Recent Research Based On Analytical Concept for Data with Structural Machine Learning For Object Oriented

Dr. Amit Kumar Chandanan
Department of Computer Science and Information Technology, Guru Ghasidas Vishwavidyalaya, Bilaspur, India

Corresponding: Dr. Amit Kumar Chandanan

Abstract

Using structural learning algorithms to combine database knowledge with prior information from domain experts can be helpful when building a Bayesian network. If the prior information is too hazy to be encoded as properties that are local to families of variables, conventional learning algorithms cannot easily incorporate it. In object-oriented fields like computer networks, large pedigrees, and genetic analysis, for instance, conventional algorithms do not take advantage of prior information about repetitive structures. A strategy for conducting structural learning in object-oriented domains is presented in this paper. It is argued that this method supports a natural approach for expressing and incorporating prior information provided by domain experts, and it is demonstrated that this method is more efficient than conventional algorithms in such domains.

Keywords: Knowledge fusion, object orientation, structural learning, and Bayesian networks
DOI Number: 10.14704/nq.2022.20.9.NQ44618 **Neuro Quantology 2022; 20(9):5320-5325**

1. INTRODUCTION

The framework of the Bayesian network (BN) (Pearl, 1988;In many areas of artificial intelligence, it has established itself as a potent instrument (Jensen, 1996, 2001). However, obtaining a BN from a domain expert can be time-consuming and laborious. As a result, approaches to learning a BN's structure from data have received a lot of attention in recent years. See Buntine (1996) and Krause (1998) for an overview. The structure of database-based BNs has been successfully learned using current learning techniques. Sadly, however, only a small amount of the prior information provided by domain experts is incorporated into these methods. Because prior information is typically encoded by specifying a prior BN, it is limited to the presence or absence of edges between particular pairs of variables.

In domains that an object-oriented language can accurately describe (Mahoney and Laskey, 1996; Mathiasen and others,2000), we typically discover substructures that are either repetitive or that are able to naturally be arranged in a super class–subclass hierarchy. The expert is typically able to provide information about these properties for such

domains. However, the aforementioned practice makes it difficult for current learning strategies to make use of this information.

According to Mahoney and Laskey object-oriented versions of the BN framework have recently been proposed in the literature. Even though these object-oriented frameworks alleviate some of the issues that arise when modeling large domains, eliciting the model's parameters and structure may still be challenging. Although offer a solution for effectively learning the parameters of an object-oriented domain model, the issue of specifying the structure persists.

Based on the OOBN framework, we present a strategy for conducting structural learning in an object-oriented domain in this paper. We contend that OOBNs provide a natural framework for encoding prior information regarding the domain's general structure. Furthermore, we demonstrate how structural learning can benefit from this kind of prior knowledge. In object-oriented domains, empirical findings demonstrate that the proposed learning algorithm is more effective than conventional learning algorithms.

2. OBJECT ORIENTED BAYESIAN NETWORKS

When constructing large Bayesian networks, a common strategy is to construct a complex model by employing small, "easy-to-read" pieces as building blocks.

As the formal foundation for the proposed learning strategy, the framework of Bangs and Wuillemin (2000b) will be discussed in the following paragraphs. Take note that the parts of the framework that are relevant to the learning algorithm are the only ones covered in the description.

2.1 The OOBN Framework

Assume that we are interested in modeling the impact of the environment on the production of milk and meat by two milk cows and two meat cows on a farm. In accordance with the object-oriented concept, we develop a generic cow class that enumerates the fundamental characteristics shared by all cows (Figure 1):In particular, because we are interested in the production of milk and meat, we make Milk and Meat the class's output nodes (shown by shaded ellipses).That is, nodes from a class that can be used outside of the class's instantiations. We designate Mother and Food as input nodes (depicted by dashed ellipses) of the class, assuming that both a cow's mother and the food it consumes have an impact on the production of milk and meat. A reference to a node that is not included in the class's instantiations is known as an input node. Normal nodes are nodes that are neither input nor output nodes. It is important to keep in mind that an instantiation's interface to its context is made up of its input and output nodes. All nodes are assumed to be discrete throughout the remainder of this paper.

