Interval Probability Propagation

Bjørnar Tessem
University of Bergen, Bergen, Norway

ABSTRACT

Belief networks are tried as a method for propagation of singleton interval probabilities. A convex polytope representation of the interval probabilities is shown to make the problem intractable even for small parameters. A solution to this is to use the interval bounds directly in computations of the propagation algorithm. The algorithm presented leads to approximative results but has the advantage of being polynomial in time. It is shown that the method gives fairly good results.

KEYWORDS: ***intelligent systems, uncertainty, interval probabilities, belief updating, network propagation***

1. INTRODUCTION

The most common method for representation of uncertainty in knowledge-based systems has been the use of probabilities. On arrival of new information, these probabilities are updated by the use of Bayesian statistics.

Kyburg [1] discusses the idea of interval probabilities and represents uncertainty by a convex set of probability distributions. He claims that the requirement that degrees of uncertainty be given as points is too strong, and proposes interval probabilities as an alternative. We shall use this idea of Kyburg and try to combine it with the ideas of Judea Pearl [2].

Pearl shows how one can construct networks that represent the dependency relationships between variables. To reflect knowledge in the networks, one assigns probabilities distributions to some of the variables and conditional probability distributions between variables that directly influence each other. This structure gives us a very efficient procedure for propagation of new knowledge. It is interesting to see how Kyburg's interval probabilities can be incorporated into Pearl's belief networks and

Address correspondence to Bjørnar Tessem, Department of Information Science, University of Bergen, N-5020 Bergen, Norway.

Received July 1991; accepted January 29, 1992.

655 Avenue of the Americas, New York, NY 10010 0888-613X/92/$5.00

what computational consequences this leads to. Similar problems are addressed by Fertig and Breese [3].

In the next two sections we shall present the concept of interval probabilities and discuss some of their features, look at Pearl's work, and give a summary of the equations he has developed. In Section 4 we try to use the ideas of convex sets of probability distributions in belief networks. This first try, however, is shown to be computationally inefficient, so we introduce an approximation algorithm in Section 5. In Section 6 we discuss errors resulting from the approximation.

2. INTERVAL PROBABILITIES

The use of intervals as a means of representing uncertainty has some history. Both possibility theory as given by Dubois and Prade in [4] with its necessity–possibility pairs and the theory of evidence (belief functions, Shafer [5]) with belief–plausibility pairs can be interpreted as interval representations of uncertainty. Driankov [6, 7] more explicitly uses intervals as a way of representing belief and plausibility of sentences. In [8] a kind of interval valued truth over an infinite-valued logic is investigated by Tessem. In [9] Eick discusses several inference schemas using interval representation.

However, most of the work with ideas of this type has been within probability theory. The best-known advocate of interval probabilities in recent years has been Kyburg [1, 10, 11]. He argues that the single-value probability approach used by standard Bayesians has a severe drawback: It cannot represent ignorance with respect to probabilities of events. As an alternative, Kyburg proposes the use of a convex set of probability distributions constrained by intervals, either obtained from subjective opinions or, preferably, extracted from statistical knowledge. The intervals would be confidence intervals as we know them from classical statistics. Other works on interval probabilities include those of Neapolitan and Kenevan [12], Snow [13], and White [14].

In general, interval probabilities are assigned to every subset of a set of possible outcomes. However, the number of constraints then becomes exponential in the size of the outcome set for a variable. We shall here, for the sake of efficiency, assume that the probabilities for any subset that is not a singleton are given implicitly by the constraints for the singletons. Thus, the constraints we shall consider are a subset of the full set of possible constraints. A more formal description of interval probabilities as we shall use them follows.

Let X be a variable with discrete values, the possible outcomes of X being the finite set Θ, where the members of Θ are $x_1, x_2, \dots, x_n$. For every event $X = x_i$, $i = 1, \dots, n$, we have upper and lower bounds on the probability $P(x_i)$ of this event. Let $P_*(x_i)$ and $P^*(x_i)$ denote the lower and upper bounds, respectively, for $P(x_i)$.

We can give a geometric interpretation of these probabilities and their bounds. The Cartesian product of the intervals $[P_*(x_i), P^*(x_i)]$, $i = 1, \ldots, n$, gives us a hyper-rectangle in the n-dimensional space R^n. The ith dimension of this space represents the value of $P(x_i)$. Within this box only points where coordinates add up to 1 are probability distributions. Thus we have a convex set of probability distributions in R^n restricted by inequalities

$$P_*(x_i) \le P(x_i) \le P^*(x_i), \qquad i = 1, \ldots, n \tag{1}$$

and

$$\sum_{i=1}^{n} P(x_i) = 1 \tag{2}$$

Not all numbers in the interval $[P_*(x_i), P^*(x_i)]$ need to be possible probabilities. Suppose $n = 2$, and $P(x_1) \in [0.0, 1.0]$ and $P(x_2) \in [0.9, 1.0]$ the given constraints. Then values of $P(x_1)$ larger than 0.1 are impossible because of (2) and the restriction that $P(x_2)$ be at least 0.9. We see that an effect of (2) is that the interval constraints impose bounds on each other. To handle this problem we introduce a concept of consistency for singleton interval probabilities.

Definition 2.1 *A set of interval constraints $P_*(x_i) \le P(x_i) \le P^*(x_i)$, $i = 1, \ldots, n$, for a probability distribution P is* consistent *if for all i, $i = 1, \ldots, n$, for every $y_i \in [P_*(x_i), P^*(x_i)]$ there exists a $y_j \in [P_*(x_j), P^*(x_j)]$, for all $j = 1, \ldots, n$; $j \ne i$, such that $\sum_{j=1}^{n} y_j = 1$.*

It is easy to see that we have consistency if and only if the following equations are satisfied:

$$P^*(x_i) + \sum_{\substack{j=1 \\ j \ne i}}^{n} P_*(x_j) \le 1 \quad \text{for all } i \tag{3}$$

$$P_*(x_i) + \sum_{\substack{j=1 \\ j \ne i}}^{n} P^*(x_j) \ge 1 \quad \text{for all } i \tag{4}$$

The first equation asserts that the upper bound is in fact a possible probability for $X = x_i$ because it shows that the lower bounds of the other probabilities need not be violated if we want to set $P(x_i)$ equal to $P^*(x_i)$. The second equation represents a corresponding argument concerning the lower bounds. Since we are in a convex set, all the values between the bounds must also be possible probabilities. Consistency follows.

There exists a simple procedure for finding consistent interval probabilities, given inconsistent probabilities. For all i, decrease all $P^*(x_i)$ such that (3) is satisfied; and for all i, increase all $P_*(x_i)$ such that (4) is satisfied. Of course, the initial intervals must be such that the box they produce has a nonempty intersection with the hyperplane given by (2), which is true when $\Sigma_{i=1}^{n} P^*(x_i) \geq 1$ and $\Sigma_{i=1}^{n} P_*(x_i) \leq 1$. A proof for the procedure is given by Tessem [15]. From now on we shall assume that interval probabilities are consistent.

The problem of updating is central in any uncertainty model. What do we do when new evidence arrives? Bayesians often use Jeffrey's rule [16]. Suppose we have a variable y with m different outcomes $y_1, y_2, \ldots, y_m$. If Y depends on another variable X, whose probability distributions vary, the probability $P(Y = y_j)$ should be

$$P(Y = y_j) = \sum_{i=1}^{n} P(Y = y_j \mid X = x_i) P(X = x_i) \tag{5}$$

The conditional probabilities $P(Y = y_j \mid X = x_i)$ remain unchanged. Jeffrey's rule has been disputed, but we shall use it adjusted to the idea of interval probabilities.

Jeffrey's rule applied to interval probabilities makes it impossible to represent the set of consistent probability distributions for Y with interval constraints like (1). This is because the resulting set is not convex. Now, nonconvex sets are very inconvenient to represent and use; therefore, good candidates for further use are the convex hulls of the resulting sets. These are convex polytopes, but they cannot be represented by singleton interval probabilities. This calls for an alternative representation of our convex sets. Observe that our convex sets are, like the sets resulting from conditioning, convex polytopes. A common choice is to represent polytopes by their vertices, and we shall use this representation in parts of this paper. The procedure for finding the vertices of the polytope given by (1) and (2) is simple to describe and is given below.

For every $i = 1, \ldots, n$, assign all 2^{n-1} possible combinations of either upper or lower bounds to $P(x_j)$, $j \neq i$. For each such combination compute $y = 1 - \Sigma_{j \neq i} P(x_j)$. If y is in the interval $[P_*(x_i), P^*(x_i)]$, then we have a vertex in the polytope given by setting the jth coordinate to the value (either upper or lower bound) used in the combination, $j \neq i$. The ith coordinate is set to the computed y.

This algorithm tries a total of $2^{n-1}n$ candidates, whereas the real maximum number $u(n)$ of vertices is given by (not proven)

$$u(n) = \begin{cases} \dbinom{n+1}{(n+1)/2}\left(\dfrac{n+1}{4}\right), & n \text{ odd} \\[2ex] \dbinom{n}{n/2}\left(\dfrac{n}{2}\right), & n \text{ even} \end{cases}$$

Observe that $u(n)$ for odd numbers are half of the $u(n)$ for the succeeding even number. When n approaches infinity, $u(n)$ for even numbers grows asymptotically as $(1/\sqrt{2\pi})2^n\sqrt{n}$, so our algorithm of $\mathcal{O}(2^n n)$ is of higher complexity than the number of vertices.

We shall leave interval probabilities at this point, turn our attention to belief networks, and give a review of Pearl's work.

3. BELIEF NETWORKS

In [2] Pearl describes how, in some special cases, one can efficiently propagate single-valued probabilities in a graphlike structure. We shall here give a review of Pearl's ideas.

The first we need to describe is the concept of a belief network. This is a directed acyclic graph where nodes stand for random variables and edges represent direct influence from one node to another. The idea is basically that values of variables that are not connected remain conditionally independent given the values of the variables between them. The only case in which they may not be independent is when the two nodes are the parents of a common node. In this case they are considered marginally independent; that is, they may become dependent given the value of their common child or one of this child's descendants. In [2], Pearl mainly discusses trees and singly connected graphs. There are efficiency problems when we cope with loops in the graphs, so we shall assume that the graphs we have are singly connected.

For every node X we have a value $\text{BEL}(x_i)$. This value is the probability of the variable being x_i given the evidence in the belief network called D.

$$\text{BEL}(x_i) = P(X = x_i \mid D) \tag{6}$$

Since X makes its descendants and ancestors conditionally independent, (6) can be written as

$$\text{BEL}(x_i) = \alpha P(D^- \mid X = x_i) P(X = x_i \mid D^+) = \alpha\lambda(x_i)\pi(x_i) \tag{7}$$

where D^+ is evidence from above in the graph and D^- is evidence from below. α is a normalizing constant to make the sum of the values of BEL sum up to 1.

For every node there is stored a conditional probability tensor that represents the conditional probabilities for the values of the node given the values of its parents. If we assume that X has two parents Y and Z, the entries of this tensor are

$$P\big(X = x_i \mid Y = y_j, Z = z_k\big) = M(X \mid Y, Z)_{jki} \tag{8}$$

From this we compute

$$\begin{aligned}\pi(x_i) &= P(X = x_i \mid D^+) \\ &= \sum_{j,k} M(X \mid Y, Z)_{jki} P\big(Y = y_j \mid D^{Y+}\big) P(Z = z_k \mid D^{Z+}) \qquad (9)\end{aligned}$$

D^{Y+} is the evidence stored in the graph beyond Y seen from X. We denote $P(Y = y_j \mid D^{Y+})$ by $\pi_X(y_j)$.

For evidence from below, we have (assuming two children A and B)

$$\begin{aligned}\lambda(x_i) &= P(D^- \mid X = x_i) \\ &= P(D^{A-} \mid X = x_i) P(D^{B-} \mid X = x_i) = \lambda_A(x_i) \lambda_B(x_i). \qquad (10)\end{aligned}$$

What we now have left to define are the parameters $\lambda_X(y_i)$ given to a node Y from its child X and $\pi_A(x_i)$ given to a node A from its parent X. The latter becomes

$$\begin{aligned}\pi_A(x_i) &= P(X = x_i \mid D^{X+}) \\ &= \alpha P(X = x_i \mid D^+) P(D^{B-} \mid X = x_i) = \alpha \pi(x_i) \lambda_B(x_i) \qquad (11)\end{aligned}$$

$\lambda_X(y_j)$ has the equation

$$\lambda_X(y_j) = \sum_k \left[\pi_X(z_k) \sum_i \lambda(x_i) M(X \mid Y, Z)_{jki} \right] \qquad (12)$$

In the equations above, we have assumed two children and two parents. This is, however, not a problematic restriction, as the equations are easily generalized to any number of children or parents.

There are several types of nodes in a network of this type:

Root—A node that has no parents. The values $\pi(x_i)$ of these nodes are a priori probabilities of the node.

Anticipatory—A node without children. The values $\lambda(x_i)$ of these nodes are all set to 1.

Ordinary—A node with both parent and child nodes.

Data—A node that is instantiated. That is, we know the value of the variable the node represents. In this case, $\lambda(x_i) = \pi(x_i) = 1$ if the node is instantiated to x_i; for all other $j \neq i$, $\lambda(x_j) = \pi(x_j) = 0$.

Dummy—A virtual node, a node C that represents a part of the complete belief network for the problem in mind that is impossible to represent in an ordinary way. It has a parent of one of the other types in the network and sends $\lambda_C(x_i)$ to its parent X.

Every node in the graph stores the parameters $M(X \mid Y, Z)$, $\lambda_A(x_i)$, $\lambda_B(x_i)$, $\pi_X(y_j)$, and $\pi_X(z_k)$. When one of these is changed for some reason, the node computes and sends a message to its neighbor nodes (except the one that triggered the computation) so they can update their parameters. Neighbors then proceed in the same way. The way the equations function ensures that evidence sent from a node is not sent back to that node. Before any data are put into the system, the a priori evidence in the root nodes is propagated through the system. After that, propagation starts when a node becomes a data node or when a dummy node is instantiated. For a more detailed description of the ideas we refer to Pearl [2] or to Pearl's book [17].

4. INTERVAL PROBABILITIES IN BELIEF NETWORKS

We are now ready to mix interval probabilities and belief networks. We shall use interval probability distributions instead of the standard single-valued probabilities in the a priori probabilities and the conditional probability tensors of belief networks. We have, however, observed earlier that resulting BEL, λ, and π distributions may not be representable as interval probabilities, and that they will be represented by the vertices of a convex polytope in multidimensional space. We shall discuss the computational consequences of this more complex representation.

Pearl shows how one can build standard belief networks on the basis of joint distributions on a set of variables. However, in many cases one would get multiply connected networks from this starting point, so Pearl proposes that the qualitative relations between variables may be based on subjective opinions. In the case of interval probabilities it is even worse to base the construction of the graph on joint distributions. If we got a joint distribution with real interval probabilities it would be impossible to get a network of the type we would like to have. The reason for this is that, when there are (even very small) changes in the joint probability distribution, the dependencies among the variables change. This again leads to completely different dependency graphs. So the main strategy for building the networks should be to depend on subjective judgments for the qualitative relations and on statistical data and/or subjective estimates for specifying conditional dependency tensors and a priori probabilities.

We shall illustrate the ideas of propagation with an example similar to one presented by Pearl [2]. Suppose there is a murder and there are three possible candidates for murderer. One and only one of them has committed the crime. We can assign probabilities to each candidate to represent our a priori belief that this particular person is the murderer. Let X represent the murderer variable, and let $\Theta_X = \{x_1, x_2, x_3\}$ be the set of possible outcomes. Suppose our a priori knowledge tells us that x_1 is more likely to be the murderer than the two others, else there is no knowledge as to what these probabilities might be. We can then choose the following a priori interval probabilities for X:

$$P(X = x_1) \in [0.7, 0.9]$$
$$P(X = x_2) \in [0.0, 0.3]$$
$$P(X = x_3) \in [0.0, 0.3]$$

Observe that the probabilities are consistent with respect to Definition 2.1. The interpretation of these intervals is that any probability distribution over X that satisfies these constraints may be the one and only correct distribution for X. We will not commit ourselves to any particular one of these, as our knowledge is not specific enough.

We can also specify the convex set of distributions in another way. In our example the set comprises the points of a polygon (two-dimensional) in R^3. A polygon can be represented by its vertices, and in our example there are four vertices, (0.7, 0.0, 0.3), (0.7, 0.3, 0.0), (0.9, 0.1, 0.0), and (0.9, 0.0, 0.1). In the general case the convex set given by interval probabilities for an n-valued variable consists of the points inside an $(n - 1)$-dimensional polytope in R^n. The number of vertices needed to represent this polytope is $\mathscr{O}(2^n \sqrt{n})$, so for large n the computation of the vertices becomes impractical.

Let X be the root in a dependency graph, and let Y be the only child of X. Let Y represent the last person of the three to hold the pistol with which the murder was committed. Then we need a matrix $M(Y \mid X)$ to represent the conditional probabilities for Y given X. This introduces the problem of representing interval conditional probabilities.

First, we give a general discussion on the use of interval probabilities in the conditional probability tensors. What will happen when a convex set of probabilities is multiplied by a convex set of tensors? Will the resulting set be convex or not? In fact, it will not.

Suppose we have a convex set of vectors $\mathscr{V}$ and a convex set of matrices $\mathscr{M}$. Now we want to describe the set of vectors $\mathscr{V}_1$ resulting from multiplying matrices from $\mathscr{M}$ and vectors from $\mathscr{V}$. Let $\mathbf{q}, \mathbf{r} \in \mathscr{V}$ and $C, B \in \mathscr{M}$. Then clearly $B_{\mathbf{q}}, C_{\mathbf{r}} \in \mathscr{V}_1$. Now, consider an arbitrary point $\eta B\mathbf{q} + (1 - \eta)C\mathbf{r}, 0 \leq \eta \leq 1$. If $\mathscr{V}_1$ is to be convex, this point must be in $\mathscr{V}_1$ and must be the result of

$$[\mu B + (1 - \mu)C][\lambda \mathbf{q} + (1 - \lambda)\mathbf{r}], \qquad 0 \leq \mu; \lambda \leq 1$$

But in general there exist no values for λ and μ that satisfy these constraints for an arbitrary η. This because the points $B\mathbf{q}$, $B\mathbf{r}, C\mathbf{q}, C\mathbf{r}$ are not necessarily linear dependent. However, a weaker result says that the convex hull of $\mathscr{V}_1$ is a polytope and that the extreme points of this polytope are the extreme points of $\mathscr{V}_1$.

The argument generalizes to tensors of any rank with the operations we use. As an approximation to the result set, we shall use the convex hull of the set. So the set of points we are going to proceed with is the set of points of an n-dimensional convex polytope in R^n. It is interesting that if we choose to use only single-valued probability tensors the convexity is maintained. We shall, however, proceed with interval probability tensors, as this is the most general case.

The convex set of matrices can also be represented by the vertices of its convex polytope. They can be found by finding the vertices for the interval probability distributions $P(X \mid Y = y_j,\ Z = z_k)$ and then combining them with each other in every possible way, picking one vertex from each distribution $P(X \mid Y = y_j,\ Z = z_k)$. We get from our restrictions an $n_x m$-dimensional convex polytope, where $m = n_y n_z$ and n_x is the size of the outcome set of X.

The number of vertices for these polytopes grows very fast as the number of parameters increases. The number of possible value combinations for variables conditioned on is m, as above, and the number of vertices for each combination is of order $2^n\sqrt{n}$. This gives us a maximum of $\mathscr{O}((2^n\sqrt{n})^m)$ vertices for the polytope, so it soon becomes impossible to compute the vertices. However, for small n and one parent, as in our example, we can determine this polytope.

The method for finding the vertices of the convex hull of $\mathscr{V}_1$ is to combine all vertices of the polytopes of $\mathscr{M}$ and $\mathscr{V}$ to get a candidate set for new vertices. In our problem we use the vertices of the conditional probability tensor polytopes and the vertices in the λ- or π-polytopes to compute the convex sets resulting from (7), (9), (10), (11), and (12). When we have computed the set of candidate vertices by combining all vertices from one polytope with the vertices from the other, we can use a convex hull algorithm (Edelsbrunner [18]) to find the real vertices of the polytope.

It is now time to return to our example. The convex set of matrices $M(Y \mid X)$ can be represented by the interval matrix

$$M(Y \mid X) = \begin{pmatrix} [0.8, 1.0] & [0.0, 0.2] & [0.0, 0.2] \\ [0.0, 0.2] & [0.8, 1.0] & [0.0, 0.2] \\ [0.0, 0.2] & [0.0, 0.2] & [0.8, 1.0] \end{pmatrix}$$

What are the possible π-vectors sent from X to Y? There are 27 vertices in the polytope that represents the set of matrices. And when the number of vertices in the polygon representing the $\pi_Y(X)$-vector is four, we get a total of 108 candidate vertices. After we have used the convex hull algorithm, the number of resulting vertices in our example is reduced to four—(0.56, 0.00, 0.44), (0.56, 0.44, 0.00), (0.92, 0.00, 0.08), and (0.92, 0.08, 0.00). In this example the interval restriction structure is also kept because the conditioning matrix has a nice symmetric structure.

When programming λ's it is possible to reduce the number of vertices one has to combine with. In this case it is fully possible to normalize every vertex in the polytope representing the λ's to get a polytope in the plane $\Sigma_i \lambda(x_i) = 1$. This is so because it is only the ratios between the $\lambda(x_i)$'s that are interesting, as we will always normalize when we compute beliefs. To see that convexity is maintained under normalization, observe that the normalizing process is like projecting the convex polytope onto the plane $\Sigma_i \lambda(x_i) = 1$ with the origin as the center of projection. It is as if we were standing at the origin looking toward a convex polytope. What we see is the projection onto a plane normal to the direction of view. Clearly the polytope we see is convex when the object observed is a convex polytope. Also when we compute BEL and $\pi_Y(X)$ vectors the same argument is valid, so the computational effort lies in computing vertices of convex polytopes from singleton intervals, multiplying them to get the candidate vertices for the resulting convex hull, and then using convex hull algorithms to find the real vertices.

