(*Train strike*)

+ p(*Student_A late* | *no Train strike*) *p(*no Train strike*) 4.

= (0.8 * 0.1) + (0.1 * 0.9) = 0.17

This is called the *marginal* probability. Similarly, we can calculate the marginal probability that *Student_B* is late to be 0.51.

However, the most important use of BBNs is in *revising* probabilities in the light of actual observations of events. Suppose, for example, that we *know* there is a train strike. In this case we can **enter the evidence** (called *instantiation*) that *train strike* = 'true'. The conditional probability tables already tell us the revised probabilities for *Student_A* being late (0.8) and *Student_B* being late (0.6). Suppose, however, that we do not know if there is a train strike but do know that *Student_A* is late. Then we can instantiate '*Student_A late*' = *true* and we can use this observation to determine:

a) the (revised) probability that there is a train strike; and

b) the (revised) probability that *Student_B* will be late.

To calculate a) we use a basic law of probability known as Bayes rule. For any two events, *A* and *B*, Bayes rule can be written as:

p(*A*|*B*) = p(*B*|*A*) x p(*A*) / p(*B*) **5.**

Bayes' rule addresses the question, "Given our previous beliefs about an event, how should we revise the probability assigned to the event in light of the new evidence at hand?" The general idea is that, if we had a high degree of belief in the likelihood of Event A based on past experience (i.e. $p(A)$), and we now observe data (Event B) that would be likely to occur if Event A occurs (i.e. $p(B|A)$), then our after the evidence confidence (i.e. $p(A|B)$) in Event A should be strengthened.

For our train strike example:

p(*Train strike* | *Student_A late*)

= p(*Student_A late* | *Train strike*)*p(*Train strike*) / p(*Student_A late*) 6.

= (0.8*0.1)/0.17

= 0.47

Thus, the observation that *Student_A* is late significantly increases the probability that there is a train strike (up from 0.1 to 0.47). Moreover, we can use this revised probability to calculate b):

p(*Student_B late*) = p(*Student_B late* | *Train strike*) * p(*Train strike*)

+ p(*Student_B* | *no Train strike*) *p(*no Train strike*) 7.

= (0.6 * 0.47) + (0.5 * 0.53) = 0.55

Thus, the observation that *Student_A* is late has slightly increased the probability that *Student_B* is late (from 0.5 to 0.55). When we enter evidence and use it to update the probabilities in this way we call it *propagation*.

Although the calculation of the prior probabilities and the after-the-evidence revised probabilities is relatively straightforward for our simple example, imagine a larger net with many dependencies and nodes that can take on more than two values. Doing the propagation in such cases is generally very difficult. In fact, there are no universally efficient algorithms for doing these computations (the problem is NP-hard). This observation, until relatively recently, meant that BBNs could not be used to solve realistic problems. However, in the 1980s researchers discovered propagation algorithms that make it possible to break the overall graph down into smaller sub-sets within which information flows are largely self-contained. This approach allows the propagation of information to proceed much more efficiently. More details of this topic can be found in Lauritzen and Spiegelhalter (1988). With the introduction of software tools that implement these algorithms (as well as providing a graphical interface to draw the graphs and fill in the probability tables) it is now possible to use BBNs to solve complex problems without doing any of the Bayesian calculations by hand. This is the reason why the popularity of BBNs has mushroomed in recent years.

Data and data relationships

A BBN requires data in three distinct categories:

Evidence to parameterise CPTs. Quantitative information is required to 'parameterise' the CPTs of a BBN (i.e. define the conditional probabilities linking parent and child nodes combinations). One of the major benefits of BBNs stems from the fact that this quantitative information can come from both subjective judgements (elicited from domain experts) or probabilities based on objective data (i.e. the frequency with which each configuration of variables is found in the data). When data is limited, it is a common approach to use subjective judgements to initially supply probability distributions and then update this to encapsulate the information contained in the data. This is commonly achieved with an Expectation Maximisation algorithm (see for example, Chickering and Heckerman, 1996).

