Chapter 6: Model Building with Belief Networks and Influence Diagrams

Ross D. Shachter

Belief networks and influence diagrams are directed graphical models for representing models of probabilistic reasoning and decision making under uncertainty. They have proven to be effective at facilitating communication with decision makers and with computers. Many of the important relationships among uncertainties, decisions, and values can be captured in the structure of these models, explicitly revealing irrelevance and the flow of information. We explore a variety of examples illustrating some of these basic structures, along with some algorithms that efficiently analyze their model structure. We also show how algorithms based on these structures can be used to resolve inference queries and determine the optimal policies for decisions.

Keywords: Decision analysis; graphical models; influence diagrams; belief networks; bayesian networks; causal networks; probabilistic inference; model structures

We have all learned how to translate models, as we prefer to think of them, into arcane representations that our computers can understand, or to simplify away key subtleties for the benefit of clients or students. Thus it has been an immense pleasure to work with graphical models where the representation is natural for the novice, convenient for computation, and yet powerful enough to convey difficult concepts among analysts and researchers.

The graphical representations of belief networks and influence diagrams enable us to capture important relationships at the structural level of the graph where it easiest for people to see them and for algorithms to exploit them. Although the diagrams lend themselves to communication, there remains the challenge of synthesis, and building graphical models is still a challenging art. This chapter presents many examples of model structures within these representations and some of their most important properties. It is designed for students and practitioners with a basic understanding of decision analysis.

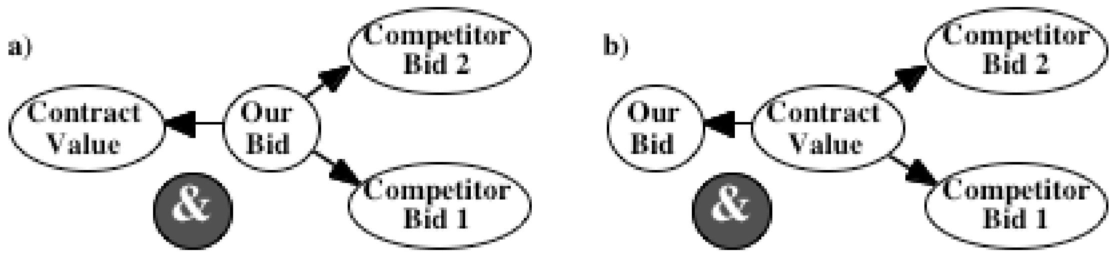

Figure 1. Different models for contract bidding

An example illustrates the value of a clear representation. For many years the accepted model for contract bidding (Friedman, 1956) corresponded to the network shown in Figure 1a. Buried in the mathematics was an incorrect assumption that if we know our bid then the bids of our competitors provides us no new information about the value of the contract. It was only after analyzing bidding patterns that had cost oil companies many extra millions of dollars in Gulf Coast leases that Capen, Clapp, & Campbell (1971) discovered the mistake and presented the model corresponding to the network shown in Figure 1b. In their model, the value of the contract renders our bid and the bids of our competitors irrelevant to each other, assuming no collusion. As a result we have the “winner’s curse,” because observing that we won the bid makes it more likely that our bid is in error. In hindsight and with the insights from graphical models, it is disturbing that the error went undetected for so many years.

The use of a directed graph factorization of a joint distribution is usually credited to (Wright, 1921, 1934) and (Good, 1961a, 1961b), although it was first fully described by (Howard & Matheson, 1984; Miller, Merkofer, Howard, Matheson, & Rice, 1976). It came to computer science and statistics attention through (Kim & Pearl, 1983; Lauritzen & Spiegelhalter, 1988; Pearl, 1988). Since then, there have many significant developments in model building and analysis, particularly with the use of undirected graphs, but our focus here is on the basic models in directed graphs.

In the next section we present examples of probabilistic models represented by belief networks, followed by a section featuring algorithms to explore the belief network structure and use simple directed graphical transformations to analyze inference queries. The final two sections present examples of decision models represented by influence diagrams, and algorithms and directed transformations to analyze them.

1. Probabilistic Models

In this section we consider examples of probabilistic models represented by belief networks. Such networks are known by many names, including Bayesian networks, relevance diagrams, causal probabilistic networks, and recursive causal models. The structure of the graph captures irrelevance information about the underlying probabilistic model through deterministic variables and the arcs that could be present but are missing. The data associated with the nodes in the graph provides a complete probabilistic description of the uncertainties in the model and allows an updating of that description as variables are observed.

The structure of a belief network $B = (U, A, F \mid \&)$ consists of undirected nodes U, directed arcs A between those nodes, and a subset F of the nodes that are determined (functionally) by other nodes. Corresponding to each node j is an uncertain variable X_j, and X_J refers to the variables corresponding to the set of nodes J. The background state of information & represents the perspective of the decision maker (DM) at a point in time. It captures her beliefs, biases, and life experiences in the context of the current situation.

Consider the belief network shown in Figure 2a, representing a contest in which the prize is determined by two coin spins. *Spin 1* and *Spin 2* are oval probabilistic nodes and *Prize* is a double oval deterministic node. If both spins are observed then we will know the prize with certainty, but if they are not then we might be uncertain about *Prize*. The background state of information & is represented by a dark oval node, which might be labeled with the date and/or the DM's name. In Figure 2a, *Spin 1* and *Spin 2* direct arcs toward *Prize*, so they are its parents and it is their child. We denote the parents of node j by *Pa(j)* and following the familial analogy, the set of nodes along directed paths from node j are called its descendants, *De(j)*.

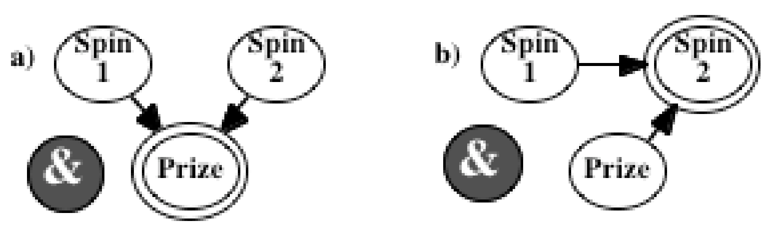

Figure 2. Different models for a coin spinning contest

A belief network B is completely specified if data is assigned to each node in the network. For each node j there is a set of possible values $x_j \in \Omega_j$, and we denote the possible values for a set of variables *J* by $x_J \in \Omega_J = \times_{j \in J} \Omega_j$. If X_j has been observed, then it only has only possible value x_j. We assume in this chapter that Ω_j is finite, but most results can be extended to continuous variables as well. For each X_j there is also a conditional probability distribution for X_j given its parents $X_{Pa(j)}$ (and &), $P\{X_j \mid X_{Pa(j)}, \&\}$. If *j* is a deterministic variable, $j \in F$, then this conditional distribution can be represented by a function, $X_j = f_j(X_{Pa(j)} \mid \&)$. If X_j has been observed then this conditional distribution is a likelihood function, $P\{X_j = x_j \mid X_{Pa(j)}, \&\}$.

It is sometimes tempting to construct a model with a directed cycle, that is, a path following directed arcs taking us back to where we started. Although work has been done to analyze such models, any joint probability distribution can be represented without a directed cycle and we will therefore assume here that there is no such cycle in the network. These cycles can arise when we have a dynamic equilibrium process, such as the relationship between prices and quantities shown in Figure 3a, which can be represented instead as a sequence of prices and quantities over time as shown in Figure 3b. A simultaneous system, such as the one shown in Figure 3c, can usually be triangulated into the system shown in Figure 3d. The cases where it cannot be triangulated are the same as for deterministic systems, that is, when the system is either inconsistent or indeterminate.

When there is no directed path in the network, there must be at least one ordered list of the nodes such that any descendants of a node follow it in the list. Such a list can be constructed in linear time by iteratively selecting a node without a parent and deleting the arcs it directs to its children. For example, in the belief network shown in Figure 2a there are two ordered lists, (*Spin 1*, *Spin 2*, *Prize*) and (*Spin 2*, *Spin 1*, *Prize*), but there is only one for the network shown in Figure 3d, (*X*, *Y*, *Z*).

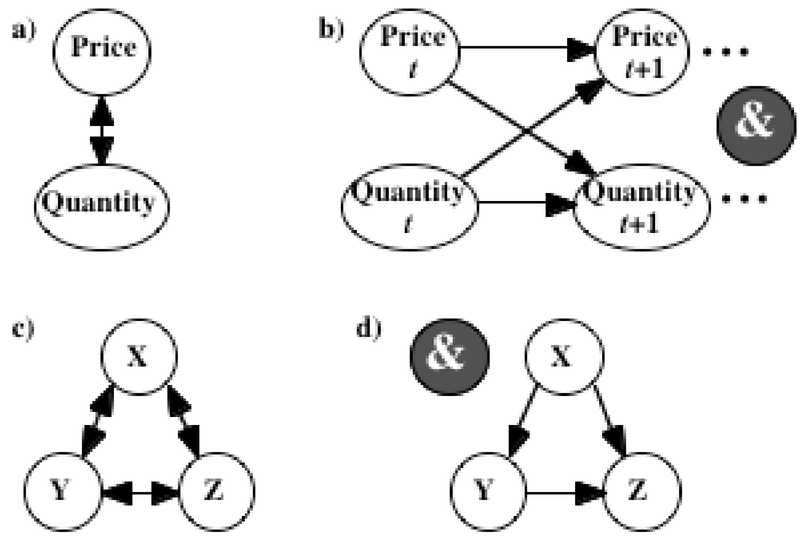

Figure 3. Directed cycles are not permitted

It is convenient to think in terms of an <u>assessment order</u> of the uncertain variables, an ordered list such that the conditional probability of X_j given the variables preceding it in the list is obtained by conditioning X_j on its parents (and the background state of information) alone, $P\{X_j \mid X_1,\ldots,X_{j-1},\&\} = P\{X_j \mid X_{Pa(j)},\&\}$, where we have assumed that the assessment order is 1, 2, … . The conditional irrelevance (or independence) embodied in the belief network *B* is satisfied by a joint probability distribution, and the distribution is said to admit a <u>directed factorization</u> with respect to *B*, if X_j is a deterministic function of $X_{Pa(j)}$ for each *j* in *F* and

$$P\{X_U \mid \&\} = \prod_{j \in U} P\{X_j \mid X_{Pa(j)},\&\}.$$

Given the joint probability distribution $P\{X_U \mid \&\}$ for X_U, X_J and X_L are said to be <u>probabilistically irrelevant</u> (or independent) given X_K (and &) if

$P\{X_J \mid X_K, X_L,\&\} = P\{X_J \mid X_K,\&\}$.

