Knowledge-Based Systems 62 (2014) 84–97

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Compatible and incompatible abstractions in Bayesian networks

Barbaros Yet *, D. William R. Marsh

School of Electronic Engineering and Computer Science, Queen Mary, University of London, London, UK

ARTICLE INFO

Article history:
Received 6 August 2013
Received in revised form 9 January 2014
Accepted 26 February 2014
Available online 12 March 2014

Keywords:
Bayesian networks
Knowledge engineering
Abstraction
Knowledge-based models
Graphical probabilistic models

ABSTRACT

The graphical structure of a Bayesian network (BN) makes it a technology well-suited for developing decision support models from a combination of domain knowledge and data. The domain knowledge of experts is used to determine the graphical structure of the BN, corresponding to the relationships and between variables, and data is used for learning the strength of these relationships. However, the available data seldom match the variables in the structure that is elicited from experts, whose models may be quite detailed; consequently, the structure needs to be abstracted to match the data. Up to now, this abstraction has been informal, loosening the link between the final model and the experts' knowledge. In this paper, we propose a method for abstracting the BN structure by using four 'abstraction' operations: node removal, node merging, state-space collapsing and edge removal. Some of these steps introduce approximations, which can be identified from changes in the set of conditional independence (CI) assertions of a network.

1. Introduction

A knowledge-based Bayesian network (BN) aims to model the data-generating process of a problem domain by encoding knowledge about influences and independences between the important variables of the domain. The first step in building the BN is for a knowledge engineer to elicit the structure of the BN from domain experts. When the structure is finalised, any available data can be used to learn the parameters of the BN or, if no data are available, the parameters can also be elicited. This paper is about the knowledge engineering techniques used in the first stage of this process: the development of the BN structure.

Knowledge-engineered BNs are often developed through multiple stages as the knowledge engineers and the domain experts refine the model iteratively [8]. The initial knowledge model is often large and detailed, and some elements of the model may need to be simplified or *abstracted* as data is lacking or the parameters are too difficult to elicit. However, even simple abstraction operations, such as removing a node, can result in numerous and complicated alternative BNs which are difficult for the knowledge engineers to evaluate without a structured method. The effects of these abstractions must be carefully examined by domain experts to prevent any unwanted changes in the modelled knowledge of the data generating process. Moreover, the way that the final BN has been derived needs to be presented thoroughly so that the knowledge base of the model and its derivation is understandable.

Our aim is to present a method of abstracting a BN structure. The method is developed for knowledge engineers developing a BN structure with domain experts. The method provides a set of abstraction operations which together:

1. Allow a BN to be simplified by removing and merging nodes, removing edges and reducing the number of states.
2. Distinguish abstractions that add to the knowledge base from those compatible with the knowledge elicited so far, so that the added knowledge can be confirmed by domain experts.
3. Provide a way to show the link between the initial and abstracted models, in the form of a derivation that captures the complete sequence of abstraction operations.

The method can be used to help knowledge engineers to select the most suitable model refinements by evaluating alternative abstractions, in consultation with domain experts. The selection may also be guided by considering the availability of data or compatibility with causal relationships.

Our knowledge engineering method is based on well-known techniques mainly used for learning and inference problems [17,22,21,3]. Our main contribution is to explore knowledge engineering aspect of these operations.

* Corresponding author. Address: Risk and Information Management (RIM) Research Group, Room CS332, School of Electronic Engineering and Computer Science, West Square, Queen Mary, University of London, London E1 4NS, UK. Tel.: +44 20 7882 8027.

E-mail address: b.yet@qmul.ac.uk (B. Yet).

http://dx.doi.org/10.1016/j.knosys.2014.02.020
0950-7051/

The remainder of this paper is organised as follows: Section 2 reviews the related work. Section 3 gives an overview of the relation between knowledge and conditional independences (CI) in BNs. Section 4 introduces abstraction as a knowledge engineering method and Section 5 describes the abstraction operations of this method and examines their compatibility properties. These operations are illustrated by a medical case-study in Section 6. Section 7 shows the graphical representation of the abstraction operations, and Section 8 presents the conclusions.

2. Related work

In this section, we review the previous studies about abstraction and knowledge engineering of causal models and BNs. The abstraction methodology proposed in this paper was initially motivated by ABEL [12]. Section 2.1 discusses this motivation by illustrating the similarities and differences between ABEL and BNs. Section 2.2 reviews the existing knowledge engineering techniques for eliciting and abstracting a BN structure. Since this paper focuses on the BN structure, techniques for eliciting parameters are not discussed.