The flexible nature of BBNs also makes possible the parameterization of local conditional probability relationships using existing models – say for example, a linear regression model, or output from a simulation model.

Two interesting points to remember about the elicitation of the CPTs for BBNs are;

<u>1. The conditional probabilities need not be exact to be useful</u>. Some people have shied away from using BBNs because they imagine they will only work well, if the probabilities upon which they are based are exact. For many applications, this is not true. It turns out very often that approximate probabilities, even subjective ones that are guessed at, give very good results. BBNs are generally quite robust to imperfect knowledge. Often the combination of several strands of imperfect knowledge can allow us to make surprisingly strong conclusions.

<u>2. Causal conditional probabilities are easier to estimate than the reverse</u>. Studies have shown people are better at estimating probabilities "in the forward direction". For example, doctors are quite good at giving the probability estimates for "if the patient has lung cancer, what are the chances their X-ray will be abnormal?", rather than the reverse, "if the X-ray is abnormal, what are the chances of lung cancer being the cause?"

Evidence to set the prior probability distribution of hypothesis variables. Information is needed to assign the appropriate prior probability distributions of all hypothesis variables such that they are consistent with the prevailing situation. Because of the dependence structure of the network, such evidence travels along the links of the network to set the prior probabilities of all other variables.

The problem of converting a *state of knowledge* to a probability assignment is the problem that lies at the heart of Bayesian probability theory. As with CPTs elicitation, where historical records exists the relative frequency of observing a hypothesis variable in a particular state can be used to guide probability assignments, else expert opinion can also be elicited. For situations in which little is known, it is common practice to assume a uniform prior distribution across all the states of a hypothesis variable (i.e. an 'uninformative' prior).

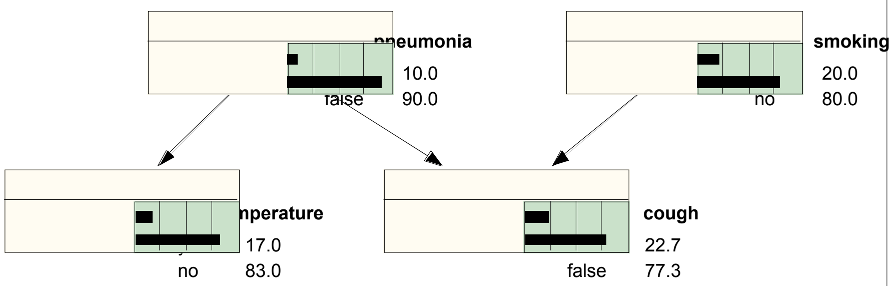
Evidence describing the state of information variables. Evidence pertaining to information variables is essential for belief updating of the hypothesis variables, with the new evidence travelling against the links to update the prior probabilities to posterior (i.e. after-the-evidence) probabilities according to Bayes rule. Information variables are typically chosen to represent elements of a system for which observable data is available (e.g. from sensors).

Key outputs and interpretation

Once fully parameterised, there are several ways in which a BBN can be used to make *inference* about the domain being modelled. Consider the hypothetical BBN in Fig. 6, which enables inference about the likely cause of a cough.

pneumonia	
true	0.1
false	0.9

smoking	
yes	0.2
no	0.8



	temperature	
pneumonia	yes	no
yes	0.9	0.1
no	0.2	0.8

		cough	
pneumonia	smoking	true	false
true	yes	0.95	0.05
true	no	0.8	0.2
false	yes	0.6	0.4
false	no	0.05	0.95

Figure 1. BBN model to enable inference about the likely cause of a cough. Note that for the case displayed, no evidence has been instantiated, therefore the displayed probabilities of a node being in a particular state (e.g. true or false) simply refer to the m distribution across all nodes

Three potential types of inference exist:

Diagnostic Inference. With diagnostic inference, we use evidence of an effect to infer the most likely cause. This is often referred to as "bottom-up" reasoning, since it goes from effects to causes; it is a common task in expert systems. For example, in the model domain described by Fig. 6, if we instantiate *cough* = 'true' (see Fig. 7) we find that p(*pneumonia* | *cough*) = 0.366 (up from 0.1 in the unconditional case).