Given the belief network structure *B* and any sets of nodes *J*, *K*, and *L*, X_J and X_L are irrelevant in *B* given X_K if X_J and X_L are probabilistically irrelevant given X_K (and &) for any joint probability distribution $P\{X_U \mid \&\}$ that admits a directed factorization with respect to *B*.

Therefore, if X_J and X_L are irrelevant in *B* given X_K then, no matter what probability distributions are assigned consistent with the network structure, they are probabilistically irrelevant, and we are assured that, having observed X_K we can learn nothing about either X_J or X_L by observing the other. This condition is equivalent to X_j and X_l are irrelevant in *B* given X_K for all *j* in *J* and *l* in *L*, even though such a decomposition is not true in general for probabilistic independence (Pearl, 1988).

As a result, the belief network makes strong assertions about conditional irrelevance but weak ones about relevance, based on which nodes are deterministic and which arcs are present. Each deterministic node and missing arc implies probabilistic irrelevance, while there might always be additional probabilistic irrelevance not captured in the belief network structure. Thus we can be sure which irrelevance statements must be true and what node data is not needed to resolve a query, but we can only say which relevance statements might be true and what node data might be needed. In the next section we will present an algorithm to answer these questions.

To explore this difference between probabilistic irrelevance and irrelevance in *B*, consider the network for the coin spinning contest shown in Figure 2a. DM believes that the two spins are irrelevant and that the prize is completely determined by the two spins, and will only consider probability distributions consistent those beliefs. If she also believes that the spins are both equally likely to land “heads” or “tails” and that the prize will be determined by whether the two coins match, then the network shown in Figure 2b would also be valid, since winning the prize and either coin spin are probabilistically irrelevant in that case. Even in that case, the network shown in Figure 2a still seems more informative! Note that probabilistic independence cannot be decomposed in the same way as irrelevance in *B*, since seeing both coins determines whether we win the prize, even when they each provide no information about the prize.

Now that we have a formal definition of irrelevance in *B*, we can apply it to the belief network shown in Figure 4. The is the simplest example of conditional irrelevance, since the missing arc from *Stock Value Yesterday* to *Stock Value Tomorrow* assures us that, no matter what conditional distributions DM uses consistent with the diagram, yesterday's value and tomorrow's value are irrelevant given today's. This embodies the "Markov property," namely, the present stock value tells us everything we need to know from the present and the past to predict the future. A Markov process, whether it is stationary or not, can be represented by such a "sausage link" belief network, with a node for each stage or time period.

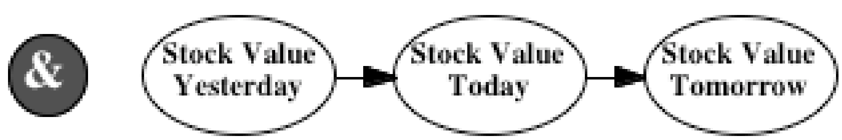

Figure 4. Markov chain for stock values

Another fundamental example involves repeated trials as represented by the belief networks shown in Figure 5. If DM spins a coin *n* times and observes each time whether it lands "heads" or "tails" she might be tempted to represent this experiment by the belief network shown in Figure 5a, in which all of the different spins are irrelevant (given &). However, many coins are not equally likely to land "heads" or "tails" after being spun because of precession, and there is a real opportunity to learn about *Spin n* from the first *n*-1 spins. Therefore, she might prefer the belief network shown in Figure 5b, in which the spins would be irrelevant if she knew some properties of the coin (and &), in this case the probability that it will land “heads” after being spun. This network allows her to learn about the coin from the spins she has observed in order to revise her beliefs about the other spins. Of course, our interest in this model does not arise from a desire to spin coins but because its structure shows up in so many important problems. Consider an experimental drug being tested on some patients, represented by the belief network shown in Figure 5c. If DM assumed that the patients’ recoveries were irrelevant then there would be no benefit to knowing what happened to the earlier patients before she received (or administered) the treatment.

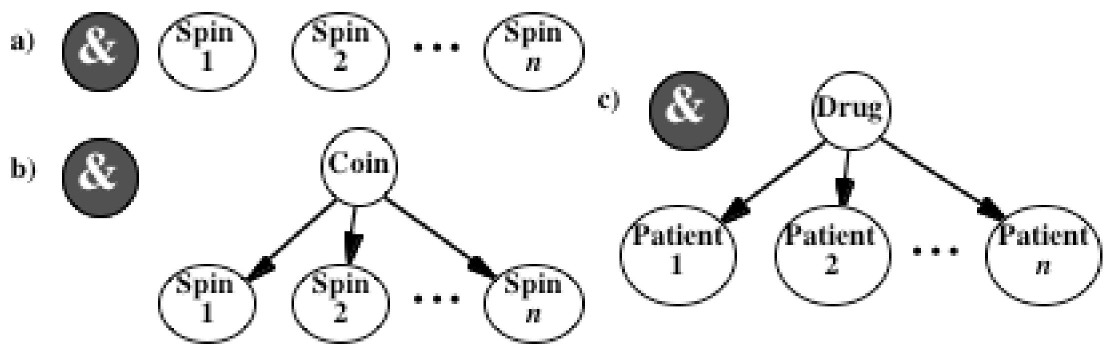

Figure 5. Repeated trials that are conditionally irrelevant

The belief network facilitates the updating of DM’s beliefs about all of the variables when she obtains evidence. Evidence is represented as the observation of one of the variables in the network. Consider the network shown in Figure 6a, which represents how the alignment of a production process affects the quality of the product. It is possible to sample some of the product for testing in order to learn whether the process is still in alignment. If DM observes the test results, she can update her beliefs about alignment and make a decision whether to ship the product. We represent the observed variable with shading and recognize that it now only has one possible value—the value observed. (If it had children in the network, then she could also cut those arcs, since the children can condition on the observed value for this variable.)

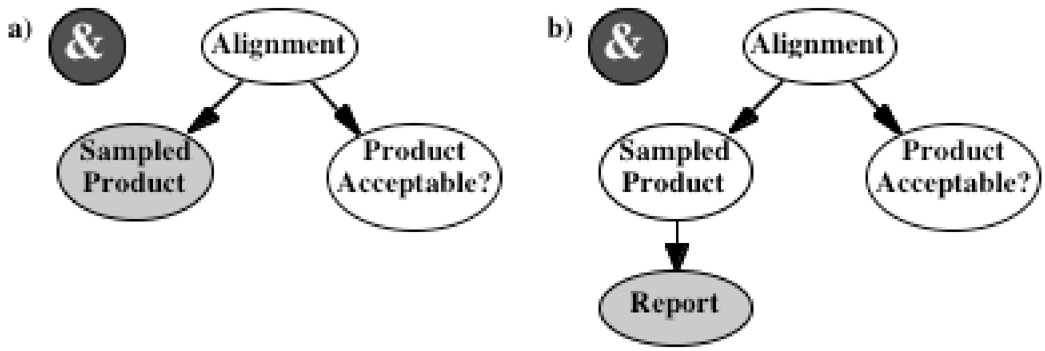

Figure 6. Evidence enters through its likelihood of observation

Treating a variable as observed is absolute. If DM is not sure that it has that value, then she can represent that uncertainty by creating a new node, such as *Report* in the network shown in Figure 6b, to represent what she has observed. In this case, she would need to specify how likely this report would be for all of the possible values of the sample test results. Many famous paradoxes are avoided by this simple mechanism of observation rather than directly updating DM's beliefs about *Alignment*. In fact, she is using this mechanism whether she observes *Sampled Product* or *Report*. Of course, if she is not sure what is in the report, then she can create still another node, its child, that she has observed instead.

The DM's background state of information & is a special case of an observed variable. There is only one possibility for & in the network and everything is implicitly conditioned on it.

In this example and the ones that follow, the observed node can be considered an "effect" and its parent(s) are the "causes." Most updating situations in belief networks involve observing a effect child to learn about a causal parent. This is because it is usually more natural to construct a belief network in the causal direction and because we often obtain evidence about a variable of interest through its effects (Shachter & Heckerman, 1987).

We saw in the coin spinning contest example that two spins DM considered irrelevant could become relevant if she observed a prize determined by them. This situation can arise whenever she observes a common descendant of uncertainties that she had believed to be irrelevant. Suppose DM is visiting a friend's house and suddenly starts sneezing. She worries she is coming down with a cold until she notices some scratches on the furniture. If her friend has a cat, her sneezing might be due to her allergy to cats rather than a cold (Henrion & Druzdzel, 1990). The belief network shown in Figure 7 represents this story.

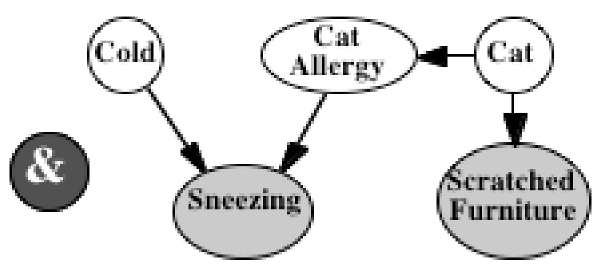

Figure 7. Observing a common child can make irrelevant uncertainties relevant

Before she started sneezing, DM thought that *Cold* and *Cat Allergy* were irrelevant and that observing one would tell her nothing about the other. Once she observes *Sneezing*, however, they become relevant—if she observed that one were absent then the other would become more likely. When she also observes *Scratched Furniture*, *Cat* and *Cat Allergy* become more likely, and *Cold* becomes less likely than it was after she first observed *Sneezing*.

