International Journal of Approximate Reasoning 92 (2018) 198–211

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

An empirical study of Bayesian network inference with simple propagation ☆

Cory J. Butz [a,*], Jhonatan S. Oliveira [a], André E. dos Santos [a], Anders L. Madsen [b,c]

[a] University of Regina, Department of Computer Science, Regina, S4S 0A2, Canada
[b] Aalborg University, Department of Computer Science, Aalborg, DK-9000, Denmark
[c] HUGIN EXPERT A/S, Aalborg, DK-9000, Denmark

ARTICLE INFO

Article history:
Received 30 September 2016
Received in revised form 14 August 2017
Accepted 2 October 2017
Available online xxxx

Keywords:
Bayesian networks
Exact inference
Join tree propagation

ABSTRACT

We propose *Simple Propagation* (SP) as a new join tree propagation algorithm for exact inference in discrete Bayesian networks. We establish the correctness of SP. The striking feature of SP is that its message construction exploits the factorization of potentials at a sending node, but without the overhead of building and examining graphs as done in *Lazy Propagation* (LP). Experimental results on optimal (or close to optimal) join trees built from numerous benchmark Bayesian networks show that SP is often faster than LP.

1. Introduction

Bayesian networks (BNs) [1] are an elegant approach to uncertainty management. By blending probability theory and graph theory, these probabilistic graphical models provide a rigorous foundation for making rationale decisions. A BN consists of a *directed acyclic graph* (DAG) and a set of *conditional probability tables* (CPTs) matching the structure of the DAG. Each node in the DAG represents a random variable in the problem domain being modeled, while the edges of the DAG encode probabilistic conditional independence information. Since the product of the CPTs is a joint probability distribution, sound inference algorithms answer a query by manipulating the CPTs to yield the same correct result that would have been obtained had the query been answered directly from the joint probability distribution.

Join tree propagation (JTP) is central to the theory and practice of probabilistic expert systems [2]. Here, exact inference in a discrete BN is conducted on a secondary structure, called a join tree, built from the directed acyclic graph of a BN. Even though the computational and space complexity of JTP is exponential in the tree-width of the network, in general, we care not about the worse case, but about the cases we encounter in practice [3]. For real-world BNs, several JTP approaches appear to work quite well. Given observed evidence, messages are systematically propagated such that posterior probabilities can be computed for every non-evidence variable. More specifically, each BN CPT, having been updated with observed evidence, is assigned to precisely one join tree node containing its variables. Classical JTP algorithms [2] form one potential per node by multiplying together the tables at each node.

☆ This paper is part of the Virtual special issue on Uncertainty Reasoning, Edited by Robert E. Mercer and Salem Benferhat.

* Corresponding author.
E-mail addresses: butz@cs.uregina.ca (C.J. Butz), oliveira@cs.uregina.ca (J.S. Oliveira), dossantos@cs.uregina.ca (A.E. dos Santos), anders@hugin.com (A.L. Madsen).

https://doi.org/10.1016/j.ijar.2017.10.005
0888-613X/

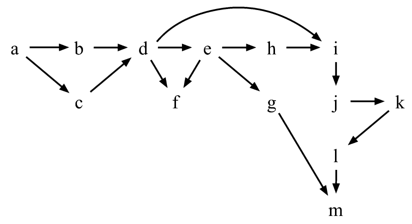

Fig. 1. A BN extended from [4].

In contrast, *Lazy Propagation* (LP) [4,5] keeps a multiplicative factorization of potentials at each node. This allows LP to remove two kinds of irrelevant potentials during message construction. Irrelevant potentials involving *barren variables* [4] are removed first from the factorization. Next, irrelevant potentials based on testing independencies induced by evidence are removed from the factorization. Here, a potential is irrelevant if and only if certain variables are separated in the moralization $\mathcal{G}_1^m$ of the *domain graph* [5] $\mathcal{G}_1$ built from the factorization. In the remaining relevant potentials, all variables not appearing in the separator need to be marginalized away. The order in which these variables are marginalized, called an *elimination ordering* [6], is determined by examining the moralization $\mathcal{G}_2^m$ of the domain graph $\mathcal{G}_2$ built from the factorization of relevant potentials. The resulting factorization is the message propagated.

