International Journal of Approximate Reasoning ••• (••••) •••–•••

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

An empirical study of testing independencies in Bayesian networks using rp-separation ☆

Cory J. Butz [a], André E. dos Santos [a], Jhonatan S. Oliveira [a], Christophe Gonzales [b]

[a] University of Regina, Department of Computer Science, Regina, S4S 0A2, Canada
[b] Université Pierre et Marie Curie, LIP6 – département DESIR, Paris, F-75005, France

ARTICLE INFO

Article history:
Received 12 October 2016
Received in revised form 19 October 2017
Accepted 20 October 2017
Available online xxxx

Keywords:
Bayesian networks
d-Separation
Conditional independence

ABSTRACT

Directed separation (d-separation) played a fundamental role in the founding of *Bayesian networks* (BNs) and continues to be useful today in a wide range of applications. Given an independence to be tested, current implementations of d-separation explore the *active* part of a BN. On the other hand, an overlooked property of d-separation implies that d-separation need only consider the *relevant* part of a BN. We propose a new method for testing independencies in BNs, called *relevant path separation* (rp-separation), which explores the intersection between the active and relevant parts of a BN. Favourable experimental results are reported.

1. Introduction

Directed separation (d-separation) [1] continues to be useful in a wide range of areas, including causal inference in statistics [2], cause and correlation in biology [3], extrapolation across populations [4], handling missing data [5], bioinformatics [6], and deep learning [7]. The d-separation algorithm is a graphical method for determining which *conditional independence* relations are implied by the *directed acyclic graph* (DAG) of a *Bayesian network* (BN) [1]. With respect to a given independence to be tested, current implementations, including Bayes-Ball [8] and Reachable [9], find all nodes reachable along active paths, called the *active* part of a BN. [10] was the first linear method for testing independencies in a BN. [8] emphasizes that improvements can still be made upon the linear method in [10]. However, the current implementations overlook a crucial property of d-separation, described next. Another method for testing independencies in BNs is *m-separation* [11]. In the proof of correctness, it is established that all active paths of interest can only appear in what we call the *relevant* part of a BN. Roughly speaking, the relevant part of a BN for a given independence is the set of variables in the independence statement together with their ancestors. This property warrants attention in itself, since it has both theoretical and practical ramifications.

In this paper, we propose *relevant path separation* (rp-separation) as a new method for testing independencies in BNs. The salient feature of rp-separation is that it explores the *intersection* between the active and relevant parts of a BN. We introduce the notion of a *relevant* path and establish that *irrelevant* paths are either active paths that are doomed to

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

E-mail addresses: butz@cs.uregina.ca (C.J. Butz), dossantos@cs.uregina.ca (A.E. dos Santos), oliveira@cs.uregina.ca (J.S. Oliveira), christophe.gonzales@lip6.fr (C. Gonzales).

https://doi.org/10.1016/j.ijar.2017.10.026
0888-613X/

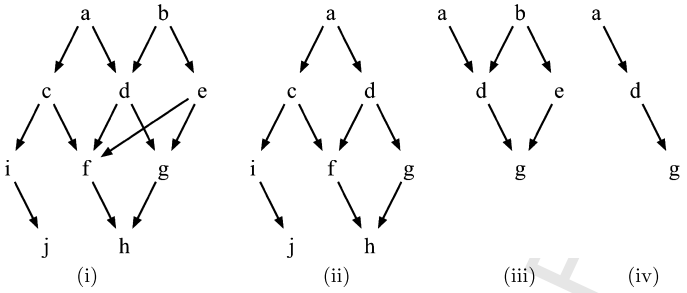

Fig. 1. (i) Testing independence $I(a, e, g)$ in a Bayesian network $\mathcal{B}$; (ii) the active part of $\mathcal{B}$; (iii) the relevant part of $\mathcal{B}$; (iv) the intersection of the active and relevant parts.

become blocked or active paths that terminate before reaching a variable of interest. Rather than exploring all active paths, rp-separation displays impressive performance in practice by only exploring active paths that are relevant. In real-world or benchmark BNs, rp-separation is faster in 17 of 19 cases with an average time savings of 53%, culminating with being nearly twice as fast in the largest BN.

This paper extends our seminal work on *i-separation* [12]. Whereas i-separation avoids traversing one type of irrelevant path, *rp-separation* [13] avoids traversing all irrelevant paths. This paper adds to the theoretical foundation of rp-separation by providing proofs of Lemma 1 and Theorem 5. Lastly, we compare rp-separation with Bayes-Ball, a simple algorithm often used in practice to test independencies in BNs. Our experimental results suggest that rp-separation is faster than Bayes-Ball especially in large BNs.

