Neurocomputing 123 (2014) 3–12

Contents lists available at ScienceDirect

Neurocomputing

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

Efficient methods for learning Bayesian network super-structures

Edwin Villanueva, Carlos Dias Maciel *

Department of Electrical Engineering, Sao Carlos School of Engineering, University of Sao Paulo, Brazil

ARTICLE INFO

Article history:
Received 12 April 2012
Received in revised form
15 October 2012
Accepted 22 October 2012
Available online 8 April 2013

Keywords:
Bayesian networks
Structure learning
Super-structure

ABSTRACT

Learning large Bayesian networks (BN) from data is a challenging problem due to the vastness of the structure space. An effective way to turn this problem affordable is the use of super-structures—SS (undirected graphs that contain the BN skeleton). However, the literature has been lacking of specialized methods for estimating SS. We present here two algorithms intended for such purpose in the hybrid approach of BN structure learning. The first one, called Opt01SS, learns SS using only zero-and-first-order conditional independence (CI) tests in a way that allows dealing with the presence of approximate-deterministic relationships and inconsistent CIs, commonly found in small samples. The second algorithm, called OptHPC, is a computational optimized version of the recent HPC algorithm (De Morais and Aussem 2010, [17]) that showed an attractive accuracy for SS recovery. Results on various benchmark networks showed that the proposed algorithms achieve a balance between sensitivity and specificity clearly more favorable for the task of SS estimation than several representative state-of-the-art methods. The computational cost was also found to be reasonable, being Opt01SS one of the most competitive among the analyzed algorithms.

1. Introduction

A Bayesian Network (BN) is a powerful tool for representing complex probabilistic knowledge. It has been widely applied in a variety of domains, including medicine [1], bioinformatics [2], economics [3], among others [4]. The wide popularity of this tool is largely due to its great expressive power, allowing the simultaneous analysis of complex relationships between many variables. The knowledge in a BN is intuitively encoded via a directed acyclic graph — DAG (model structure), where the nodes represent random variables of a problem and the edges represent direct dependencies among variables.

Frequently, the construction of the model structure is too complex (if not impossible) for humans, requiring the estimation of it from training data. This is a challenging problem, since exact inference of the DAG is a NP-hard problem [5]. Several methods have been proposed to this end over the last two decades, being two approaches dominants: the constraint-based (CB) methods [6] and the score-and-search (SS) methods [7]. In CB methods, the structure is found via conditional independence (CI) tests. In SS methods, the network is found by optimizing a function that measures how well the network fits the data. Both approaches have drawbacks. CB methods are inaccurate in dense networks and few data because the CI tests become unreliable in such cases. SS methods are more accurate, but they do not scale up to high-dimensional problems due to a super-exponential growth of the search space [8]. Hybrid methods have emerged to overcome such limitations [9–12]. In this approach, a *super-structure* (an undirected graph assumed to contain all true edges) is rapidly estimated with a CB approach, which is then used in a subsequent SS phase to constraint the search space, i.e., the final DAG is searched considering only the edges on that super-structure.

Despite the scalability gain with the hybrid approach, some concerns has been raised recently regarding the suitability of current methods used to learn super-structures [11,12]. Most of such methods were not specifically devised for that purpose. Some are found as subroutines in CB approaches, responsible for getting the BN skeleton to be further directed [6,13,8,14]. Some other methods were designed for learning the local structure around a target variable [10,15] (e.g. for classification applications). The main concern of these methods when used as super-structure estimators is that they give equal importance to the rate of false-negative (FNE) and false-positive errors (FPE) (most of them are correct in the sample limit). However, for a hybrid approach is more important to maintain the FNE rate as low as possible in the super-structure estimation phase, since it sets the lower bound of the FNE rate of the whole learning process [11] (the rate of false-positive errors can be lowered in the SS phase). Based on this, it was suggested recently [11,12] that the problem of super-structure estimation should be addressed as a whole, since it is a key step to improve the accuracy and scalability of the inference of BN structures from data.