Diagnostic inference is typical in medical and industrial applications. For example, many industrial applications of BBNs are for determining the chance of component failure. Diagnostic inference also encompasses the class of problems known as sensor fusion, where data from various sources must be integrated to arrive at an interpretation of a situation.

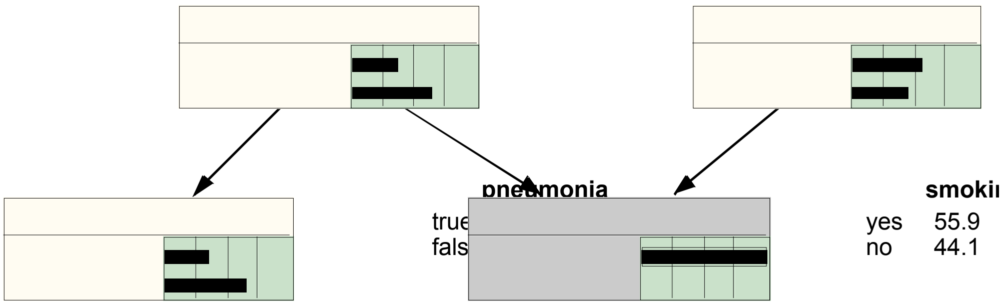


Figure 1. Instantiation of *cough* = 'true' allowing for <u>diagnostic inference</u> of

p(*pneumonia* | *cough*) = 0.366

Causal Inference. With casual inference, or "top-down" reasoning, we endeavour to identify the most likely cause of and effect. For example, in the model domain described by Fig. 6, if we instantiate *pneumonia* = 'true' (see Fig. 8) we find that p(*cough* | *pneumonia*) = 0.83 (up from 0.227 in the unconditional case).

Such causal inference often finds application in weather forecasting, stock market prediction, ecological modelling, etc., where you can supply evidence of past events, and then run the BBN to see what the most likely future outcomes will be.

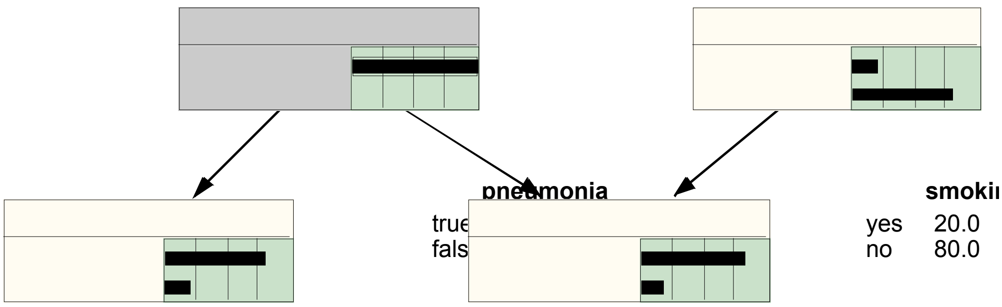


Figure 1. Instantiation of *pneumonia* = 'true' allowing <u>causal inference</u> of

p(*cough* | *pneumonia*) = 0.83

Inter-causal inference. With inter-causal inference, we attempt to "explain away" potentially competing causes of a shared effect. For example, in the model domain described by Fig. 6, the observation of *smoking* (see Fig. 9) would partially "explain away" *pneumonia* as the likely cause of a *cough* (i.e. p(*pneumonia* | *cough* and *smoking*) = 0.15 compared to the p(*pneumonia* | *cough*) = 0.366.)

It should be remembered that whatever the form of inference utilised, the output for the hypothesis (i.e. query) variable is a probability distribution (representing the degrees-of-belief in each state) rather than a simple scalar or vector.

