Constructing Bayesian Networks Automatically using Ontologies

Ann Devitt, Boris Danev and Katarina Matusikova
Network Management Research Centre
Ericsson, Ireland.
E-mail: {ann.devitt, boris.danev, katarina.matusikova}@ericsson.com

Abstract. Bayesian Networks are probabilistic structured representations of domains which have been applied to monitoring and manipulating cause and effects for modelled systems as disparate as the weather, disease and mobile telecommunications networks. Although useful, Bayesian Networks are notoriously difficult to build accurately and efficiently which has somewhat limited their application to real world problems. Ontologies are also a structured representation of knowledge, encoding facts and rules about a given domain. This paper outlines an approach to harness the knowledge and inference capabilities inherent in an ontology model to automate the construction of Bayesian Networks to accurately represent a domain of interest. The approach was implemented in the context of an adaptive, self-configuring network management system in the telecommunications domain. In this system, the ontology model has the dual function of knowledge repository and facilitator of automated workflows and the generated BN serves to monitor effects of management activity, forming part of a feedback look for self-configuration decisions and tasks.

Keywords: Bayesian Network, Inference, Knowledge representation

1. Introduction

In today’s world of digital compression and storage, there is a wealth of data accessible from a vast range of domains and topics and in a huge variety of formats and structures. Over the last number of years, there have been great strides made in how this data can be accessed, indexed and searched. The challenge remains how such data can be exploited as knowledge, facilitating and enhancing new and existing applications. Ontologies have emerged as a means of providing a structured representation of knowledge which can range from generic real world to strictly domain-specific. The purpose of employing an ontological representation is to capture concepts in a given domain in order to provide a shared common understanding of this domain, enabling interoperability and knowledge reuse but also machine-readability and reasoning about information through inferencing. They are deterministic in nature, consisting of concepts and facts about a domain and their relationships to each other. Bayesian Networks have emerged as a means of estimating complex probabilities of states based on graphical models of a domain. They also are a structured representation of knowledge and specify the relationships between concepts (or variables) of a domain. These relationships denote the dependencies and independencies that hold between the concepts or variables. They are probabilistic in nature, encoding the probability that variables assume particular values given the values of their parent variables in the Bayesian Network structure. These two tools for knowledge representation and manipulation have independently been used to facilitate machine reasoning and decision-making. This paper describes an approach to harness the knowledge representation and inference capabilities of ontologies in order to construct automatically a Bayesian Network which accurately represents a given domain and can then be used to support machine decision-making processes. The research is being undertaken as part of a program to develop adaptive, self-configuring functionality for devices within the mobile telecommunications network management domain [Baliosian et al., 2006]. Within this context, the ontology model is designed to provide the self-configuring functionality, facilitating automation of configuration workflows. It also serves as a repository of knowledge for the construction of a machine learning Bayesian Network component. The Bayesian Network component is designed to
provide the *adaptive* functionality, monitoring and learning the effects of configuration actions and closing the feedback loop on management activity.

Section 2 gives a brief introduction to Bayesian Networks and outlines current approaches to Bayesian Network construction and how ontologies have been exploited to date for this purpose. Section 3 sets out how in this research the structure and inference capabilities of ontologies have been exploited to automate the construction of a Bayesian Network. Section 4 describes an implementation of this approach in the telecommunications network management domain. Finally, section 5 discusses some conclusions of this research and directions for future work.

2. Background

Korb and Nicholson [2004, p.21] state that the ultimate goal of Bayesian AI, of which Bayesian Networks are an integral part, is to “create a thinking agent which does as well or better than humans in such [reasoning] tasks, which can adapt to stochastic and changing environments, recognize its own limited knowledge and cope sensibly with these varied sources of uncertainty.” Bayesian Networks provide a means of capturing existing knowledge about a domain, learning the stochastic properties of that domain and thereby adjusting its model of the domain over time. They are currently being exploited in several domains notably for estimating effects of different types of behaviour and as support for human or automated decision tasks. Some sample applications include using BNs to reduce power consumption of machines with reference to user behaviour [Harris and Cahill, 2005] or to diagnose faults in industrial processes [Arroyo-Figueroa and Sucar, 1999]. The ultimate goal of an autonomous thinking agent is not yet realised but BNs have become state-of-the-art for modelling, monitoring and adapting stochastic processes.

BNs consists of a Directed Acyclic Graph (DAG) structure. The nodes of this graph represent variables from an application domain, for example, performance counters in a telecommunications network or weather indicators in the climate domain. The arcs represent the dependencies that hold between these variables, for example, a drop in parameter X triggers alarm Y or high atmospheric pressure is associated with warm weather. Additionally, there is an associated conditional probability distribution over these variables which encodes the probability that the variables assume their different values given the values of their parent variables in the BN graph structure. For example, the probability of alarm Y being triggered when parameter X is above a given threshold is \( p = 1 \) or the probability of the weather being good when the atmospheric pressure is high is \( p = 0.65 \). It should be noted that the arcs of the Bayesian Network do not necessarily denote a causal relationship between two variables but only that the distribution of the child variable values is dependent on its parents value. In some instances, this may be a causal relationship but not in all cases. Figure 1 shows a sample Bayesian Network for a set of eight variables from the telecommunications network domain. It consists of a Key Performance Indicator (KPI) for a telecommunications device, the performance counters which contribute to that KPI, a service workflow which is triggered by degradation in the KPI levels and two temporal variables, day of the week and peak time. It is a structure such as this which the approach outline in this paper aims to build based on an ontology model of the telecommunications domain.