Now, let us introduce dummy nodes into our network. Suppose the pistol is sent to some specialists on fingerprints. Their knowledge is modeled by a dummy node. This dummy node sends information to its parent node giving ratios for $P(C \mid Y = x_i)$, $i = 1, 2, 3$. C is here the evidence given by the fingerprints. We also denote these ratios by $\lambda_C(Y)$. There is very little left of the fingerprints on the gun, so the specialists are very uncertain about whose they are. However, they come up with a set of interval probabilities,

$$\lambda_C(Y) = \begin{pmatrix} [0.3, 0.8] \\ [0.5, 0.6] \\ [0.5, 0.9] \end{pmatrix}$$

These probabilities do not need to sum up to 1, so the convex set of λ's is represented by the prism given by the upper and lower limits. If we normalize, we get a convex hexagon represented by the following vertices:

$$\left(\frac{4}{11}, \frac{5}{22}, \frac{5}{22}\right), \left(\frac{4}{9}, \frac{5}{18}, \frac{5}{18}\right), \left(\frac{8}{19}, \frac{6}{19}, \frac{5}{19}\right), \left(\frac{3}{14}, \frac{3}{7}, \frac{5}{14}\right),$$
$$\left(\frac{1}{6}, \frac{1}{3}, \frac{1}{2}\right), \left(\frac{3}{17}, \frac{5}{17}, \frac{9}{17}\right)$$

As this is the only child of Y in the network, this hexagon also represents $\lambda(Y)$. To find the BEL(Y), we combine all vertices of $\lambda(Y)$ with all vertices of $\pi(Y)$ and normalize, to get a total number of candidate vertices of 24.

The resulting polygon is a quadrangle with vertices

$$\left(\frac{14}{47},0,\frac{33}{47}\right),\left(\frac{92}{97},0,\frac{5}{97}\right),\left(\frac{92}{97},\frac{5}{97},0\right),\left(\frac{7}{18},\frac{11}{18},0\right)$$

So Bel(Y) is one of the points in this polygon. The polygon is shown in baricentric coordinates in Figure 1. The corner marked with (1,0,0) represents Bel(Y) = (1,0,0), and correspondingly for the other corners.

We can go on and propagate the information from the fingerprints up to the murder variable X by multiplying the set of λ's by the set of $M(Y \mid X)$'s. From this we get a set of 162 possible candidates for the vertices for the polygon of $\lambda(X)$. The convex polytope is given by 10 vertices:

$$\left(\frac{41}{111},\frac{25}{111},\frac{15}{37}\right),\left(\frac{41}{107},\frac{25}{107},\frac{41}{107}\right),\left(\frac{4}{9},\frac{5}{18},\frac{5}{18}\right),\left(\frac{40}{97},\frac{32}{97},\frac{25}{97}\right),$$
$$\left(\frac{18}{71},\frac{30}{71},\frac{23}{71}\right),\left(\frac{15}{68},\frac{15}{34},\frac{23}{68}\right),\left(\frac{5}{29},\frac{11}{29},\frac{13}{29}\right),\left(\frac{5}{31},\frac{11}{31},\frac{15}{31}\right),$$
$$\left(\frac{15}{83},\frac{23}{83},\frac{45}{83}\right),\left(\frac{21}{89},\frac{23}{89},\frac{45}{89}\right)$$

Combining this with $\pi(X)$, which is the set of a priori probabilities of X, we get a quadrange that contains all possible probability distributions for X, given all the evidence. The quadrangle is given by the four vertices

$$\left(\frac{72}{77},0,\frac{5}{77}\right),\left(\frac{369}{394},\frac{25}{394},0\right),\left(\frac{35}{68},\frac{33}{68},0\right),\left(\frac{7}{16},0,\frac{9}{16}\right)$$

Figure 2 shows the convex set in baricentric coordinates.

The combinatorial explosion in the number of vertices in the tensor polytope severely restricts the maximal number of parents in a dependency

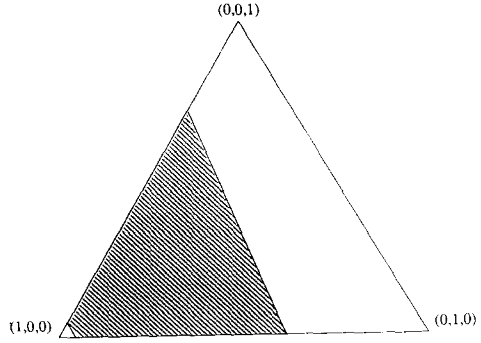

Figure 1.

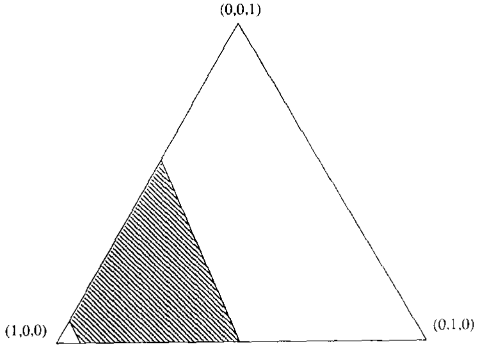

Figure 2.

graph and also the number of possible values for a variable. In Table 1 we show the maximal number of vertices for the tensor polytope when variables have at most n possible values and at most k parents. We see that the number can be accepted for $n = 2$ and $k \leq 3$, and, depending on the computer, we can let $k = 1$ and n be up to 3. Any other parameters seem impractical if we want a fast response.

5. AN APPROXIMATIVE METHOD

As we have seen, the number of vertices in our polytopes grows very fast as the number of critical parameters increases. Hence, it is neither tempting nor practical to use the algorithm of the previous section to do propagation. There seems to be a need for an approximation algorithm.

The idea we present here is to maintain the singleton interval representation of the convex set of distributions. We find the minimum and maximum bounds for all singleton probabilities and use these as resulting intervals. Together with the plane $\Sigma_i P(X = x_i) = 1$, these intervals give us a representation compatible with the one we started with.

Table 1

	k				
n	1	2	3	4	5
2	4	16	256	65536	4.29×10^9
3	216	1.01×10^7	1.02×10^{21}		
4	20736	1.85×10^{17}			
5	2.43×10^7				

Let us start with the two-valued case. Suppose X is a root variable that takes one of the values x_0 and x_1. Then the set represented by the interval restrictions is a straight line between two points in two-dimensional space. In particular, this is also true for $\pi(X)$, and also for $\lambda(X)$ after normalization. If we suppose that X has a two-valued child Y, Y's only parent being X, then the transformation done by (9) of a normalized $\pi_X(Y)$ is the same as rotating and stretching the line given by the interval restrictions on $\pi_X(Y)$. This new set of possible $\pi(Y)$'s is represented by a line, and interval restrictions can trivially be found from the bounding points.

The formulation of the problems can be done in a linear programming manner. For example, find the minimum $\pi_*(y_0)$ of

$$\pi(y_0) = \pi_Y(x_0)P(y_0 \mid x_0) + \pi_Y(x_1)P(y_0 \mid x_1)$$

given the inequalities

$$\pi_{Y*}(x_0) \le \pi_Y(x_0) \le \pi_Y{}^*(x_0)$$

$$P_*(y_0 \mid x_0) \le P(y_0 \mid x_0) \le P^*(y_0 \mid x_0)$$

$$P_*(y_0 \mid x_1) \le P(y_0 \mid x_1) \le P^*(y_0 \mid x_1)$$

and the equality

$$\pi_Y(x_0) + \pi_Y(x_1) = 1$$

The solution in this case is found when

$$P(y_0 \mid x_0) = P_*(y_0 \mid x_0)$$

$$P(y_0 \mid x_1) = P_*(y_0 \mid x_1)$$

$$\pi_Y(x_0) = \begin{cases} \pi_{Y*}(x_0) & \text{if } P_*(y_0 \mid x_0) \ge P_*(y_0 \mid x_1) \\ \pi_Y{}^*(x_0) & \text{else} \end{cases}$$

To see why, observe that what we do in this case is multiply the largest $P(y_0 \mid x_i)$ by the smallest factor to minimize its influence in the sum. It is a kind of *annihilation* of the probabilities, and we shall use this name for the idea later.

