Knowledge-Based Systems 30 (2012) 95–103

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A search problem in complex diagnostic Bayesian networks

Dayou Liu, Yuxiao Huang, Qiangyuan Yu, Juan Chen *, Haiyang Jia

College of Computer Science and Technology, Jilin University, Changchun 130012, China
Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

ARTICLE INFO

Article history:
Received 18 May 2011
Received in revised form 12 December 2011
Accepted 20 December 2011
Available online 28 December 2011

Keywords:
Diagnostic Bayesian networks
Exact inference
Computation sharing
Inference simplifying
Search

ABSTRACT

Inference in Bayesian networks (BNs) is NP-hard. We proposed the concept of a node set namely Maximum Quadruple-Constrained subset $MQC(A,a-e)$ to improve the efficiency of exact inference in diagnostic Bayesian networks (DBNs). Here, A denotes a node set in a DBN and $a-e$ represent five real numbers. The improvement in efficiency is achieved by computation sharing. That is, we divide inference in a DBN into the computation of eliminating $MQC(A,a-e)$ and the subsequent computation. For certain complex DBNs and $(A,a-e)$, the former computation covers a major part of the whole computation, and the latter one is highly efficient after sharing the former computation.

Searching for $MQC(A,a-e)$ is a combinatorial optimization problem. A backtracking-based exact algorithm Backtracking-Search (BS) was proposed, however the time complexity of BS is $O(n^3 2^n)$ ($n=|A|$). In this article, we propose the following algorithms for searching for $MQC(A,a-e)$ especially in complex DBNs where $|A|$ is large. (i) A divide-and-conquer algorithm Divide-and-Conquer (DC) for dividing the problem of searching for $MQC(A,a-e)$ into sub-problems of searching for $MQC(B_1, a-e),\ldots,MQC(B_m,a-e)$, where $B_i \subseteq A(1 \leqslant i \leqslant m, 1 \leqslant m \leqslant |A|)$. (ii) A DC-based heuristic algorithm Heuristic-Search (HS) for searching for $MQC(B_i,a-e)$. The time complexity of HS is $O(n^6)$ ($n=|B_i|$). Empirical results show that, HS outperforms BS over a range of networks.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Bayesian networks (BNs) [1] play a central role in a wide range of automated inference applications, including in diagnosis [2–11]. In diagnostic Bayesian networks (DBNs), nodes are divided into three categories: the diagnostic nodes D, the measurement nodes M, and the intermediate nodes I [4][1]. Inference in DBNs is to compute the posterior probability for a set of diagnostic variables $Q(Q \subseteq D)$, given the observations e to a set of measurement variables $E(E \subseteq M)$[2]. Inference in BNs includes exact and approximate inference, and both exact and approximate inference are NP-hard [12–14]. In this article, we only study the exact one, and term "inference" hereafter refers to exact inference.

To improve the efficiency of inference in DBNs, Huang et al. [15] proposed the concept of a node set namely Maximum Quadruple-Constrained subset $MQC(A,a-e)$, where $A \subseteq I$ and $a-e$ denote five real numbers. For simplicity, in the rest of this article, we shorten the notation $MQC(A,a-e)$ to MQC. The improvement in efficiency is accomplished by sharing the computation of eliminating MQC. Specifically, for a query $P(Q|E=\mathrm{e})$ in a DBN, we can divide the inference of computing $P(Q|E=\mathrm{e})$ into the inference of eliminating MQC and the subsequent inference, that is, the one of eliminating $N-MQC-(Q \cup E)$. Here, we let N be the node set of a DBN, that is, $N=D \cup M \cup I$. For certain complex DBNs and $(A,a-e)$, the former inference covers a major part of the total computation (computing $P(Q|E=\mathrm{e})$), and the latter one is highly efficient after sharing the former computation. We can see this in Example 3.

Moreover, if we eliminate MQC offline and eliminate the remaining variables online, then (i) all $P(Q|E=\mathrm{e})$ can be computed online, and (ii) there are circumstances where online inference can be highly efficient. Note that one might also divide inference into offline and online inference in certain conventional inference algorithms, for instance, Clique Tree Propagation (CTP) [1,16–18][3]. Consequently, the efficiency of online inference can also be improved through sharing the offline computation. However, the improvement in these algorithms can not be as enormous as in ours; see this in Example 3. It is because the offline inference in the algorithms can not carry out most of the computation. Our offline–online method might find applications in complex tasks that need to be executed in real-time, for instance, object tracking in video sequences.