In this paper, we present two optimized CB methods for the task of super-structure estimation: the Optimized Zero-First-Order

* Corresponding author. Tel. +55 16 33739350.
E-mail addresses: carlos.maciel@usp.br, maciel@sc.usp.br (C.D. Maciel).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2012.10.035

Super-Structure (Opt01SS) and the Optimized Hybrid Parents and Children[1] (OptHPC). Opt01SS is constructed to get a safe super-structure in situations of limited data. For this end, only zero and first order CI tests are performed in a way that facilitates the identification of promising edges under the presence of approximate deterministic relationships (ADRs) [17] and inconsistent CI tests [18], common problems in small sample scenarios. The second algorithm, OptHPC, is a computational optimization of the recent Hybrid Parents and Children algorithm (HPC) [17], a sound local method proposed to ameliorate the large FNE rates that most CB methods present in small data. Among the optimizations are the use of a cache and a global graph to store and share zero-first order CI computations, which aid to gain efficient in the super-structure learning (a direct application of HPC to this task leads to many repeated computations). The performance of the algorithms is experimentally compared against various state-of-the-art methods, including the HPC, the Max-Min Parents and Children (MMPC) [10], the Heuristics PC algorithm (HeuPC)[2] [8], the GetPC algorithm [15], and the conventional zero-first-order CI tests (01SS) [9]. Results on benchmark networks show the suitability of our algorithms in the super-structure estimation task.

In the next section we give a brief revision of the main principles and concepts used in the paper. Sections 3 and 4 describe respectively the Opt01SS and OptHPC algorithms. Section 5 details our experimental setup and analyzes the obtained results. Finally, Section 6 presents our conclusions.

2. Preliminaries

Only discrete random variables and complete datasets are considered in this paper. We use upper-case, X, to denote a single variable, and the same lower-case, x, for its value. For a variable set, we use upper-case bold-face, $\mathbf{Z}$, and lower-case bold-face, $\mathbf{z}$, for a particular assignment. We use $X \perp Y | \mathbf{Z}$ to denote that variables X and Y are CI given a variable set $\mathbf{Z}$.

Formally, A BN [19] is a model $\langle G, \Theta \rangle$ for representing the joint probability distribution P of a set of random variables $\mathbf{U} = \{X_1, \ldots, X_n\}$. G is a directed acyclic graph — DAG (model structure) whose nodes have a one-to-one correspondence to the random variables in $\mathbf{U}$ and edges represent conditional dependence relationships among variables. Θ is a set of parameters that defines for each node X_i a conditional probability distribution $P(X_i | \mathbf{Pa}_i)$, where $\mathbf{Pa}_i$ denotes the parents of X_i in G. All BN satisfies the *Markov condition (MC)* [19]: every node X_i is conditionally independent on any subset of its non-descendants given its parents $\mathbf{Pa}_i$. From this condition, the joint distribution over $\mathbf{U}$ can be efficiently factored as: $P(\mathbf{U}) = \prod_{X_i \in \mathbf{U}} P(X_i | \mathbf{Pa}_i)$.

A BN encodes a set of CI statements that can be identified from its DAG by using the *d-separation* criterion [19]: two nodes X and Y are *d-separated* by a subset of nodes $\mathbf{Z}$ in a DAG G, denoted by $Dsep_G(X, Y | \mathbf{Z})$, if every path between X and Y is *blocked* by $\mathbf{Z}$. A path l is *blocked* by $\mathbf{Z}$ if: (i) $\mathbf{Z}$ contains a node V that is in l in the form $U \to V \to W$ or $U \leftarrow V \to W$; or (ii) l constains a 'collider' $U \to V \leftarrow W$ with V and all its descendants out of $\mathbf{Z}$. Each d-separation, $Dsep_G(X, Y | \mathbf{Z})$, readed from a BN DAG G implies a CI in P in the form $X \perp Y | \mathbf{Z}$. The converse, however, is not necessarily true. A BN that entails all and only the CIs in P by the d-separation criterion (i. e., $Dsep_G(X, Y | \mathbf{Z}) \Leftrightarrow X \perp Y | \mathbf{Z}$) is said to be a *faithful* BN of the distribution P. A *faithful distribution* P is one for which exists a faithful BN $\langle G, \cdot \rangle$ and G is said to be a *perfect map* of P.

