Relevance-based Sequential Evidence Processing in Bayesian Networks

Yan Lin & Marek J. Druzdzel
Decision Systems Laboratory
School of Information Sciences
and Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
{yan,marek}@sis.pitt.edu

Abstract

Relevance reasoning in Bayesian networks can be used to improve efficiency of belief updating algorithms by identifying and pruning those parts of a network that are irrelevant for the computation. Relevance reasoning is based on the graphical property of *d*-separation and other simple and efficient techniques, the computational complexity of which is usually negligible when compared to the complexity of belief updating in general.

This paper describes a belief updating technique based on relevance reasoning that is applicable in practical systems in which observations are interleaved with belief updating. Our technique invalidates the posterior beliefs of those nodes that depend probabilistically on the new evidence and focuses the subsequent belief updating on the invalidated beliefs rather than on all beliefs. Very often observations invalidate only a small fraction of the beliefs and our scheme can then lead to substantial savings in computation. We report results of empirical tests that demonstrate practical significance of our approach.

Introduction

Emergence of probabilistic graphs, such as Bayesian belief networks (BBNs) (Pearl 1988) and closely related influence diagrams (Howard and Matheson 1984) has made it possible to base uncertain inference in knowledge-based systems on the sound foundations of probability theory and decision theory. However, as many practical models tend to be large, the main problem faced by the decision-theoretic approach using probabilistic graphs is the complexity of probabilistic reasoning, shown to be NP-hard both for exact inference (Cooper 1990) and for approximate inference (Dagum and Luby 1993). The critical factor in exact inference schemes is the topology of the underlying graph and, more specifically, its connectivity. The complexity of approximate schemes may, in addition, depend on factors like the a-priori likelihood of the observed evidence or asymmetries in probability distributions. There are a number of ingeniously efficient algorithms that allow for fast belief updating in moderately sized models.[2] Still, each of them is subject to the growth in complexity that is generally exponential in the size of the model.

Belief updating algorithms can be enhanced by schemes based on relevance. Relevance reasoning in Bayesian networks can be used to improve the efficiency of belief updating algorithms by identifying and pruning those parts of a network that are irrelevant for the computation. This approach helps to reduce the size and the connectivity of the network. Relevance reasoning is based on the graphical property of *d*-separation and other simple and efficient techniques, the computational complexity of which is usually negligible when compared to the complexity of belief updating in general. Relevance reasoning is always conducted with respect to a set of nodes of interest, that we will call subsequently *target nodes*. Target nodes are all nodes whose posterior probability will be queried by the user. For example, in a medical decision support system, target nodes may be all disease nodes, as the user may be only interested in how likely these diseases are given observed evidence (i.e., symptoms and/or test results). In addition to removing computationally irrelevant nodes that are probabilistically independent from the target nodes $\mathcal{T}$ given the observed evidence $\mathcal{E}$, relevance-based methods can also remove passively relevant nodes, for example, nuisance nodes (Suermondt 1992; Druzdzel and Suermondt 1994). Furthermore, the technique called *relevance-basc decomposition* proposed in (Lin and Druzdzel 1997) decomposes networks into partially overlapping subnetworks by focusing on their parts, then updates beliefs in each subnetwork. This technique makes reasoning in some intractable networks possible and often results in significant speedup. In this paper, we show the speedup in belief updating of a new technique which is dealing with a different situation: when belief updating is interleaved with evidence gathering. We would like to make it clear that the current technique is significantly different from the previously proposed methods and it can be combined with them for a further speed-up of

[2] For an overview of various exact and approximate approaches to algorithms in BBNs see (Henrion 1990).

belief updating.

Some decision support systems based on graphical probabilistic models are used in environments where evidence is collected gradually rather than coming all at once and is interleaved with belief updating. It is desirable in such systems to incrementally update beliefs rather than recomputing the posterior probability distribution over all nodes. In this paper, we describe a belief updating technique based on relevance reasoning that is applicable in such systems. Our technique, called *relevance-based incremental updating*, is based on invalidating the posterior beliefs of those nodes that depend probabilistically on the new evidence. The result of previous computations remain valid for the rest of the network. Subsequent belief updating focuses on updating those nodes whose beliefs are invalid. In most reasonably sparse topologies, only a small fraction of the beliefs are invalidated. The sub-networks that need updating, as determined by our scheme, can be significantly smaller than the entire network and our scheme can then lead to substantial savings in computation. We demonstrate empirically that our scheme can lead to significant speedups in large practical models even in the clustering algorithm (Lauritzen and Spiegelhalter 1988; Jensen *et al.* 1990), that is believed to be the best suited for sequential evidence processing.

