Knowledge-Based Systems 24 (2011) 963–976

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A partial correlation-based Bayesian network structure learning algorithm under linear SEM

Jing Yang *, Lian Li *, Aiguo Wang

Department of Computer Science and Technology, Hefei University of Technology, Hefei 230009, China

ARTICLE INFO

Article history:
Received 27 September 2010
Received in revised form 4 April 2011
Accepted 6 April 2011
Available online 13 April 2011

Keywords:
Partial correlation
Bayesian network
Structure learning
Local learning
Linear SEM (simultaneous equation model)

ABSTRACT

A new algorithm, the PCB (partial correlation-based) algorithm, is presented for Bayesian network structure learning. The algorithm effectively combines ideas from local learning with partial correlation techniques. It reconstructs the skeleton of a Bayesian network based on partial correlation and then performs a greedy hill-climbing search to orient the edges. Specifically, we make three contributions. First, we prove that in a linear SEM (simultaneous equation model) with uncorrelated errors, when the datasets are generated by linear SEM, subject to arbitrary distribution disturbances, we can use partial correlation as the criterion of the CI test. Second, we perform a series of experiments to find the best threshold value of the partial correlation. Finally, we show how partial correlation can be used in Bayesian network structure learning under linear SEM. The effectiveness of the method is compared with current state of the art methods on eight networks. A simulation shows that the PCB algorithm outperforms existing algorithms in both accuracy and run time.

1. Introduction

Bayesian networks (BN) are widely used to represent probabilistic relationship among random variables. They have been successfully applied to many domains such as medical diagnosis, gene data analysis, and hardware troubleshooting, rare event predictions, scenario analysis [28,4,6].

Learning the structure of a Bayesian network from a dataset **D** is useful; unfortunately, it is an NP-hard problem [5]. Consequently, many heuristic techniques have been proposed. One of the most basic search algorithms is a local greedy hill-climbing search over all DAG structures. The size of the search space of the greedy search is a super-exponential function of the number of variables. One approach is to place constraints on the search to improve its efficiency, as in the K2 algorithm [7], the SC algorithm [10], the MMHC algorithm [25], and the L1MB algorithm [18].

One drawback of the K2 algorithm is that it requires a total variable ordering. The SC algorithm is based on the idea of local learning and uses a two-phase framework including a **Restriction** step and a **Search** step. In the **Restriction** step, the SC algorithm uses mutual information to find a set of potential neighbors (parents and children) for each node and achieves fast learning by restricting the search space. One drawback of the SC algorithm is that it only allows a variable to have a maximum of up to k parents. However, a common parameter k for all nodes will have to sacrifice either efficiency or quality of reconstruction [25]. The MMHC algorithm uses the max–min parents–children (MMPC) algorithm to identify a set of potential neighbors [25]. Our experiments show that the MMHC algorithm has high accuracy, but one drawback of it is that it requires conditional independency tests on exponentially large conditioning sets. Therefore, the MMHC algorithm is very slow on high dimensional complex networks. The L1MB algorithm uses L1 techniques to learn DAG structure and uses the LARS algorithm [9] to find a set of potential neighbors [18]. The L1MB algorithm has good time performance. However, the L1MB algorithm evaluates the effects of a set of variables, not a single variable. The method can describe the correlation between a set of variables and a variable but not the correlation between two variables. It is not reasonable to use this method to select potential neighbors, and our experiments show that the L1MB algorithm has low accuracy.

In fact, many algorithms, such as the K2, SC, PC [22], TPDA [3], and MMHC, can be implemented efficiently with discrete variables but are not directly applicable to continuous variables. Some algorithms including the SC, PC, and TPDA, have been designed for discrete variables. Even though they can be used for continuous variables, our experiments show that they have many structural errors. The L1MB algorithm has been designed for continuous variables. However, it uses L1 regression to find a set of potential neighbors for one variable once. However, it cannot precisely capture the causal relation between two variables, so the selection of potential neighbors is somewhat unreasonable, and our

* Corresponding authors. Tel.: +86 13866185496; fax: +86 05512901001 (J. Yang), tel./fax: +86 05512901001 (L. Li).
E-mail addresses: jsjyj0801@163.com (J. Yang), llian@hfut.edu.cn (L. Li).

0950-7051/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2011.04.005

experiments show that its accuracy is relatively low. In this paper, we propose a new heuristic technique using local learning, and we show experimentally that it outperforms several existing approaches for continuous and binary variables.

The correlations among multiple correlative variables are complex. To a certain extent, there is a correlation between any two variables, but that correlation is affected by the other correlative variables. The simple correlation method does not consider these influences, so it cannot reveal the true correlation between two variables. The true correlation between two variables can be obtained only after the influences of the other correlative variables are removed. The partial correlation method can eliminate the influences of other correlative variables by holding them unchanged in the analysis and thus reveal the true correlation between the two variables of interest [27]. For example, the findings in [27] suggest that the simple correlation coefficient between NmF2 and h(100) is affected by other influence factors and therefore cannot reveal the true correlation between NmF2 and h(100); the partial correlation method can eliminate the influences of F107, Ap and the seasonal variation factors and can thus reveal the true correlation between the two variables of interest by eliminating the influences of the other correlative variables. Partial correlation has been widely used to describe the relative importance of variables in multivariate regression analysis, and it has been successfully applied to many fields such as medicine [13,26], economics [23], and geology [27]. In causal discovery, it has been used (as transformed by Fisher's z [17]) as a continuous replacement for CI tests in the PC algorithm. Pellet et al. introduced the partial-correlation-based CI test into causation discovery, on the assumption that the data follow a multivariate Gaussian distribution for continuous variables [14]. However, when the data do not follow a multivariate Gaussian distribution, can partial correlation be used as a CI test?

