Efficient Probabilistic Reasoning in Bayes Nets with Mutual Exclusion and Context Specific Independence[*]

Carmel Domshlak
Computer Science Department
Cornell University
Ithaca, NY 14853, USA
dcarmel@cs.cornell.edu

Solomon E. Shimony
Computer Science Deptartment
Ben-Gurion University
Beer-Sheva 84105, Israel
shimony@cs.bgu.ac.il

Abstract

Prior work has shown that context-specific independence (CSI) in Bayes networks can be exploited to speed up belief updating. We examine how networks with variables exhibiting mutual exclusion (e.g. "selector variables"), as well as CSI, can be efficiently updated. In particular, singly-connected networks, that have an additional common selector variable, can be updated in linear time, where quadratic time would be needed without the mutual exclusion requirement. The above result has direct applications, as such network topologies can be used in predicting the ramifications of user selection in some multimedia systems.

Introduction

Using Bayes networks (BNs) to model uncertain knowledge, and performing inference in this model, are of major interest in both theoretical and applied AI research (Pearl 1988). As inference over BNs is hard in the general case (Cooper 1990; Dagum & Luby 1993; Shimony 1994), efficient algorithms for sub-classes of Bayes networks are of extreme importance. Numerous inference algorithms on BNs use a reduction to a known tractable class, in order to perform inference. The reduction is usually exponential in some aspect of the problem instance. Conditioning algorithms (Diez 1996; Horvitz, Suermondt, & Cooper 1989; Pearl 1988) use a cutset whose removal reduces inference into a number of (easy) inference problems on polytrees - the number of polytree inference problems is exponential in the cutset size. Similarly, clustering schemes (Jensen, Olsen, & Andersen 1990; Lauritzen & Speigelhalter 1988) aggregate nodes into macro-nodes organized as a tree, and problem reformulation cost is exponential in the number of nodes in each macro-node.

Relatively recent work has used local independence structure, also known as context-specific independence (CSI) (Boutilier *et al.* 1996) to improve performance of belief updating in Bayes networks. We argue here than in addition to taking advantage of CSI, some application networks exhibit distributions with specific types of mutual exclusion, which can also be used to improve performance. In particular, we examine Bayes networks that are singly connected, except for one additional multiple-valued "selector" variable S (see below for the precise definitions). S may have multiple children, creating an arbitrary number of undirected cycles.

Intuitively, a selector variable models user selection from a large set of options. In some user-interface applications, the user can force one of a set of variables into a particular state. System behaviour is such that a user action can cause more than one system action. The goal is to predict system behaviour, when a distribution over user actions is known. The prediction is necessary in order to achieve better system performance, for example by attempting to optimize actions that are more likely to be executed in the near future. The unknown user selection is modeled by a selector variable, and the system is modeled by the rest of the network.

If the network has n nodes, the selector variable has $O(n)$ possible values. Since the selector may have all network nodes as its children, a natural way to evaluate such a network is by cutset conditioning, with node S being a singleton cutset, resulting in complexity $O(n^2)$ (linear time for each singly connected network problem instance, and $O(n)$ singly connected problem instances - one for each value of S). Other known algorithms can do no better. However, by carefully taking advantage of singly connected network properties, and the CSI and mutual exclusion properties of the selector, probability updating can be done in time linear in the size of the network - the main contribution of this paper.

The rest of the paper is organized as follows. We begin with a formal definition of the problem (network structures, as well as selector variables). The equations and algorithm for computing marginal probabilities for all nodes (also called belief updating) for the null evidence case are then developed. This is followed by extending the results to arbitrary conjunctive evidence. Finally, an application for the presented results is mentioned, and related work on belief updating is examined, suggesting some future work.

Problem Definition

As excellent introductions to Bayes networks abound (Charniak 1991; Neapolitan 1990; Pearl 1988), it suffices to briefly define our notation, as well as to overview the standard inference problems on BNs. A Bayes network $\mathcal{B} = (G, P)$ represents a probability distribution as a directed

[*] Partially supported by a grant from the Israeli Science Ministry, and the Paul Ivanier Center for Robotics and Production Management, Ben-Gurion University.