We will see a general algorithm for recognizing irrelevance in the next section, but this example provides some intuition. In some cases, an observation creates irrelevance while in other cases it can eliminate it. Before she observed *Sneezing*, *Cold* and *Scratched Furniture* were irrelevant, but once she sneezed, they became relevant. On the other hand, if she then also observed *Cat* or *Cat Allergy* then *Cold* and *Scratched Furniture* would again be irrelevant, since the scratches on the furniture are relevant to cold only because they suggest to us the presence of a cat and our cat allergy.

Another common and important model is the partially observable Markov chain, as shown in Figure 8, in the context of an optical character recognition system. The system observes pen strokes and tries to infer a sequence of characters, using a Markov model to capture the relevance of adjacent characters. The strokes for each character represent an imperfect observation of the character. Adding more layers of meaning could refine this model, much as people employ in deciphering handwriting, but it would become considerably more complex.

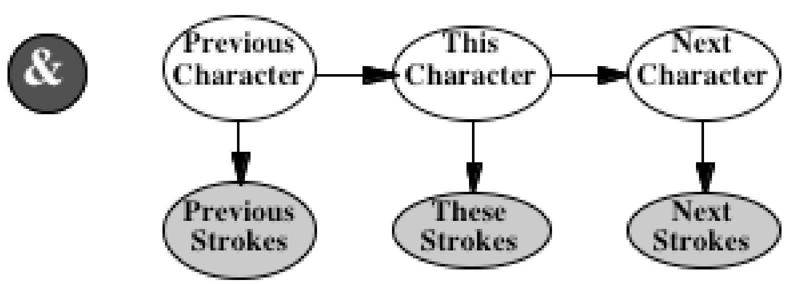

Figure 8. Partially observable Markov chain

The belief network structure shown in Figure 8 also corresponds to Kalman filters and hidden Markov models. The models all feature a Markov process backbone with imperfect observations of the Markov variables. Usually they are used to model a process over time but they can also be applied to other sequences as well.

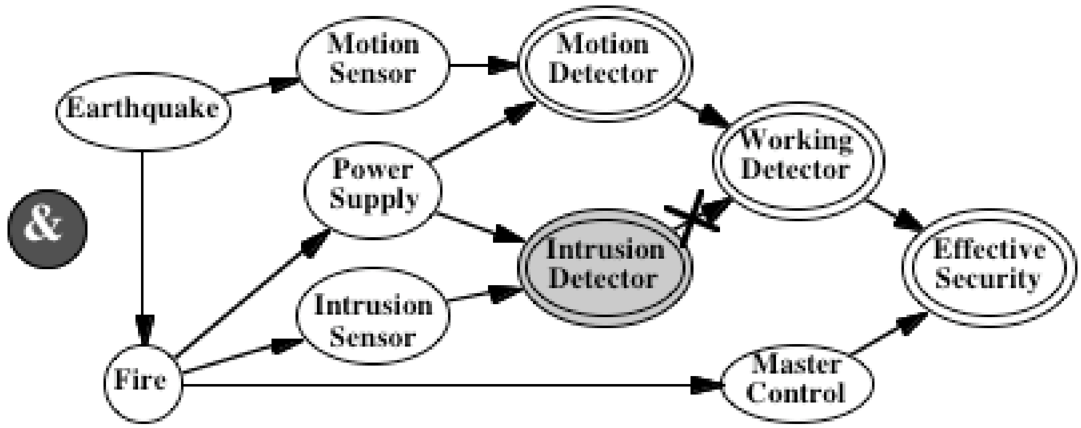

Figure 9. Fault belief network

Many applications involve whether a critical system is functioning. Suppose a security system has two kinds of sensors, one for intrusion and one for motion, both sharing a common power supply. The system can function as long as one of the sensors and the master control are working. There are potential failures of all of these systems in the event of an earthquake or a fire. A belief network for this example is shown in Figure 9, where the deterministic nodes have "and" and "or" functions. Note that if the DM could observe *Motion Sensor*, *Power Supply*, *Intrusion Sensor*, and *Master Control*, then she would know the values of all of the deterministic variables. If she observed that the intrusion sensor were not working, she could update her beliefs about all of the other variables, both to know whether the overall system is functioning and how to plan repairs for the parts that have failed. At that point, *Intrusion Detector* would have only one possible value and the arc to its child can be cut. Even though it is a deterministic function of its parents, there are multiple values of the parents that might yield that value, so the diagnostic task is not deterministic in general, even for a system with determinacy.

Traditionally, these systems have been modeled using fault trees, which must be augmented to capture common cause failure, the relevance among components. The belief network makes it easier to represent these relationships. There are three fundamentally different kinds of tasks that one can perform using this type of network. First is prediction – given the observations available (and &), how likely is the system to remain effective until repairs are completed. Second is diagnosis – given those same observations, which components are likely to have failed. Both of these tasks lead to decisions about information collection, such as troubleshooting, and repair strategies. Third is system design – looking prospectively at the response of the system to internal and external failures, such as whether there is sufficient redundancy to deal with the impacts of fires or earthquakes.

In the next section we will see how we can analyze these belief networks, to understand the implications of the model structure and how we can update our beliefs.

2. Analyzing Probabilistic Models

Much of the popularity of belief networks arises from our increasing computational power and the development of efficient algorithms. This section presents some of those algorithms, primarily to determine the implications of our belief network structure for irrelevance and data requirements, but also to show how we could update our beliefs after making observations. There is a large and growing literature on the efficient solution of these inference problems, mostly using undirected graphical structures and beyond the scope of this chapter (Jensen, Lauritzen, & Olesen, 1990; Lauritzen & Spiegelhalter, 1988; Shenoy & Shafer, 1990). Our focus instead will be on how we can explore and modify the belief network structure in order to gain some intuition for modeling and analysis.

A useful and efficient algorithm, Bayes-Ball, identifies for any query $P\{X_J \mid X_K, \&\}$ which variables in a belief network must be irrelevant to X_J given X_K (and &), and which conditional distributions and observation might be needed to resolve the query (Shachter, 1998). We call the information that might be needed the requisite distributions and requisite observations for *J* given *K*. The Bayes-Ball algorithm runs in time linear in the size of the belief network and is based on the metaphor of a bouncing ball. The ball sometimes bounces off a node, sometimes passes through it, and sometimes gets stuck, depending on the type of node and whether the ball is approaching the node from a parent or from a child, as shown in Figure 10.

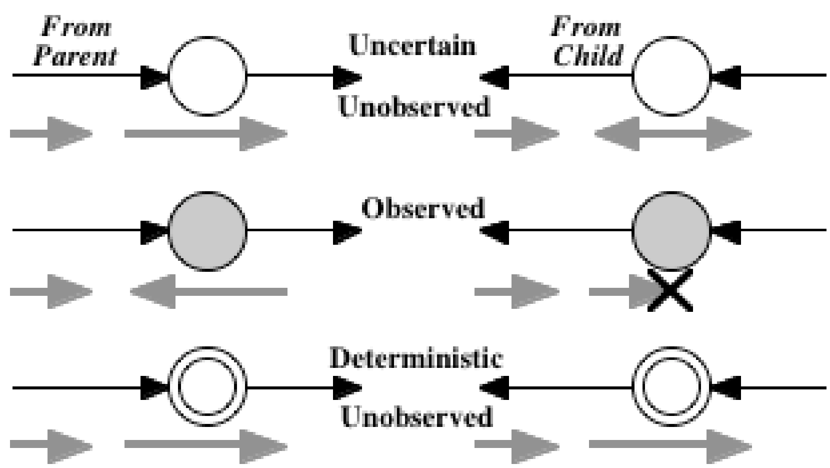

Figure 10. Bayes-Ball

The Bayes-Ball Algorithm explores a structured belief network $B = (N, A, F \mid \&)$ with respect to the expression $P\{X_j \mid X_K, \&\}$ and constructs the sets of irrelevant and requisite nodes.

1. Initialize all nodes as neither visited, nor marked on the top, nor marked on the bottom.

2. Create a schedule of nodes to be visited, initialized with each node in *J* to be visited as if from one of its children.

3. While there are still nodes scheduled to be visited:

   a. Pick any node *j* scheduled to be visited, mark it as visited (√) and remove it from the schedule.

   b. If $j \notin K$ and *j* was scheduled for a visit from a child:

      i. If the top of *j* is not yet marked, then mark its top and schedule each of its parents to be visited;

      ii. If $j \notin F$ and the bottom of *j* is not yet marked, then mark its bottom and schedule each of its children to be visited.

   c. If *j* was scheduled for a visit from a parent:

      i. If $j \in K$ and the top of *j* is not yet marked, then mark its top and schedule each of its parents to be visited;

      ii. If $j \notin K$ and the bottom of *j* is not yet marked, then mark its bottom and schedule each of its children to be visited.

4. The irrelevant nodes, $N^I(J \mid K)$, are those nodes <u>not</u> marked on the bottom.

5. The requisite distribution nodes, $N^P(J \mid K)$, are those nodes marked on the top.

6. The requisite observation nodes, $N^E(J \mid K)$, are those nodes in K marked as visited.

An example of the Bayes-Ball Algorithm is shown in Figure 11 for the query $P\{X_{10}, X_{11} \mid X_1, X_4, X_7, \&\}$. The algorithm determines irrelevant set N^I={1, 2, 3, 4, 7, 12}, requisite distributions N^P={3, 5, 6, 7, 10, 11, 13} and requisite observations N^E={4, 7}. The non-requisite nodes $N-(N^P \cup N^E)$={1, 2, 8, 9, 12} could be deleted from the belief network and we would still be able to resolve the query.

