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* (sp), 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 joint trees.

Introduction

Exact inference is a method used to answer queries asked to a bayesian network. One of the ways to go around exact inference is through junction trees. The junction 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 [1] keeps a multiplicative factorization of potentials at each node. This helps to remove irrelevant potencials

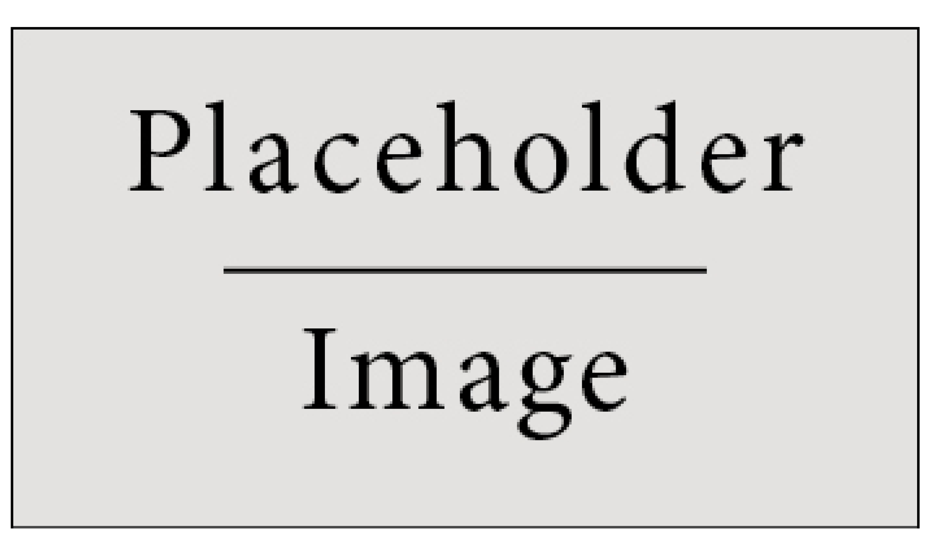

Figure 1: Figure caption

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. Then we compare the average results in computation time of LP and SP.

Only for optimal junction trees

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

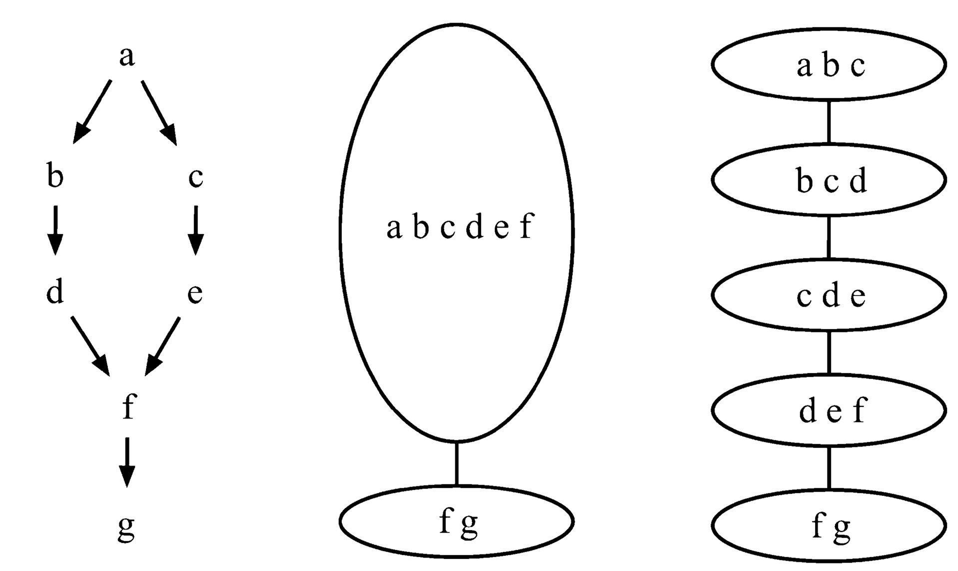

Figure 2: Non-optimal and optimal join trees of a BN

Mathematical Section

Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin. Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin.

$$E = mc^2 \tag{1}$$

Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin. Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin.

$$\cos^3\theta = \frac{1}{4}\cos\theta + \frac{3}{4}\cos 3\theta \tag{2}$$

Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin. Nam quis odio enim, in molestie libero. Vivamus cursus mi at nulla elementum sollicitudin.

Methods

Lorem ipsum dolor **sit amet**, consectetur adipiscing elit. Sed laoreet accumsan mattis. Integer sapien tellus, auctor ac blandit eget, sollicitudin vitae lorem. Praesent dictum tempor pulvinar. Suspendisse potenti. Sed tincidunt varius ipsum, et porta nulla suscipit et. Etiam congue bibendum felis, ac dictum augue cursus a. **Donec** magna eros, iaculis sit amet placerat quis, laoreet id est. In ut orci purus, interdum ornare nibh. Pellentesque pulvinar, nibh ac malesuada accumsan, urna nunc convallis tortor, ac vehicula nulla tellus eget nulla. Nullam lectus tortor, *consequat tempor hendrerit* quis, vestibulum in diam. Maecenas sed diam augue.

Results

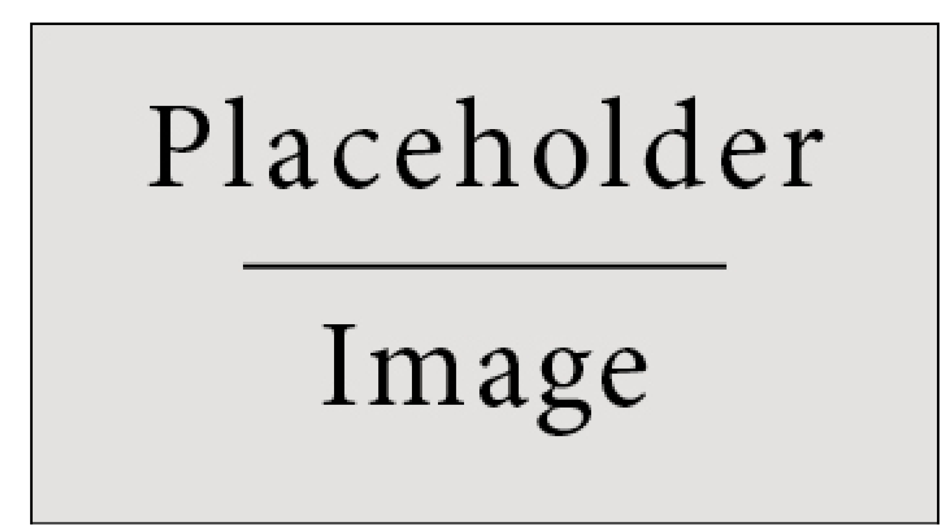

Figure 3: Figure caption

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.

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

Table 1: Table caption

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 non-optimal join trees.
- It is still an exact inference method, so its use is restricted to tractable bayesian networks.

Artículo real

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.

Referencias

[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

Nam mollis tristique neque eu luctus. Suspendisse rutrum congue nisi sed convallis. Aenean id neque dolor. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.