acyclic graph G (see Figure 1), where its set of nodes V stands for random variables (assumed discrete in this paper), and P, a set of tables of conditional probabilities (CPTs) - one table for each node $X \in V$. For each possible value $x \in D(X)$ (where $D(X)$ denotes the domain of X - the set of possible values for X), the respective table lists the probability of the event $X = x$ given each possible value assignment to (all of) its parents. Thus, the table size is exponential in the in-degree of X. Usually, it is assumed that this in-degree is small - otherwise, representation of the distribution as a Bayes network would not be a good idea in the first place. (We thus assume that the in-degree is bounded by a constant, whenever algorithm runtime results are claimed in this paper.) The joint probability of a complete state (assignment of values to all variables) is given by the product of $|V|$ terms taken from the respective tables (Pearl 1988). That is, with $\Pi(X)$ denoting the parents of X in G, we have:

$$P(V) = \prod_{X \in V} P(X|\Pi(X))$$

(Directed-path) singly connected (DP-singly connected, as an abbreviation) Bayesian networks are networks where for every pair of nodes (s,t) in the directed acyclic graph (DAG) of the network, there is at most one *directed* path from s to t (see Figure 1a). The notion defined here is somewhat more general than polytree topology, where the requirement is that there be at most one path from s to t in the *underlying undirected graph.* Reasoning on polytrees is known to be easy (Kim & Pearl 1983). All polytrees are DP-singly connected, but not vice versa. For example, the network in Figure 1a is *not* a polytree, even though it *is* DP-singly connected.

Probabilistic reasoning (inference) is usually in one of two forms: belief updating, and belief revision (Pearl 1988). In either case, a distinction can be made between a problem with conjunctive *evidence*, which is a partial assignment $\mathcal{E}$ to some of the variables (presumably *observed* values for some of the variables), and a reasoning problem with no evidence (or null evidence).

The belief updating problem is: compute marginal distributions for all variables given the evidence, i.e. compute $P(X = x|\mathcal{E})$ for all $X \in V$ and for each value $x \in D(X)$. Belief revision, also called most probable explanation (MPE), is finding the assignment A to all the variables that maximizes $P(A|\mathcal{E})$. We discuss only the belief updating problem, although our method and results may also be applicable to belief revision.

We consider applications where the reasoning system needs to address an external event (e.g. a user) making a selection, i.e. forcing *no more than* one of the network variables into a specific state, with $\perp$ denoting that no variable is forced. The a-priori selection distribution, or its dependence on some variables, is known - but for simplicity we initially assume the former. The a-priori distribution is modeled by a selector variable S with no parents. Without loss of generality, let the set of children of S be all the other nodes in the network. The domain of S is the sum of the domains of all its children, plus a special value $\perp$. When S has a specific value (some value other than $\perp$), one of its children is forced to the respective value.

Formally, let $\mathcal{B}$ be a Bayes network over the variables $\{X_1, \ldots, X_n, S\}$ with S being a selector variable (formally defined below). Assume for clarity that the domains of the X_i variables are disjoint, and that their values are denoted $x_{i,j}$, with $1 \leq j \leq |D(X_i)|$, respectively. The domain of S is $D(S) = \bigcup_{i=1}^{n} D(X_i) \cup \{\perp\}$, where $S = x_{i,j}$ means that variable X_i is forced to have value $x_{i,j}$, while $S =\perp$ means no variable is forced. The semantics of having the selector as a parent is:

$$P(X_i = x_{i,j}|\Pi'(X_i)) = \begin{cases} 1, & S = x_{i,j} \\ 0, & S = x_{i,k}, k \neq j \\ p(x_{i,j}, \Pi(X_i)), & \text{otherwise} \end{cases}$$

where $\Pi'(X_i)$ are the parents of X_i in $\mathcal{B}$, and $\Pi(X_i) = \Pi'(X_i) \setminus \{S\}$. The probabilities $p(x_{i,j}, \Pi(X_i))$ are an arbitrary probability table denoting conditional probabilities of X_i given its parents for the case that S does not select X_i. Observe that there is no need to maintain a complete probability table for X_i given all its parents - it is sufficient to keep the table for X_i given its parents *excluding* S (for the state $S = \perp$).

Note the mutual exclusion - only at most one variable is forced. Also, note the following context-specific independence - if S is known not to select (force) some variable X_i, then X_i becomes independent of S given the state of the other parents of X_i. Our algorithm takes advantage of both properties.

Algorithm for Null Evidence

Let $\mathcal{B}$ be a Bayes network over the variables $\{X_1, \ldots, X_n, S\}$, with S being a parent-less selector variable as defined above, and such that $\mathcal{B} \setminus \{S\}$ is DP-singly connected (see Figure 1b). Our problem here is to compute $P(X_i)$ with no evidence.