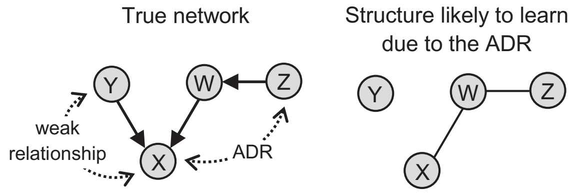

Fig. 1. Example to illustrate how an ADR (X and Z) may hamper the detection of a true edge (Y–X) that has a weak support on the data. This edge is missed when testing $Y \perp X | Z$ and it is accepted (due to the strong influence that Z exert on X and the faint dependency between X and Y).

All CIs readable from a BN with the d-separation criterion are outlined by its *skeleton* (the graph resulting of undirecting the DAG) and *v-structures*, i.e., colliders $U \to V \leftarrow W$ with U and W being not adjacent in the DAG (U and W are called *spouses*). Thus, two BNs with equal skeletons and v-structures encode the same CIs, they are said *equivalent models* [20].

The *Markov blanket* of a variable X in a BN, $\mathbf{MB}_X$, is the minimal set of variables that make X independent from the rest of variables in the BN, given $\mathbf{MB}_X$. In a faithful BN, the MB of a variable X is composed by the parents of X, the children of X (denoted by $\mathbf{Ch}_X$) and the spouses of X (denoted by $\mathbf{SP}_X$).

We adopt the super-structure concept [11]: *a sound or complete super-structure of a DAG G is any undirected graph S that contains the skeleton of G, otherwise it is said incomplete*.

3. Optimized zero-first-order super-structure (Opt01SS)

Opt01SS was conceived to perform a safe and efficient estimation of super-structures in limited data situations. Two main problems make the learning task challenging when the sample is small: (i) presence of approximate deterministic relationships (ADRs)[3] [21,17]; and (ii) inconsistency in CI testing [18]. The first problem leads to commit errors in the learning process due to the unreliability of CI tests under ADRs [17]. To illustrate this, consider the structure in Fig. 1, in which Z and X have an ADR and X and Y have a relationship with weak support on the data. If a CI test is performed between these latter variables conditioned on Z, the most likely result would be $Y \perp X | Z$, since the values of Z exert such strong influence on X that hide the dependency of Y on X (because the very few samples supporting this dependency), thus missing the edge $Y \to X$.

The second problem is originated by the uncertainties of the small sample. Such uncertainties are reflected in the learning process as inconsistent conditional independence and dependence statements — CIDS [18] (statements that cannot be simultaneously represented in a perfect map). To exemplify, consider a BN: $X \to Y \leftarrow Z$, in which, due to the scarcity of the data, two inconsistent CIs are detected: $X \perp Z | Y$ and $Y \perp Z | X$ (the rest of CI tests give dependency). Most CB methods would end learning only the edge X–Y, since they normally remove all edge associated with a CI [6,13,8,14]. If such methods are used as super-structure estimators in a hybrid learning approach, the true network would never be achieved in the SS stage.

To overcome the referred problems, Opt01SS (Algorithm 1) implements 3 phases, the first two face the ADR problem and the third one faces the CI inconsistencies. Phase 1 (1–5 lines) begins with a fully connected undirected graph S and iteratively remove edges with marginal independence (as in PC and HeuPC [6,8]). Function *Dep* (Table 1) implements the CI testing, returning the degree of dependence, $dep_{X,Y|\mathbf{Z}}$, between variables X and Y given

[1] Some preliminary results of OptHPC were presented in [16].

[2] In the original paper the Heuristics PC algorithm is abbreviated as *AlgorithmHPC*, but here we call it as HeuPC to avoid confusion with the HPC, another algorithm referenced in the paper.

[3] An ADR [17] is a strong association between two variables, where only a small portion of samples exhibit a non-deterministic relation for that variables.