RELEVANCE-BASED INCREMENTAL BELIEF UPDATING IN BAYESIAN NETWORKS

YAN LIN and MAREK J. DRUZDZEL

Decision Systems Laboratory, School of Information Sciences and Intelligent Systems Program, University of Pittsburgh Pittsburgh, PA 15260, USA
E-mail: {yan,marek}@sis.pitt.edu

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 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 and model revisions are interleaved with belief updating. Our technique invalidates the posterior beliefs of those nodes that depend probabilistically on the new evidence or the revised part of the model and focuses the subsequent belief updating on the invalidated beliefs rather than on all beliefs. Very often observations and model updating invalidate only a small fraction of the beliefs and our scheme can then lead to sub stantial savings in computation. We report results of empirical tests for incremental belief updating when the evidence gathering is interleaved with reasoning. These tests demonstrate the practical significance of our approach.

Keywords: Bayesian networks, belief updating, relevance.

1. INTRODUCTION

The emergence of probabilistic graphs, such as Bayesian networks (BNs)[13] and closely related influence diagrams[7] 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[1] and for approximate inference.[2] 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.[a] 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

[a]For an overview of various exact and approximate approaches to algorithms in BNs, see Ref. 6.

International Journal of Pattern Recognition and Artificial Intelligence Vol. 13 No. 2 (1999) 285–295

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 ε, relevance-based methods can also remove passively relevant nodes, e.g. *nuisance nodes*.[4,15] Only the prior distributions of *nuisance nodes* are needed for the computation. Furthermore, the technique called *relevance-based decomposition* proposed in Ref. 11 decomposes networks into several partially overlapping subnetworks by focusing on their parts, then updates beliefs in each subnetwork, where all subnetworks combined cover the entire network. This technique makes reasoning possible in some intractable networks and often results in significant speedup, because subnetworks can be significantly smaller than the original network. In this paper, we show the speedup in belief updating due to a new technique, called *relevance-based incremental belief updating*, which is dealing with a different situation — when belief updating is interleaved with evidence gathering and model revision. 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 speedup of belief updating.

Some decision support systems based on graphical probabilistic models are used in environments where evidence is collected gradually rather than in bulk and is interleaved with belief updating. Evidence can be of various types, including new observations, retraction of previous observations, or changes in the previously observed values. Furthermore, the users may wish to revise the model by adding new variables, by removing variables or arcs, or by modifying the conditional probability tables. 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 or the changes to the model. The results 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 subnetworks 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,[8,10] that is believed to be the best suited for sequential evidence processing.

2. SEQUENTIAL EVIDENCE PROCESSING

Incremental updating that we implemented in our framework is a practical application of the lazy evaluation principle.

Each piece of evidence, including retraction or changes of previously observed values, can be viewed as invalidating some of the previously computed marginal distributions (we compute the marginal probabilities of all nodes in the network beforehand), viz. 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. Similarly, each modification to the model invalidates those beliefs that were based on the modified part of the model. We will discuss the model modification in the next section, focusing here on sequential evidence processing. 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 relevance-based incremental updating is listed in Fig. 1.

```
Given: A Bayesian belief network net,
        a set of target nodes T, a set of evidence nodes 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, E, e)
    For all nodes n that are not d-separated from e by E,
      perform n.valid=:false .
  end
void Incremental_Update(net, T, E)
    Construct a set T' by removing from T all nodes t
      such that t.valid=true .
    Using relevance reasoning, remove from net all nodes
      irrelevant to updating T' given E
  end
main()
  Construct a junction tree J for net.
  Initialize the set E' to be empty.
  For each e in E
    New_Evidence(net, E', e);
    Add e to E';
    Incremental_Update(net, T, E');
    If (predicted cost for inference on pruned net
        is larger than the cost for incremental updating on J
      Perform belief updating on J.
    otherwise
      Perform inference on net.
end
```

Fig. 1. The algorithm for relevance-based incremental updating.

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 the invalidated part of the network.