Consider any node X_i. We distinguish between the three following cases, and analyze them separately:

1. H_i (where S forces (selects) X_i),
2. H_i^+ (where S forces an ancestor of X_i in $\mathcal{B}$),
3. $\overline{H}_i$ (S does not force X_i, nor its ancestors).

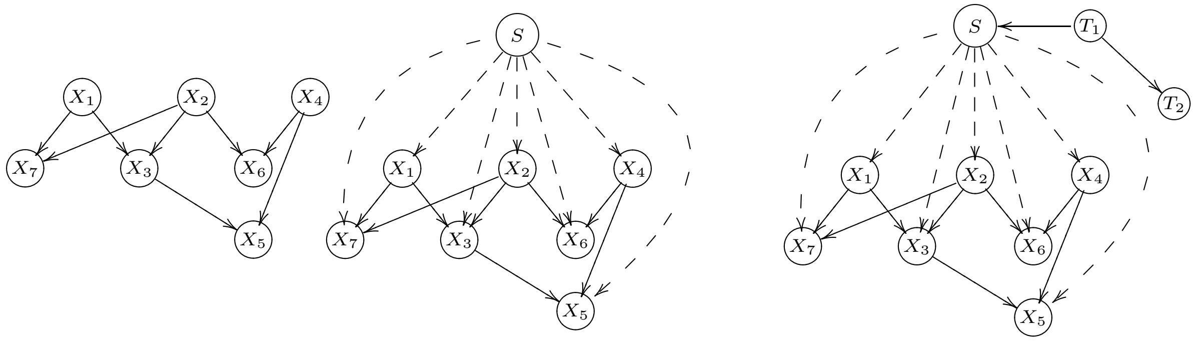
Clearly, the hypotheses H_i, H_i^+, and $\overline{H}_i$ constitute a disjoint cover of all possible values of S, and thus:

$$\begin{aligned} P(X_i = x_{i,j}) =& P(X_i = x_{i,j}|H_i)P(H_i) \\ &+ P(X_i = x_{i,j}|H_i^+)P(H_i^+) \\ &+ P(X_i = x_{i,j}|\overline{H}_i)P(\overline{H}_i) \end{aligned} \quad (1)$$

The case of H_i is straightforward:

$$P(X_i = x_{i,j}|H_i) = P(S = x_{i,j}|H_i) = \frac{P(S = x_{i,j})}{P(H_i)} \quad (2)$$

Now consider the case of $\overline{H}_i$. Denote by $\mathcal{A}(\Pi(X_i))$ the set of all complete assignments (of values to variables) on $\Pi(X_i)$. Observe that (by definition of $\overline{H}_i$):



(a) DP-singly connected (b) DP-singly connected + selector (c) DP-singly connected + non-root selector

Figure 1: Several topologies of Bayes networks

$$\begin{aligned} P(X_i = x_{i,j}|\overline{H}_i) &= P(X_i = x_{i,j}|S = \perp) \\ &= \sum_{\mathsf{A}\in\mathcal{A}(\Pi(X_i))} p(x_{i,j},\mathsf{A}) \prod_{X_m\in\Pi(X_i)} P(X_m = \mathsf{A}(X_m)|S = \perp) \end{aligned} \quad (3)$$

Evaluating Eq. 3 is equivalent to standard null-evidence belief propagation for polytree Bayes networks (i.e. passing only π-messages (Kim & Pearl 1983)), resulting from conditioning on $S = \perp$.

Finally, consider H_i^+, assuming $P(H_i^+) > 0$. By conditioning on all possible assignments on $\Pi(X_i)$ we get:

$$\begin{aligned} P(X_i = x_{i,j}|H_i^+) &= \sum_{\mathsf{A}\in\mathcal{A}(\Pi(X_i))} P(X_i = x_{i,j}|\mathsf{A}, H_i^+) P(\mathsf{A}|H_i^+) \\ &= \sum_{\mathsf{A}\in\mathcal{A}(\Pi(X_i))} P(X_i = x_{i,j}|\mathsf{A}) P(\mathsf{A}|H_i^+) = \\ \frac{1}{P(H_i^+)} & \sum_{\mathsf{A}\in\mathcal{A}(\Pi(X_i))} P(X_i = x_{i,j}|\mathsf{A}) P(\mathsf{A}, H_i^+) \end{aligned} \quad (4)$$