The principle is that we find the interval bounds by distributing the probability masses within the conditioning matrix and $\pi_X(Y)$ such that we get the minimum or maximum of the entries in $\pi(Y)$. By a probability mass we mean a part of the total probability. The total probability mass sums up to 1 and is distributed to singleton sets during the approximation algorithm.

From now on we assume that the possible π-vectors are normalized and consistent according to Definition 2.1. We proceed with the computation of $\lambda_Y(X)$. To find $\lambda_{Y*}(x_0)$ we have the following problem. Find the minimum of

$$\lambda_Y(x_0) = P(y_0 \mid x_0)\lambda(y_0) + P(y_1 \mid x_0)\lambda(y_1)$$

given the constraints

$$\begin{aligned} \lambda_*(y_0) \leq {} & \lambda(y_0) & \leq \lambda^*(y_0) \\ \lambda_*(y_1) \leq {} & \lambda(y_1) & \leq \lambda^*(y_1) \\ P_*(y_0 \mid x_0) \leq {} & P(y_0 \mid x_0) & \leq P^*(y_0 \mid x_0) \end{aligned}$$

and

$$P(y_0 | x_0) + P(y_1 | x_0) = 1$$

The solution is given by

$$\lambda_*(y_0)\mu + \lambda_*(y_1)(1 - \mu)$$

where

$$\mu = \begin{cases} P_*(y_0 \mid x_0) & \text{if } \lambda_*(y_0) \geq \lambda_*(y_1) \\ P^*(y_0 \mid x_0) & \text{else} \end{cases}$$

Equivalent ideas can be used to find all lower and upper bounds for both π's and λ's. When we want to find upper bounds, we do the opposite of annihilation. The components with the largest upper bounds are multiplied by numbers as large as possible. In the following we call this process *reinforcement*. When we have the bounds it is time to combine π and λ to get the intervals for the probability distributions of our variable or, equivalently, find the lower and upper bounds for BEL(X). Suppose now we have an interval representation of both $\pi(X)$ and $\lambda(X)$. Then $\text{BEL}_*(x_0)$ can be found by solving the following problem. Find the minimum of

$$\text{BEL}(x_0) = \frac{\lambda(x_0)\pi(x_0)}{\lambda(x_0)\pi(x_0) + \lambda(x_1)\pi(x_1)}$$

under the constraints

$$\begin{aligned} \lambda_*(x_0) &\leq \lambda(x_0) \leq \lambda^*(x_0) \\ \lambda_*(x_1) &\leq \lambda(x_1) \leq \lambda^*(x_1) \\ \pi_*(x_0) &\leq \pi(x_0) \leq \pi^*(x_0) \\ \pi_*(x_1) &\leq \pi(x_1) \leq \pi^*(x_1) \end{aligned}$$

and

$$\pi(x_0) + \pi(x_1) = 1$$

The solution is

$$\mathrm{BEL}_*(x_0) = \frac{\pi_*(x_0)\lambda_*(x_0)}{\pi_*(x_0)\lambda_*(x_0) + \pi^*(x_1)\lambda^*(x_1)}$$

We also use the annihilation/reinforcement strategy to find the solution. To find upper bounds we have to make the denominator of the fraction as small as possible, so we use annihilation on the denominator. For lower bounds we use reinforcement. This way, we find all the bounds of BEL.

Throughout these computations we have assumed that we know the bounds on both $\pi_Y(X)$ and $\lambda(X)$. To compute them is as simple as what we did above. The intervals for $\lambda(X)$ we get from simple interval multiplication (Moore [19]) of the intervals from the different $\lambda_Y(X)$'s. The resulting interval for $\lambda(x_0)$ is $[\prod_{i=1}^{k}\lambda_{Y_i*}(x_0), \prod_{i=1}^{k}\lambda_{Y_i}{}^*(x_0)]$. To get the $\pi_Y(X)$ we first do an interval multiplication of the $\lambda_{Y_i}(X)$ intervals from the children Y_i of X not equal to Y. We denote this product by $\lambda_{-Y}(X)$. A normalized $\pi_{Y*}(x_0)$ is then

$$\pi_{Y*}(x_0) = \frac{\lambda_{-Y*}(x_0)\pi_*(x_0)}{\lambda_{-Y*}(x_0)\pi_*(x_0) + \lambda_{-Y}{}^*(x_1)\pi^*(x_1)}$$

The other bounds are found similarly.

In the case where a node Y has more than one parent, we need more general algorithms for computing $\pi(Y)$ and $\lambda_Y(X)$. The annihilation/reinforcement idea is central here also.

The idea is to compute a joint distribution $P_X^+(D_{X_1}^+, \ldots, D_{X_k}^+)$ over the X_i's that are parents of Y. We get this from the different $\pi_{X_i}(Y)$'s. Since they are marginally independent, this distribution is given by multiplying the $\pi_{X_i}(Y)$'s together, which is simply done by interval multiplication.

The problem of finding $\pi_*(y_0)$ can be described as follows. Find the minimum of $\pi(y_0)$ given the constraints given by the intervals and the fact that the $\pi_Y(X_i)$'s have to be normal. The method to use is to first compute the joint distribution from all parents of Y. Then sort in increasing order the entries $M(Y \mid X_1, X_2, \ldots, X_k)_{i_1, i_2, \ldots, i_k, 0}$, where the indices vary over 0 and 1. Then distribute the probability mass to the joint distribution such that the probability of the event that has the smallest $M(Y \mid X_1, X_2, \ldots, X_k)_{i_1, i_2, \ldots, i_k, 0}$ gets as much mass as possible. A more precise algorithm is given in Figure 3. (In the computation of upper bounds, the sorting in the algorithm should be in decreasing order on the upper bounds of the conditional probabilities. And in the last step one should use the upper rather than lower bounds for the conditional probabilities in the equation.)

1. Create a joint probability distribution $\pi_{X_1,\ldots,X_k}(x_{1,i_1},\ldots,x_{k,i_k})$ from the individual $\pi_{X_j}(x_{j,i_j})$'s by using interval multiplication.
2. Pick all entries $M_*(Y|X_1,\ldots,X_k)_{i_1,\ldots,i_k,0}$ for all index sequences $i_1,\ldots,i_k$ and sort in increasing order. Give the order by the index sequences placed in an array I(1:2$^{\mathtt{k}}$).
3. Let $P(X_1,\ldots,X_k)$ be a single value joint distribution over $X_1,\ldots,X_k$. Distribute masses such that for all entries
$$P(X_1 = x_{1,i_1},\ldots,X_k = x_{k,i_k}) = \pi_{X_1,\ldots,X_k}(x_{1,i_1},\ldots,x_{k,i_k}).$$
Let S be the sum over $P(X_1,\ldots,X_k)$. S is the mass spent so far. Let **count**= 1.
4. Loop while $S < 1$. If $S \geq 1$ go to step 7.
5. Get indices $i'_1,\ldots,i'_k$ from I(count) and set
$$\begin{aligned} P(X_1 = x_{1,i'_1},\ldots,X_k = x_{k,i'_k}) \\ = \min(\pi_{X_1,\ldots,X_k*}(x_{1,i'_1},\ldots,x_{k,i'_k}) + 1 - S, \\ \pi_{X_1,\ldots,X_k}{}^*(x_{1,i'_1},\ldots,x_{k,i'_k})). \end{aligned}$$
6. Add $\pi_{X_1,\ldots,X_k}{}^*(x_{1,i'_1},\ldots,x_{k,i'_k}) - \pi_{X_1,\ldots,X_k*}(x_{1,i'_1},\ldots,x_{k,i'_k})$ to S. Increase count by 1 and go to step 4.
7. Use the $P(X_1,\ldots,X_k)$-distribution to compute and return
$$\pi_*(y_0) = \sum_{i_1,\ldots,i_k} M_*(Y|X_1,\ldots,X_k)_{i_1,\ldots,i_k,0} P(X_1 = x_{1,i_1},\ldots,X_k = x_{k,i_k}).$$

Figure 3. Algorithm for computing $\pi_*(y_0)$.