It is believed that environments in which evidence is gathered incrementally are particularly well supported by clustering algorithms (e.g. Ref. 16). 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 add very little overhead, while the scheme 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 subnetwork 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 subnetwork by the number of potentials generated from a triangulation algorithm. If the predicted cost plus the overhead of relevance reasoning outweigh the complexity of incremental evidence updating on the original junction tree (which was constructed and saved initially), we simply discard the relevant subnetwork 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 subnetwork.

3. REPRESENTATION INCREMENTALITY

D'Ambrosio[3] pointed out that there are three types of incrementality of inference: *query incrementality, evidence incrementality, and representation incrementality.* The first two are related to sequential evidence processing. When a new query comes, we can first check the validity flag of the corresponding node. If it is not valid, we can apply the relevance-based incremental updating scheme in pretty much the same way as we do when a new evidence enters. The third type of incrementality involves interleaving inference within a partial problem representation with representation extension operations (i.e. model revisions). We have built into our system also this type of incrementality with minor modifications to the algorithm presented in Sec. 2. When a part of our network is modified, we invalidate those beliefs in the model that were based on the modified part. For example, removing or adding a node involves removing or adding its incoming and outgoing arcs. Removal or addition of an arc will, in turn, change the conditional probability table of the node that the arc points to, and this change invalidates all correspondingly relevant nodes. When the conditional probability table of a node is modified, the marginal distributions of all the children of this node will be invalidated, so will all other nodes that are relevant to these children nodes given existing evidence. In addition, if this node has evidence nodes as its descendants, then the marginal distributions of all

```
void Remove_Arc(net, E, a)
    n=:the node which arc a points to
    n.valid=false .
    Change_CPT(net, n, E)
    Remove a.
end
void Change_CPT(net, n, E)
    For every child c of node n:
      c.valid=:false .
      For every node m in net which is relevant to c given E
        m.valid=false .
    If n or its descendants belong to E
    Then for every parent p of node n:
        p.valid=:false .
        For every node m in net which is relevant to p given E
          m.valid=false .
    end
    Modify the CPT of n.
end
```

Fig. 2. The algorithms for incremental network modification.

its parents and their relevant nodes will be invalidated as well. The algorithms are listed in Fig. 2. In networks containing deterministic nodes, a simple patch to the algorithm is to first find out which nodes are functionally determined,[5] add them to the evidence set. 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.

Furthermore, the incremental scheme can be extended to influence diagrams. Influence diagrams are extensions of Bayesian networks that include decision and value (utility) nodes. The result contained in a value node is valid when the computed expect utility of every decision option is valid. Validity of decision nodes depends on validity of value nodes. That is, any changes which impact value nodes will invalidate decision nodes if they are not d-separated from the value nodes. The *relevance-based invalidation* algorithm should be modified for influence diagrams as follows: (1) ignore all incoming and out-going arcs of decision nodes; (2) run relevance-based invalidation when any changes occur; if a decision node is set to an option, then start the invalidation from its children; (3) if value nodes are not impacted, terminate; otherwise, restore the links between decision nodes and their children, find those decision nodes that are not d-separated from the impacted value nodes, and invalidate them.

4. EMPIRICAL RESULTS

In this section, we present the results of an empirical test of relevance-based incremental updating for Bayesian network inference. Our tests are based on the enhancement to incremental updating in the clustering algorithm. Since model revision invalidates the junction tree, clustering algorithm cannot naturally support representation incrementality. The relevance-based scheme will always outperform a plain belief updating scheme, because it updates beliefs on smaller subnetworks. Therefore, tests of relevance-based processing of model revision will necessarily come out favorably to our approach. Similarly, when a small set of nodes in the network is designated as targets, our scheme will outperform the clustering algorithm, because it is able to remove many nodes that do not need to be updated. We focus our tests on sequential evidence processing in the context of updating the entire network.

