Bayesian network inference with simple propagation

David Quesada López

Computational Intelligence Group, Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Spain

Objectives

We propose *simple propagation*, a new algorithm for tree propagation in exact inference in discrete bayesian networks. It aims to:

- Outperform lazy propagation in terms of efficiency.
- Have a good performance in optimal join trees.

Introduction

Exact inference is a method used to answer queries asked to bayesian networks. One of the ways to go around exact inference is through jion trees. This trees are built from the DAG, and tipically at each node of the tree a potencial is computed by multiplying the probability tables inside it.

- Lazy propagation (LP) [1] keeps a multiplicative factorization of potentials at each node. This helps to identify irrelevant potencials at the moment of sending a message from a node. When it identifies the relevant potencials, LP builds a graph to determine elimination ordering of the variables.
- Simple propagation (SP) uses a "one in, one out" property at the moment of sending a message from a node to eliminate variables. This way, it works greedily towards the elimination of variables, not needing to build a graph to determine the elimination order

Materials

To conduct a comparison between the performance of LP and SP we used 28 benchmark bayesian networks. Then we generate 100 sets of evidence randomly and compute the message passing to calculate the posterior probabilities given the evidence for each non-evidence variable. Finally, we compare the average results in computation time of LP and SP.

Only for optimal join trees

Simple propagation performance significantly degrades in non-optimal join trees.

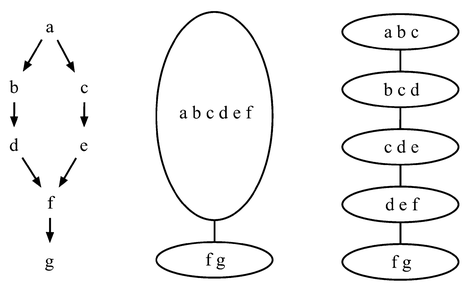

Figure 1: Non-optimal and optimal join trees of a BN

One in, one out

This process takes place in the join tree, at the time when we can start the message passing in a node. By “one in, one out”, we mean a potential in F has at least one non-evidence variable in the separator with other node and another non-evidence variable not in the separator. SP then eliminates each “out” variable.

Heuristics proposed

We propose some heuristics to select the order in wich we evaluate potentials satisfying the “one in, one out” property based on:

- Increasing number of variables in X.
- Decreasing number of variables in X.
- Increasing number of variables of both X and S.
- Decreasing number of variables of both X and S.

Results

Our study shows that SP is faster than LP in 18 cases, equal in 5 cases and slower in 5 cases. When SP is slower, further research showed that it is because LP has more elimination orderings to choose from, and when it is considerably better than the one SP uses, it is faster even though it spent more time building graphs to determine this ordering. This is also the case why LP is better for non-optimal join trees.

Heuristic	Wins
Arbitrary	6
Increasing X	1
Decreasing size S	1

Table 1: Heuristic selection results

We also tried our heuristics in selecting the order of evaluation of potencials: out of 29 BNs, heuristic and arbitrary order tied in 13 cases, arbitrary won in 6 and heuristic won in 2. This means that the selection of the order is not really relevant to SP.

Conclusion

- Simple propagation for exact inference performs in general better than lazy propagation when provided with an optimal join tree.
- It’s performance doesn’t seem to improve even with our proposed heuristics in optimal or non-optimal join trees.
- It is still an exact inference method, so its use is restricted to tractable bayesian networks.

Original paper

Butz C. J., Oliveira J. S., dos Santos A. E., Madsen, A. L. (2018). An empirical study of Bayesian network inference with simple propagation. *International Journal of Approximate Reasoning*, 92, 198-211.

References

[1] A.L. Madsen, F.V. Jensen, Lazy propagation: a junction tree inference algorithm based on lazy evaluation, Artif. Intell. 113 (1–2) (1999) 203–245

POLITÉCNICA