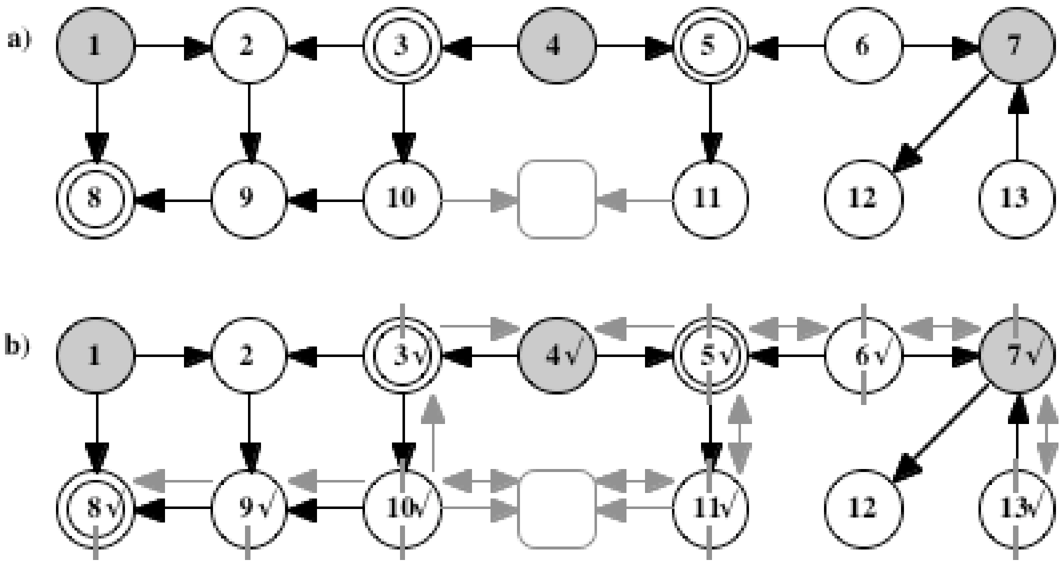

Figure 11. Bayes-Ball Algorithm example

Probabilistic inference can be performed by graphical operations on the belief network that alter the ordered list of the requisite nodes. This allows us to transform from the assessment order ideal for model construction and communication with DM to the ordered list needed to resolve a query, such as $P\{X_J \mid X_K, \&\}$. Once the nodes *K* (some of which might already be observed) are the only other parents of the nodes *J*, we have solved the problem (Shachter, 1988).

Consider the belief network shown in Figure 12a representing an HIV test. *HIV+* is an uncertain variable indicating whether DM has been infected with the HIV virus and *"HIV+"* is her test result. Her test result and *Risk Behavior* are irrelevant given her infection status, and the type of *Test* performed is irrelevant to both her risk behavior and her infection status, at least until she observes her test result. Suppose she would prefer the ordered list (*Risk Behavior*, *Test*, *"HIV+," HIV*). Bayes-Ball can be used to find the requisite observations for *P*{*"HIV+"* | *Risk Behavior*, *Test*, &} and both *Risk Behavior* and *Test* are requisite. All of the variables that precede *HIV* are also requisite, so we obtain the belief network in Figure 12b. This is intuitive, since the type of test is now relevant to her infection status, and her risk behavior is relevant to the test results.

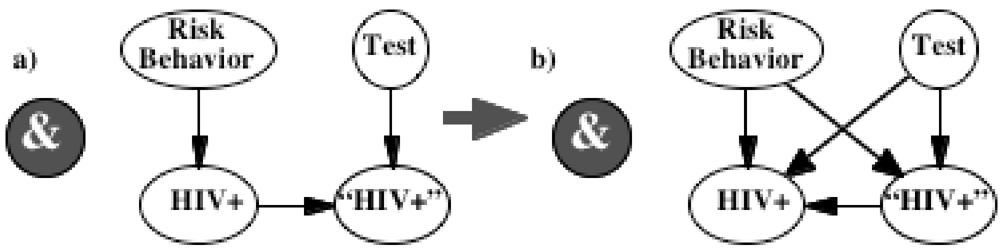

Figure 12. HIV test and arc reversal

This graphical transformation is called arc reversal. The arc from node *i* to node *j* is reversible if there is no other directed path from *i* to *j*; if there were, then an arc from *j* to *i* would create a directed cycle. Reversing the direction of the arc so that it goes from *j* to *i* is exactly the same operation as flipping a probability tree. In the process, the two nodes inherit each other's parents, so arc reversal can add arcs as shown in Figure 13a. The first step is to compute the joint distribution of the two corresponding variables,

$$P\{X_i,X_j \mid X_J,X_K,X_L,\&\} = P\{X_i \mid X_J,X_K,\&\}P\{X_j \mid X_i,X_K,X_L,\&\},$$

where $J = Pa(i) - Pa(j)$, $K = Pa(i) \cap Pa(j)$, and $L = Pa(j) - (\{i\} \cup Pa(i))$.

From that joint distribution we can compute new distributions for X_i and X_j,

$$P\{X_j \mid X_J,X_K,X_L,\&\} = \sum_{x_i \in \Omega_i} P\{X_i,X_j \mid X_J,X_K,X_L,\&\}$$

and $$P\{X_i \mid X_j,X_J,X_K,X_L,\&\} = \frac{P\{X_i,X_j \mid X_J,X_K,X_L,\&\}}{P\{X_j \mid X_J,X_K,X_L,\&\}}.$$

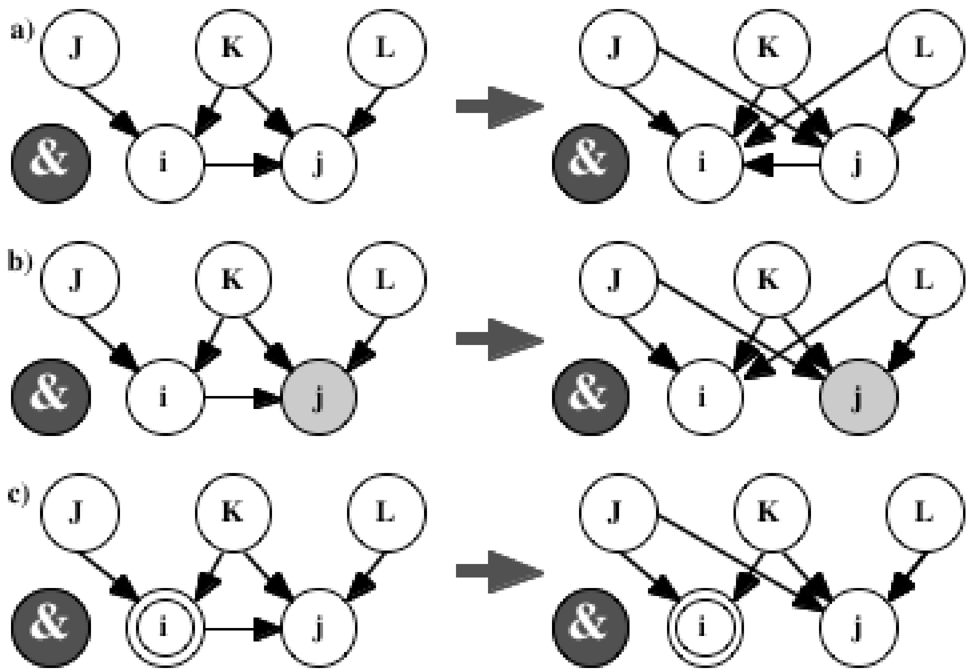

Figure 13. The three kinds of arc reversal

There are two special cases for arc reversal. The first, <u>evidence reversal</u>, arises when X_j has been observed and thus has only one possible value. The only difference is that there is no need for an arc from *j* to *i* afterward as shown in Figure 13b. The same reversibility condition applies nonetheless, since any other path from *i* to *j* would go through the set *L* of nodes becoming parents of *i*, thus creating a directed cycle. The other special case, <u>deterministic propagation</u>, arises when *i* is a deterministic node. In this case, there is no uncertainty about X_i given $X_{Pa(i)}$, so there is no need to add additional parents, either *j* or *L*, as shown in Figure 13c. In this case there is no reversibility condition at all since the only arcs being added are from *J* to *j* replacing the arc from *i* to *j*, and corresponding to the substitution,

$$P\{X_j \mid X_J, X_K, X_L, \&\} = P\{X_j \mid f_i(X_J, X_K), X_K, X_L, \&\} \ .$$

Given arc reversal, computing $P\{X_J \mid X_K, \&\}$ is then a matter of reordering the graph so that the nodes in *K* precede the nodes in *J* and they both precede the other nodes.

In fact, once a node outside of $J \cup K$ has no children, it can be removed from the network as a barren node. Any node outside of $J \cup K$ can be removed from a model by reversing the arcs to its children in order until it is barren (Shachter, 1988).

3. Decision Models

In this section we consider examples of decision models represented by influence diagrams. Influence diagrams express choices and preferences as well as beliefs and information. They consist of belief networks with two additional node types, representing decisions and criteria for making choices. In addition to the beliefs at the time the network is constructed, the influence diagram represents the prospective observation of variables that are still uncertain.

The structure of an influence diagram consists of nodes, arcs, and &, as in the belief network but now there are additional types of nodes. Decision nodes *D* correspond to variables complete under DM's control, and value nodes *V* represent the criteria she uses to make those choices. Consider the influence diagram shown in Figure 14a, representing an investor's decision problem. She has complete control over *Stock Purchase* and she will choose it to obtain *Profit* that will maximize the expected value of her *Satisfaction*. Her decision variable *Stock Purchase* is drawn as a rectangle and her value variable *Profit* is drawn as a rounded rectangle. (In the literature, value variables are drawn in a variety of shapes.)