2.2 Algorithm 1 (Underlying BN)

1. BNI ought to be the blank graph.
2. Add a node to BNI for each normal, output, and input node in I. Prefix the name of the instantiation with the node name (INSTANTIATION-NAME) and add a node to BNI for each of the instantiations encapsulated in I's input, output, and normal nodes. Name of node: Apply the same procedure to the instantiations that are contained in these instantiations, and so on.
3. Repeat as previously for all instantiations, adding a link for each normal link in I.
4. Merge all of the nodes in each reference tree into a single node. According to the normal links, this node's family includes all of the reference tree's parents and children. Because all other nodes are references to this one, only the tree's root can have parents.

   A normal node with a default potential will be created from an input node that does not reference any other nodes.

*e*ISSN1303-5150

This is also shown in Figure 3, which shows the underlying BN of a Stock-class instantiation.

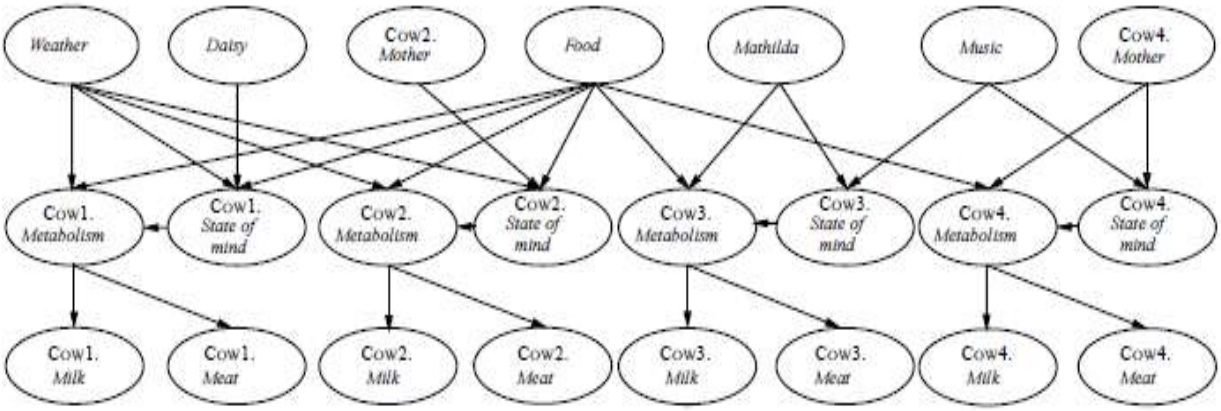

Figure 1 The underlying BN for the OOBN depicted in Figure.

It is important to keep in mind that default potential nodes (COW2.Mother and COW4.Mother) can be marginalized out of the underlying BN because they do not have any effect. Additionally, it is important to emphasize that an OOBN is merely a compact representation of a (unique) BN that satisfies the OO assumption, namely the underlying BN (as is evident right away from Algorithm 1).

3 LEARNING FROM INCOMPLETE DATA

Assumption 3 of the BD metric (and the BDe metric) is likely to be violated in real-world problems because we rarely have access to a complete database. As a result, known closed-form expressions cannot be used to determine the data's marginal likelihood because the model's parameters become dependent. When attempting to approximate the integral in Equation 1, these approximations are based on the assumption that the posterior over the parameters is peaked and that the maximum a posteriori (MAP) parameters are utilized. Therefore, we must locate the MAP parameters in order to apply these approximations, for example by employing the expectation-maximization (EM) algorithm, before we can determine a model's score. As a result, in order to evaluate each candidate model, we may need to devote a significant amount of time.

The Structural EM (SEM) algorithm, as described by Friedman (1998), basically "fills in" the missing values prior to searching the joint space of network structures and parameters. As a result, we avoid the time-consuming and costly step of calculating the MAP parameters for each candidate model. The legitimacy of the SEM calculation depends with the understanding that the information is absent (Close to nothing and Rubin, 1987), which is likewise accepted in the rest of this paper. Informally, this indicates that the missingness pattern may only be affected by the observed variable values7. The SEM algorithm maximizes $P(D, B_S/\varepsilon)$but instead of maximizing this score directly it maximizes the expected score. Let o be the set of observations from the database D, and let h be the set of unobserved entries in D. The general algorithm can then be outlined as:

Algorithm 2 (SEM)