* Corresponding author at: Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China.

E-mail address: chenjuan@jlu.edu.cn (J. Chen).

[1] In this article, we use italic upper and lower cases for node sets and nodes respectively.

[2] In this article, we use term "node" and "variable" interchangeably.

[3] The offline inference in CTP is also referred to as offline compilation [2].

0950-7051/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.12.011

The problems of choosing $(A, a - e)$ and searching for the corresponding *MQC* especially in complex DBNs are critical. In this article, we focus on the study of algorithms for searching for *MQC* when $(A, a - e)$ is given. We will discuss the algorithms for choosing $(A, a - e)$ in another work. Searching for *MQC* is a combinatorial optimization problem [15]. Huang et al. [15] proposed a backtracking-based exact algorithm Backtracking-Search (BS) for searching for *MQC*. The time complexity of BS is $O(n^3 2^n)$ ($n = |A|$), therefore it is computationally hard to use BS when $|A|$ is large.

In this article, we propose two algorithms for searching for *MQC* particularly in complex DBNs where $|A|$ is large. (i) A divide-and-conquer algorithm Divide-and-Conquer (DC) for dividing the problem of searching for *MQC* into sub-problems of searching for $MQC(B_1, a - e), \ldots, MQC(B_m, a - e)$, where $B_i \subseteq A$ $(1 \leqslant i \leqslant m, 1 \leqslant m \leqslant |A|)$. (ii) A DC-based heuristic algorithm Heuristic-Search (HS) for searching for $MQC(B_i, a - e)$. The time complexity of HS is $O(n^6)$ ($n = |B_i|$), and that of DC is $O(n_1^2 + n_1 n_2^3 2_2^n)$ or $O(n_1^2 + n_1 n_2^6)$ ($n_1 = |A|$, $n_2 = \max(|B_1|, \ldots, |B_m|)$) when searching for $MQC(B_i)$ using BS or HS respectively. Based on computational experiments, it is shown that HS is superior to BS over a range of networks.

The remainder of this article is organized as follows. We give a compact introduction to the topic of BN in Section 2. The definition of *MQC* and the idea of BS will also be provided in this section. We propose algorithm DC and HS in Sections 3 and 4. Empirical results are reported in Section 5. Conclusion and future work are discussed in Section 6.

2. Background

A BN is a Directed Acyclic Graph (DAG) composed of network structure and conditional probabilities [19]. In a BN, nodes represent variables in the modeled data set and are annotated with Conditional Probability Tables (CPTs), while edges represent causal relations between variables [20]. In this article, we restrict ourselves to discrete random variables, and "node" thus means "discrete node"; we let $dom(x)$ be the domain of variable x.

BNs not only explicitly represent conditional probabilities, but also implicitly represent the following conditional independence assertions: let $x_1, \ldots, x_n$ be an enumeration of all the nodes in a BN such that each node x_i $(1 \leqslant i \leqslant n)$ appears after its parents $\pi(x_i)$, then x_i is conditionally independent of $\{x_1, \ldots, x_{i-1}\} - \pi(x_i)$ when $\pi(x_i)$ is given [21]. Therefore according to the chain rule, we have:

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | x_1, \ldots, x_{i-1}) = \prod_{i=1}^{n} P(x_i | \pi(x_i)).$$

In BNs, the computation of posterior probabilities is of great interest [2]. One may use Bayesian network inference to achieve this. Most inference algorithms belong to either the class of query-based or the class of all-marginals algorithms [22]. The former includes, for example, Variable Elimination (VE) [23–27], Belief Propagation [1], Arc-Reversal [23,24,28], Symbolic Probabilistic Inference [29], Recursive Decomposition [30], Bucket Elimination [31], the Fusion operator [32], Query DAGs [33], Recursive Conditioning [34] and Value Elimination [35]. The latter contains, for instance, CTP. VE and CTP are the most representative algorithms in the two categories respectively. Major problem of conventional algorithms is, they are highly time consuming or unfeasible in complex BNs. It is because, as we mentioned, inference in BNs is NP-hard.