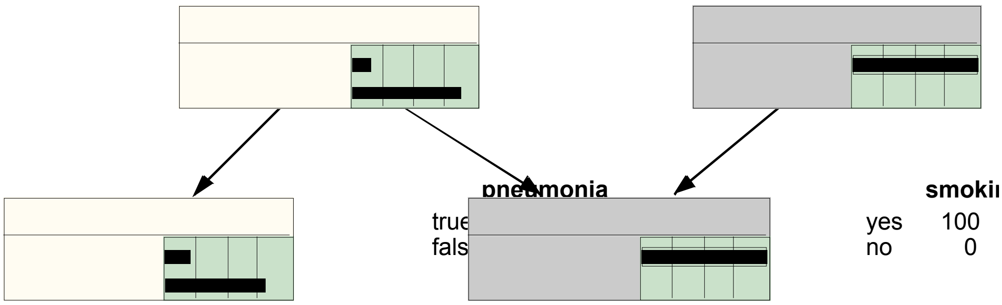


Figure 1. Instantiation of *cough* = 'true' and *smoking* = 'true' allowing <u>inter-causal inference</u> of p(*pneumonia* | *cough* and *smoking*) = 0.15 to partially "explain away" pneumonia as the likely cause of a cough

Limitations of the method or reservations about the method

In spite of their remarkable power, there are a number of potential limitations that require consideration:

Potential to over-emphasize expert opinion. In some cases where objective (i.e. empirical) data are lacking, a BBN is no better than a simpler statement of an educated guess, and in such situations can easily reflect personal bias. Even in such a situation however, a BBN may prove valuable for organizing thinking and posing testable hypotheses. When only subjective (expert) judgment is available, it is advantageous to rely on multiple experts and average their estimated probabilities to reflect the relative uncertainty in collective beliefs. Any disagreement among the experts would then produce a more uniform distribution of probabilities across states reflecting greater uncertainty in the conditional dependencies.

Large BBNs can become unmanageable. Along with significant computational overheads, BBNs become difficult to apply to large problems because the number of conditional probabilities that must be specified can quickly become extremely large as the conceptual scope of a problem increases. In such situations, model design not only becomes difficult to manage but many probabilities will not be well characterized and will therefore need to be supplied directly by expert judgment.

BBNs behave rigidly to unforeseen events. BBNs prove to be most useful for developing a consistent and transparent interpretation of 'likely' system responses when some knowledge of the dependency (causal) structure is known. They provide little insight however regarding unknown dependencies. Another important consequence of the rigid structure is that it is difficult to capture relationships between variables which have a temporal element (i.e. change over time).

How could the method be enhanced ?

BBNs are an active area of research, especially in the artificial intelligence community. Two areas where we can expect to see advances in the next few years are:

Learning the Graphical Structure from data. The dependency (causal) structure of a BBN can be crafted after an expert's mental map of how a system operates. For situations in which the current level of system understanding does not permit the informed development of the structure, the potential exists to try and learn the BBN from available data sets.

In recent years, learning the structure of belief networks has become a very active research topic and many algorithms have been developed for it. For an introductory paper on belief network learning, interested readers are referred to Buntine (1996). By far the most common methods are based on *dependency analysis* algorithms which attempt to discover dependencies from the data (based on conditional independence tests), and then use these dependencies to infer the structure. To increase efficiency and robustness, such algorithm's also usually attempt to incorporate "expert" domain knowledge where possible, such as the partial ordering of nodes.

Temporal Reasoning. A problem with the standard theory of belief networks is that there is no natural mechanism for representing time. For example, it is difficult to represent a situation such as the variability of when an employee arrives at work and the causal relationships between the time of arrival and later events.