Loop for n = 0; 1; : : : until convergence

Compute the posterior $P\left(\theta_{B\,\frac{n}{S}}/B_S^n, o\right)$

1) **E-step:** For each BS, compute:

$$Q(B_s: B_S^n) = E_h[log\ P\ (h, o, B_s)/B_S^n, o]$$

$$= \sum_h P(g/o\ , B_S^n) log P(h, o, B_S).$$

2) **M-step:** Choose $B_S^{n+1} \leftarrow B_S$that maximizes $Q(B_s: B_S^n)$:

$$Q(B_S^n: B_S^n) = Q(B_S^{n+1}: B_S^n)\ then$$

3) If return B_S^n

In the E-step, the algorithm completes the database by "filling-in" the unobserved entries based on the observations o, the current best model B_S^n and the posterior over the parameters for B_S^n(calculated in step 1). From the completed database the best candidate model is then selected in the M-step, which ensures that $Q\left(B_S^{l+1}: B_S^l\right) - Q\left(B_S^l: B_S^l\right) \geq 0$. According to Friedman (1998), the algorithm converges when the expected score is increased at every iteration, resulting in a better marginal score for the network.

4. SPECIFYING PRIOR INFORMATION

A prior distribution over the discrete space of all candidate structures and a prior distribution over the continuous space of probability parameters for each model represent the domain's prior information when learning a Bayesian network. The use of prior information regarding the structure of BNs and OOBNs will be the primary focus of this section, following a brief description of a prior for the probability parameters in Section 3.

4.1 Structural Priors in OOBNs

The additional sources of prior information that are accessible when learning in object-oriented domains are the subjects that this section examines. The OOBN framework, as we will argue, is a natural language for specifying prior information. As we will see, for a significant portion of the model space, the fundamental object-oriented modeling assumptions naturally result in zero prior probabilities.

4.2 Relations Among Variables

The domain expert typically has the ability to group the variables into substructures with high internal coupling and low external coupling when modeling object-oriented domains. An OOBN instantiation naturally corresponds to these substructures. In addition, instantiations of the same kind are categorized in a manner that is analogous to the classification of similar substructures .A collection of substructures, for instance, might refer to the same kind of physical object or they might describe a collection of entities that take place at the same time.

A (partial) OOBN specification—also known as a prior model—can represent these kinds of prior data. An OOBN's a priori specification includes a list of class specifications and a classification of the nodes into instantiations based on the classes. Following that, this prior OOBN model can be utilized in the same manner as conventional prior models, and we can in principle use any of the definitions of $P(X_i, \prod i)$outlined above.

The domain expert may have difficulty indicating the presence or absence of edges between particular nodes in the model when specifying the variables' relations. The domain expert may be unsure whether another node Z in the encapsulating context of I should be the parent of X or Y, even though he believes that Z should influence at least one of them, for instance, if two variables X and Y in an instantiation I are strongly correlated. The interface between the instantiation I and its encapsulating context can be specified in the OOBN framework to encode this prior information. The domain expert, for instance, can indicate which instantiations are permitted to reference a particular node and, more importantly, which are not (see Figure 2). Particularly, questions such as "Do you think it is possible that a variable Z directly influences any of the variables in instantiation I?" could be posed to the domain expert.

*e*ISSN1303-5150

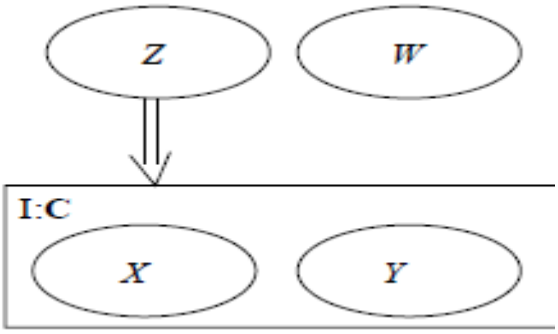

Figure 2 The figure depicts a possible way to describe knowledge about the structure of the domain.

5 THE EMPIRICAL RESULTS

Take into consideration the insurance network's OOBN version once more, as described in Section 2. Also keep in mind the brief experiment that our research group conducted to obtain object-oriented prior information in this domain. The experiment's objective was to locate OOBN edges that we could a priori disregard. As a result, only 253 of the 702 possible network structure-incorporating edges were designated as possible, including all 52 actual network structure-incorporating edges. We used the algorithm to see how much this previous information could help the search algorithm based on this experiment.

Figure 3a depicts the empirical results for the insurance domain. The object oriented prior information regarding the interfaces was coded as absolutely certain $\left(w_{ij}^{+} = \infty \; if \; an \; edge \; X_i \rightarrow X_j\right)$ required a larger interface than given by the prior information). The results based on "OO priors" are superior to those based on "conventional priors" (that is, the standard), as the KL divergence decreases with the level of information available to the learning algorithm. This is consistent with what was anticipated.