Incremental Updating

Incremental updating that we implemented in our framework is a practical application of the lazy evaluation principle. Each piece of evidence can be viewed as invalidating some of the previously computed marginal distributions (we compute the marginal probabilities of all nodes in the network beforehand), namely those to which it is relevant. Each network node in our system is equipped with a flag *valid* that is set to the value **true** when the node's marginal probability is computed and set to the value **false** when it is invalidated by a new piece of evidence. Invalidating the distribution is based on the condition of *d*-separation — a marginal distribution of a node is invalid if the observed evidence node is not *d*–separated from it given the previously observed evidence. Given a subsequent query, the system excludes from the computation those target nodes whose marginal probability distributions are still valid (i.e., those for which *valid*=**true**). In addition, the system does not recompute the distributions of nodes that are invalid but are not needed in updating the distributions of the target nodes that need updating. The algorithm used in incremental updating is listed in Figure 1.

A decision support system based on this scheme can improve its reactivity by producing the requested answer in a much shorter time if this answer is available (note that generally not all nodes in the target set are invalidated by every new piece of evidence). The system can also update its beliefs in a generally shorter time by focusing only on invalidated part of the network.

```
Given: A Bayesian belief network net,
          a set of target nodes $\mathcal{T}$,
          a set of evidence nodes $\mathcal{E}$,
          new evidence node e.
          Each node has a flag valid that is true
            if the node's marginal distribution
            is valid and false otherwise.
void New_Evidence(net, $\mathcal{E}$, e)
     For all nodes n that are not d–separated
         from e by $\mathcal{E}$,
       perform n.valid=:false .
  end
void Incremental_Update(net, $\mathcal{T}$, $\mathcal{E}$)
     Construct a set $\mathcal{T}'$:
       by removing from $\mathcal{T}$ all nodes t
       such that t.valid=true .
     Using relevance reasoning, remove from net all
       nodes that irrelevant to updating $\mathcal{T}'$ given $\mathcal{E}$
  end
main()
   Construct a junction tree $\mathcal{J}$ for net.
   Initialize the set $\mathcal{E}'$ to be empty.
   For each e in $\mathcal{E}$
     New_Evidence(net, $\mathcal{E}'$, e);
     Add e to $\mathcal{E}'$;
     Incremental_Update(net, $\mathcal{T}$, $\mathcal{E}'$);
     If (predicted cost for inference on pruned net
         > the cost for incremental updating on $\mathcal{J}$
       Perform belief updating on $\mathcal{J}$.
     otherwise
       Perform inference on net.
  end
```

Figure 1: The algorithm for relevance-based incremental updating.

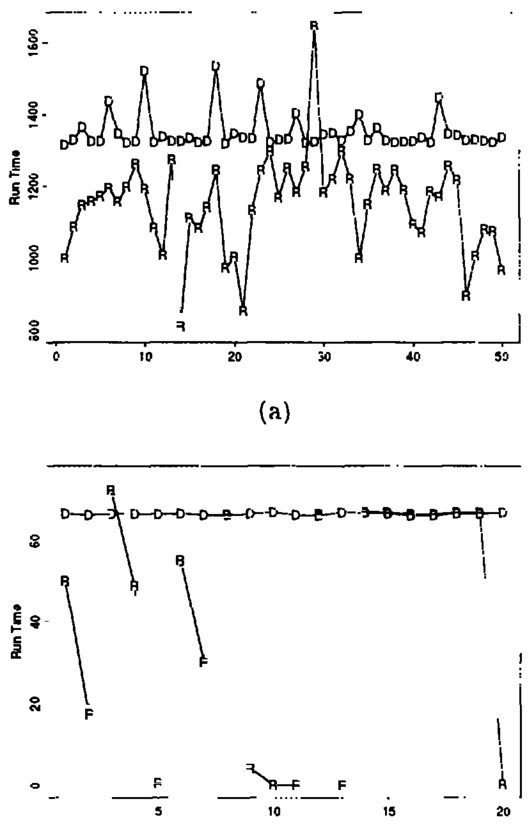
It is believed that environments in which evidence is gathered incrementally are particularly well supported by clustering algorithms (e.g., (Zhang and Poole 1996)). While we agree with this statement, we believe that this can be enhanced even more by the incremental updating scheme proposed above. Maintaining the validity flags and pruning parts of the network before the real inference adds very little overhead, while it can substantially reduce the size and the connectivity of the network, hence reduce the computational complexity. The cost that has to be paid for using this scheme is the need for recompiling the relevant sub-network into a clique tree each time computation needs to be performed. However, we can reasonably predict this cost with a very fast (not necessarily optimal) triangulation algorithm guided by a simple heuristic. That is, we can predict the computational complexity of inference on the pruned sub-network by the number of potentials generated from a triangulation algorithm. If the predicted cost plus the overhead of relevance reasoning outweight the complexity of incremental evidence updating on the original junction tree (which was constructed and saved initially), we simply discard the relevant sub-network and reason on the original junction tree instead. The overhead introduced by relevance reasoning and simple triangulation is very small, and can often pay off by a very fast inference on the resulting pruned sub-network.