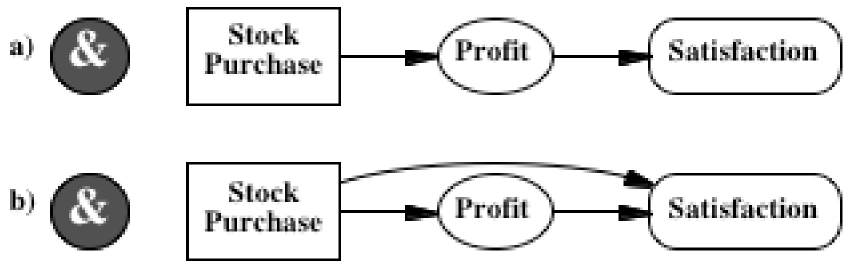

Figure 14. Investor's decision opportunity

An influence diagram is completely specified if possible values are assigned for each node and conditional probability distributions for all the nodes in *U* and *V*. The value variables *V* are assumed to be deterministic functions of their parents, just like the uncertain variables in *F*, and we assume that the total value is the sum of all of the value functions. (Some results still apply even if the value functions are multiplied instead of added.) Conditional probability distributions are not needed for the decision nodes *D* – these will be determined so as to maximize the expected total value.

We assume that all of the decision variables can be ordered in time, $d_1, d_2, \ldots, d_m$, and that the parents of each decision variable represent all of the other variables that will have been observed before that choice must be made. We assume the no forgetting condition, that the parents of each decision include all earlier decisions and the variables observed before them (Howard & Matheson, 1984). As a result, we can partition the uncertainties into m+2 sets, decision windows $W_0, \ldots, W_{m+1}$, where the uncertain variables in W_0 have already been observed, those in W_i will be observed before the choice of d_i but after the choice of d_{i-1}, and those in W_{m+1} will not be observed before any of the decisions, as shown in Figure 15. The windows can be thought of as one-way, so future decisions can observe anything from earlier windows but nothing from the future. More formally, $Pa(d_1) = W_0 \cup W_1$ and $Pa(d_i) = Pa(d_{i-1}) \cup \{d_{i-1}\} \cup W_i$ for i=2, …, m. To reduce clutter in the diagram, arcs into decisions will not be drawn when they can be inferred by no forgetting.

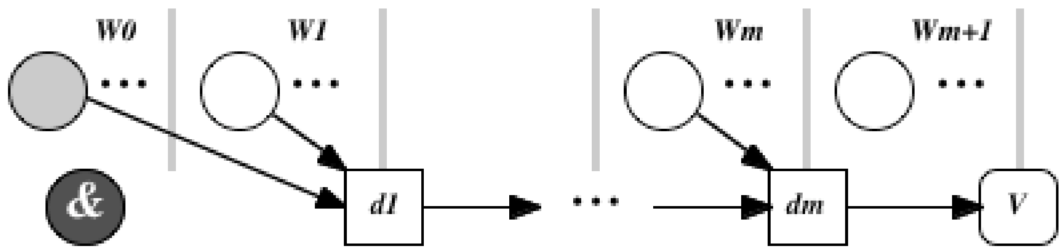

Figure 15. Decision windows and the no forgetting condition

Consider again the influence diagram drawn in Figure 14a. As stated above, DM completely controls the decision *Stock Purchase* so as to maximize the expected value of her *Satisfaction*. This is a causal notion, because she is convinced that her choice will have some impact or effect on the world, in this case on *Profit*. She anticipates the prospect of satisfaction based on that profit. Therefore, her choice of stocks affects every descendant of her decision, her satisfaction as well as her profit. If she could not affect any of her value variables by making her choice, why would she devote any attention to that choice? Now, suppose that she wants to base her purchase decisions on factors besides profit, that is, she is willing to forgo some potential profit to own companies she prefers for other reasons. The influence diagram shown in Figure 14b represents this situation, since *Stock Purchase* and *Satisfaction* are no longer irrelevant given *Profit*.

Now that we have seen the role that cause plays in influence diagrams we can recognize a particular type of belief network called a causal network, in which all of the arcs have a causal interpretation (Heckerman & Shachter, 1994; Pearl, 2000). A belief network is said to be a causal network if, were DM able to completely control one (or more) of the uncertain variables, the influence diagram obtained by cutting any incoming arcs from parents, and changing it (them) to decision node(s), is valid. (We should also have a meaningful value node, too, although this is not usually mentioned.)

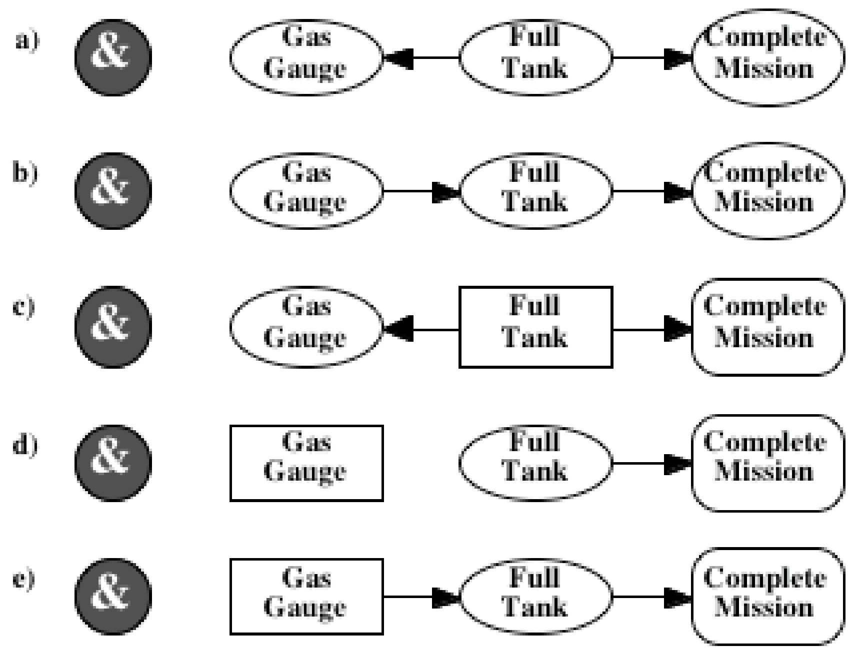

Figure 16. Causal networks are special belief networks

Consider the belief networks shown in Figure 16a and16b. A full gas tank is necessary to complete the mission and the gas gauge indicates whether the tank is full. Probabilistically, these two networks are indistinguishable, representing identical irrelevance relationships. However, they have quite different meanings as causal networks. If we could control the tank in the first network we obtain the influence diagram shown in Figure 16c. Filling (or emptying) the tank would affect the gauge and the mission. If, instead we control the gas gauge, we get the influence diagram shown in Figure 16d. Manipulating the gauge has no affect on either the tank or the mission. On the other hand, if we could control the gauge in the second belief network we get the influence diagram shown in Figure 16e. According to this diagram, manipulating the gauge affects both the tank and the mission. Clearly, the belief network shown in Figure 16a is a causal network and the one shown in Figure 16b is not.

The influence diagram shown in Figure 17a represents a prototypical decision opportunity. The DM is choosing a vacation activity but her favorite activity if the weather were nice would not be her choice if the weather were bad. She can obtain a long-range weather forecast before she has to put down a deposit on her choice. She has no ability to change the weather, but she does have an opportunity to learn about it before she makes her choice. Her satisfaction is based on the weather and her choice of activity.

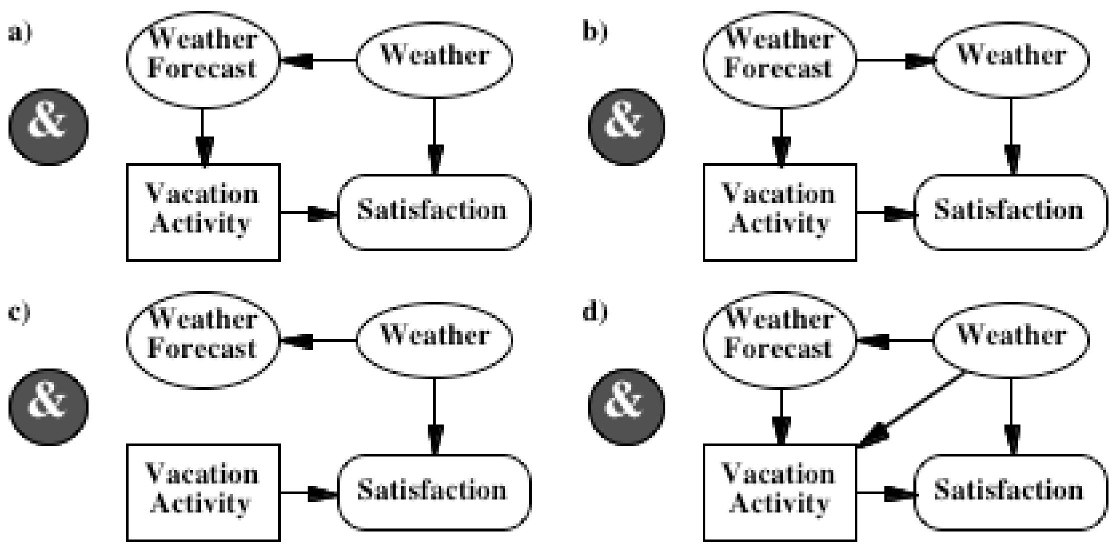

Figure 17. Choosing a vacation activity

The arc from *Weather Forecast* to *Vacation Activity* is an informational arc, indicating that she will observe the forecast before she must commit to an activity. Because the informational arc is explicit rather than implicit, as it is in decision trees, the assessment order does not represent the order in which uncertain variables are observed. Thus, even though *Vacation Activity* is a descendant of *Weather*, there is no claim that *Weather* will be known at the time of *Vacation Activity*. This also means that this problem cannot be represented as a decision tree without reversing the direction of the arc from *Weather* to *Weather Forecast* to obtain the influence diagram shown in Figure 17b.

It is common to ask what a decision "depends upon," and on reflection it is clear that all of the elements of the influence diagram contribute to making a good decision. But the arcs do not indicate what the decision "depends upon," any more than they represent a flow chart. The arcs into the decision represent what other variables will have been observed before the choice is made and the arcs into the value node(s) indicate what criteria should be used in making that choice, all based on our background state of information &.

Consider now the influence diagram shown in Figure 17c. It only differs from the one shown in Figure 17a by the deletion of the informational arc from *Weather Forecast* to *Vacation Activity*. If the forecast were available at no cost then DM would be at least as well off in the earlier situation as in this one. Observing *Weather Forecast* cannot make her worse off, since it gives her the opportunity to make a different choice of activity for each possible forecast instead of having to make just one choice. If she chose the same activity regardless which forecast she observed then she should be indifferent between the two situations and there is no benefit to making the observation. The difference in value between these two diagrams represents the value to DM of observing *Weather Forecast* before making the *Vacation Activity* decision.

Consider instead the influence diagram shown in Figure 17d. It only differs from the one shown in Figure 17a by the addition of the informational arc from *Weather* to *Vacation Activity*. We saw that this additional information cannot make DM worse off, and in this particular case we would expect her to be much better off, since the weather is the key uncertainty in making her decision. This new diagram represents perfect information or clairvoyance in her choice and we can measure that change by the difference in the value of the two diagrams. If the value for each diagram represents her willingness to pay to be in that decision situation then the difference represents her willingness to pay a clairvoyant to tell her what the weather would be before she makes her choice. Of course, such a clairvoyant does not exist in real life, but the value provides an upper bound on the amount she should pay for any information about the weather. In fact, she might achieve the same benefit by choosing a resort that offers a variety of different activities that she enjoys, so that she can make her choice each day knowing what the weather will be. In this particular example the value of clairvoyance is clearly the benefit of observing *Weather* before choosing *Vacation Activity*, but when there are multiple decisions and uncertain variables, there can be multiple values of clairvoyance, depending on which variables will be observed and when they will be observed.

We can extend these concepts to situations with multiple decisions and, in particular, a decision about collecting information. The influence diagram shown in Figure 18 represents a decision by a patient with suspected Herpes encephalitis. Left untreated, the patient is unlikely to survive, but the standard therapy, treatment with vidarabine, has serious side effects. To test for the presence of the disease, doctors can perform a biopsy, removing some of her brain tissue for examination, a procedure that can also have serious side effects. There are two decisions, *Perform Biopsy?* And *Treat with Vidarabine?*, both posing potential complications. The key uncertainty is *Herpes Encephalitis*, and she cares about her *Quality of Life*. Her treatment decision affects her quality of life both through possible *Recovery* and *Severe Complication*, while her biopsy decision affects it directly through *Severe Complication* and indirectly through the information available about *Herpes Encephalitis* when the treatment decision is made. (If she does not perform the biopsy then *Biopsy Indication* will have the value "not available.") Note that if the arc from *Perform Biopsy?* to *Treat with Vidarabine?* had been omitted, it would have been inferred through no forgetting.

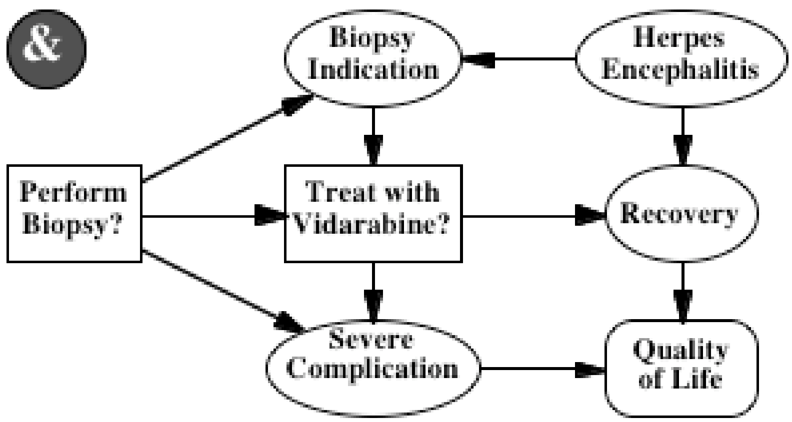

Figure 18. Treating suspected Herpes encephalitis

In this situation, there is considerable value to clairvoyance on *Herpes Encephalitis?* before *Treat with Vidarabine?*, since otherwise brain cells might be taken from DM to gather information. In medical problems, the cost of testing can be expensive in dollars but also in terms of pain, morbidity, and mortality, both from the test itself and from the delay in treatment. In this influence diagram it is not possible to determine the value of clairvoyance on *Recovery* before *Treat with Vidarabine?*, because an arc from *Recovery* to *Treat with Vidarabine?* would create a directed cycle. Recovery is affected by her treatment choice, so it would not make sense for her to know whether she recovers before making that choice. We resolve this problem by adding more data and structure to the influence diagram until it is in canonical form, as explained below.

Another example of an information gathering decision is faced by an oil wildcatter DM considering a potential drilling location (Raiffa, 1968), represented by the influence diagram shown in Figure 19. The wildcatter must choose whether to perform a seismic test and then whether to drill, and her key uncertainty is the amount of oil. There are costs for testing and drilling and there are potential revenues if she drills and finds oil. Before she chooses whether to drill, she will know whether she ordered a test and its results. *Test Report* will be "not available" if she orders no test, equal to *Seismic Structure* if she orders that test, or otherwise equal to the results of a cheaper experimental seismic test.

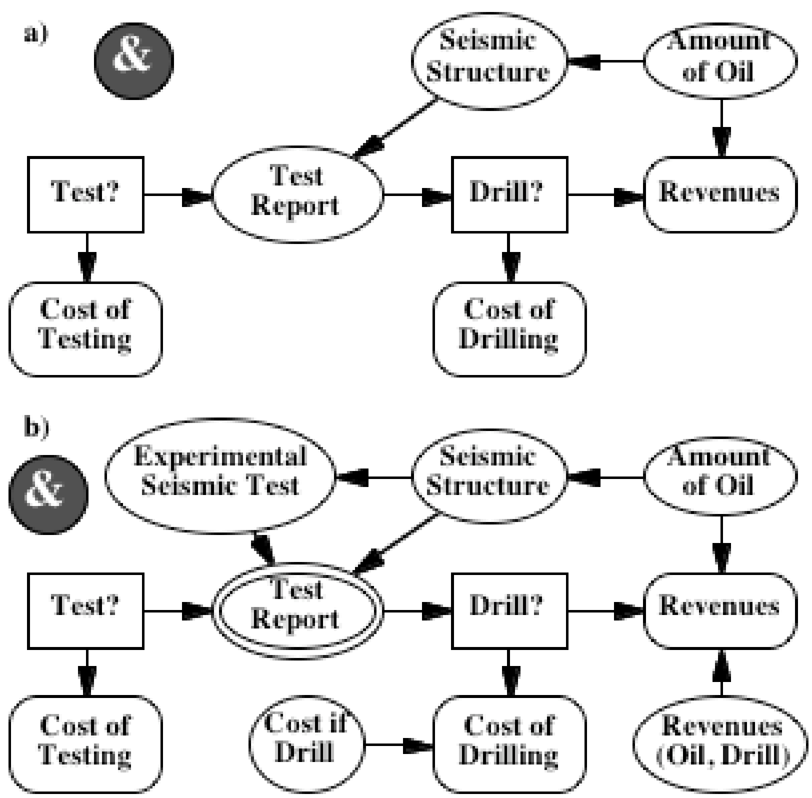

Figure 19. The oil wildcatter and canonical form

The wildcatter can use the influence diagram shown in Figure 19a to make and value her decisions. It would even permit her to compute her value of clairvoyance on the amount of oil or the seismic structure before either of her decisions. However, there are things that are hidden about the test report, and she is unable to compute her value of clairvoyance on the revenues or the cost of drilling. We can address these issues by adding information and structure to the diagram so that it is in canonical form as shown in Figure 19b.

An influence diagram is in canonical form if every uncertainty that is affected by a decision is represented by a deterministic node descendant of that decision in the influence diagram (Heckerman & Shachter, 1995; Howard, 1990; Matheson, 1990). The cost of drilling depends on whether she chooses to drill and how much it would cost if she drills. Thus it is both uncertain and somewhat under the control of the decision maker. To make *Cost of Drilling* deterministic, we must the remove the uncertainty to another node, *Cost if Drill*, unaffected by DM, and for which she can value clairvoyance. Likewise, to make *Revenues* deterministic, we must remove the uncertainty to another node, *Revenues(Oil, Drill)*, which is an uncertain table of values corresponding to all possible values of *Amount of Oil* and *Drill?*. This table is unaffected by DM so she can value clairvoyance for it. Similarly, to make *Test Report* deterministic, we must explicitly represent the results from the different tests as uncertain variables, unaffected by DM and for which she can value clairvoyance. In the process of transforming an influence diagram into canonical form, the causal mechanisms must be explicitly represented for every uncertain variable affected by decisions.

A classic model of decision making over time is the Markov decision process (Howard, 1960) represented by the influence diagram shown in Figure 20. There are multiple stages, usually thought of as epochs in time, *t*, *t*+1, …, and in each stage there is a state of the process, an action to be chosen, and a (discounted) reward for that period. The state for the next stage is affected by the current action and the reward for each period is based on the current state, the current action, and the next state. It is well known that even though all of the states and decisions in the past are observable now through no forgetting, the current state contains all of the information from the past and present needed to value and make decisions in the present and in the future. In the terminology of the next section, it is the requisite observation for the current action.

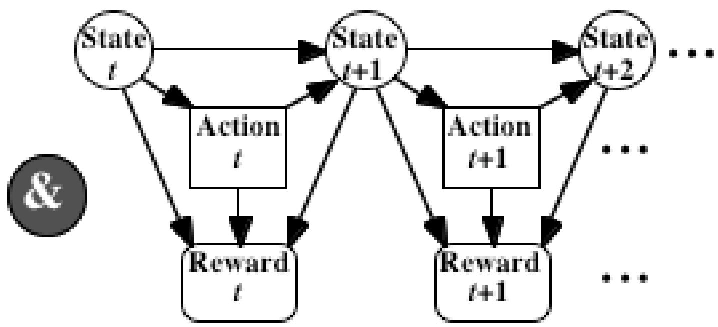

Figure 20. Markov decision process

There are many situations where DM wants to analyze the decisions of others. Imagine a health care provider deciding whether to offer a screening test. If DM were the patient receiving the test, she might represent the situation using the influence diagram shown in Figure 21a, similar to the one in Figure 17a. The focus when considering a screening test is typically on the sensitivity and specificity of the test, *P*{*Test Results*|*Disorder*, &}. However every aspect of the influence diagram matters. What treatment alternatives are available and how effective are they in terms of the health outcomes? What is the prevalence of the disorder, *P*{*Disorder*, &}? If the DM is not the patient, but rather the health care provider offering the screening test, then *Treatment* is someone else's choice, not under her control and therefore uncertain, and we get the diagram shown in Figure 21b.

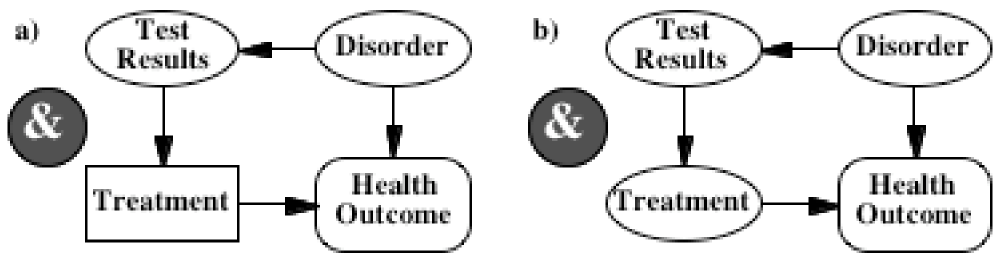

Figure 21. Screening for disease

The focus in the examples so far has been on the relationships among the uncertain and decision variables, but influence diagrams can be used to capture the structure of preferences as well. Consider the objectives hierarchy for automobile safety (Keeney, 1992) represented by the influence diagram shown in Figure 22. This could easily be incorporated into a more complex decision model, harnessing the graphical intuition of the influence diagram for models with multiple attributes and objectives.

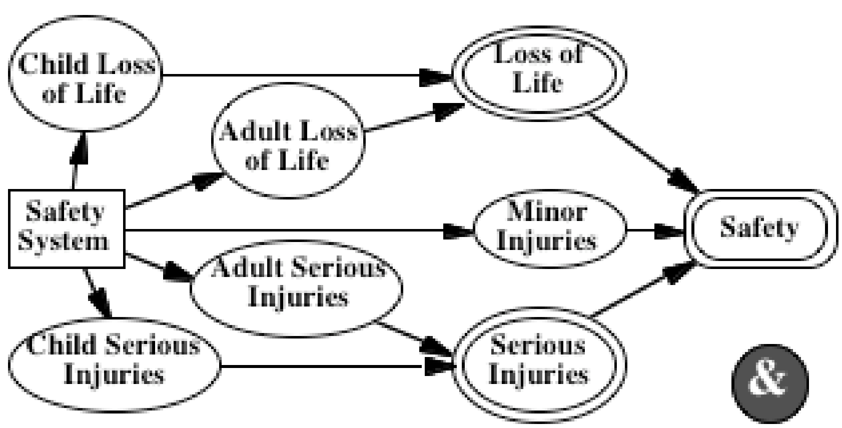

Figure 22. Objectives Hierarchy for Automobile Safety

4. Analyzing Decision Models

Although influence diagrams have not yet achieved the popularity of belief networks, they benefit from many of the same improvement in computational power and the development of efficient algorithms (Jensen, Jensen, & Dittmer, 1994; Keeney, 1992; Shachter & Peot, 1992). This section presents some of those algorithms, not in their most efficient implementations, but primarily to gain some intuition for modeling and analysis.

There is a straightforward extension of the Bayes-Ball Algorithm for decisions that explores a structured influence diagram to determine the requisite observations for each decision (Shachter, 1999). Visiting each decision node d_i in reverse chronological order, $i = m, \ldots, 1$, replace d_i with a deterministic node with parents $Pa(i) = N^E(V \cap De(d_i) \mid \{d_i\} \cup Pa(d_i)) - \{d_i\}$, the requisite observations for d_i with respect to the values it affects. (Note that if d_i is not in its requisite observation set it can be eliminated, since it has no effect on DM's total value.) At this point, the requisite observations to determine the total value are given by $N^E(V \mid W_0)$. Of course, while gathering the requisite observations, we can also determine the requisite distributions.

As an example of the Bayes-Ball Algorithm for decisions, consider a situation in which both historical data and new experimental evidence about an uncertain state are available to DM in making a decision, and she also chooses which evidence to collect. There is a cost of the evidence and a benefit based on her choice and the state. The influence diagram for this situation is shown in Figure 23a. Bayes-Ball is run on the *Act* decision, the *Design* decision, and on the situation before the *Design* decision in the diagrams shown in Figures 23b, 23c, and 23d. In this situation all of the variables are requisite.

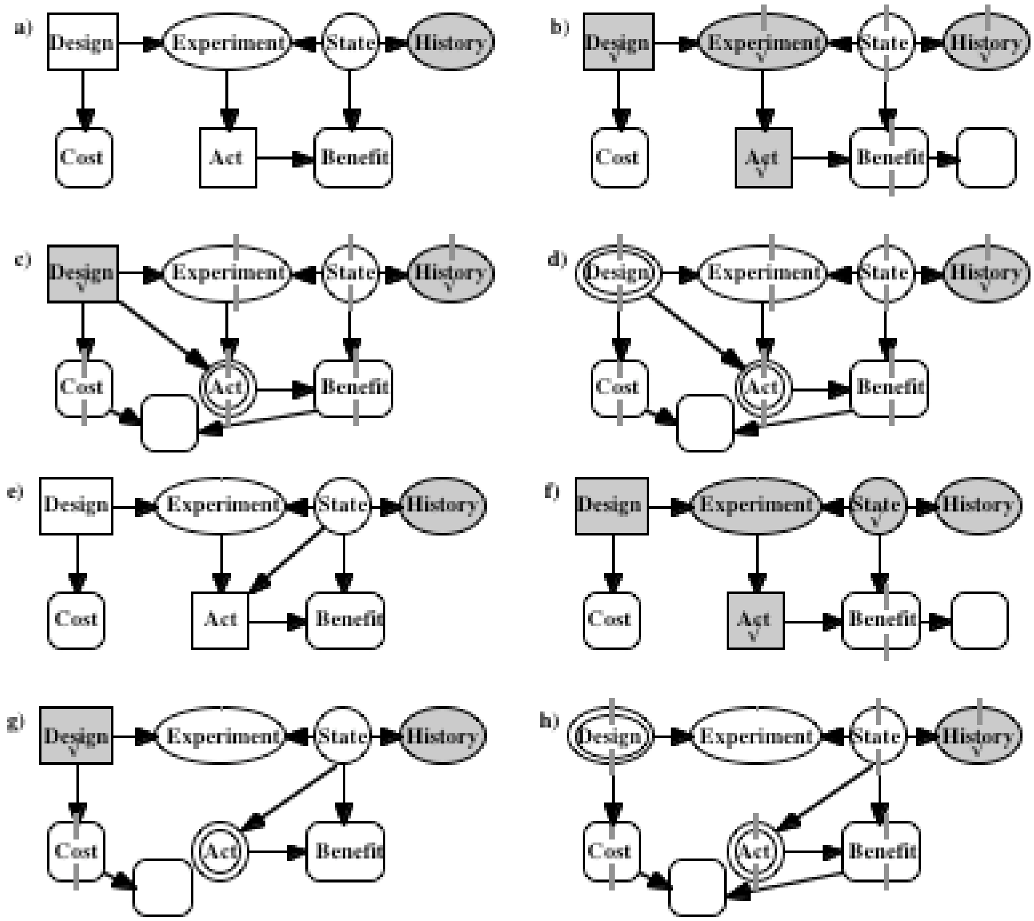

Figure 23. Bayes-Ball Algorithm for Decisions

The same decision situation with clairvoyance on *State* before *Act* is represented by the influence diagram shown in Figure 23e and the Bayes-Ball diagrams are shown in Figures 23f, 23g, and 23h. Now the only requisite information to make the optimal choices are the value functions and the *State* observation. Considerably more information is needed to value this decision situation prospectively. In that case the only non-requisite variable is *Experiment*.

The task of analyzing influence diagrams, actually determining the optimal policies and valuing the decision situation can be accomplished by graphical operations on the diagrams that alter the ordered list and successively remove nodes from the diagram. The diagram is reduced until all that remains is a single value node so that we can value the decision situation.

The simplest transformation, barren node removal, can be used at any time that a decision or unobserved uncertain node has no children – at that point, it can be simply removed from the diagram, since it no longer affects the total value. The requisite parent sets for decisions computed by Bayes-Ball can also be used to simplify the problem.

There are three other graphical transformations needed, one to remove decision nodes, one to remove uncertain nodes, and one to remove value variables (when there are multiple value nodes).

When a decision node *i* has only one child, a value node *j*, and all of the other parents of the value node are also parents of the decision node, $Pa(i) \supseteq Pa(j) - \{i\}$, then optimal policy determination replaces the decision variable with a deterministic variable, as shown in Figure 24a. The new deterministic function for node *i* is given by

$$f_i(x_K \mid \&) = \arg\max_{x_i \in \Omega_i} f_j(x_i, x_K) \text{ for all } x_K \in \Omega_K,$$

where $K = Pa(i) \cap Pa(j) = Pa(j) - \{i\}$. The choice of x_i can be arbitrary when there are ties. This optimal policy should be recorded before it is itself removed. Note that any parents of *i* that are not parents of *j* are ignored in choosing the optimal policy and might become barren. (These are all identified by Bayes-Ball.) The other parents of *i*, given by *K*, are the requisite observations for *i*.

