Explaining Predictions in Bayesian Networks and Influence Diagrams

Urszula Chajewska

Stanford University
Computer Science Department
Stanford, CA 94305-9010
urszula@cs.stanford.edu

Denise L. Draper

Rockwell Palo Alto Laboratory
444 High Street, Suite 400
Palo Alto, CA 94301
draper@rpal.rockwell.com

Abstract

As Bayesian Networks and Influence Diagrams are being used more and more widely, the importance of an efficient explanation mechanism becomes more apparent. We focus on *predictive explanations*, the ones designed to explain predictions and recommendations of probabilistic systems. We analyze the issues involved in defining, computing and evaluating such explanations and present an algorithm to compute them.

Introduction

As knowledge-based reasoning systems begin addressing real-world problems, they are often designed to be used not by experts but by people unfamiliar with the domain. Such people are unlikely to accept system's prediction or advice without some explanation. In addition, the systems' ever increasing size makes their computations more and more difficult to follow even for their creators. This situation makes an explanation mechanism critical for making these systems useful and widely accepted.

Probabilistic systems, such as *Bayesian Networks* (Pearl 1988) and *Influence Diagrams* (Howard and Matheson 1984), need such a mechanism even more than others. Human judgment under uncertainty differs considerably from the idealized rationality of probability and decision theories. Tversky and Kahneman (1974) have shown that people asked for a probability assessment rely on a limited number of heuristics to simplify complex domains. They also exhibit biases which often lead to serious errors. It is not surprising that the results of the system's computations often leave them baffled.

The notion of explanation, once investigated only by philosophers (Hempel and Oppenheim 1948, Hempel 1965, Salmon 1984, Gärdenfors 1988), found its way into Artificial Intelligence in recent years (Spiegelhalter and Knill-Jones, 1984, Pearl 1988, Henrion and Druzdzel 1990, Shimony 1991, 1993, Suermondt 1992, Boutilier and Becher 1995, Chajewska and Halpern 1997). While philosophers have been working mostly on defining the notion of explanation, the emphasis in the AI contributions has been on the efficient computation of explanations.

In this paper, we will restrict our attention to probabilistic systems, specifically Bayesian Networks (BNs) and Influence Diagrams (IDs). We will assume the system to have full knowledge of the domain: both its causal structure and probability distribution over it. The user's knowledge will typically be incomplete (i.e., a subset of the system's knowledge) and/or his or her reasoning ability inadequate to the size of the domain. We will also assume that some observations have been made and these observations are available to both the system and the user.

In most of the literature on explanation, it is assumed that the item which requires an explanation (the *explanandum*) is an observed fact: something known, but unexpected. The agent's uncertainty may concern the current state of the world or the causal structure of the domain. It is often assumed that the explanation should consist of both causal and factual information. It is also assumed that what is known, should be excluded from the explanation—restating the obvious will not satisfy the user.

Most of the AI research dealing with explanation in probabilistic systems restricted the explanandum to a subset of observations and the explanation to factual information about the state of the world (Pearl 1988, Henrion and Druzdzel 1990, Shimony 1991 and 1993)[1]. We call the explanations provided under these restrictions *diagnostic explanations.* The intuition behind this name comes from medical applications—we observe "symptoms" and make a "diagnosis".

There is another situation, in which a user may request an explanation from a probabilistic system: A user may be surprised by a change in a prediction or a recommendation made by a system (i.e., a change in the probability distribution over a node of interest) caused by the input of new observations. The confusion in this case is caused either by the amount of new data or the lack of the user's understanding of the relationship between observations and the new prediction. The explanandum is the change in the probability distribution (which may be considered a generalization of "observation", which is used in diagnostic explanation to mean an instantiated node). It seems, however, that in this case we need to relax our as-

[1] There is no reason to exclude the information about the causal structure of the domain in diagnostic explanation, however, as pointed out in (Chajewska and Halpern, 1997) most of the work in this area does so.

sumption that nothing known to the user should be used in the explanation. By asking why the observed facts changed the prediction, the user informs us that although he knows of these facts, he doesn't understand all of their ramifications. We are faced with the fact that users are not perfect reasoners. The goal of the explanation mechanism is therefore to find the smallest possible subset of data (known, observed facts!) which would account for the change in the prediction and the paths in the causal structure most influential in transmitting its impact. We will call the result of this process a *predictive explanation.* It is this type of explanation that we will focus on in this paper.

The two types of explanation share a lot of properties: They both require causal and factual information and they both have to deal with the fact that people often don't realize all implications of their knowledge. It is possible to reason about them within the same framework. However, by considering them separately, we can take advantage of some additional information. The two types of explanation differ in the nature of uncertainty they try to alleviate, and consequently, in the type of computations they require. The need for diagnostic explanations arises from the uncertainty about the current state of the world. Predictive explanations are required when the uncertainty concerns the causal structure and dynamics of the domain.