Observe that the hypothesis H_i^+ can be further decomposed as follows:

$$H_i^+ = \bigcup_{X_m\in\Pi(X_i)} (H_m \cup H_m^+)$$

Since $\mathcal{B} \setminus \{S\}$ is DP-singly connected, the above decomposition is a *disjoint cover*, and thus, for each $\mathsf{A} \in \mathcal{A}(\Pi(X_i))$, the term $P(\mathsf{A}, H_i^+)$ in Eq. 4 can be decomposed as follows:

$$\begin{aligned} P(\mathsf{A}, H_i^+) = & \sum_{X_m\in\Pi(X_i)} P(\mathsf{A}|H_m) \cdot P(H_m) \\ & + \sum_{X_m\in\Pi(X_i)} P(\mathsf{A}|H_m^+) \cdot P(H_m^+) \end{aligned} \quad (5)$$

The first and the second terms in Eq. 5 distinguish between forcing a parent X_m of X_i, and forcing one of the ancestors of X_m, respectively. Now we rewrite Eq. 4 using the decomposition in Eq. 5, while changing the order of summation, to get:

$$\begin{aligned} P(X_i = x_{i,j}|H_i^+) &= \frac{1}{P(H_i^+)} \times \\ \times \sum_{X_m\in\Pi(X_i)} \Bigg(& \sum_{\mathsf{A}\in\mathcal{A}(\Pi(X_i))} P(X_i = x_{i,j}|\mathsf{A}) P(\mathsf{A}|H_m) P(H_m) + \\ & \sum_{\mathsf{A}\in\mathcal{A}(\Pi(X_i))} P(X_i = x_{i,j}|\mathsf{A}) P(\mathsf{A}|H_m^+) P(H_m^+) \Bigg) \end{aligned} \quad (6)$$

The only terms in Eq. 6 that are not trivial are $P(\mathsf{A}|H_m)$ and $P(\mathsf{A}|H_m^+)$. Recall that $\mathcal{B} \setminus \{S\}$ is DP-singly connected, and thus the parents of X_i are independent given H_m, and likewise for H_m^+. Thus we have:

$$\begin{aligned} P(\mathsf{A}|H_m) &= P(X_m = \mathsf{A}(X_m)|H_m) P(H_m) \times \\ & \times \prod_{X_k\in\Pi(X_i)\setminus\{X_m\}} P(X_k = \mathsf{A}(X_k)|\overline{H}_k) \\ P(\mathsf{A}|H_m^+) &= P(X_m = \mathsf{A}(X_m)|H_m^+) P(H_m^+) \times \\ & \times \prod_{X_k\in\Pi(X_i)\setminus\{X_m\}} P(X_k = \mathsf{A}(X_k)|\overline{H}_k) \end{aligned}$$

All terms can now be computed recursively. To compute marginal probabilities for all variables efficiently, proceed top-down instead, as shown in Figure 2.

Theorem 1 *Given a Bayes net $\mathcal{B}$ over the variables $\{X_1, \ldots, X_n, S\}$ with S a selector variable, such that $\mathcal{B} \setminus \{S\}$ is DP-singly connected, algorithm* BeliefProp *computes the marginal probabilities $P(X_i)$, for $1 \leq i \leq n$, with no evidence, in time linear in the size of $\mathcal{B}$.*

Proof: Immediate - observe that the algorithm loops n times, and each equation takes constant time to compute, using previously computed terms (assuming in-degree and variable domain size (excluding S) bounded by a constant). Topological sort also takes linear time. □

The complexity of BeliefProp is a factor of n better than the optimal respective cut-set conditioning scheme, where the cut-set would be the singleton set $\{S\}$, and where we would need to perform one propagation for each possible state of S. Assuming bounds on the in-degree and domain

```
BeliefProp
Topologically sort the Bayes net $\mathcal{B}$,
  i.e. make $X_m \in \Pi(X_i) \Rightarrow i > m$.
for $i = 1$ to $n$ do
  $P(H_i) = \sum_{x_{i,j} \in D(X_i)} P(S = x_{i,j})$
  $P(H_i^+) = \sum_{X_m \in \Pi(X_i)} (P(H_m) + P(H_m^+))$
  $P(\overline{H}_i) = 1 - P(H_i) - P(H_i^+)$
  foreach $x_{i,j} \in D(X_i)$ do
    Compute $P(X_i = x_{i,j} | H_i^+)$ using Eq. 6.
    Compute $P(X_i = x_{i,j} | \overline{H}_i)$ using Eq. 3.
    Compute $P(X_i = x_{i,j})$ using Eq. 1.
```

Figure 2: Algorithm for belief propagation

cardinality allows us to take $O(n)$ as the size of the network. In the standard polytree algorithm, taking the size of the network to include the size of the CPTs (possibly exponential in the in-degree), allows one to drop these assumptions. Although a similar argument can be made in our case, our algorithm has a further multiplicative factor (the in-degree) in the equations, and we thus cannot quite claim linear time for unbounded in-degree, but can drop the requirement on domain size.