In this paper, we propose *Simple Propagation* (SP) as a new JTP algorithm for exact inference in discrete BNs. SP consists of three steps. First, remove irrelevant potentials based on barren variables. Second, while the factorization at a sending node contains a potential with a non-evidence variable in the separator and another not in the separator, then the latter must be marginalized away. Third, propagate only those potentials exclusively containing variables in the separator. We establish the correctness of SP. Thus, SP is equivalent to LP, but without following key tenets of LP. SP never explicitly tests independencies, nor does it determine elimination orderings. This means SP saves the overhead of having to build and examine graphs. In experimental results on 28 benchmark cases, SP is faster than LP in 18 cases, ties LP in 5 cases, and is slower than LP in 5 cases.

This paper extends our seminal work on SP [7] in several ways. We establish that the relevant potentials in SP are exactly those in LP [8]. This is a stronger result that in [7], where it was only shown that SP is equivalent to LP. We propose and evaluate eight heuristics for determining elimination orderings in SP [8]. Our experimental results suggest that heuristics do not help SP run faster in *optimal join trees* [9]. We also investigate the role the type of join tree plays in SP inference. The performance of SP degrades dramatically when non-optimal join tree are used [10]. The use of heuristics do not help SP in non-optimal join trees either. Thereby, SP's clear advantage over LP relies on the use of optimal join trees.

This paper is organized as follows. Section 2 reviews BNs. SP is introduced in Section 3. Its correctness is established in Section 4. We establish a one-to-one correspondence between the relevant potentials in SP and LP in Section 5. An empirical comparison of SP versus LP on optimal join trees is reported in Section 6. Section 7 presents heuristics for SP and we empirically evaluate the heuristics in optimal join trees. We analyze the performance of SP in non-optimal join trees in Section 8. Section 9 empirically analyzes heuristics for SP in non-optimal join trees. Conclusions are drawn in Section 10.

2. Background

Let $U = \{v_1, v_2, \ldots, v_n\}$ be a finite set of variables, each with a finite domain, and V be the domain of U. A *potential* on V is a function ϕ such that $\phi(v) \geq 0$ for each $v \in V$, and at least one $\phi(v) > 0$. Henceforth, we say ϕ is on U instead of V. A *joint probability distribution* is a potential P on U, denoted $P(U)$, that sums to one. For disjoint $X, Y \subseteq U$, a *conditional probability table* (CPT) $P(X|Y)$ is a potential over $X \cup Y$ that sums to one for each value y of Y. For simplified notation, $\{v_1, v_2, \ldots, v_n\}$ may be written as $v_1 v_2 \cdots v_n$, and $X \cup Y$ as XY.

A *Bayesian network* (BN) [1] is a *directed acyclic graph* (DAG) $\mathcal{B}$ on U together with CPTs $P(v_1|Pa(v_1)), P(v_2|Pa(v_2)), \ldots, P(v_n|Pa(v_n))$, where $Pa(v_i)$ denotes the parents (immediate predecessors) of v_i in $\mathcal{B}$. For example, Fig. 1 depicts a BN, where CPTs $P(a), P(b|a), \ldots, P(m|g,l)$ are understood. We call $\mathcal{B}$ a BN, if no confusion arises. The product of the CPTs for $\mathcal{B}$ on U is a joint probability distribution $P(U)$.

The *conditional independence* [1] of X and Z given Y holding in $P(U)$ is denoted $I(X, Y, Z)$, where X, Y, and Z are pairwise disjoint subsets of U. If needed, the property that $I(X, Y, Z)$ is equivalent to $I(X - Y, Y, Z - Y)$ [1] can be applied to make the three sets pairwise disjoint; otherwise, $I(X, Y, Z)$ is not well-formed.

There are different types of probabilistic inference that one might be interested in, including belief update, finding most probable explanation (MPE), and finding maximum a-posterior hypothesis (MAP) [3]. Moreover, queries can be classified into different sessions [11]. In a diagnostic setting, query variables tend to be at the "top" of the BN, while in a prediction setting query variables tend to be at the "bottom" of the BN. Another setting, called incremental, is when evidence is received incrementally, while the query variables remain the same. In sensitivity analysis, the query variables remain the same, but the evidence can be incremental or even completely different. In this paper, we focus on belief update, namely, computing posterior probabilities after other variables have been observed as evidence.

A *join tree* [2] is a tree with sets of variables as nodes, and with the property that any variable in two nodes is also in any node on the path between the two. The *separator* [2] S between any two neighboring nodes N_i and N_j is $S = N_i \cap N_j$.