2.1. ABEL

ABEL [12,13] was a pioneering clinical expert system that was developed for diagnosing acid–base disturbances of patients. Given laboratory data about a particular patient, ABEL generates the relevant causal diagrams from its knowledge-base, which is called a patient specific model (PSM). It reasons by abstracting and elaborating these causal models to make diagnostic inferences, which is considered to be similar to how clinicians express their decisions. The causal models at the higher levels are directed acyclic graphs like BNs. Although widely referenced, PSMs have not become a commonly used approach for developing clinical decision support models.

The causal diagrams of ABEL have several differences from the BN formalism. First, each node in a PSM represents a single state of a variable, whereas each node in a BN represents a variable that can have multiple states. Second, the lower abstraction levels of a PSM can have feedback loops which are always eliminated at higher levels; but BNs are acyclic graphs. Third, a PSM does not reason probabilistically and its reasoning mechanism does not take the prior probabilities of diseases into account; BNs have superior probabilistic reasoning algorithms that are able to calculate complex learning and inference problems. Fourth, BNs are lacking techniques for abstracting their knowledge-base for different levels of detail as used in ABEL. Abstraction is clearly necessary for developing knowledge-based BNs for complex domains, and for explaining these models to external users. Our work in this paper was initially motivated by ABEL, notably its hierarchical structure and abstraction operations.

2.2. Knowledge engineering approaches for BNs

Wu and Poh [23] propose a set of operations that change the abstraction level of knowledge-based influence diagrams. The ‘extend’ and ‘retract’ operations respectively add and remove the parents of a variable. The ‘abstract’ operation merges a set of variables that share a single parent and child. The ‘refine’ operation is the opposite of ‘abstract’, dividing a variable with a single parent and child into multiple variables. These operations can be applied to limited and simple modelling tasks. For example, Wu and Poh [23] do not discuss merging variables that do not share parents.

Srinivas [20] proposes a hierarchical BN approach for the fault diagnosis problem in engineering systems. In this approach, functional schematics can be defined in multiple levels of abstraction between the inputs and outputs of the system. Shachter’s node removal operation [17] is used to reach to higher level schematics. The different abstraction levels of schematics must have the same inputs and outputs.

Neil et al. [10] use specific BN fragments called idioms for representing common types of uncertain reasoning. Knowledge engineers and domain experts select the most appropriate idioms for their modelling problems and use these idioms as building blocks for their BN structure. Idioms are reused for similar modelling tasks in order to develop BNs efficiently and consistently. Koller and Pfeffer [6] describe object-oriented Bayesian Networks (OOBN), representing BNs with inter-related objects. OOBN are particularly useful for complex models that contain repeated fragments, where objects can be reused to decrease the modelling effort. Laskey and Mahoney [7] also use object-oriented concepts to construct a BN by using semantically meaningful fragments as the basic building blocks. Laskey and Mahoney [8] propose a system engineering approach that uses a spiral lifecycle model for the development of BNs. Their approach starts by defining objectives and building initial prototypes with simple features. These prototypes are evaluated and rebuilt. As this process proceeds the knowledge engineer understands the domain and the domain experts understand the principles of BN. The systems engineering approach uses network fragments [7] as basic elements of model building.

Heckerman [5] describes similarity networks that can be used for diagnosing a single hypothesis that has mutually exclusive and exhaustive states. In this approach, each pair of similar hypotheses is connected in a similarity network. A separate BN network structure is elicited for each pair of these similar hypotheses. Then, the separate BN structures are merged to form the final BN structure. This approach divides the task of network building into pieces that are easier to manage. However, it can only be applied when the hypotheses are mutually exclusive and exhaustive and the hypothesis variable has no parents. Parent divorcing approach can reduce the parameter space by adding a variable to the BN [11]. As the parameter space of a variable increases exponentially with the number of its parents, adding an intermediate variable between the variable and its parents can make parameter space smaller.

3. Conditional independences in BNs

3.1. Bayesian networks

A BN represents a joint probability distribution compactly in a factorised way. The structure of a BN is a directed acyclic graph that consists of nodes representing variables and directed edges encoding a set of CI conditions about these variables. Every node in a BN is independent of its non-descendants given that the state of its parents is known. Therefore, each node has a conditional probability distribution that defines its probabilistic relation with its parents. A probability distribution P_X factorises over a BN structure G_X if P_X can be decomposed into the product of factors $P_X = P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | PA_{X_i}^{G_X})$ where $X_1, \ldots, X_n$ are a set of variables, $PA_{X_i}^{G_X}$ is the set of parents of X_i in G_X.

We say that G_X asserts the set of CIs $I(G_X)$. P_X can factorise on G_X if $I(G_X)$ is a subset of $I(P_X)$ where $I(P_X)$ is the set of CIs in P_X.

3.2. Domain knowledge and conditional independences

The aim of a knowledge-based BN is to model the data-generating process for the domain by encoding knowledge about influences and independences in the BN structure. A satisfactory