The concept of *MQC* was proposed to improve the efficiency of inference in DBN. Fig. 1 demonstrates a DBN namely DBN 1. One interesting feature of DBN is, one may simplify inference in a DBN by pruning the network. The following definition of a node set plays an important role in pruning a DBN. In this article, we let N be the node set of a DBN and $an(x)$ be the ancestor set of node $x \in N$.

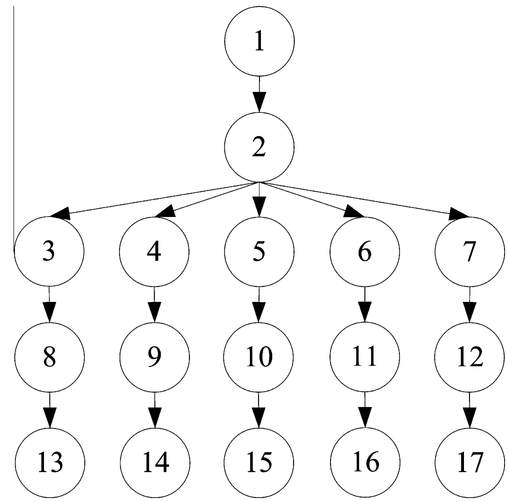

Fig. 1. DBN 1, where $D = \{1\}$, $M = \{8–12\}$, $I = \{2–7, 13–17\}$, $|dom(1)| = 8$, $|dom(x)| = 2000$ ($x \in \{2–5, 11, 13–17\}$), $|dom(y)| = 2$ ($y \in \{6, 7\}$), $|dom(z)| = 4$ ($z \in \{8–10\}$), $|dom(12)| = 200$, conditional probability of each variable is evenly distributed.

Definition 1 (*Minimal ancestral set [36]*). Consider node set $X \subseteq N$ and $Y \subseteq N$. X is the minimal ancestral set of Y $man(Y)$, if X is the minimal node set which adheres to the following conditions: (i) $Y \subseteq X$; (ii) $\forall x \in X$, then $an(x) \subseteq X$.

Consider DBN 1, then $man(D) = D$, $man(M) = \{1–12\}$, $man(I) = N$. With the concept from Definition 1 in hand, we introduce the following result according to which we prune a DBN. In this article, we let $I_p = I \cap man(D \cup M)$, $N_p = D \cup M \cup I_p$.

Theorem 1 [36]. $\forall Q \subseteq D$ and $\forall E \subseteq M$, *then* $P(Q|E) = (\sum_{N_p-(Q \cup E)} P(N_p))/P(E)$.

According to Theorem 1, by removing I-$man(D \cup M)$ (as well as edges connected to them) from a DBN, we obtain a sub-network which possesses the following features: (i) the value of $P(Q|E)$ ($Q \subseteq D$, $E \subseteq M$) computed in the sub-network is equal to that computed in the original one; (ii) less computation is needed to compute $P(Q|E)$ in the sub-network compared with the original one. Fig. 2 demonstrates the pruned sub-network obtained by removing I-$man(D \cup M)$ from DBN 1. This sub-network, namely DBN 2, will be used throughout the article.

MQC is a node set in the pruned sub-network of a DBN; specifically, $MQC \subseteq I_p$. The definition of *MQC* relies on the following

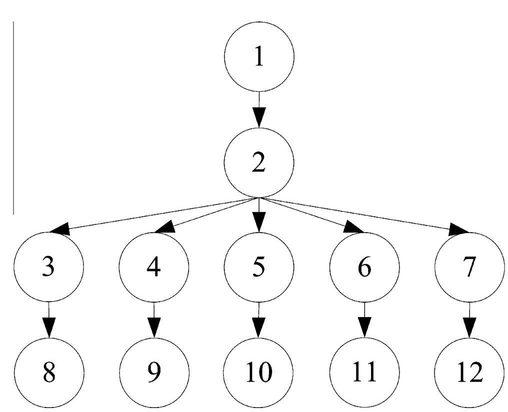

Fig. 2. DBN 2, obtained by removing I-$man(D \cup M)$ from DBN 1.

متن کامل مقاله
دریافت فوری

ISIArticles
مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات