Local Monotonicity in Probabilistic Networks

Johan Kwisthout Hans Bodlaender Gerard Tel

Department of Information and Computer Sciences, University of Utrecht,
P.O. Box 80.089, 3508 TB Utrecht

The full version of this extended abstract appeared in the proceedings of the Ninth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-07), Hammamet, Tunisia, 2007 (LNAI 4724, pp. 548-559). The first author was (partly) supported by the Netherlands Organisation for Scientific Research (NWO).

1 Introduction

In many probabilistic networks [3] that are used for classification in real problem domains, the variables of the network can be distinguished into observable input variables, non-observable intermediate variables, and a single output variable. For example, in a medical domain the observable variables represent clinical evidence such as symptoms and test results, the output variable functions as a classification of a disease, and the intermediate variables model non-observable facts that are relevant for classification. Often, the relations between observable symptoms and the classification variable are monotone, e.g., higher values for a particular observable variable makes higher values of the classification variable more likely, independent of the value of other observations. Such a network is *monotone in distribution* [4] if higher-ordered configurations of the observable variables make higher-ordered outputs more (isotone) or less (antitone) likely.

When a domain expert insists that a certain relation ought to be monotone, the joint probability distribution should be such, that this property is reflected in the network [5]. If monotonicity is violated, the probability distribution in the network can be revised in cooperation with the expert. Unfortunately, determining whether a network is monotone is highly intractable [4]. One approach to overcome this unfavorable complexity, is by approximating the decision (i.e, sometimes have 'undecidable' as outcome) like the algorithm discussed in [4]. This algorithm uses qualitative influences (QPNs, see e.g. [7]) that summarize the direction of the influence of variables by signs. However, the use of these signs of course requires an ordering on the values of the variables under consideration.

Such an ordering might be implicit, for example *large* $>$ *medium* $>$ *small* or *true* $>$ *false*. But in practice, there are often variables in a network which do not have such 'natural' orderings, e.g., in a network in the medical domain, there may be a variable which denotes the organs infected by a particular disease. As it is desirable to have as many as possible monotone influences (to minimize the offending context), it is important to *choose* an optimal ordering for the values of these variables. Such an optimal ordering maximises the number of monotone arcs, or, equivalently, minimizes the number of '?' signs in the corresponding QPN.

2 Complexity results

We introduce the notion of an *interpretation* of a variable to denote a particular ordering on the values of that variable, and *monotonicity functions* which determine whether a certain combination of interpretations for the two nodes of an arc make the arc isotone or antitone. We formalise the optimisation problem (finding an optimal ordering) and show that is is infeasible in general. In our paper, we prove NP-hardness with a reduction from GRAPH 3-COLORABILITY. Furthermore, we show that this problem is hard to approximate as well. We prove that the problem is APX-hard (see e.g. [2]), a very strong indicator that there exists no polynomial time approximation scheme (PTAS) for this problem, and that the problem can only be approximated within a fixed ratio. Since this ratio might be very large, an approximation algorithm might not be very useful to address this problem.

3 A Branch-and-bound algorithm

In our paper we sketch an exact algorithm for this problem, based on a so-called branch-and-bound strategy. In such a strategy, the set of possible solutions is partitioned (the branch step), and upper (or lower, for minimalization problems) bounds for this partition are calculated. Whenever these bounds are lower than or equal to the current best solution (i.e., further exploration of these branches will not lead to a better solution) the branch is terminated, and other, yet unvisited branches are explored. This procedure continues until all branches terminate (we can return an optimal solution), or a given ratio between current best solution and upper bound is reached (we can return a ’good enough’ solution).

There are many degrees of freedom in this branch-and-bound strategy; the performance of these algorithms is highly dependent on the tightness of the bounds and the choice of the node to branch on. In our paper, we chose to compute rather loose bounds; one can compute tighter bounds by considering more arcs that can be made monotone. Nevertheless, the constraints imposed by these arcs might require re-evaluation of all constraints in the network, so there is a tradeoff between the tightness of the bounds – and thus the number and depth of the branches – and the time needed to calculate such bounds.

4 Conclusion

Optimising the number of monotone arcs in a network is a computationally hard problem, and hard to approximate as well. We propose a branch-and-bound approach to calculate optimal orderings. This approach may work rather well in practice with ’real world’ networks, provided that the number of values per node is small. For example, in the ALARM network [1], the number of values is at most four, and in the OESOPHAGEAL network [6] it is at most six. However, for networks where some nodes have a large range of possible values, this approach will be infeasible, since there are $k!$ possible orderings for a variable with k values. Other methods must be used in such cases to calculate or approximate an optimal solution. These issues are subject to further research.

References

[1] I. Beinlich, G. Suermondt, R. Chavez, and G. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on AI and Medicine*. Springer-Verlag, 1989.

[2] P. Crescenzi and A. Panconesi. Completeness in approximation classes. *Information and Computation*, 93:241–262, 1991.

[3] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, Palo Alto, 1988.

[4] L. C. van der Gaag, H.L. Bodlaender, and A. Feelders. Monotonicity in Bayesian networks. In *Twentieth Conference on Uncertainty in Artificial Intelligence*, pages 569–576. AUAI Press, 2004.

[5] L. C. van der Gaag, P. L. Geenen, and H. Tabachneck-Schijf. Verifying monotonicity in Bayesian networks with domain experts. In L. C. van der Gaag and R. Almond, editors, *Proceedings of the 4th Bayesian Modelling Applications Workshop: Bayesian Models Meet Cognition*, pages 9–15, 2006.

[6] L. C. van der Gaag, S. Renooij, C. L. M. Witteman, B. M. P. Aleman, and B. G. Taa. Probabilities for a probabilistic network: a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25:123–148, 2002.

[7] M. P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303, 1990.