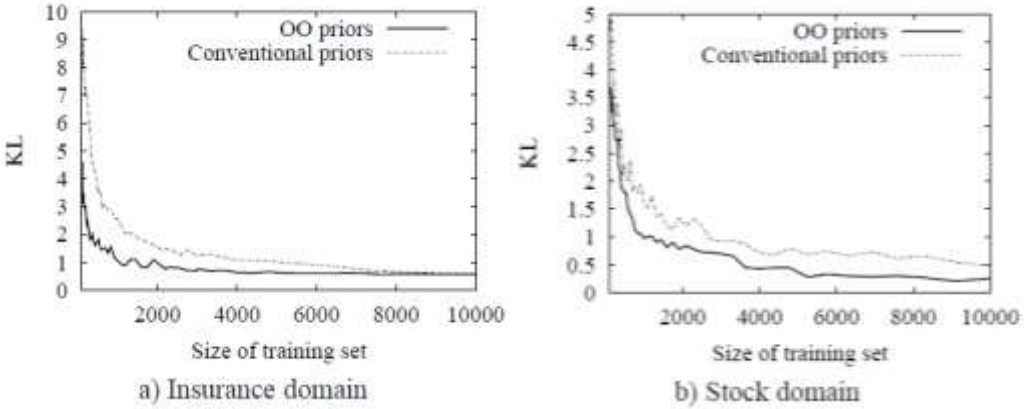

Figure 3 The KL divergence of the gold standard model vs. the generated models

6 CONCLUSION

A method for structural learning in object-oriented domains has been proposed in this paper. The proposed learning algorithm exploits an intuitive way of expressing prior information in object-oriented domains and was shown to be more efficient than conventional learning algorithms in this setting. It has been implemented using a tailor-made version of the Structural EM algorithm. The learning algorithm is based on the OOBN framework.

REFERENCES

1. Olav Bangsø, Helge Langseth, and Thomas D. Nielsen. Structural learning in object oriented domains. In Proceedings of the Fourteenth International Florida Artificial

Intelligence Research Society Conference, pages 340–344. AAAI Press, 2001.
2. Olav Bangsø and Pierre-Henri Wuillemin. Object oriented Bayesian networks. A framework for topdown specification of large Bayesian networks with repetitive structures. Technical report CIT-87.2-00-obphw1, Department of Computer Science, Aalborg University, 2000a.
3. Olav Bangsø and Pierre-Henri Wuillemin. Top-down construction and repetitive structures representation in Bayesian networks. In Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference, pages 282–286. AAAI Press, 2000b.
4. Langseth et al, "Fusion of Domain Knowledge with Data for Structural Learning", JMLR 2003.
5. John Binder, Daphne Koller, Stuart Russell, and Keiji Kanazawa. Adaptive probabilistic networks with hidden variables. Machine Learning, 29(2–3):213–244, 1997.
6. Wray L. Buntine. A guide to the literature on learning probabilistic networks from data. IEEE Transactions on Knowledge and Data Engineering, 8:195–210, 1996.
7. Philipp Bachy, Victor Chernozhukov, Malte S. Kurz, Martin Spindler, "DoubleML – An Object-Oriented Implementation of Double Machine Learning in Python", arXiv:2104.03220v2 [stat.ML] 20 Dec 2021.
8. Gil Gonçalves, Umberto Andriolo, Luísa Gonçalves, Paula Sobral and Filipa Bessa, "Quantifying Marine Macro Litter Abundance on a Sandy Beach Using Unmanned Aerial Systems and Object-Oriented Machine Learning Methods", Remote Sens. 2020, 12, 2599; doi:10.3390/rs12162599.
9. Amandeep Kaur1, Sushma Jain, Shivani Goel and Gaurav Dhiman, "A Review on Machine-Learning Based Code Smell Detection Techniques in Object-Oriented Software System(s)", Recent Advances in Electrical & Electronic Engineering, 2021, 14.
10. Andrea Tassi and Marco Vizzari, "Object-Oriented LULC Classification in Google Earth Engine Combining SNIC, GLCM, and Machine Learning Algorithms", Remote Sens. 2020, 12, 3776; doi:10.3390/rs12223776.