Empirical Results

In this section, we present the results of an empirical test of relevance-based incremental updating for Bayesian belief network inference. We focused our tests on the enhancement to incremental updating in the clustering algorithm. The clustering algorithm that we used in all tests is an efficient implementation that was made available to us by Alex Kozlov. See Kozlov and Singh (1996) for the details of the implementation. We have enhanced Kozlov's implementation with relevance techniques described in (Lin and Druzdzel 1997) except for the *relevance-base decomposition*. We tested our algorithms using the CPCS network, a multiply-connected multi-layer network consisting of 422 multi-valued nodes and covering a subset of the domain of internal medicine (Pradhan *et al.* 1994). Among its 422 nodes, 14 nodes describe diseases, 33 nodes describe history and risk factors, and the remaining 375 nodes describe various findings related to the diseases. The CPCS network is among the largest real networks available to the research community at present time. Our computer (a Sun Ultra-2 workstation with two 168Mhz UltraSPARC-1 CPU's, each CPU has a 0.5MB L2 cache, the total system RAM memory of 384 MB) was unable to load, compile, and store the entire network in memory and we decided to use a subset consisting of 360 nodes generated by Alex Kozlov for earlier benchmarks of his algorithm. This network is a subset of the full 422 node CPCS network without predisposing factors (like gender, age, smoking, etc.). This reduction is realistic, as history nodes can usually be instantiated and absorbed into the network following an interview with a patient. We updated the marginal probabilities of all nodes in this model, i.e., all nodes in the networks were treated as target nodes.

We constructed 50 test cases, each of which consists of twenty randomly generated evidence nodes from among the finding nodes defined in the network. For each of the test cases, we first constructed a junction tree for the whole network and computed the prior probability distributions. Then we recorded the time for belief updating when each piece of evidence was entered and, at the end, computed the total time for 20 sequential evidence processing interleaved with belief updating by adding these times. We compared the relevance-based incremental updating with the incremental updating directly on the original junction tree. In case of relevance-based incremental updating, when each piece of evidence came in, we (1) ran the relevance-based incremental updating algorithm to obtain a pruned relevant subnet, (2) predicted the size of the junction tree for this subnet (in terms of the number of potentials in the junction tree), compared it with the size of the original junction tree, (3) if the estimated time for updating on the subnet was less than the time needed to update on the original junction tree, we ran clustering algorithm on the subnet, otherwise, we discarded the subnet and ran the incremental updating on the original junction tree.

Our earlier experiments had shown that it takes roughly three times less to update beliefs on an existing junction tree than to build a junction tree. We used this finding in estimating the updating time. Our simple heuristic used in the tests was to continue with relevance-based incremental updating only when the predicted number of potentials generated from an identified subnetwork was less than one third of the number of potentials in the original junction tree. This heuristic can be tuned up for the best performance in individual networks.



(a)

(b)

Figure 2: Comparison of the performance of relevance-based incremental belief updating (which "R" stands for) with plain clustering algorithm (which "D" stands for). (a) The total computing time for each of the 50 test cases and (b) time series of one case with 20 incrementally coming evidence.

The results of our tests are presented in Figure 2 with the summary data in Table 1. Note that relevance-based incremental belief updating introduces negligible overhead and practically always leads to faster belief updating than the plain clustering algorithm. There is only one case of an outlier in Figure 2–a. It is apparent that the relevance-based schemes in combination with the clustering algorithm performed on average 15% faster than direct incremental updating using clustering algorithm. The individual case in Figure 2–b shows that the overhead of relevance-based schemes is almost negligible, even when the belief updating is not chosen on the resulting relevant subnets (or the subnets are not small enough). But a few big savings from inference on small relevant subnets improves the overall performance. A question that one might ask is whether conditional dependencies introduced by multiple observations will enhance or reduce the benefits of relevance-based incremental updating. We performed tests that aimed at investigating this question but we did not find any evidence for the influence of the amount of evidence on the performance of the algorithm.

	Relevance	Direct
μ	1145.22	1349.66
σ	135.18	50.40
Min	810.22	1315.09
Median	1167.93	1329.07
Max	1647.25	1534.18

Table 1: Summary simulation results for the CPCS network, $n = 50$.

In addition to the CPCS network, we tested the relevance-based incremental updating algorithm on several other Bayesian networks. One of these was a randomly generated highly connected network A.ergo (Kozlov and Singh 1996). Summary results of this test are presented in Table 2. Here, the savings introduced by our scheme were even larger.

	Relevance	Direct
μ	188.23	344.83
σ	63.26	16.57
Min	90.19	325.66
Median	180.85	342.16
Max	341.11	425.12

Table 2: Summary simulation results for the A.ergo network, $n = 50$.

