Knowledge-Based Systems 121 (2017) 185–197

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A new hybrid method for learning bayesian networks: Separation and reunion

Hui Liu[a], Shuigeng Zhou[b,*], Wai Lam[c], Jihong Guan[d]

[a] Lab of Information Management, Changzhou University, Jiangsu, China
[b] Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China
[c] Department of Systems Engineering and Engineering Management, the Chinese University of Hong Kong, Shatin, Hong Kong, China
[d] Department of Computer Science and Technology, Tongji University, Shanghai, China

ARTICLE INFO

Article history:
Received 3 June 2016
Revised 13 January 2017
Accepted 20 January 2017
Available online 23 January 2017

Keywords:
Bayesian network
Conditional independence
Scoring function
d-separation

ABSTRACT

Most existing algorithms for learning Bayesian networks (BNs) can be categorized as constraint-based or score-based methods. Constraint-based algorithms use conditional independence (CI) tests to judge the presence or absence of an edge. Though effective and applicable to (high-dimensional data) large-scale networks, CI tests require a large number of samples to determine the independencies. Thus they can be unreliable especially when the sample size is small. On the other hand, score-based methods employ a score metric to evaluate each candidate network structure, but they are inefficient in learning large-scale networks due to the extremely large search space. In this paper, we propose a new hybrid Bayesian network learning method, SAR (the abbreviation of **S**eparation **A**nd **R**eunion), which maintains the merits of both types of learning techniques while avoiding their drawbacks. Extensive experiments show that our method generally achieves better performance than state-of-the-art methods.

1. Introduction

Bayesian network (BN) is a type of probabilistic graphical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). BNs have been widely used in the representation of knowledge with uncertainty and efficient reasoning, and has been widely used in real-world applications, including document classification, image processing, semantic search and modeling of biological systems [27]. The BN learning problem consists of the DAG recovery and numerical parameters characterizing it. Many learning algorithms have been proposed to learning BNs from increasingly available data in many domains.

Existing BN learning methods can be categorized into three types: constraint-based, score-based, and hybrid approaches. Constraint-based methods, such as PC [45], TPDA [11] and sparse candidate [19], often start with a complete graph, and then carry out conditional independence (CI) tests to remove undesired edges as many as possible. A drawback of constraint-based methods is that they require a large number of samples to determine the independencies leading to unreliable CI tests and incorrect decision of the presence or absence of graph edges. A more worse problem is that an error in the early stage has a cascading effect that causes a drastically different network. Though some methods have been proposed to cope with this problem, it has not yet solved well [10,52]. Score-based methods employ a score metric, such as K2 [14], BDe [22] and BIC [41], to evaluate each candidate network structure, and try to search the optimal structure that fits best to the sample data. One major problem with score-based methods is that the number of possible network structures grows exponentially with the number of variables and exhaustive search for the optimal structure becomes prohibitively costly. Therefore, almost all score-based methods employ heuristic search to reduce the search space, which often converges to a local optima. In addition, some hybrid methods have been proposed to combine the features of the constraint-based and score-based techniques [48,51] . For example, the CB algorithm first uses CI tests to derive an ordering of nodes, and then use a modified version of K2 to learn the network [44]. The EGS algorithm first searches numbers of essential graphs using the PC algorithm [45] and then randomly converts each essential graph to a DAG and scores the DAG using the Bayesian metric, and finally uses a Bayesian scoring metric to search for a maximum-scored DAG [16]. The IMAPR algorithm also use CI tests to find a more promising starting point to restart the local structural search [17]. Acid and de Campos [1] proposed a discrepancy-based scoring metric and a heuristic search strategy to search for good network structure, meanwhile the au-

* Corresponding author.
E-mail addresses: liuhui@fudan.edu.cn (H. Liu), sgzhou@fudan.edu.cn (S. Zhou).

http://dx.doi.org/10.1016/j.knosys.2017.01.029
0950-7051/

thors employed low-order conditional independence tests to reduce the overhead of structure search.

However, these hybrid methods cannot obtain desirable performance for complex networks since they just combine these two types of techniques in a straightforward manner.

Xie and Geng [53] innovatively proposed a *recursive algorithm* to learn Bayesian network structures. They showed that the structural learning task for a Bayesian network can be decomposed into some subtasks, provided that legal undirected independence graphs are given at each decomposition step. The recursive algorithm pushes the CI tests into smaller and smaller conditioning sets. Encouraging experimental result was obtained. Nevertheless, this recursive algorithm is in essence constraint-based and suffers from the inherent weaknesses of CI tests.