One approach to the temporal problem has been to develop BBNs that are made up of interconnected *time slices* of smaller (usually structurally identical) static belief networks (Kjaerulff, 1995). The evidence and inferred beliefs of previous time slices are used to estimate beliefs in the current and future (or prediction) time slices. This approach however results in large and complex networks, requiring considerable computational time and resources. Current research is focussed on providing more efficient methods to provide temporal reasoning.

Judging the success of the method

A robust evaluation of a BBN should include three basic elements:

Model walk-through: Given the importance of the dependency (causal) structure, once constructed it is beneficial to present the model (or particular submodel components) to "fresh" experts for peer review. The consistency of the model in delivering the intuition of domain experts should be confirmed.

Sensitivity Analysis: Like all models, BBNs can be overfit. To avoid over-fitting, it is common practice to use sensitivity tests to measure the effect of one variable on another. Variables for which the model output is particularly insensitive should be removed in order to produce the most parsimonious model description.

Case-based evaluation: Perhaps the most rigorous form of testing is to use a set of real cases to test how well the predictions or diagnosis match the actual cases. This could involve testing local model fragments (component testing) or global model behaviour (whole model testing). Typical performance measures might involve calculating a confusion matrix, error rate, calibration table, quadratic (Brier) score, logarithmic loss score etc.

Tools for operationalising the method

Several research and commercially developed products exist that contain both an editor (for creating the graphical BBN structure) and a runtime module which takes care of evidence propagation. A very useful website containing an up-to-date listing of the available packages can be found at:

(http://www.cs.berkeley.edu/~murphyk/Bayes/bnsoft.html)

By far the two most popular commercial products are:

1. Netica (http://www.norsys.com), and
2. HUGIN (http:// www.hugin.com).

Note: Netica provides a free demo version of their product (full functionality), but which is limited to models with less than 15 nodes.

Another useful commercial software package for transforming a database into a network of dependencies is Bayesware Discover (http://bayesware.com/). It is an automated modelling tool that searches for the most probable BBN responsible for the observed data.

Applications

Given the remarkable flexibility of BBNs, it is possible to implement them for most model domains. However, BBNs are most suited for situations;

1. Where some underlying understanding exists of the model domain, and we are interested in knowing whether observed information on some event should influence our belief in other events.

2. Where the model domain can be conceptually represented by a flow diagram of nodes and linkages, and where the nature of these linkages (even if uncertain) is reasonably stable through time.

3. Where it is beneficial to have a highly visual and transparent model structure (e.g. to engage stakeholders groups more effectively in resource management decisions).

4. Where we would like to implement a formal decision support framework, but where the majority of the understanding about the model domain is derived from the cognitive experience of human 'experts', as opposed to empirical observations.

5. Where our understanding of the behaviour for the model domain is derived from a variety of disparate information sources (e.g. empirical data, other models, expert opinion), and we are interested in providing a fusion-type 'meta' framework for developing weight-of-evidence decision support.

Example 1: Microsoft Office Assistant

One of the most celebrated users of BBN technology is the Decision Theory & Adaptive Systems Group (DTAS) at Microsoft (http://research.microsoft.com/research/dtg/). The aim of their research is to create software which can automatically and intelligently interact with the users, anticipating the goals and needs of these users. One of the more obvious products can be seen as "Office Assistant" (Fig. 10) implemented in the Microsoft Office suite of productivity applications.