When an uncertain node *i* has only one child, value node *j*, then uncertain node removal removes the node *i* from the diagram, as shown in Figure 24b. The new function for node *j* is obtained by taking expectation,

$$f_j(X_J, X_K, X_L \mid \&) = \sum_{x_i \in \Omega_i} P\{x_i \mid X_J, X_K, \&\} f_j(x_i, X_K, X_L),$$

where $J = Pa(i) - Pa(j)$, $K = Pa(i) \cap Pa(j)$, and $L = Pa(j) - (\{i\} \cup Pa(i))$. Similarly, when there are multiple value nodes, *i* and *j*, value node removal removes the node *i* by replacing the function in node *j* with their sum, as shown in Figure 24c. The new function for node *j* is given by

$$f_j(X_J, X_K, X_L \mid \&) = f_i(X_J, X_K) + f_j(X_K, X_L).$$

This operation computes the partial sum familiar to dynamic programmers, but it should be delayed as long as possible, so that decisions and uncertain nodes can be removed into smaller value functions, ideally waiting until *J* or *L* is the empty set (Tatman & Shachter, 1990).

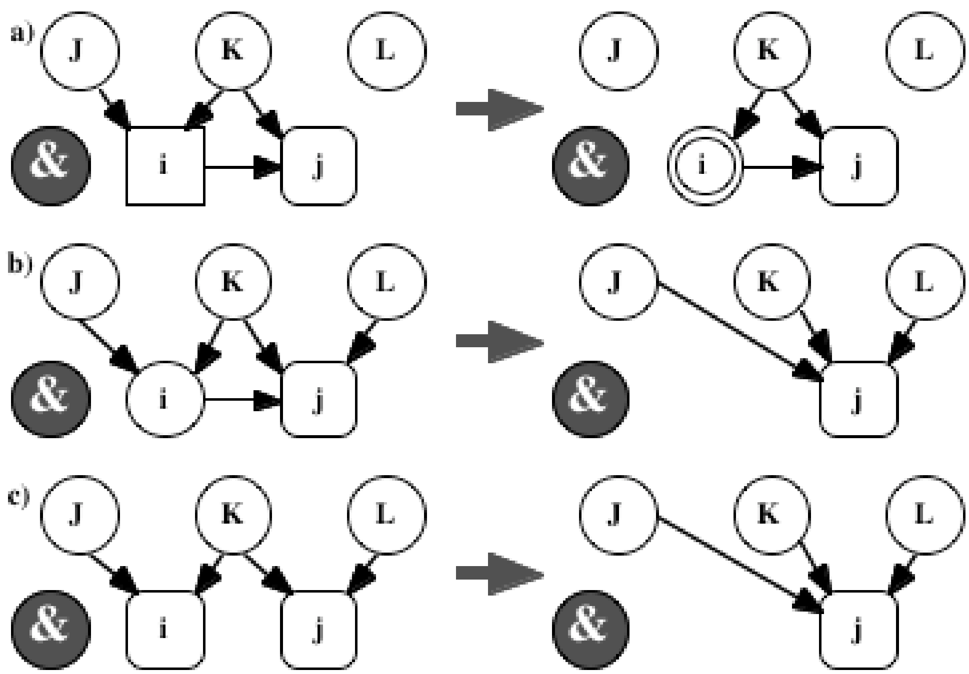

Figure 24. Influence diagrams operations to remove nodes

Given these operations, an influence diagram with no directed cycles and ordered decisions with no forgetting can be solved by the following algorithm (Shachter, 1986).

1. If there is a value node with children, turn it into an uncertain node and give it a value node child with value equal to its own.
2. While there are nodes remaining besides those in V or W_0:
   a. If there is a barren node, remove it;
   b. Otherwise, if conditions are satisfied to remove the latest decision then determine its optimal policy;
   c. Otherwise, if there is chance node that is not observed before the latest decision and has at most one value child then remove it, after reversing any arcs to its non-value children in order;
   d. Otherwise, a value node needs to be removed, preferably a descendant of the latest decision.

Many people use influence diagrams to construct their decision models but prefer to solve them using decision trees. The conversion to decision trees is immediate if there are no non-sequential arcs (Howard & Matheson, 1984). An arc is said to be non-sequential if it goes from an uncertain node in a later decision window to one in an earlier decision window. Whenever there are such arcs in an influence diagram at least one of them will be reversible (Shachter, 1990). Although it might not be the optimal algorithm, a simple and effective process is to iteratively find reversible non-sequential arcs and reverse them until there are no more non-sequential arcs, and the diagram can be converted to a tree.

Acknowledgements

I am grateful to the many colleagues and students with whom I have learned this subject, the editors and, most especially, Ward Edwards, who was a constant inspiration to all of us working to bring decision analysis "out of the boutique."

References

Capen, E. C., Clapp, R. V., & Campbell, W. M. (1971). Competitive Bidding in High-Risk Situations. *Journal of Petroleum Technology, 23*(6), 641-653.

Friedman, L. (1956). A Competitive Bidding Strategy. *Operations Research, 4*(1), 104-112.

Good, I. J. (1961a). A causal calculus--I. *Br. J. Phil Sci, 11*, 305-318.

Good, I. J. (1961b). A causal calculus--II. *Br. J. Phil Sci, 12*, 43-51.

Heckerman, D., & Shachter, R. (1995). Decision-Theoretic Foundations for Causal Reasoning. *Journal of Artificial Intelligence Research, 3*, 405-430.

Heckerman, D. E., & Shachter, R. D. (1994). A Decision-Based View of Causality. In R. Lopez de Mantaras & D. Poole (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Tenth Conference* (pp. 302-310). San Mateo, CA: Morgan Kaufmann.

Henrion, M., & Druzdzel, M. J. (1990). *Qualitative propagation and scenario-based approaches to explanation of probabiliistic reasoning*. Paper presented at the Sixth Conference on Uncertainty in Artificial Intelligence, Cambridge, MA.

Howard, R. A. (1960). *Dynamic Programming and Markov Processes*. Cambridge, MA: MIT Press.

Howard, R. A. (1990). From Influence to Relevance to Knowledge. In R. M. Oliver & J. Q. Smith (Eds.), *Influence Diagrams, Belief Nets, and Decision Analysis* (pp. 3-23). Chichester: Wiley.

Howard, R. A., & Matheson, J. E. (1984). Influence Diagrams. In R. A. Howard & J. E. Matheson (Eds.), *The Principles and Applications of Decision Analysis* (Vol. II). Menlo Park, CA: Strategic Decisions Group.

Jensen, F., Jensen, F. V., & Dittmer, S. L. (1994). From Influence Diagrams to Junction Trees. In R. Lopez de Mantaras & D. Poole (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Tenth Conference* (pp. 367-373). San Mateo, CA: Morgan Kaufmann.

Jensen, F. V., Lauritzen, S. L., & Olesen, K. G. (1990). Bayesian Updating in Causal Probabilistic Networks by Local Computations. *Comp. Stats. Q., 4*, 269-282.

Keeney, R. L. (1992). *Value-focused thinking*. Cambridge, MA: Harvard University Press.

Kim, J. H., & Pearl, J. (1983). *A computational model for causal and diagnostic reasoning in inference engines*. Paper presented at the 8th International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany.

Lauritzen, S. L., & Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *JRSS B, 50*(2), 157-224.

Matheson, J. E. (1990). Using Influence Diagrams to Value Information and Control. In R. M. Oliver & J. Q. Smith (Eds.), *Influence Diagrams, Belief Nets, and Decision Analysis* (pp. 25-48). Chichester: Wiley.

Miller, A. C., Merkofer, M. M., Howard, R. A., Matheson, J. E., & Rice, T. R. (1976). *Development of Automated Aids for Decision Analysis*: Stanford Research Institute, Menlo Park, CA.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufmann.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*: Cambridge University Press.

Raiffa, H. (1968). *Decision Analysis*. Reading, MA: Addison-Wesley.

Shachter, R. D. (1986). Evaluating Influence Diagrams. *Ops. Rsrch., 34*(November-December), 871-882.

Shachter, R. D. (1988). Probabilistic Inference and Influence Diagrams. *Ops. Rsrch., 36*(July-August), 589-605.

Shachter, R. D. (1990). An Ordered Examination of Influence Diagrams. *Networks, 20*, 535-563.

Shachter, R. D. (1998). Bayes-Ball: The Rational Pastime (for Determining Irrelevance and Requisite Information in Belief Networks and Influence Diagrams). In *Uncertainty in Artificial Intelligence: Proceedings of the Fourteenth Conference* (pp. 480-487). San Francisco, CA: Morgan Kaufmann.

Shachter, R. D. (1999). Efficient Value of Information Computation. In *Uncertainty in Artificial Intelligence: Proceedings of the Fifteenth Conference* (pp. 594-601). San Francisco, CA: Morgan Kaufmann.

Shachter, R. D., & Heckerman, D. E. (1987). Thinking Backwards for Knowledge Acquisition. *AI Magazine, 8*(Fall), 55-61.

Shachter, R. D., & Peot, M. A. (1992). Decision Making Using Probabilistic Inference Methods. In *Uncertainty in Artificial Intelligence: Proceedings of the Eighth Conference* (pp. 276-283). San Mateo, CA: Morgan Kaufmann.

Shenoy, P. P., & Shafer, G. (1990). Axioms for Probability and Belief-Function Propagation. In R. D. Shachter, T. S. Levitt, J. F. Lemmer & L. N. Kanal (Eds.), *Uncertainty in Artificial Intelligence 4* (pp. 169-198). Amsterdam: North-Holland.

Tatman, J. A., & Shachter, R. D. (1990). Dynamic Programming and Influence Diagrams. *IEEE SMC, 20*(2), 365-379.

Wright, S. (1921). Correlation and Causation. *J. of Agricultural Research, 20*, 557-585.

Wright, S. (1934). The method of path coefficients. *Ann. Math. Stat., 5*, 161-215.