The clustering algorithm that we used in all tests is an efficient implementation that was made available to us by Alex Kozlov. See Ref. 9 for the details of the implementation. We have enhanced Kozlov's implementation with relevance techniques described in Ref. 11 except for the *relevance-base decomposition*. We tested our algorithms using the CPCS network, a multiply-connected multilayer network consisting of 422 multivalued nodes and covering a subset of the domain of internal medicine.[14] 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 consisted of 20 pieces of randomly generated evidence from among the finding nodes defined in the network. We restricted our tests to observing one new piece of evidence each time, assuming persistence at the previously observed evidence. This assumption does not have much impact on the generality of our results as retracting or changing evidence in a node invalidates precisely the same set of nodes as observing a value at this node does. We also assumed that the order of evidence, which of course can affect performance, is given externally by the user; thus, in return, mimics the diagnosis process. 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 pieces of sequential evidence processing interleaved with belief updating by adding up 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.

The results of our tests are presented in Fig. 3 with the summary data in Table 1. Note that with one exception (Fig. 3) relevance-based incremental belief updating was faster than the plain clustering algorithm (the means are different due to chance at $p < 1.85 * 10^{-14}$). 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 cases in Figs. 4(a) and 4(b) show 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

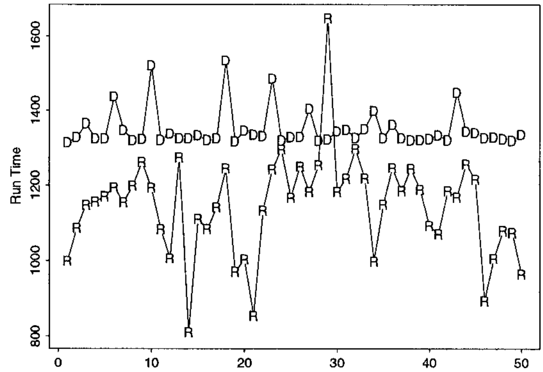


Fig. 3. Comparison of the performance of relevance-based incremental belief updating (which "R" stands for) with plain clustering algorithm (which "D" stands for). The total computing time for each of the 50 test cases.

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

	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

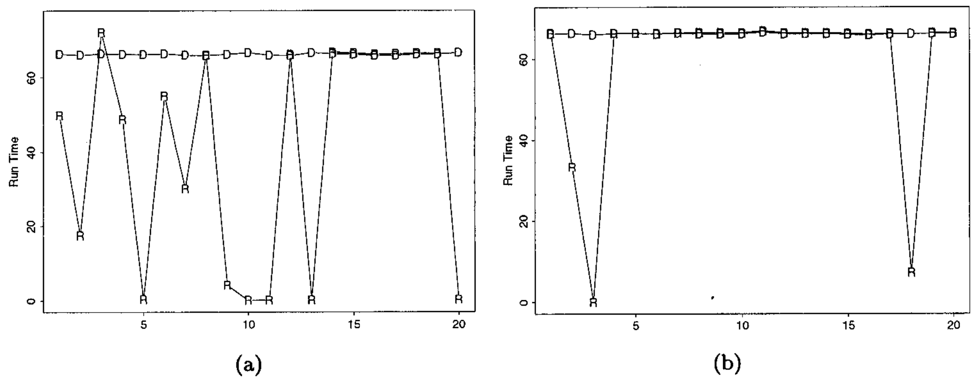


Fig. 4. Comparison of the performance of relevance-based incremental belief updating (which "R" stands for) with plain clustering algorithm (which "D" stands for). (a) Time series of a good case with 20 pieces of incrementally coming evidence. (b) Time series of a bad case with 20 pieces of incrementally coming evidence.

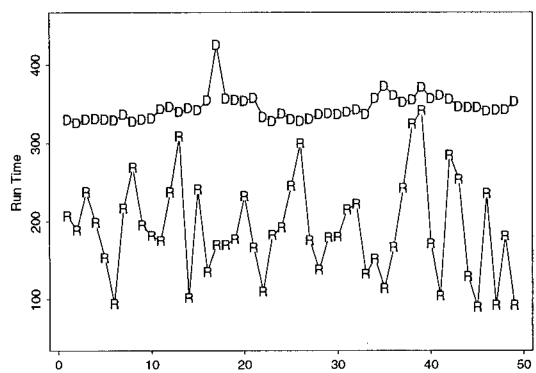


Fig. 5. The total computing time for each of the 50 test cases.

whether conditional dependencies introduced by multiple observations will enhance or reduce the benefits of relevance-based incremental updating. We run the test with as many as 40 evidence nodes, but observed essentially the same performance and the same advantage of relevance-based incremental updating algorithm over clustering algorithm. We report tests with 20 evidence in this paper, because we believed that more than 20 findings would not be realistic.

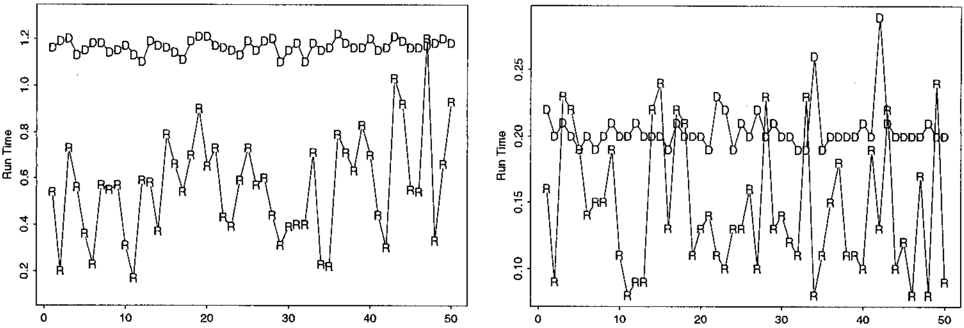
In addition to the CPCS network, we tested the relevance-based incremental updating algorithm on several other Bayesian networks. One of these was a randomly generated highly connected network A.[9]

The results of our tests are presented in Fig. 5 with the summary in Table 2. Here, the savings introduced by our scheme were even larger.

Another two networks we tested were Hailfinder and Imports. The results are presented in Fig. 6. As the belief updating is fast over small networks anyway, the advantage of relevance-based scheme becomes less evident.

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

	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



(a) (b)

Fig. 6. The total computing time for each of the 50 test cases. (a) Network Hailfinder with 57 nodes. (b) Network Imports with 15 nodes.

5. DISCUSSION

In this paper, we introduced an incremental belief updating technique based on relevance reasoning that is applicable in systems in which model revision is interleaved with computation or 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 changes of 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. Since the complexity of the relevance algorithms is linear in the number of arcs in the network,[4,5,11] 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 changes of evidence, the answer may be available with no computation. Even in case the changes of evidence invalidate 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. But if all the observations are available at the same time, it would not be more efficient to apply them one by one using the proposed algorithm. Our scheme can also easily enhance approximation algorithms, as the pruned network is 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 very densely connected networks.

All methods described in this paper have been implemented in SMILE (Structural Modeling, Inference, and Learning Environment) and its Windows user interface, GeNIe, both available free of charge for research, teaching, and personal use at Decision Systems Laboratory's WWW site at http://www2.sis.pitt.edu/~genie.

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. An early version of this paper was presented in the Eleventh International Florida Artificial Intelligence Research Symposium (FLAIRS–98).[12]

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 kind permission from the developers of the Internist system at the University of Pittsburgh. We are indebted to Jaap Suermondt for his insightful suggestions and Daniel Garcia for helpful discussions.