This paper is organized as follows. Section 2 gives background information. We propose rp-separation in Section 3. Section 4 includes an empirical evaluation and analysis. Section 5 compares rp-separation with Bayes-Ball. Conclusions are drawn in Section 6.

2. Background

Let $U = \{v_1, v_2, \ldots, v_n\}$ be a finite set of variables. Let $\mathcal{B}$ denote a *directed acyclic graph* (DAG) on U. A *directed path* from v_1 to v_k is a sequence $v_1, v_2, \ldots, v_k$ with directed edges (v_i, v_{i+1}) in $\mathcal{B}$, $i = 1, 2, \ldots, k-1$. For each $v_i \in U$, the *ancestors* of v_i, denoted $An(v_i)$, are those variables having a directed path to v_i. For a set $X \subseteq U$, we define $An(X)$ in the obvious way. The *children* $Ch(v_i)$ and *parents* $Pa(v_i)$ of v_i are those v_j such that $(v_i, v_j) \in \mathcal{B}$ and $(v_j, v_i) \in \mathcal{B}$, respectively. An *undirected path* in a DAG is a path ignoring directions. A directed edge $(v_i, v_j) \in \mathcal{B}$ may be written as (v_j, v_i) in an undirected path. A variable v_k is called a *v-structure* [9] in a DAG $\mathcal{B}$, if $\mathcal{B}$ contains directed edges (v_i, v_k) and (v_j, v_k), but not a directed edge between variables v_i and v_j. A singleton set $\{v\}$ may be written as v, $\{v_1, v_2, \ldots, v_n\}$ as $v_1 v_2 \cdots v_n$, and $X \cup Y$ as XY.

A *Bayesian network* (BN) [1] is a DAG $\mathcal{B}$ on U together with *conditional probability tables* (CPTs) $P(v_1|Pa(v_1))$, $P(v_2|Pa(v_2)), \ldots, P(v_n|Pa(v_n))$. For example, Fig. 1 (i) shows a BN, where CPTs $P(a), P(b), \ldots, P(j|i)$ are not provided. 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* $P(U)$ [1]. The *conditional independence* [1] of X and Z given Y holding in $P(U)$ is denoted $I_P(X, Y, Z)$, where X, Y, and Z are pairwise disjoint subset of U. It is known that if $I(X, Y, Z)$ holds in $\mathcal{B}$, then $I_P(X, Y, Z)$ holds in $P(U)$.

d-Separation [1] tests independencies in BNs and can be presented as follows [14]. Let X, Y, and Z be pairwise disjoint sets of variables in a BN $\mathcal{B}$. We say X and Z are *d-separated* by Y, denoted $I(X, Y, Z)$, if at least one variable on every undirected path from (any variable in) X to (any variable in) Z is closed. On a path, there are three kinds of variable v: (i) a *sequential* variable means v is a parent of one of its neighbours and a child of the other; (ii) a *divergent* variable is when v is a parent of both neighbours; and (iii) a *convergent* variable is when v is a child of both neighbours. A variable v is either open or closed. A sequential or divergent variable is *closed*, if $v \in Y$. A convergent variable is *closed*, if $(v \cup De(v)) \cap Y = \emptyset$. A path with a closed variable is *blocked*; otherwise, it is *active*.

Example 1. Let us test $I(a, e, g)$ in the BN $\mathcal{B}$ of Fig. 1 (i) using d-separation. Here $X = \{a\}$, $Y = \{e\}$, and $Z = \{g\}$. The path (a, d), (d, b), (b, e), (e, g) from X to Z is blocked by closed convergent variable d, since $d \cup De(d) = \{d, f, g, h\}$ and $\{d, f, g, h\} \cap Y = \emptyset$. On the contrary, the path (a, d), (d, g) from X to Z is active, since d is an open sequential variable. As there exists an active path from a to g, $I(a, e, g)$ does not hold in $\mathcal{B}$ by d-separation.

Geiger et al. [10] were the first to provide a linear time complexity algorithm for implementing d-separation. Their method, however, always explores the entire DAG. *Bayes-Ball* [8] only explores the *active* part of the DAG. As will be discussed in Section 5, Bayes-Ball works using a hypothetical ball bouncing in a DAG and can be used for purposes outside the scope of our paper. Instead, we use the REACHABLE algorithm [9], since it also explores the active part of a DAG, but with d-separation terminology. To test $I(X, Y, Z)$ in a BN $\mathcal{B}$, REACHABLE takes X, Y, and $\mathcal{B}$ as input and returns the set of all variables reachable from X along active paths. For pedagogical purposes, Example 2 will mention active paths explicitly.

Please cite this article in press as: C.J. Butz et al., An empirical study of testing independencies in Bayesian networks using rp-separation, Int. J. Approx. Reason. (2017), https://doi.org/10.1016/j.ijar.2017.10.026