Note that constraint-based methods are effective in learning large-scale graphs but need a large sample dataset, while score-based methods perform efficiently only in learning small-scale graphs but is computationally prohibitive for large-scale graphs. These observations motivate us to explore new methods that can combine the merits of the two types of learning techniques while avoiding their drawbacks. In this paper, we propose a new hybrid learning algorithm SAR (**S**eparation **A**nd **R**eunion) that integrates the two types of learning methods. We first describe an overview of our framework and derive some theoretical results that underlying our learning algorithm. The task of learning a large BN can be decomposed into learning some relatively small BNs, and the actual BN can be obtained by re-unifying these small BNs. Precisely, we first use only a few CI tests to construct the *lookahead-q-partial graph*, which is proposed in this paper as an extension to the *q-partial graph* introduced by Castelo and Roverato [10]. The *lookahead-q-partial graph* is then decomposed into some small subgraphs of proper size, from each of which a small BNs is learned by using a score-based algorithm. Finally, we unify these small BNs to form the actual BN. We have conducted extensive experiments to compare our method with six existing methods, and the experimental results demonstrate that our method can generally achieve better or comparable performance than state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 surveys the related work. Section 3 presents the preliminaries of BN learning. Section 4 introduces the SAR method including its framework, theoretical foundation, and overall design. Section 5 presents the detailed design of some major algorithms in the SAR method. Section 6 shows the experimental results. Section 7 concludes the paper and highlights future work.

2. Related work

Before diving into technical detail of our SAR framework, we'd like to give a concise survey of the three types of BN learning methods.

2.1. Constraint-based approaches

Classical constraint-based methods include inductive causation (IC) algorithm [37], PC algorithm [45], grow and shrink (GS) algorithm [32], and three-phase dependency analysis (TPDA) algorithm [11]. All these methods utilize statistical tests (G-square, conditional partial correlation, conditional mutual information etc.) for evaluating conditional independency (CI) between variable pairs to learn the skeleton of the graph, and then assign directions to as many edges as possible according to some orientation rules. Finally, one network is output from the equivalence class consistent with these CI tests. The IC algorithm searches for a separator for each pair of variables that are independent conditioned on this separator. The PC algorithm limits the possible separators that are adjacent to the pair of variables. The GS algorithm first detects the Markov blanket of each variable and then orients the edges. The TPDA method can only identify monotone DAG-faithful network, which requires that the conditional mutual information of two variables be a monotonic function of the "active paths" between the variables in the network structure. Under the faithfulness condition, these methods have been proven to output a graph converging to the true network as the size of data scales to infinity. Moreover, the complexity of these methods is polynomial if the maximal size of neighborhood nodes is bounded. However, in practice, the performance of these methods suffers from the inaccuracy of CI tests especially when the sample size is small.

Xie and Geng [53] have proposed a recursive algorithm to learn the structure of a BN. Their main idea is first decomposing a legal *undirected independence graph* into a number of subgraphs from each of which an essential subgraph can be learned independently, then recovering the actual essential graph by combining these essential subgraphs step by step, and finally assigning orientations to as many edges as possible by using the rules proposed by Meek [33]. Although the recursive algorithm is shown to efficiently derive an equivalence of the BN [49], it is in essence based on CI tests and its performance is inevitably affected by the accuracy of the CI tests. For a graph with large tree-width [6], a large amount of data is needed to guarantee the reliability of the CI tests. Furthermore, the recursive algorithm has to afford for the considerable overhead of constructing the undirected independence graph, especially for large-scale graphs.

2.2. Score-based approaches

This family of approaches uses a scoring function to evaluate how well a DAG fits the data, and return the one that maximizes the scoring function. The posterior probability of the network given the data is most widely used as the score function by the score-based algorithms such as BIC [41], BDe [22], and AIC [2]. Imoto et al. [24] proposed another scoring criterion by using nonparametric regression. Yang and Chang [54] evaluated five score metrics, including uniform prior score metric (UPSM), conditional uniform prior score metric (CUPSM), Dirichlet prior score metric (DPSM), likelihood-equivalence Bayesian Dirichlet score metric (BDe), and minimum description length (MDL), and found that DPSM outperformed other metrics. Recently, [9] developed a non-Bayesian scoring function based on mutual information tests (MIT in short), to evaluate the candidate networks.

For score-based approaches, the search strategy is as important as the scoring function, as the search space increases in super exponential size with regard to the number of nodes n, that is, $O(n!2^{\binom{n}{2}})$, an exhaustive search is computationally inhibitive. Various search strategies have been proposed to traverse the structural space. The sparse candidate (SC) method impose restrictions on the maximum number of parent nodes and estimates the candidate parents for each node before the search process [19]. Greedy equivalent search [13] traverses the space of the equivalence class as main scoring functions are score equivalent [12], that is, two equivalent DAGs (encoding the same set of conditional independencies) are given same scores; The optimal re-insertion (OR) method employs an optimal re-insertion transformation strategy that globally affects the graph structure at each step, and somehow enables the search to escape from local optima. Some algorithms constrain the ordering of the variables to reduce the search space [34]. The K2 algorithm is proposed to select the best parent set for each variable among the subsets of preceding variables [14]. Teyssier and Koller [46] proposed another ordering-based search strategy and obtained promising results. Several optimal search algorithms try to find the global optimal graph without explicitly checking every possible graph [36,43]. de Campos et al. [8] revealed that the search space of possible network structures can be reduced drastically without