Example 1 *Consider the well-known car-buyer's example (Howard 1976, Pearl 1988). The buyer of a used car has a choice between buying car one (C1), car two (C2) or no car at all. There are several tests available, with different costs and accuracy guarantees to be run on one or both cars. The buyer is assumed to know the chances that the cars are of good quality. The influence diagram in Figure 1 describes this problem.*

If we found out that our friend has bought car one and it turned out to be of poor quality, we might ask how could it have happened with all these tests on his disposal. Such a question could be answered by a diagnostic explanation. On the other hand, if somebody gave us an advice on the best course of action to take with respect to testing and buying cars, it would be reasonable to ask why the recommended sequence of actions was chosen. This question would be answered by a predictive explanation. Note that the question can be interpreted in two ways: why is the advice good in this particular situation or why is it better than any other (for a given prior on cars' quality). The explanation should be able to address both questions.

We treat predictive explanations as comparative. In fact, they very often are: (1) We can be given the description of the state of the world before and after a change, (2) We can consider two different plans of action, or (3) We can compare the prediction to the one which would be made in a default or average case. They can be used in all situations where it is desirable for the user to get some insight into the system's reasoning process. The possible applications include medical diagnosis and decision systems and comparison of plans represented as Influence Diagrams or Dynamic Belief Networks.

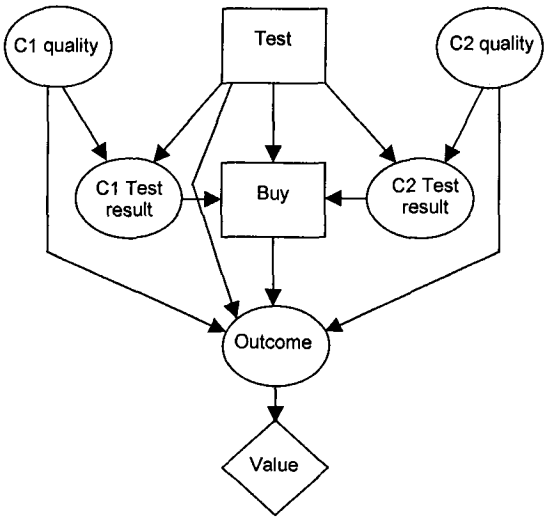


Figure 1 Car buyer's network

Predictive explanations for BNs have been studied by Suermondt (1991, 1992). His explanations consist of two parts: a subset of evidence variables with their instantiations that are considered most important to the prediction and a subset of the paths from these evidence variables to the node about which the prediction is made. The measure he uses to judge the explanation quality is the cross-entropy between the probability distribution resulting from conditioning on the proposed explanation and the posterior probability distribution resulting from conditioning on all of the evidence.

Our goal is to improve on Suermondt's work in several ways:

- Better computational complexity.
  The algorithm works by instantiating subsets of evidence and evaluating the network for each one of them. Each evaluation of the network is exponential in the number of nodes in the network (Cooper, 1990). To get the full answer, we would need an exponential number of instantiations and evaluations (exponential in the number of instantiated nodes). Suermondt suggests some heuristics which let him reduce this number to linear in some cases, with some loss of accuracy.
- Relax assumptions about conflicts between findings.
  Suermondt assumes a certain pattern in the interaction of the influence of sets of nodes. Specifically, he distinguishes two cases: (1) one in which the probability of the node of interest changes in the same direction given every node in the set separately and given the entire set (with the change given the entire set being the greatest) for all of its values, and (2) another when the change given the entire set is smaller than the change given one of the subsets, which is supposed to indicate a data conflict (the change in the probability of the node of interest given the rest of the set must be in the opposite direction). This type of data interaction is, in fact, typical for certain special types of nodes, like NOISY-OR. However, it does not hold in the general case. It is perfectly possible for two nodes to change the

probability of the node of interest in a certain direction while instantiated separately and in the opposite direction when instantiated together.

- More appropriate measure of explanation quality. Cross entropy is a function of two arguments. We can use it to compare the distribution resulting from conditioning on the proposed explanation to either prior or posterior distribution, but not both at the same time. Suermondt chooses to ignore the prior.

The rest of this paper is organized as follows: in Section 2 we discuss the issues involved in defining, evaluating and computing predictive explanations, in Section 3 we present an approximate algorithm to compute explanations and analyze it in Section 4. Section 5 discusses using our explanation mechanism in Influence Diagrams. We conclude in Section 6.

Predictive Explanations

When designing an explanation mechanism we need to ask ourselves three questions:

1. What counts as an explanation?
2. How are we going to measure the explanation quality?
3. Is there an efficient way to compute the best explanation?

We will consider each of these questions in turn.

Definition

Our definition of a predictive explanation is based on Suermondt's (1992).[2]

Let Δ be a set of nodes[3] changed by an outside intervention (either observations or actions). We will refer to the set of instantiations of the variables in Δ before the change as δ_A, and after the change as δ_B. Let O be the node of interest.

Following Suermondt, and for simplicity of presentation, we refer to the Δ nodes as though they were instantiated both before and after the change. Nothing in our analysis, however, rests on this assumption. Our algorithm can be also used in the case when there is only a change in the probability distribution caused by an intervention. In general, δ_A and δ_B should be considered to be arbitrary probability distributions over the nodes in Δ.

We would like the explanation of the node of interest O to account for most of the change in the probability distribution over O, that is, cause a similar change in that probability distribution when the rest of the Δ set and the rest of the network remain at their pre-change values. Such an explanation would consist of a subset of the nodes in Δ, called the *explanation set* Δ_X, and a subset of paths between the nodes in Δ_X and the node O. Intuitively, we can think of the explanation X as a subnetwork of the given network, consisting of the nodes in Δ_X, the node O and some of the paths between them (with internal nodes on these paths). In pruning the rest of the network, we would use the values from the pre-change instantiation δ_A to create the new conditional probability tables.

Definition 1 *A* predictive explanation X *of a change in the probability distribution over the node of interest* O *caused by the evidence* δ_B *received for the set of nodes* Δ, $O \notin \Delta$, *with respect to the prior distribution over the nodes in* Δ, δ_A, *is a conjunction* $\delta_X \wedge P_X$ *where:*

- δ_X *is an instantiation of nodes in* Δ *such that the subset of nodes* $\Delta_X \subset \Delta$ *is set to the values they assume in* δ_B *and the complementary subset of nodes* $\Delta - \Delta_X$ *is set to the values they assume in* δ_A.
- P_X *is a subset of the undirected paths* P *linking* O *to the nodes in* Δ_X.

We will refer to the set of nodes Δ_X *as the* explanation set.

Measuring Explanation Quality

Suermondt measures the explanation quality by computing cross-entropy of $P(O|\delta_B)$ and $P(O|\delta_X)$ where δ_X is defined as above. He also computes cross-entropy of $P(O|\delta_B)$ and $P(O|\delta_{\bar{X}})$ where $\delta_{\bar{X}}$ is the instantiation of nodes in Δ such that the subset of nodes $\Delta - \Delta_X$ are set to the values they assume in δ_B and the subset of nodes Δ_X are set to the values they assume in δ_A. The two are called *cost of commission* and *cost of omission*, respectively. The explanations with the cost of commission within a certain threshold Θ and the cost of omission outside the Θ are considered necessary and sufficient. They are also considered minimal if no explanation $X' = \delta'_X \wedge P'_X$ such that $\delta'_X \subseteq \delta_X$ and $\delta'_X \neq \delta_X$ is also necessary and sufficient.

Note that this measure ignores completely the prior probability of O. This may cause misleading results.

Example 2 *Consider a binary variable of interest O such that the probability of O being true becomes very high after we receive some new observations:* $P(O = true|\delta_B) = 0.9$. *An explanation X which causes the probability of O* $P(O = true|\delta_X) = 0.7$ *would be judged equally good regardless of whether the prior probability of O* $P(O = true|\delta_A)$ *was 0.1 or 0.8! In the first case the probability of* $O = true$ *given the explanation is much closer to the posterior probability than the prior. In the second, it is farther away. In other words, while in the first case we have a pretty good explanation, in the second, this explanation is worthless. Yet they are judged to be of equal quality.*

Cross-entropy is not a distance metric since it is asymmetric and doesn't obey the triangle inequality. It cannot be easily combined, i.e., given *H(P;Q)* and *H(Q;R)* we cannot say anything about *H(P;R)*. Therefore, even computing both the cross-entropy between the probability of the node of interest given the explanation and the prior and the cross entropy between the probability of the node of interest given the explanation and the posterior will not

[2] For a discussion on issues involved in defining diagnostic explanations, see Chajewska and Halpern (1997).

[3] These can be both chance and decision nodes.

provide the necessary information needed for an adequate quality measure.

Intuitively, every explanation X corresponds to a subnetwork created by pruning all the nodes not in Δ_X or on paths in P_X (with the new probability tables updated to reflect δ_A values of the deleted nodes). We would like to base the explanation quality measure on the 'closeness' of the probability distribution over the node of interest O in such subnetwork, $P'(O|\delta_X)$, to the two distributions generated in the original network, $P(O|\delta_A)$ and $P(O|\delta_B)$.

We propose a set of postulates that an explanation quality measure should satisfy:

- The measure should include a *cost function* based on the size of the explanation set. While we want the explanation to be as exhaustive as possible, we do not want it to be too complex to be easily understood. The cost function would play a role analogous to the minimum description length (MDL) principle.
- The other component of the quality measure should be based on a function f over three arguments: $P'(o|\delta_X)$, $P(o|\delta_A)$, and $P(o|\delta_B)$ (for each value of the node of interest O). (We are assuming that $P(o|\delta_A) \neq P(o|\delta_B)$.) This function should satisfy the following conditions:
  * The function should be monotonically increasing for $0 \le P'(o|\delta_X) \le P(o|\delta_B)$ and monotonically decreasing (for some applications, monotonically non-increasing may be a sufficient condition) for $P'(o|\delta_X) > P(o|\delta_B)$. Consequently, the explanation should have the highest value when it causes the prediction identical to the one made given the second instantiation δ_B:

$$\forall_{\delta_X} f(P'(o|\delta_X), P(o|\delta_A), P(o|\delta_B)) \le f(P(o|\delta_B), P(o|\delta_A), P(o|\delta_B))$$

  * The explanation, which does not change the prediction at all from what it was given the first instantiation δ_A is worthless—its value should be 0.

$$f(P(o|\delta_A), P(o|\delta_A), P(o|\delta_B)) = 0$$

We have found no sufficient reasons for any stronger conditions than these. Many possible functions could be used as quality measures. Two examples of functions that satisfy all our postulates are:

$$f_1 = \min_o \left(1 - \frac{\left| \log \frac{P(o|\delta_B)}{P'(o|\delta_X)} \right|}{\left| \log \frac{P(o|\delta_B)}{P(o|\delta_A)} \right|} \right)$$

Measures the ratio of relative differences.

$$f_2 = \sum_o \left(1 - \frac{P(o|\delta_B) - P'(o|\delta_X)}{P(o|\delta_B) - P(o|\delta_A)} \right)$$

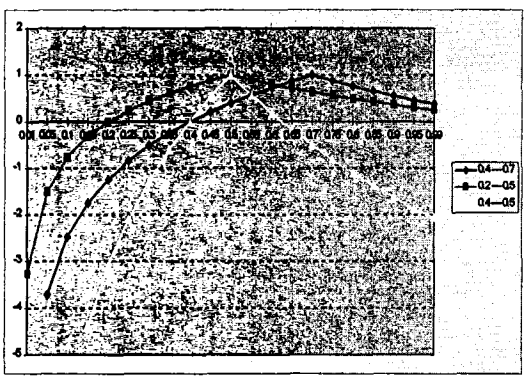
Measures the ratio of absolute differences.

Figure 2 illustrates the behavior of f_1 and f_2 for several probability distributions.

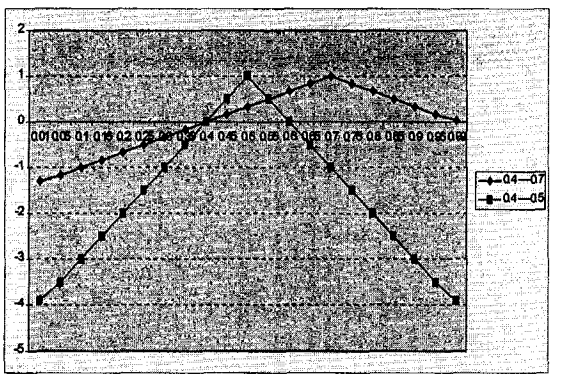
To compose the values of the function f for different values of the node of interest O, we could use any symmetric monotonic combination function such as sum or minimum.

Computational Complexity

As we mentioned above, Suermondt's algorithm requires an exponential number of network evaluations in the worst case. For complex domains, with many decision nodes or many observations, this is not feasible. But can we do much better? If we want to find the best explanation according to our explanation quality measure, the answer is probably no. We conjecture that even a simpler problem, of finding out whether there exists an explanation with the explanation set of size k and the quality value (measured,



(a)



(b)

Figure 2 Explanation quality as measured by f_1 (a) and f_2 (b) given different prior-posterior values. In both cases, the functions are computed for individual values of the node of interest O. Each line represents the explanation quality for a given combination of prior-posterior values. In (a) the prior-posterior value pairs are: 0.4-0.7 (darkest line), 0.2-0.5, and 0.4-0.5 (lightest line). In (b): 0.4-0.7 (darker line) and 0.4-0.5 (lighter line). Note that every line crosses the X-axis at the point in which the prior and the explanation-induced values are the same and has the maximum at the point where the posterior and the explanation-induced values are the same.

for example, by f_1 or f_2) above a certain threshold Θ requires examining all possible subsets of Δ.

What can we do in this situation to make computation of explanations more efficient? Suermondt makes certain assumptions about the type of parent interactions in the network and uses heuristics which reduce the complexity to linear (in Δ) in some special cases. As we mentioned above, such assumptions might be justified for some applications, but lead to disastrous results in others.

An alternative solution is to relax our requirements either for the size of the explanation set or the threshold value; in other words, look for a good explanation, but not necessarily the best one.

The Algorithm

We want the explanation to point out not only which of the changes to Δ caused the surprising result, but also the causal mechanism involved in transmitting the change to the node of interest. In some cases, there can be more than one explanation of the change. We want to find the one which involves the causal mechanism instrumental in the original situation. The idea behind our algorithm is the intuition that an explanation that would match the effect of the set of all observations on most of the nodes in the network, not only the node of interest, will ensure that the causal mechanism involved is indeed the same. Thus we will compute the explanation quality of the proposed explanation with respect to the internal nodes, as well as the node of interest.

Computing Explanation in Polytrees

We start by setting the desired accuracy threshold Θ for O, the node of interest. Then, we examine the neighbors of O. We select the smallest subset of them that has to be set to its posterior (given the instantiation δ_B) values for the probability distribution over the node of interest to reach the explanation quality above a desired threshold. What is more, the value of the node of interest must be within that threshold even if the values of the selected subset of neighbors are only an approximation of their posterior values. We can figure out how tight this approximation must be, given the number of neighbors in the subset and the threshold specified for the node of interest. This bound will constitute the accuracy threshold for the selected neighbors. Then, we repeat the process recursively for all of the nodes in the subset until we reach the members of the Δ set. At every step we adjust the explanation quality threshold for the selected set of neighbors (see the next section for analysis).

In order to be able to compute the explanation quality of different instantiations of parents efficiently, we should take advantage of messages passed during the initial evaluations given δ_A and δ_B.

We cannot guarantee that the explanation produced by this algorithm will have the smallest explanation set for the given threshold. However, it will have another important advantage. The probabilities of all the nodes in the extended explanation set (i.e., the nodes on the paths between the members of the explanation set and the node of interest) will be bounded by Θ. This will ensure that the quality of the explanation produced is not a coincidence, but it is caused by relying on the same mechanism which induced the change in the prediction in the first place.

Compute-explanation

Input: the network (a polytree), set Δ, instantiations δ_A and δ_B, the node of interest O, and the desired accuracy threshold Θ

Output: the explanation set Δ_X, the extended explanation set (EES) containing internal nodes important in transmitting the influence and the set of relevant paths P_X

Initialization:
- remove from the network all nodes not on the active paths between members of the Δ set and the node of interest.
- compute the posterior probabilities of every node in the network given δ_A and δ_B using the polytree algorithm (save all the messages passed).
- current = O
- $\Theta_{current} = \Theta$
- Δ_X = EES = EDGES = fringe = $\varnothing$

Loop:
- set the set N to include all not-visited neighbors of current (consider only the neighbors on paths from current to the members of Δ such that the node of interest is not on the path.)
- for i=1 to |N|
  - choose $S \subset N$ of size i with the highest EQ'(S, current, $\Theta_{current}$)
  - if EQ'(S, current) < $\Theta_{current}$, exit the for loop
- end for
- compute Θ_s for all $s \in S$ as a function of $\Theta_{current}$ and |S|
- fringe = fringe $\cup$ S
- for all $s \in S$, add the edge linking it with current to EDGES
- current = GetNode(fringe)
- if current = empty, exit

EQ'(set, node, Θ)
- set the probability of all $s \in$ set to $P(s|\delta_B)$
- set the probability of all other neighbors of current t to $P(t|\delta_A)$ (we will refer to this distribution over neighbors as δ_S)
- compute explanation quality, a function of $P'(node|\delta_S)$, $P(node|\delta_A)$, and $P(node|\delta_B)$, where $P'(node|\delta_S)$ is an approximation to $P(node|\delta_S)$ bounded by Θ

GetNode(set)
- find all nodes in the set which are in Δ and either are fully instantiated or are roots; remove them from set and add to Δ_X
- if set is empty, return empty
- pick a node randomly
- remove node from set
- if node $\in \Delta$, add node to Δ_X, otherwise add to EES
- return node

Figure 3 Algorithm to compute the explanation in polytrees.

Bayesian Networks with Loops

How can we extend the algorithm to work in general case, i.e., in networks with loops? If we tried to use the same technique, only replacing the polytree algorithm with the cutset conditioning, we would run into the following problem:

Example 3 *Consider the network in Figure 3.* O *is the node of interest,* X, Y *and* Z *belong to the* Δ *set. Assume that in the first iteration of our algorithm, we have determined that* U_1 *is the only important neighbor of* O *and we placed it in the explanation set. Then, in the next iteration, we found that both* X *and* Y *must be in the explanation set for the desired approximation to* U_1. *Does this guarantee the required explanation quality? Not necessarily. We do not know what the probability of* U_2 *given* Y *set to its value from* δ_B *and* Z *set to its value from* δ_A *is. It may be very different from both the probability of* U_2 *given the prior value (specified by* δ_A*) of both parents and the probability of* U_2 *given the posterior value (specified by* δ_B*) of both parents.*

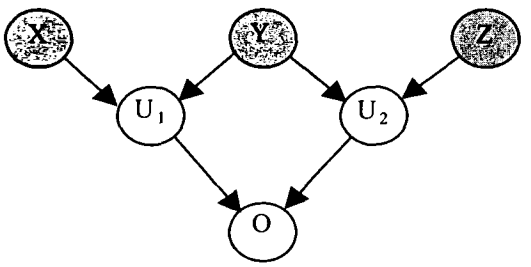


Figure 4 The network from example 3.

This problem arises whenever two neighbors of a node have a common ancestor (descendant) in the Δ set and one or both of them are put in the explanation set. The only solution to this problem, which would work in every case, would be to combine the two neighbors into one node (using arc reversal (Shachter 1986) to avoid creating directed cycles if necessary). The resulting network would become a polytree (except for the possible loops in the parts of the network considered of little importance; the loops not involving common ancestors in the Δ set would be removed in the initialization phase of the algorithm). The complexity of the algorithm would increase considerably, however, with the increase in the branching factor and the number of values per node. In order to avoid this, we should not combine the two neighbors unless it is necessary.

Let the S set be the set of selected neighbors of the current node, that is, the minimal set of neighbors which must be set to their posterior values to guarantee the required bound on the probability distribution over the current node. We do not have to combine two neighbors of the current node in the following cases:

- Neither neighbor belongs to the S set,
- Only one neighbor belongs to the S set, the two neighbors have at most one common ancestor (and no common descendants) in the Δ set and the neighbor not in S does not have any other ancestors or descendants in the Δ set,
- Both neighbors belong to the S set, they have at most one common ancestor (and no common descendants) in the Δ set, neither has another ancestor or descendant in the Δ set.

In all other cases the neighbors must be combined.

To ensure the correct threshold propagation, we need to check for the presence of loops right after selecting the subset of neighbors S and adding its members to the explanation set. In order to do that, we loop over all pairs of neighbors, merging them if necessary until no change is made.

Computational Complexity of the Algorithm

As in many Bayes Net algorithms, the complexity of our algorithm depends on the network topology. For polytrees, it is linear in the number of nodes, but exponential in the branching factor ($O(NV^B)$, where N is the number of nodes in the network, V is the number of values per node and B is the branching factor). When the networks with loops are transformed into polytrees, the number of nodes decreases, but the branching factor and the number of values per node may increase considerably. In the extreme case, we could combine together all of the interior nodes lying on the paths between the nodes in Δ and the node of interest O and the algorithm would be exponential in N and the size of the set Δ (specifically, $O(V^{N|\Delta|})$). We can hope, however, that some nodes in the network can be pruned and some loops ignored.

Analysis: accuracy threshold propagation

In this analysis, we will use the propagation algorithm proposed for the Bayes Net evaluation by Peot and Shachter (1991). In their version, instead of computing the probability of a node *given* the evidence, we compute the probability of a node *and* the evidence. This method has several nontrivial advantages, particularly for cutset conditioning. We will use it because it clarifies the role of normalization in Bayes Net propagation.

We discuss the error propagation in terms of the distance (absolute or relative) between the true and approximate probability of the node of interest conditioned on the explanation. We need this information to compute the required change in the accuracy threshold for every node visited by the algorithm. Note that the accuracy threshold can be easily converted into the bound on the chosen explanation quality measure.

One-step propagation

In order to prove a bound on the size of the error on the node of interest, we have to find the increase in the error value in a one-step propagation.

Theorem 1 *Relative error in the probability of a given value x of a node X when all of its incoming messages have*

a relative error of α each for every value (i.e., for every neighbor N the message received from N by X will be an approximation to the real message such that $\pi^*_X(n) = (1 \pm \alpha)\pi_X(n)$ *or* $\lambda^*_N(x) = (1 \pm \alpha)\lambda_N(x)$, $0 \le \alpha \le 1$*), will grow linearly in α with increase in |N|, the number of neighbors with approximate values. Specifically*

$$(1-\alpha)^{|N|} \le \frac{P^*(x,e)}{P(x,e)} \le (1+\alpha)^{|N|}.$$

The absolute error in the probability of a given value x of the node X when all incoming messages have an absolute error of at most α each (for every value), will grow linearly in α with the increase in |N|, the number of children and parents with approximate values. Specifically,

$$|P^*(x,e) - P(x,e)| = (1+\alpha)^{|N|} - 1.$$

Outgoing messages for X are bounded similarly.

Proof

Relative error:

$$P^*(x,e) = \left(\sum_u P(x|u)\prod_i \pi_X(u_i)(1\pm\alpha)\right)\prod_j \lambda_{Y_j}(x)(1\pm\alpha)$$
$$= (1\pm\alpha)^{|N|} P(x,e)$$

Absolute error:

$$\left|P^*(x,e) - P(x,e)\right| =$$
$$= \left|\left(\sum_u P(x|u)\prod_i (\pi_X(u_i)\pm\alpha)\right)\prod_j (\lambda_{Y_j}(x)\pm\alpha) - \left(\sum_u P(x|u)\prod_i \pi_X(u_i)\right)\prod_j \lambda_{Y_j}(x)\right|$$
$$\le \left(\sum_u P(x|u)\left(\sum_{k=1}^{|Parents|}\alpha^k\left(\sum_{\substack{\text{all subsets of parents of size} \\ |Parents|-k}}\prod_{i\,\text{in subset}}\pi_X(u_i)\right)\right)\right)\times \left(\sum_{k=1}^{|Children|}\alpha^k\left(\sum_{\substack{\text{all subsets of children of size} \\ |Children|-k}}\prod_{j\,\text{in subset}}\lambda_{Y_j}(x)\right)\right)$$
$$= \left(\sum_{k=1}^{|Parents|}\alpha^k\left(\sum_{\substack{\text{all subsets of parents of size} \\ |Parents|-k}}\sum_u P(x|u)\prod_{i\,\text{in subset}}\pi_X(u_i)\right)\right)\times \left(\sum_{k=1}^{|Children|}\alpha^k\left(\sum_{\substack{\text{all subsets of children of size} \\ |Children|-k}}\prod_{j\,\text{in subset}}\lambda_{Y_j}(x)\right)\right)$$
$$\le \left(\sum_{k=1}^{\#Parents}\alpha^k\binom{|Parents|}{|Parents|-k}\right)\times\left(\sum_{k=1}^{\#children}\alpha^k\binom{|Children|}{|Children|-k}\right)$$
$$= ((1+\alpha)^{|Parents|}-1)((1+\alpha)^{|Children|}-1)$$
$$\le (1+\alpha)^{|N|} - 1$$

Based on this result we can compute the accuracy bounds required for every node we add to the explanation set, for a given threshold Θ desired for the node of interest.

However, the bound on the probability of the node of interest *and* the evidence is not enough. We need to have a bound on the probability of the node of interest *given* the evidence. If Θ is the maximal error on P(O) (the node of interest) and P^* is the approximation to the true probability distribution, in case of the absolute error, we get

$$P^*(o|e) = \frac{P^*(o,e)}{P(e)} = \frac{P(o,e)\pm\Theta}{P(e)} = P(o|e) \pm \frac{\Theta}{P(e)}$$

The error term, $\Theta/P(e)$, can become unacceptable if *P(e)* is small and our threshold relatively large. But we know *P(e)* in advance! Therefore, we can set the Θ accordingly.

A possible problem we can run into with this approach would be the situation in which a very small *P(e)* forces us to set the threshold to a small value and the explanation set we get as a result of running our algorithm is close to the size of Δ.

Note that in the case of relative error, the division by *P(e)* does not present any problems.

Multi-step Propagation

Linear increase in the error in one-step propagation is not good news. Such error will grow exponentially in the length of the path and for large networks will become unreasonably large. Since, we start with the threshold over the node of interest and move away from it at every step, it means that the thresholds over the nodes further away will quickly become very small. How can we avoid it? The hope is in having small number of neighbors with approximate values at every step. For example, in case of one neighbor with an approximate value, the error will be at most the same and possibly smaller after one-step propagation.

If the change in the probability distribution over the node of interest is really influenced by most of the variables in the Δ set, there is nothing we can do (that's why the problem is requires the exponential number of network evaluations for the exact answer!). We have to settle for either a very large threshold value or a very large explanation set. In most cases, however, we hope to find only one or two neighbors for every node on the path that need to be included in the explanation set, which will let us keep the error within reasonable bounds.

Explanation in Influence Diagrams

It is our contention that we can use the same technique for computing predictive explanation in both BNs and IDs. It is usually assumed that the presence of the utility calculation in IDs requires the use of a different distance measure, namely the difference in the expected utility values, than the ones used for comparison of two probability distributions. However, while the difference in the expected utility value is the best measure to compare

the quality of two sets of decisions, it is not appropriate for explaining differences between them. Using the difference in EU could lead to misleading explanations, since the same expected utility can be associated with two very different states of the world. We will therefore use the distance between the probability distributions over the outcome node[4] in our explanation generating mechanism.

In the case of Influence Diagrams, the Δ set will consist of both observable nodes and decision nodes. The explanation will be defined exactly the same way as for Bayesian Networks and we will use the same explanation quality measures. We will assume that the decision nodes are either instantiated (possibly conditionally) or that we have a prior over all possible strategies (sequences of decisions).