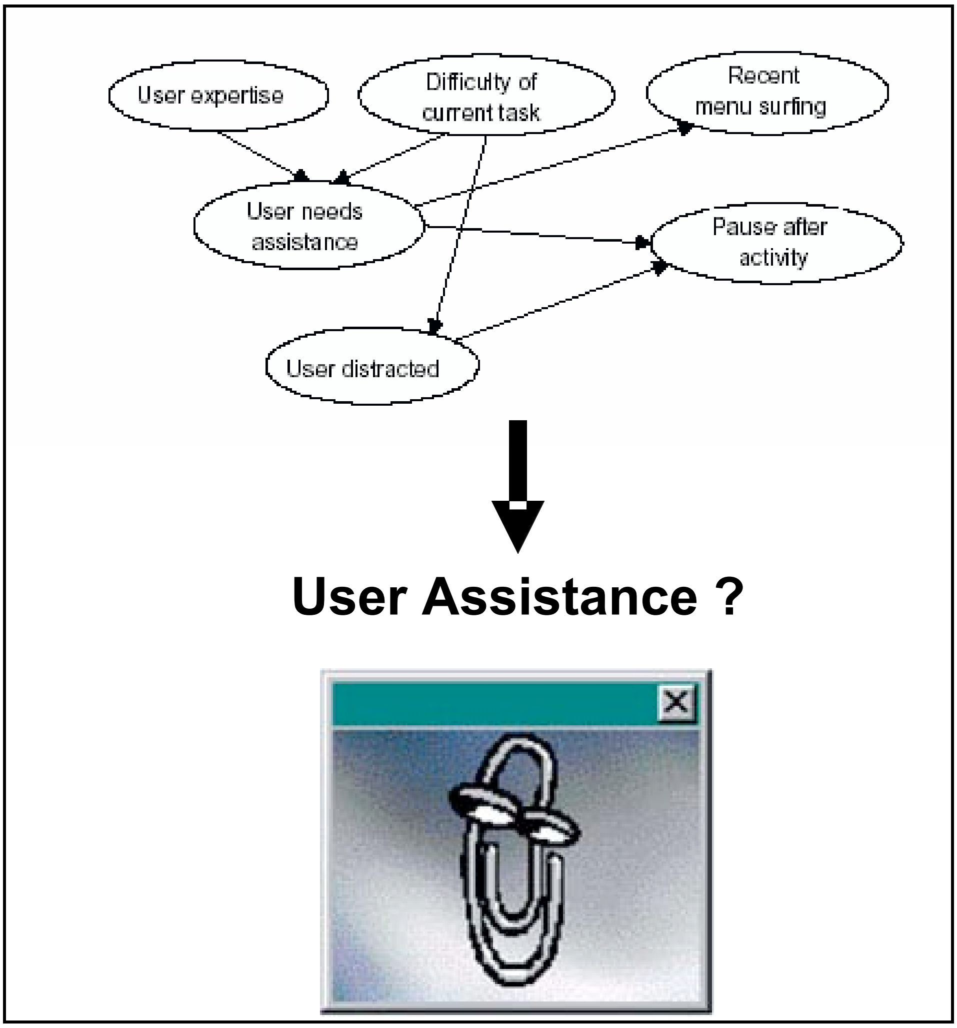


Figure 1. A portion of a BBN user model for inferring the likelihood that a user needs assistance, considering profile limitation as well as observations of recent activity

Example 2: Modelling forestry effects on stream ecosystems

Managing impacts of logging activities on stream ecosystems requires an understanding of the complex processes involved. Researchers at the National Institute of Water & Atmospheric Research, New Zealand have been investigating the dual potential of BBNs to deal with this complexity, whilst at the same time being able to provide a management tool that can be understood and implemented by resource managers and forest harvest planners.

The developed BBN as shown in Fig. 11 provides a model of how forestry management practices can affect stream health. The model was developed from a combination of survey data, literature information, and field experience. As knowledge about a site (e.g., riparian management, channel width) is entered into the model it updates the probabilities associated with states of each related node. The model can be used either to predict forest management effects on stream health or to diagnose the causes of observed impacts. For further details, contact John Quinn at NIWA Hamilton, j.quinn@niwa.co.nz