Extensions

Although the above algorithm is sufficient for one application discussed below, it is certainly of interest whether the scheme is applicable to belief updating in the presence of evidence, as well as relaxing the assumption that S is a root node. It turns out that we can extend the conditions for using the algorithm to allow selector S to be a non-root node, as well as the presence of evidence, as long as all the following conditions hold:

1. All evidence nodes are not descendants of S,
2. S together with its non-descendants constitute a polytree T, and
3. Every path, from any node in T to any descendant of S, contains S.

If these conditions hold (for example, see Figure 1c assuming the evidence is only at T_1 and/or T_2), the following simple scheme still performs correct belief updating, and takes only linear time:

1. Compute the probability of S given the evidence, by (temporarily) removing all descendents of S and performing belief updating on the resulting polytree.
2. Temporarily remove from the original network all nodes except for S and its descendents, and use our algorithm.

However, allowing *unrestricted* conjunctive evidence complicates things considerably. First, for DP-singly-connected networks, belief updating with evidence is NP-hard (Shimony & Domshlak 2002a), thus we cannot expect to solve the problem in linear time, even *without* the additional selector variable. Second, evidence at or below "converging" nodes (e.g. X_5, X_7 in Figure 1c), also called *diagnostic* evidence, tends to create further dependencies. Nevertheless, if the network (excluding S) is a *polytree*, efficient belief updating is still possible:

Theorem 2 *Given Bayes net $\mathcal{B}$ over variables $\{X_1, \ldots, X_n, S\}$ with S a selector variable, such that $\mathcal{B} \setminus \{S\}$ is a polytree, computing marginals $P(X_i|\mathcal{E})$, for $1 \leq i \leq n$, where $\mathcal{E}$ is arbitrary conjunctive evidence, can be done in time linear in the size of $\mathcal{B}$.*

Clearly, however, a different algorithm (presented elsewhere (Shimony & Domshlak 2002b) due to lack of space) must be used. Suffice it to say that the algorithm required for belief updating with unrestricted evidence uses an intricate analysis of location of pieces of evidence as in the standard polytree belief updating algorithm in conjunction with careful conditioning on some states of S, similar to that done in the algorithm presented here. However, the number of cases one needs to consider is much larger, though still a constant, i.e. independent of the size of the network.

Discussion

The issue of selectors in Bayes networks was raised in trying to predict system behaviour in user-interface type applications, where a user action can cause more than one system action. The prediction is necessary in order to achieve better system performance, for example by attempting to optimize actions that are more likely to be executed in the near future. The unknown user selection is modeled by a selector variable, and the system is modeled by the rest of the network.

In particular, (Domshlak & Shimony 2002) discusses an adaptive system that presents multimedia data items and/or multi-component web pages based on preferential constraints from the author, as well as on user selection. Due to the constraints, a selection has multiple, non-trivial ramifications. If a user-model (i.e. a distribution over user selections) is available, one can use it to predict the distribution of the resulting system actions. In this application, the actions are retrieving and transmitting certain data items, which may be time consuming. Response time to the user can be significantly improved if some of the actions are done ahead of time, but since resources are limited it is important to know which are more likely to be required, given the state of the system (evidence). The results in this paper are directly applicable there, as user selection(s) is modeled by the selector variable(s) and system constraints (whether deterministic or probabilistic) are modeled by the rest of the Bayes network. Due to lack of space (and the algorithmic focus of this paper) the exact details of this specific prediction scheme are discussed elsewhere.

We briefly discuss other possible future extensions to our algorithm. First, consider the case where there is more than one selector variable, in order to model more than one user selection (or other external event). If the topology is still DP-singly connected, except for the common selector variables, it seems that the above methods can still apply, but with a significantly larger number of cases ("H"s) to handle. For m selectors, this complexity should be nK^m for some small constant K. This is exponential in m, but complexity of existing algorithms will be $O(n^{m+1})$, so our scheme has potential here.

Further possible extensions would be to consider dependent selectors, as well as other topologies. In a general topology, with one selector, one could apply a join-tree algorithm to the network (excluding S). Possibly, some form of mutual exclusion can be used for the resulting join-tree in the presence of S. Due to clustering effects, the mutual exclusion property may hold only partially, but it may still be possible to use it to gain some performance improvement. Finally, it may be possible to take advantage of forms of mutual exclusion other than selectors.