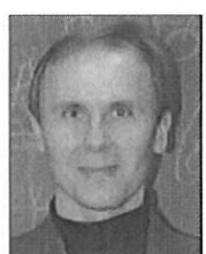
REFERENCES

1. G. F. Cooper, "The computational complexity of probabilistic inference using Bayesian networks," *Artif. Intell.* **42** 2–3 (1990) 393–405.
2. P. Dagum and M. Luby, "Approximating probabilistic inference in Bayesian networks is NP-hard," *Artif. Intell.* **60** 1 (1993) 141–153.
3. B. D'Ambrosio, "Incremental probabilistic inference," *Proc. Ninth Annual Conf. Uncertainty in Artificial Intelligence* (*UAI-93*), Washington, D.C, 1993, pp. 301–308,
4. M. J. Druzdzel and H. J. Suermondt, "Relevance in probabilistic models: 'Backyards' in a 'small world'," *Working Notes of the AAAI-1994 Fall Symp. Series: Relevance*, New Orleans, LA, 1994, pp. 60–63. (An extended version of this paper is in preparation.)
5. D. Geiger, T. S. Verma and J. Pearl, "Identifying independence in Bayesian networks," *Networks* **20**, 5 (1990) 507–534.
6. M. Henrion, "An introduction to algorithms for inference in belief nets," *Uncertainty in Artificial Intelligence 5*, eds. M. Henrion, R. D. Shachter, L. N. Kanal and J. F. Lemmer, Elsevier Science Publishers B.V., North Holland, 1990, pp. 129–138.
7. R. A. Howard and J. E. Matheson, "Influence diagrams," *The Principles and Applications of Decision Analysis*, eds. R. A. Howard and J. E. Matheson, Strategic Decisions Group, Menlo Park, CA, 1984, pp. 719–762.

8. F. V. Jensen, K. G. Olesen and S. Andersen, "An algebra of Bayesian belief universes for knowledge-based systems," *Networks* **20**, 5 (1990) 637–659.
9. A. V. Kozlov and J. P. Singh, "Parallel implementations of probabilistic inference," *IEEE Comput.* **29**, 12 (1996) 33–40.
10. S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *J. R. Stat. Soc. Series B (Methodological)* **50**, 2 (1988) 157–224.
11. Y. Lin and M. J. Druzdzel, "Computational advantages of relevance reasoning in Bayesian belief networks," *Proc. Thirteenth Annual Conf. Uncertainty in Artificial Intelligence (UAI-97)*, San Francisco, CA. Morgan Kaufmann Publishers, 1997, pp. 342–350.
12. Y. Lin and M. J. Druzdzel, "Relevance-based sequential evidence processing in Bayesian belief networks," *Proc. Uncertain Reasoning in Artificial Intelligence Track of the Eleventh Int. Florida Artificial Intelligence Research Symp. (FLAIRS-98)*, Sanibel Island, FL, 1998, pp. 446-450.
13. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, San Mateo, CA, 1988.
14. M. Pradhan, G. Provan, B. Middleton and M. Henrion, "Knowledge engineering for large belief networks," *Proc. Tenth Annual Conf. Uncertainty in Artificial Intelligence (UAI-94)*, Seattle, WA, 1994, pp. 484–490.
15. H. J. Suermondt, *Explanation in Bayesian Belief Networks*, PhD Thesis, Department of Computer Science and Medicine, Stanford University, Stanford, CA, 1992.
16. N. Zhang and D. Poole, "Exploiting causal independence in Bayesian network inference," *J. Artif. Intell. Res.* **5** (1996) 301–328.

Yan Lin received the B. S. degree in computer science form Nanjing University, People's Republic of China and the M.S. degree in intelligent system from the University of Pittsburgh, USA. She is currently a doctoral student in the Intelligent Systems Program, University of Pittsburgh.

Her research interests include reasoning under uncertainty and decision-theoretic methods in intelligent systems.



Marek J. Druzdzel is an assistant professor in the Department of Information Science, Intelligent Systems Program, and Medical Informatics Training Program at the University of Pittsburgh. He received the M.S. degree in computer science (1985) and electrical engineering (1987) from Delft University of Technology in The Netherlands (both with Distinction) and his Ph.D. in engineering and public policy (1992) from Carnegie Mellon University, Pittsburgh, USA.

His research interests concentrate on decision-theoretic reasoning in decision support systems and human interfaces to decision support systems. More details about his research interests and publications are available on line at http://www.pitt.edu/~druzdzel.