Example 4 *An interesting special case of a network with loops is an Influence Diagram in which the decision nodes are sequentially linked by the information arcs. The network in Figure 1 is of this type. The decision about testing the car(s) is known to the car buyer before the decision about buying is made. Any plan of action would have the node 'Buy' instantiated conditionally. Note that since 'Buy' is not a root node, the algorithm will not stop there, but examine its parents as well. The node 'Test' may prove to be unimportant (e.g., if the tests are cheap and not very accurate), but it can also provide the crucial information, according to which the decision about buying is made[5]. The algorithm will be able to determine that. Also, it is possible that only the node 'Test' is important and 'Buy' is not (e.g., if we are choosing between the Plan A: "Do not test, do not buy" and Plan B: "Test both cars, buy the one which passes" in a situation where the tests are very expensive and the prior indicates that the cars are most likely in very poor condition).*

Conclusion

The algorithm presented here has several advantages compared to those found in the literature (Suermondt 1991, 1992). Although the worst-case running time is the same, our algorithm works in a general case without making any assumptions about the structure of the conditional probability tables. It finds both the explanation set and the influential paths through the network in a single pass. We hope that the empirical evaluation will show a significant gain in the average running time. In addition, the explanation quality measure employed in our algorithm avoids unintuitive results by taking into account both prior and posterior distribution over the Δ nodes.

We would like to extend this work in several directions:

- We would like to obtain analogous results for diagnostic explanations and extend our algorithm to compute them.
- We plan to investigate the problem of customizing explanations to the needs of a specific user. This will involve modeling the knowledge the user has about the state and the causal structure of the domain. The advantages of a customized explanation would range from eliminating from the explanation the things that the user already knows to correcting misconceptions and biases.

Acknowledgements

We thank Mark Peot and Joe Halpern for useful discussions. Part of this research was carried out while the first author was at the Rockwell International Palo Alto Research Lab. Rockwell's support is gratefully acknowledged. This work was supported in part by the ARPI grant F30602-95-C-0251 and by the NSF, under grant IRI-95-03109.

References

Boutilier, C. and V. Becher (1995, August). Abduction as belief revision. *Artificial intelligence 77,* 43-94.

Chajewska, U. and J. Y. Halpern (1997). Defining explanation in probabilistic systems. In *Proceedings Uncertainty in Artificial Intelligence 13* (*UAI '97*), pp. 62-71.

Cooper, G. F.(1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence 42, pp 393-405.*

Gärdenfors, P. (1988). *Knowledge in Flux: Modeling the Dynamics of Epistemic States.* Cambridge, Mass: MIT Press.

Hempel, C. G. (1965). *Aspects of Scientific Explanation.* Free Press.

Hempel, C. G. and P. Oppenheim (1948). Studies in the logic of explanation. *Philosophy of Science 15.*

Henrion, M. and M. J. Druzdzel (1991). Qualitative propagation and scenario-based schemes for explaining probabilistic reasoning. In *Uncertainty in Artificial Intelligence 6* (*UAI '90*), pp 17-32.

Howard, R. A. (1976). The used car buyer. In Howard, R. A., J. E. Matheson and K. L. Miller (Eds.), *Readings in Decision Analysis,* Menlo Park, CA: Stanford Research Institute.

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

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

Peot, M. A. and R. D. Shachter (1991). Fusion and propagation with multiple observations in belief networks (research note). *Artificial Intelligence,* 48, pp.299-318.

[4] We assume that the utility node has a single parent and call this parent the *outcome node.* Since every influence diagram can be transformed to achieve this form, we can do that without loss in generality. (See Figure 1 for an example.)

[5] The node 'Test' may also be found to influence the outcome directly, not through the 'Buy' node. In both cases it will be added to the explanation set. The difference between the two cases will be reflected in the edges added to the set of relevant paths when the 'Test' node is chosen to be included in the explanation set.

Salmon, W. C. (1984). *Scientific Explanation and the Causal Structure of the World.* Princeton, NJ: Princeton University Press.

Shachter, R. D. (1986). Evaluating influence diagrams. In Glen Shafer and Judea Pearl, editors, *Readings in Uncertain Reasoning,* pages 259-273, Los Altos, CA: Morgan Kaufmann.

Shimony, S. E. (1991). Explanation, irrelevance and statistical independence. In *Proc. National Conference on Artificial Intelligence (AAAI '91)*, pp. 482-487.

Shimony, S. E. (1993). Relevant explanations: Allowing disjunctive assignments. In *Proc. Ninth Conference on Uncertainty in Artificial Intelligence (UAI '93)*, pp. 200-207.

Spiegelhalter, D. J. and R. P. Knill-Jones (1984). Statistical and knowledge-based approaches to clinical decision-support systems, with an application in gastroenterology. In *Journal Royal Statistical Society A, vol. 147,* pp. 35-77.

Suermondt, H. J. (1991). Explanation of Probabilistic Inference in Bayesian Belief Networks, *Report KSL-91-39* (Knowledge Systems Laboratory, Medical Computer Science, Stanford University)

Suermondt, H. J. (1992). *Explanation in Bayesian Belief Networks.* Ph. D. thesis, Stanford University.

Tversky, A. and D. Kahneman (1974). Judgement under uncertainty: Heuristics and biases. In Glen Shafer and Judea Pearl, editors, *Readings in Uncertain Reasoning,* pages 32-39, Los Altos, CA: Morgan Kaufmann.