Some related work relevant to taking advantage of various belief network special-case characteristics is briefly discussed below. Our algorithm can be seen as an extension of the polytree belief updating algorithm (Kim & Pearl 1983). In fact, some of the quantities computed in our algorithm are exactly π messages from (Kim & Pearl 1983), and similar quantities. Likewise, for the general evidence case, our extended algorithm (Shimony & Domshlak 2002b) uses both π and λ messages, as well as other types of messages that are specific to aggregated states of the selector S.

Alternately, one can view our algorithm within the framework of refined conditioning schemes presented in (Darwiche 1995; 2000), but where we take additional advantage of context specific independence and the selector properties of S. As written, the scheme proposed in (Darwiche 1995; 2000) would still have a quadratic runtime, even for networks that are polytrees (with one additional common selector).

Yet another way to look at our algorithm is within the framework of Symbolic Probabilistic Inference (SPI (D'Ambrosio 1993)). Their system attempts to perform (automated) factoring of the symbolic equations for belief updating, and in theory could arrive at results similar to our scheme. However, this is unlikely in practice, without adding in rules for handling mutual exclusion and specific search heuristics - an interesting issue for future research.

Summary

We examined a special case of Bayes network - DP-singly connected, except for a common selector variable S. Belief updating in Bayes networks with this topology, which are "almost" DP-singly connected, can be done by conditioning on S in quadratic time, which is the best that existing algorithms can achieve. We developed a linear-time algorithm for this problem.

It may be possible to take advantage of our ideas in networks with several selector variables, an issue for future research. Interaction of the selector variables with the other variables may also be generalized.

Networks of the type discussed in this paper can be used to predict system actions in some applications, and to improve system performance in some criteria (such as average response time). One such application, adaptive multimedia presentation, was briefly discussed.

References

Boutilier, C.; Friedman, N.; Goldszmidt, M.; and Koller, D. 1996. Context-specific independence in Bayesian networks. In *Proceedings of UAI-96*, 115–123.

Charniak, E. 1991. Bayesian networks without tears. *AI Magazine* 12(4):50–63.

Cooper, G. F. 1990. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence* 42 (2-3):393–405.

Dagum, P., and Luby, M. 1993. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence* 60 (1):141–153.

D'Ambrosio, B. 1993. Incremental probabilistic inference. In *Proceedings of UAI-93*, 301–308.

Darwiche, A. 1995. Conditioning methods for exact and approximate inference in causal networks. In *Proceedings of UAI-95*, 99–107.

Darwiche, A. 2000. Recursive conditioning. *Artificial Intelligence* 125(1-2):5–41. Special Issue on Resource Bounded Reasoning.

Diez, F. J. 1996. Local conditioning in Bayesian networks. *Artificial Intelligence* 87(1-2):1–20.

Domshlak, C., and Shimony, S. E. 2002. Improving the dynamic behavior of CP-net based multimedia systems by predicting likely components. In *AAAI/KDD/UAI Joint Workshop on Real-Time Decision Support and Diagnosis Systems*.

Horvitz, E. J.; Suermondt, H. J.; and Cooper, G. F. 1989. Bounded conditioning: Flexible inference for decisions under scarce resources. In *5th Workshop on Uncertainty in AI*, 182–193.

Jensen, F. V.; Olsen, K. G.; and Andersen, S. K. 1990. An algebra of Bayesian belief universes for knowledge-based systems. *Networks* 20:637–660.

Kim, J. H., and Pearl, J. 1983. A computation model for combined causal and diagnostic reasoning in inference systems. In *Proceedings of IJCAI-83*, 190–193.

Lauritzen, S., and Speigelhalter, D. J. 1988. Local computations with probabilities on graphical structures and their applications to expert systems. *Journal of the Royal Statistical Society* 50:157–224.

Neapolitan, R. E. 1990. *Probabilistic Reasoning in Expert Systems*. John Wiley and Sons. chapter 8.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.

Shimony, S. E., and Domshlak, C. 2002a. Complexity of probabilistic reasoning in singly connected (not polytree!) Bayes networks. *submitted for publication*.

Shimony, S. E., and Domshlak, C. 2002b. Belief updating in polytree Bayes nets with an additional selector variable. Technical Report FC-03-03, Computer Science Department, Ben-Gurion University.

Shimony, S. E. 1994. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence Journal* 68(2):399–410.