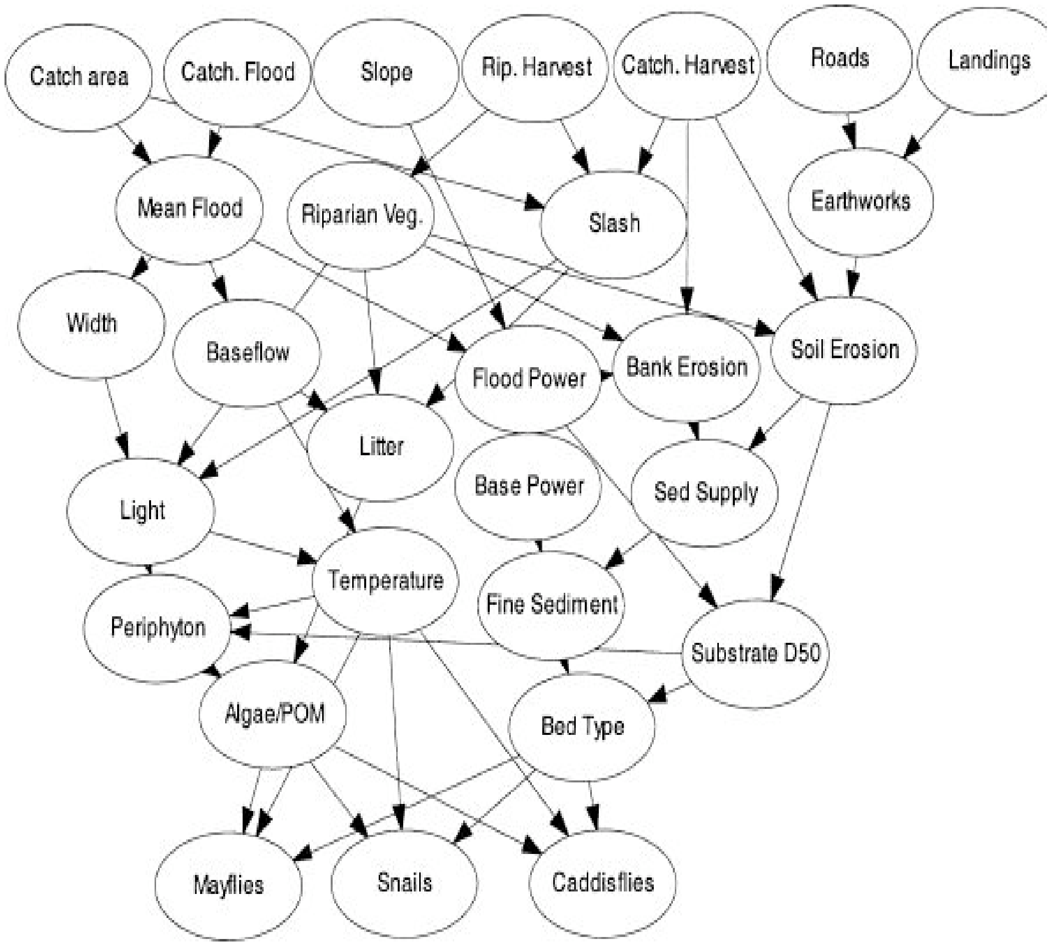


Figure 1. BBN developed to provide a model of how forestry management practices can affect stream health

Example 3: Bayesian Network Models for Search and rescue

SAR**Bayes** is a project of the Reasoning Under Uncertainty Group (a part of the Monash Data Mining Centre), with the cooperation of Victorian Police Search & Rescue and VicWalk's Bushwalkers' Search & Rescue. The project has delivered a decision support tool to aid in the optimal allocation of resources in a lost-person search and rescue. The unifying theme of SAR**Bayes** is that search and rescue is a classic case of Reasoning Under Uncertainty and the core of the problem is generating and maintaining a probability map for the current location of the lost person (e.g. see Fig. 12).

The general methodology involves:

1. Developing BBN models to predict lost-person behaviour based on a historical data-base of lost person incidents,

2. Generating probability maps of 'likely' search areas, by merging the behaviour BBN models with physical details of the current search environment,

3. Designing efficient search strategies by linking the probability maps with *Optimal Resource Allocation* algorithms.

More details on the project can be found at (http://sarbayes.org/index.html).