The algorithm for computing $\lambda_{Y*}(x_l, 0)$ is somewhat more complicated and uses methods similar to those mentioned above. An algorithm is given in Figure 4.

It should now be straightforward to extend the method to the case where there are more than two possible values for a variable. Every time there is a need to distribute probability masses to a distribution to obtain bounds, we use the above method of sorting the indices according to upper or lower bounds of the other vector of the computation. We shall exemplify this by using the ideas on our murder case.

Suppose we are given the bounds as in the previous section:

$$\pi(X) = ([0.7, 0.9], [0.0, 0.3], [0.0, 0.3])$$

and

$$M(Y \mid X) = \begin{pmatrix} [0.8, 1.0] & [0.0, 0.2] & [0.0, 0.2] \\ [0.0, 0.2] & [0.8, 1.0] & [0.0, 0.2] \\ [0.0, 0.2] & [0.0, 0.2] & [0.8, 1.0] \end{pmatrix}$$

Then to find the lower bound $\pi_*(y_1)$ we do the following. First we assign as much probability mass as possible to the $\pi_Y(x_i)$ for which $M_*(Y \mid X)_{i,1}$ is minimal. We continue with this distribution until all the mass is distributed. This is in principle the same thing we do when we work with several parents in the two-valued case. After having done this we are ready to compute

$$\begin{aligned} \pi_*(y_1) &= \sum_{i=1}^{3} \pi_Y(x_i) M_*(Y \mid X)_{i,1} \\ &= 0.7 \times 0.8 + 0 \times 0 + 0 \times 0.3 = 0.56 \end{aligned}$$

The upper bound is found by assigning as much mass as possible to the $\pi_Y(x_i)$ for which $M^*(Y \mid X)_{i,1}$ is largest. This is done repeatedly until all the mass is distributed. The result is

$$\begin{aligned} \pi^*(y_1) &= \sum_{i=1}^{3} \pi_Y(x_i) M^*(Y \mid X)_{i,1} \\ &= 0.9 \times 1 + 0.1 \times 0.2 + 0 \times 0.2 = 0.92 \end{aligned}$$

The resulting interval $\pi(Y)$-vector is

$$\pi(Y) = ([0.56, 0.92], [0.00, 0.44], [0.00, 0.44])$$

One can see that the new intervals are consistent in the sense of Definition 2.1 by observing that any new bound is the result of combining consistent values for the parameters. We also observe that the convex set obtained

1. Create a joint probability distribution

$$\pi_{-X_l}(x_{1,i_1},\dots,x_{l-1,i_{l-1}},x_{l+1,i_{l+1}},\dots,x_{k,i_k})$$

from the individual $\pi_{X_j}(x_{j,i_j})$'s by using interval multiplication.

2. For all index sequences $i_1,\dots,i_{l-1},0,i_{l+1},\dots,i_k$ find bounds for

$$\begin{aligned}&M_{X_l}(x_{1,i_1},\dots,x_{l-1,i_{l-1}},x_{l+1,i_{l+1}},\dots,x_{k,i_k})\\&=\sum_{j=0}^{1}\lambda(y_j)M(Y|X_1,\dots,X_k)_{i_1,\dots,i_{l-1},0,i_{l+1},\dots,i_k,j}.\end{aligned}$$

by annihilation/reinforcement.

3. Pick all entries in the M_{X_l*}-tensor and sort in increasing order. Give the order by the index sequences placed in an array `I(1:2^{k-1})`.

4. Let $P(X_1,\dots,X_{l-1},X_{l+1},\dots,X_k)$ be a single value joint distribution over $X_1,\dots,X_{l-1},X_{l+1},\dots,X_k$. Distribute masses such that for all entries

$$\begin{aligned}&P(x_{1,i_1},\dots,x_{l-1,i_{l-1}},x_{l+1,i_{l+1}},\dots,x_{k,i_k})\\&=\pi_{-X_l*}(x_{1,i_1},\dots,x_{l-1,i_{l-1}},x_{l+1,i_{l+1}},\dots,x_{k,i_k}).\end{aligned}$$

Let S be the sum over $P(X_1,\dots,X_{l-1},X_{l+1},\dots,X_k)$. S is the mass spent so far. Let `count=1`.

```
5. Loop while $S < 1$. If $S \geq 1$ go to step 8.
6. Get indices $i'_1, \ldots, i'_{l-1}, i'_{l+1}, \ldots, i'_k$ from I(count) and set
       $P(x_{1,i'_1}, \ldots, x_{l-1,i'_{l-1}}, x_{l+1,i'_{l+1}}, \ldots, x_{k,i'_k})$
         $= \min(\pi_{-X_l *}(x_{1,i'_1}, \ldots, x_{l-1,i'_{l-1}}, x_{l+1,i'_{l+1}}, \ldots, x_{k,i'_k}) + 1 - S,$
             $\pi_{-X_l}{}^*(x_{1,i'_1}, \ldots, x_{l-1,i'_{l-1}}, x_{l+1,i'_{l+1}}, \ldots, x_{k,i'_k}))$.
7. Add
       $\pi_{-X_l}{}^*(x_{1,i'_1}, \ldots, x_{l-1,i'_{l-1}}, x_{l+1,i'_{l+1}}, \ldots, x_{k,i'_k})$
         $-\pi_{-X_l *}(x_{1,i'_1}, \ldots, x_{l-1,i'_{l-1}}, x_{l+1,i'_{l+1}}, \ldots, x_{k,i'_k})$
   to $S$. Increase count by 1 and go to step 5.
8. Use the $P(X_1, \ldots, X_{l-1}, X_{l+1}, \ldots, X_k)$-distribution to compute and return
       $\lambda_{Y*}(x_{l,0}) =$
         $\sum_{i_1, \ldots, i_{l-1}, i_{l+1}, \ldots, i_k} M_{X_l *}(x_{1,i_1}, \ldots, x_{l-1,i_{l-1}}, x_{l+1,i_{l+1}}, \ldots, x_{k,i_k})$
             $\cdot P(x_{1,i_1}, \ldots, x_{l-1,i_{l-1}}, x_{l+1,i_{l+1}}, \ldots, x_{k,i_k})$
```

Figure 4. Algorithm for computing $\lambda_{Y*}(x_{l,0})$.

with these particular interval restrictions is the same set we get when we use the method of the previous section on the same example.

Now from our fingerprint experts we have the $\lambda(Y)$-distribution given by

$$\lambda(Y) = \begin{pmatrix} [0.3, 0.8] \\ [0.5, 0.6] \\ [0.6, 0.9] \end{pmatrix}.$$

We do not normalize this vector to get some kind of normal interval distribution for the λ's. This is so because then we would have to add the constraint that the λ's sum up to 1 to our problem, which in fact leaves the problem unnecessarily more difficult. When we were normalizing λ's in the method of the previous section, the normalizing process had no impeding effect because it only created a new set of ratios with the same results, but with normality implicit.