Discussion

In this paper, we introduced an incremental belief updating technique based on relevance reasoning that is applicable in systems in which evidence is collected gradually in different phases of interaction with the system and interleaved with belief updating. Our technique, called *relevance-based incremental updating*, is based on invalidating the posterior beliefs of those nodes that depend probabilistically on the new evidence. Subsequent belief updating focuses on updating those target nodes whose beliefs are invalid. Our algorithm identifies the smallest subnetwork that is relevant to those target nodes that need updating, predicts the cost of inference on the identified subnetwork, and then decides whether to perform inference on this subnetwork or to perform incremental belief updating on the original junction tree. Because the complexity of relevance algorithms is linear in the number of arcs in the network (Geiger *et al.* 1990; Druzdzel and Suermondt 1994; Lin and Druzdzel 1997), our scheme can predict its speed at almost no cost. When applied, it obtains significant gains by reducing the size and the connectivity of the network. In those cases where no target nodes are influenced by the new evidence, the answer may be available with no computation. Even in case when the new evidence invalidates all target nodes, the cost for predicting the efficiency of inference on the network is negligible with a fast triangulation algorithm. It is always possible to switch back to the incremental updating on the original junction tree. The relevance-based incremental belief updating improves system reactivity on average. Our scheme can also easily enhance approximation algorithms, as the pruned network almost always smaller than the original one. Of course, all relevance-based schemes are sensitive to the topology of networks and their performance can deteriorate in the worst case.

D'Ambrosio (1993) pointed out that there are three types of incrementality of inference: *query incrementality*, *evidence incrementality*, and *representation incrementality*. The first two are naturally built into our scheme. The third type of incrementality involves interleaving inference within a partial problem representation with representation extension operations. We have built in into our system also this type of incrementality. When a part of our network is modified, the modification also leads to invalidating those parts of the model that are not d–separated from the modified part. Incrementality with respect to representation extension enables a system to reuse results from prior computations even when the representation on which those computations are based is modified between queries.

Acknowledgments

This research was supported by the National Science Foundation under Faculty Early Career Development (CAREER) Program, grant IRI-9624629, by the Air Force Office of Scientific Research under grant number F49620–97–1–0225, and by ARPA's Computer Aided Education and Training Initiative under grant number N66001-95-C-8367. We are grateful to Alex Kozlov for making his implementation of the clustering algorithm and several of his benchmark networks available to us. Malcolm Pradhan and Max Henrion of the Institute for Decision Systems Research shared with us the CPCS network with a kind permission from the developers of the Internist system at the University of Pittsburgh. We are indebted to Jaap Suermondt for insightful suggestions. We also thank the anonymous reviewer for

helpful comments.

References

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

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

Bruce D'Ambrosio. 1993. Incremental probabilistic inference. In *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI 93)*, pages 301–308, Washington, D.C.

Marek J. Druzdzel and Henri J. Suermondt. 1994 Relevance in probabilistic models: "Backyards" in a "small world". In *Working notes of the AAAI-1994 Fall Symposium Series: Relevance*, pages 60–63, New Orleans, LA (An extended version of this paper is in preparation.).

Dan Geiger, Thomas S. Verma, and Judea Pearl. 1990. Identifying independence in Bayesian networks. *Networks*, 20(5):507–534.

Max Henrion. 1990. An introduction to algorithms for inference in belief nets. In M. Henrion, R.D. Shachter, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 5*, pages 129–138. Elsevier Science Publishers B.V., North Holland.

Ronald A. Howard and James E. Matheson. Influence diagrams. In Ronald A. Howard and James E. Matheson, editors, *The Principles and Applications of Decision Analysis*, pages 719 762. Strategic Decisions Group, Menlo Park, CA, 1984.

Finn Verner Jensen, Kristian G. Olesen, and Stig Kjær Andersen. 1990. An algebra of Bayesian belief universes for knowledge-based systems. *Networks*, 20(5):637 659.

Alexander V. Kozlov and Jaswinder Pal Singh. 1996. Parallel implementations of probabilistic inference. *IEEE Computer*, pages 33–40.

Steffen L. Lauritzen and David J. Spiegelhalter. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B (Methodological)*, 50(2):157–224.

Yan Lin and Marek J. Druzdzel. 1997. Computational advantages of relevance reasoning in Bayesian belief networks. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-97)*, pages 342–350, San Francisco, CA. Morgan Kaufmann Publishers, Inc.

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

Malcolm Pradhan, Gregory Provan, Blackford Middleton, and Max Henrion. 1994. Knowledge engineering for large belief networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–94)*, pages 484–490, Seattle, WA.

Henri J. Suermondt. 1992. *Explanation in Bayesian Belief Networks.* PhD thesis, Department of Computer Science and Medicine, Stanford University, Stanford, CA.

Nevin Zhang and David Poole. 1996. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301 328.