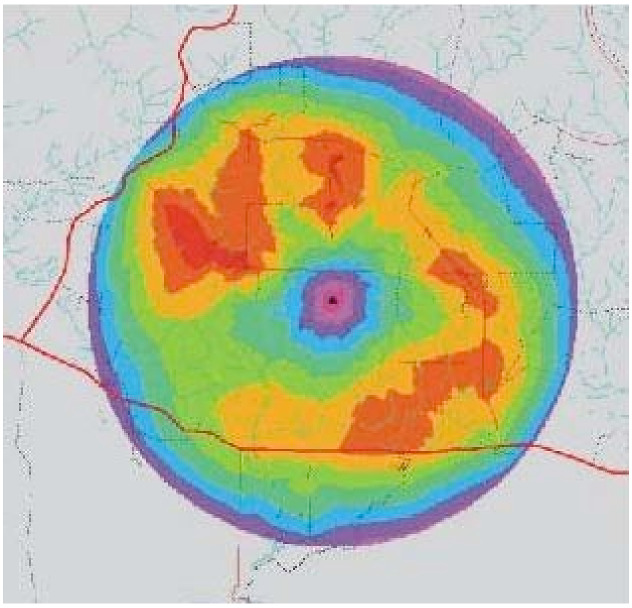


Figure 1. An example of a BBN generated probability map, demonstrating 'likely' search areas for a lost person incident

Sources of more information

Key publications

Charniak, E., 1991. Bayesian Networks without Tears. *AI magazine* (Winter), 50-63.
Heckerman, D., 1995. A tutorial on learning Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research.
Jensen, F.V., 2001. Bayesian Networks and Decision Graphs, Springer. 2001
Special Issue on Bayesian Networks: Communications of the ACM., March, 1995, vol 38, no. 3.

Useful websites

Kevin Murphy (Berkley University) maintains a website that provides an excellent introduction to BBNs. The site also contains links to relevant web literature on BBNs. http://www.ai.mit.edu/~murphyk/Bayes/bnintro.html
Norman Fenton (London University) has developed a very helpful online tutorial on the theory and development of BBNs. http://www.dcs.qmul.ac.uk/~norman/BBNs/BBNs.htm
Russell Greiner (University of Alberta) maintains a website that provides links to numerous articles, research groups, software, and applications associated with BBNs. http://www.cs.ualberta.ca/~greiner/bn.html
Amos Storkey (University of Edinburgh) also maintains a useful website on BBNs http://www.anc.ed.ac.uk/~amos/belief.html

Key contacts

Name	Organisation	Email/web address
Scott Wooldridge	Australian Institute of Marine Science	s.wooldridge@aims.gov.au
Bruce Marcot	USDA Forest Service	http://www.spiritone.com/~brucem/bbns.htm
Kevin Korb	School of Computer Science, Monash University	http://www.csse.monash.edu.au/~korb/
Ann Nicholson	School of Computer Science, Monash University	http://www.csse.monash.edu.au/~annn/

References

Buntine, W., 1996. A guide to the literature on learning probabilistic networks from data, *IEEE Trans. Knowledge Data Engrg*., 8(2), 195-210.

Chickering, D., and Heckerman, D., 1996. Efficient approximations for the marginal likelihood of incomplete data given a Bayesian network. *Technical Report MSR-TR-96-08*, Microsoft Research, Redmond, WA.

Kjaerulff, U., 1995. dHugin: a computational system for dynamic time-sliced Bayesian networks. *J. Forecast*., 11, 89-111.

Lauritzen, S.L., and Spiegelhalter, D.J., 1988. Local computations with probabilities on graphical structures and their application to expert systems. *J. Royal Statistics Society B*, 50(2), 157-194.

Pearl, J., 1986. Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29, 241-288

Pearl, J., 2000. Causality: models, reasoning and inference. Cambridge University Press.

Table of Contents

List of Tables.