Let us now combine $\pi(Y)$ and $\lambda(Y)$. Then the lower bound $\mathrm{Bel}_*(y_1)$ is found by reinforcement. Assign as little mass to $\pi(y_1)$ as possible, that, is $\pi_*(y_1)$. We also have to assign $\lambda_*(y_1)$ to $\lambda(y_1)$. Then to ensure that the normalizing factor becomes as small as possible, we must assign the upper bounds to the other elements in the λ-vector and distribute the rest of the probability mass to elements $\pi(y_i)$ with the largest $\lambda^*(y_i)$. The result in our case is

$$\begin{aligned} \mathrm{BEL}_*(y_1) &= \frac{\lambda(y_1)\pi(y_1)}{\Sigma_{i=1}^{3}\lambda(y_i)\pi(y_i)} \\ &= \frac{0.56 \times 0.3}{0.56 \times 0.3 + 0 \times 0.6 + 0.44 \times 0.9} = \frac{14}{47}. \end{aligned}$$

To find the upper bound of $\mathrm{BEL}(y_1)$ we use annihilation. Assign as much as possible to $\pi(y_1)$ and $\lambda(y_1)$. To maximize the normalizing factor, the other λ's become equal to their lower bounds and the rest of the mass for the π-vector is distributed so that the $\pi(y_i)$'s with the smallest corresponding $\lambda_*(y_i)$'s are as large as possible. The result is

$$\mathrm{BEL}^*(y_1) = \frac{0.92 \times 0.8}{0.92 \times 0.8 + 0.08 \times 0.5 + 0 \times 0.5} = \frac{92}{97}$$

The complete BEL vector is

$$\mathrm{BEL}(Y) = \left(\left[\frac{14}{47}, \frac{92}{97}\right], \left[0, \frac{11}{18}\right], \left[0, \frac{33}{47}\right] \right).$$

The convex set given by these interval restrictions is somewhat larger than the set given by the convex set method, and the polygon contains an extra vertex, namely (14/47, 11/18, 77/846). The extra area introduced in the baricentric triangle in Figure 5 represents the extra probability distributions introduced by our algorithm.

We shall now compute $\lambda_Y(X)$, the information sent to X from Y. We use the same type of generalization as when we computed $\pi(Y)$. To get the lower bound of $\lambda_Y(x_1)$ we first assign values as small as possible to the $\lambda(y_i)$'s. Afterwards we distribute the probability mass so that the $M(Y \mid X)_{1,i}$'s with the smallest corresponding $\lambda(y_i)$'s get as much mass as possible. The result is

$$\lambda_{Y*}(x_1) = 0.3 \times 1.0 + 0.5 \times 0 + 0.5 \times 0 = 0.3.$$

The complete $\lambda_Y(X)$ vector is

$$\lambda_Y(X) = ([0.30, 0.82], [0.46, 0.66], [0.46, 0.90]).$$

Combining this vector with the a priori probabilities of X gives us

$$\text{BEL}(X) = \left(\left[\frac{7}{16}, \frac{369}{392}\right], \left[0, \frac{33}{68}\right], \left[0, \frac{9}{16}\right]\right).$$

The polygon resulting from these limits is given in Figure 6. The area shaded with a grid shows the error introduced.

The method sketched for computing upper and lower bounds of BEL is sufficiently general to work for all numbers of possible values for a variable. The idea is to sort on the lower or upper bounds of the λ-vector, not including the element that corresponds to the one we are computing a bound of. We use the order to distribute the rest of the mass to minimize or maximize the denominator in the equation for BEL. A minimal denominator gives us the upper bound, and a maximal denominator gives us the lower bound. If a many-valued variable has more than one parent, the methods to use to compute the messages $\pi(Y)$ and $\lambda_Y(X)$ from and to

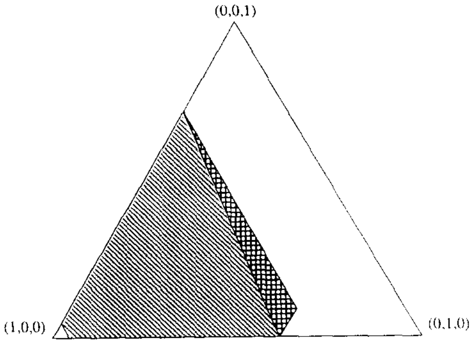

Figure 5.

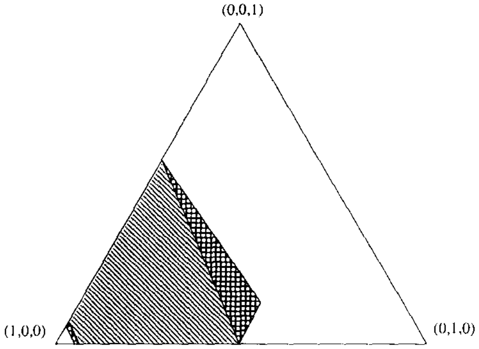

Figure 6.

the parents are generalizations of the algorithms given in Figures 3 and 4. This approximation method works in polynomial time and is thus an interesting alternative to the intractable method presented in the previous section.

In his book [17], Pearl discusses multiply connected networks and how one can propagate constraints in them. This subject has not been discussed in this paper. However, the methods of clustering and conditioning presented there should be easy to generalize for use with the methods given here. This would be a topic for further research.

6. ERROR ANALYSIS

As we have seen from the example, the annihilation/reinforcement (A/R) algorithm introduces an error in the intervals after propagation. No extra error is introduced into intervals if we use convex hulls to represent the nonconvex sets of distributions. This is so because the vertices of the convex hull of the nonconvex set will also be extremes of the correct set. Any point in the interior of the convex hull will not be able to become an extreme point during further propagation because the exact propagation uses linear transformations. When transforming a convex polytope P, a linear transformation creates a new polytope P' whose vertices are transformations of vertices of P, and thus only extreme points of the original nonconvex set will become possible extreme points of new nonconvex sets.

As a result of this, an interesting way of looking at the errors may be to look at the overestimation of the intervals given by the A/R algorithm. We shall first look at how the total and relative errors in intervals change through propagation of *initial data* in a tree-structured belief network. Afterwards we shall see what happens when we introduce dummy nodes into the same type of network.

The first experiments are done in belief networks where variables can take three values. Any higher number of values leads to so many possible vertices in the polytopes we must compute that experiments are difficult to perform.

The first experiments performed were designed to discover how the error from one source of data in singleton intervals changes through propagation. We tested 101 random belief networks (chains) with interval probabilities in all initial data. The idea was to watch the error caused by the initial a priori probability. Each network consisted of 20 nodes in a chain (no node had more than one child), and each node could take only three values.

When the errors at each node were compared with the errors at all other nodes, the analysis showed that both total and relative errors have a clear tendency to grow the first few four or five steps of propagation, but after this there seems to be no significant (0.05-level) change in the error size. After the tenth step, however, there is a significant reduction in both total and relative errors. This last observation is to be expected, as interval probabilities after a long chain of propagation will converge to the interval $[0, 1]$.

The next experiments did the same type of analysis of the error using variables that could take five values and use *point* matrices. The data we propagated were the interval a priori probabilities in the root node. The results showed that the relative error in this case seems to grow large, whereas the total error seems to go toward zero.

Furthermore, the correct probability distribution seemed to converge to point distributions as we want away from the root. This can be explained by the use of single-valued matrices and a result from the theory of nonstationary Markov processes (Isaacson and Madsen [7, Ch. 5]) The relation to our problem is that a belief network that is a chain can be seen as a nonstationary Markov process where the conditional probability matrices correspond to the transition matrices, and the nodes correspond to the variables we observe at a certain time. If you multiply a set of m transition matrices of a Markov processes, P_t, $t = 1, \dots, m$, to get the matrix $P^{(m)}$, then the number $\delta(P^{(m)}) = 1 - \min_{i,k} \Sigma_{j=1}^{n} \min(p_{ij}, p_{kj})$ converges to 0 when $m \to \infty$. The constraints on the matrices P_t are not very restrictive and include most of the random matrices we have used in our experiments. (If a matrix that is excluded by the constraints is used, the convergence is postponed only one time step.) The convergence result says that the maximal distance between elements in columns becomes zero after infinite time. So what we would get in our case is that the maximum distance between elements in columns of the matrix product of the conditional probability matrices converges to zero as we get far down in a chain. In a way, we can say that we forget the a priori probability of the root node as we move away from it.

The property that the A/R algorithm also seems to converge is also explainable. If we let $w_i = \pi^*(x_i) - \pi_*(x_i)$, then the sum $\Sigma_{i=1}^{n} w_i$ will decrease as we move X away from the root node. To suggest why it is so, let us first use the standard interval approach and multiply the intervals of the π-vectors by the point conditional probability matrices. Then the sum of the widths of intervals, $\Sigma_{i=1}^{n} w_i$, will be constant, as we multiply every interval by a total of 1. When the A/R algorithm is used to compute lower and upper bounds, however, some of the singleton probabilities will take lower bounds and others upper bounds as values at the same time. This will almost always reduce the number each interval is multiplied by to lower than (and never larger than) 1 and thus reduce the total width. There are special cases where this does not work. But, at least in the case where matrices contain only elements smaller than 1 and there are at least $n - 1$ nonzero elements larger than a constant ϵ in each row, this algorithm will also be less affected by the root node value the farther away we move.

The experiments showed that the relative error grows and the total error decreases as we move away from the root node. This, together with the theoretical results mentioned, implies that both methods converge to the same point but that the correct method converges faster. This again suggests that when interval matrices are used, the relative and total errors will be acceptable. Use of interval matrices can be seen as the use of all combinations of point matrices from the convex sets. All of these combinations will converge to give single-valued results from the interval starting point, both in the correct case and in the A/R algorithm case. When we look at nodes far away from sources of data, the difference set between the polytope given by the A/R algorithm and the correct set can thus be said to be a thin layer of incorrect probability distribution. The largest errors will be found when one is relatively close to the sources of data. Whether these are acceptable or not depends on the problem.

In the last experiments we introduced dummy nodes into the networks. In this case only trivalued variables were used, and a maximum of five random dummy nodes were generated and connected to the network at random points. The random trees contained 10 standard-type nodes, and every node had a maximum of four children.

In these experiments both relative and total errors grew when the dummy nodes were added. Some of the experiments also gave as a result that the errors became smaller when the number of dummy nodes approached 5. This last effect is presumably caused by the fact that intervals tend to grow toward [0, 1]. But the errors always had acceptably small values. The mean size of the A/R intervals taken over the whole graph after propagation of new data was at the most 1.4 times larger than the correct size, and the variance of this number was no larger than 0.1.

Aìl the experiments mentioned here were performed with initial intervals of mean width 0.2. It would be natural to assume that the relative error would grow larger when intervals were narrower and also that this relative error would start to diminish (because of the effect that intervals tend to grow to [0, 1]) at nodes farther away from the source of data.

The conclusion of the error analysis must be that the errors introduced do not seem to be of a dimension that makes the A/R algorithm useless. The relative errors did not exceed an unacceptable limit during the experiments, even after several dummy nodes were added to the data.

7. CONCLUSION

It has been shown that when we accept the limitation that we are allowed to specify only upper and lower probabilities for singleton sets, it is in fact possible to propagate these constraints efficiently in a singly connected belief network. This is under the assumption that we accept some errors in the interval beliefs.

The error analysis shows that the annihilation/reinforcement algorithm presented here should be of interest to anyone considering the use of interval probabilities. Of course, there are cases where too much information is lost, but by and large the errors introduced by the algorithm seem acceptable.

The limitation that we should only consider singleton interval probabilities, however, may be an important restriction in many applications. It should be of interest to try to use general interval constraints. In [21] Tessem attempts to use linear programming techniques.

ACKNOWLEDGMENTS

This research was supported by the Royal Norwegian Council of Scientific and Industrial Research grant IT2.23.18229.

References

1. Kyburg, H. E., Jr., Bayesian and non-Bayesian evidential updating, *AI*, **31**, 271–293, 1987.

2. Pearl, J., Fusion, propagation, and structuring in belief networks, *AI*, **29**, 241–288, 1986.

3. Fertig, K. W., and Breese, J. S., Interval influence diagrams, in *Uncertainty in Artificial Intelligence,* Vol. 5 (M. Henrion, R. D. Schachter, L. N. Kanal, and J. F. Lemmer, Eds.), North-Holland, Amsterdam, 149–161, 1990.

4. Dubois, D., and Prade, H., *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum, New York, 1986.

5. Shafer, G., *A Mathematical Theory of Evidence*, Princeton Univ. Press, Princeton, N.J., 1976.

6. Driankov, D., A calculus for belief-intervals representation of uncertainty, in *Uncertainty in Knowledge-Based Systems*, Lect. Notes Comput. Sci. 286 (B. Bouchon and R. R. Yager, Eds.), Springer-Verlag, Berlin, 205–216, 1987.

7. Driankov, D., A many valued logic of belief: detachment operators, in *Uncertainty and Intelligent Systems* (B. Bouchon, L. Saitta, and R. R. Yager, Eds.), Springer-Verlag, Berlin, 265–272, 1988.

8. Tessem, B., Truth maintenance in infinitely-valued logic, Tech. Rep. 35, Dept. of Informatics, Univ. Bergen, Norvay, 1989.

9. Eick, C. F., Uncertainty management for fuzzy decision support systems, *Proceedings of the 4th Workshop on Uncertainty in AI*, 98–108, 1988.

10. Kyburg, H. E., Jr., Representing knowledge and evidence for decision, in *Uncertainty in Knowledge-Based Systems* (B. Bouchon and R. R. Yager, Eds.), Springer-Verlag, Berlin, 30–40, 1987.

11. Kyburg, H. E., Jr., Knowledge, in *Uncertainty in Artificial Intelligence*, Vol. 2 (J. F. Lemmer and L. N. Kanal, Eds.), North-Holland, Amsterdam, 263–272, 1988.

12. Neapolitan, R. E., and Kenevan, J., Justifying the principle of interval constraints, *Proceedings of the 4th Workshop on Uncertainty in AI*, 266–274, 1988.

13. Snow, P., Bayesian inference without point estimates, *Proc. AAAI-86, Proceedings of the 5th Conference on AI*, 233–237, 1986.

14. White, C. C., A posteriori representations based on linear inequality descriptions of a priori and conditional probabilities, *IEEE Trans. Syst., Man Cybern.*, **16**(4), 570–573, 1986.

15. Tessem, B., Interval probability propagation, Tech. Rep. 39, Dept. of Informatics, Univ. Bergen, Norway, 1989.

16. Jeffrey, R., *The Logic of Decision*, McGraw-Hill, New York, 1965.

17. Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, New York, 1988.

18. Edelsbrunner, H., *Algorithms in Combinatorial Geometry*, Springer-Verlag, Berlin, 1987.

19. Moore, R. E., *Interval Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1966.

20. Isaacson, D. L., and Madsen, R. W., *Markov Chains—Theory and Applications*, Wiley, New York, 1976.

21. Tessem, B., Extending the A/R algorithm for interval probability propagation, Tech. Rep. 42, Dept. of Informatics, Univ. Bergen, Norway, 1989.