Our first contribution is to prove that partial correlation can be used as the criterion for a CI test under the linear simultaneous equation model (SEM), which includes the multivariate Gaussian distribution as a special case. Our second contribution is that we propose an effective algorithm, called PCB (partial correlation-based), which effectively combines ideas from local learning with partial correlation techniques. The PCB algorithm works in the continuous or binary variable settings under the assumption data generated by linear SEM. The computational complexity of PCB is $O(3mn^2 + n^3)$(where n is the number of variables and m is the number of cases). One advantage of PCB is its time performance. The time complexity of our PCB is bounded by a polynomial in the number of variables. Another advantage of the PCB algorithm is its quite high accuracy. The third advantage of the PCB algorithm is that it uses a relevance threshold to evaluate the correlation to alleviate the drawback of SC algorithm (common parameter k for all nodes), and we also find the best relevance threshold by a series of extensive experiments. Empirical results show that PCB outperforms the above existing algorithms in both accuracy and time performance.

The remainder of the paper is structured as follows. In Section 2, we present the background of learning structure. In Section 3, we present the PCB algorithm and its computational complexity analysis. Some empirical results are presented and discussed in Section 4. Finally, we conclude this work and address some issues for future work in Section 5.

2. Background: learning structure

Consider the problem of analyzing the distribution over some set $\mathbf{X}$ of random variables $X_1,\ldots,X_n$, each of which takes values in some domain $Val(X_i)$, where the variables are either discrete-valued or continuous-valued. A Bayesian network is an annotated directed acyclic graph that encodes a joint probability distribution over $\mathbf{X}$. Formally, a Bayesian network for $\mathbf{X}$ is a pair $B = \langle G, \Theta\rangle$. The first component, G, is a directed acyclic graph with vertices that correspond to the random variables $X_1,\ldots,X_n$; the second component, Θ, represents the set of parameters that quantifies the network. It contains a parameter $\theta_{x_i|\mathbf{pa}(X_i)} = P(x_i|\mathbf{pa}(X_i))$, for each possible value x_i of X_i, $\mathbf{pa}(X_i)$ of $\mathbf{Pa}(X_i)$. Here $\mathbf{Pa}(X_i)$ denotes the set of parents of in G and $\mathbf{pa}(X_i)$ is a particular instantiation of the parents. A Bayesian network B specifies a unique joint probability distribution over $\mathbf{X}$ given by $P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i|\mathbf{pa}(X_i))$.

The problem of learning a Bayesian network can be stated as follows. Our input is a fully observed data set $D = \{\mathbf{x}^1,\ldots,\mathbf{x}^m\}$ of instances of $\mathbf{X}$, where each $\mathbf{x}^i$ is a complete assignment to the variables $X_1,\ldots,X_n$ in $Val(X_1,\ldots,X_n)$. Our goal is to find a network structure G that is a good predictor for the data. The most common approach to this task is to define it as an optimization problem. We define a scoring function score $(G : D)$, which evaluates different networks relative to the data D. We must then solve the combinatorial optimization problem of finding the network that achieves the highest score. An important characteristic of the score is its decomposability [24], i.e., that it is the sum of the scores associated with individual families (where a family is a node and its parents):

$$score(G) = \sum_{i=1}^{n} score(\{X_i, \mathbf{Pa}(X_i)\}).$$

where$\{X_i, \mathbf{Pa}(X_i)\}$ is a set composed by the union of $\mathbf{X}$ and its parents. Given a scoring function, our task is to find

$$argmax(score(G)).$$

This task is a difficult combinatorial problem. Several of its specific instances have been shown to be NP-hard, even when the maximum number of parents per node is at most two [5].

Consequently, many heuristic techniques have been proposed. Three main approaches to the problem are the search-and-score, constraint-based, and hybrid approaches. In general, constraint-based algorithms use tests of conditional independence and measures of association to impose constraints on the network structure and infer the final *DAG*. This is usually done using a statistical hypothesis test, such as the G^2 test, and mutual information. Examples of this approach include the SGS [21], PC [22], and TPDA [3] algorithms. Search-and-score methods search over a space of structures, employing a scoring function to guide the search. Some of the standard scoring functions are Bayesian Dirichlet (specifically BDe with uniform priors, BDeu) [9], the Bayesian Information Criterion (BIC)[19], the Akaike Information Criterion (AIC) [1], and Minimum Description Length (MDL) [16,12]. Examples of this approach include the GES, K2, SC, and L1MB algorithms. Hybrid algorithms combine the search-and-score and constraint-based techniques. The first hybrid algorithm to appear is the CB algorithm [20]. Perhaps the most successful algorithm of this kind is the MMHC algorithm. The MMHC was shown to outperform many other methods in a series of extensive experiments [18].

3. PCB algorithm

In this section, we first give the framework of the PCB algorithm, we then discuss each step individually, and finally, we give an analysis of the PCB algorithm.

3.1. Outline of the algorithm

We can now explain our algorithm, called the PCB (Partial Correlation-Based) algorithm, which works under the two-phase framework. PCB first identifies the undirected skeleton of a