The task of building the structure and assigning the probability distributions of a Bayesian Network is complex and knowledge-intensive. It requires the identification of relevant statistical variables in the application domain, the specification of causality relations between these variables and assignment of initial probability distributions [Druzdzel and der Gaag, 2000]. Both the structure and parameters, or probability distribution, of such a network can be assigned by an expert or learnt off-line from historical data. The parameters may also be learnt on-line incrementally from a live feed of data. Parameter estimation is a mature field of study and several algorithms exist to derive conditional probability tables for a fixed network.

---

1Terminology: Ontologies represent knowledge in terms of concepts and relations. Ontological concepts, in Bayesian Network terms, are domain variables which can take certain values and have an associated probability distribution which are represented as nodes in the BN graph. In this paper, we use the terms concept, variable and node interchangeably to denote concepts in the ontology model and variables in the Bayesian Network directed acyclic graph.
structure efficiently and accurately from data [Spiegelhalter and Lauritzen, 1990, Geman and Geman, 1984, Dempster et al., 1977]. In more recent years, structure learning for Bayesian Networks has become a hot topic in the data mining community. Initial BN applications involved defining the network structure and learning the parameters from data. There are a number of methodologies proposed for facilitating building BNs by hand [Laskey and Mahoney, 2000, Neil et al., 2000] and indeed most BN software tools today include a GUI component for defining BN structures. However, in addition to expert knowledge in the application domain, the human may require some understanding of the statistical principles and in particular the notion of conditional dependence and independence underlying a Bayesian Network in order to correctly specify relations between the variables in the domain. To address this knowledge bottleneck and the inherent difficulties of building BNs by hand, several algorithms have been developed to derive the structure of the network from data, such as the K2 algorithm [Cooper and Herskovits, 1991], MDL (Minimum Description Length) [Lam and Bacchus, 1993] and CAMML [Wallace et al., 1996]. While learning causal structure for Bayesian Networks can eradicate some of the bottlenecks which impede the application of BNs to real world problems, the learning algorithms are not without their own drawbacks such as making over-simplifying assumptions about the input data or output structure, inability to deal with missing data, requirement for huge input datasets or intractability for complex multivariate input data. Furthermore, it may be non-trivial to integrate existing knowledge sources which in today’s world of multiple data sources would constitute an unnecessary waste of resources.

Ontologies provide a potential knowledge source which could be exploited to build the BN structure. Helsper and der Gaag [2002] outline an approach which uses ontologies to facilitate the building of Bayesian Networks in the medical domain. However, the ontologies are used only as means of representing knowledge to facilitate the manual creation of the BN structure. The ontology constitutes shared and agreed domain knowledge to be used to derive the BN graph structure that is close to the ontological descriptions in the given domain. Due to the complexity of the medical domain and the high impact of
misclassifications, the derivation of the graph structure is still done manually by expert analysis. The following sections outline how the BN–from–ontology building process has been automated and enhanced using the inference capabilities offered by the formal ontological representation.

3. Automating Bayesian Network Construction

This section outlines how our approach uses the inheritance structure and inference capabilities of an ontology to build the structure of a Bayesian Network (BN). This addresses two difficulties associated with the construction of BNs:

– complexity of hand-coding BN structure, requiring both domain and BN knowledge;
– integration of existing knowledge sources.

Like a hand-coded approach, the algorithm relies wholly on expert knowledge, rather than evidence derived from data. However, here the work of the expert is simplified. They no longer have to master the principles of BNs, they must only classify their domain knowledge in the familiar ontological world of concepts and relations. While this is not a trivial task, it is a more straightforward one. Furthermore, where there is an existing ontology knowledge source, little extra input is required to build a BN for this domain.

The task of building a BN can be decomposed into several subtasks:

1. identify the variables of interest;
2. specify the values these variables can take;
3. define the relations between the variables;
4. assign a conditional probability distribution.

The following sections set out how the use of an ontology model facilitates each of these steps.

3.1. Identifying Variables of Interest

The approach described here assumes that an ontology of concepts for the domain of interest has been or can be built. Some of these concepts may be of interest to include in a BN which models causal relations in that domain and some may not. In order to distinguish between these, we have defined a new ontology of BN concepts and link this to the original domain ontology. All concepts of interest for the Bayesian Network then inherit from a node in this new BN ontology. The root concept of the BN ontology is the BNnode. In order to create the Bayesian Network, an instance of each leaf class which inherits from the BNnode class is created. The description of the generic BNnode concept, its properties and relations are set out in figure 2. The concept has two types of relations:

– **hasParentNode**: BN nodes have a directed link from themselves to at least one parent node;
– **hasDelayParentNode**: this is a directed time-delay link which can be used to generate a Dynamic Bayesian Network (a BN which includes a temporal dimension) [Nicholson, 1992].

When a BNnode instance is created, these relations define the influential links between this BNnode instance and other BNnode instances (see section 3.3). The BNnode concept properties listed in the figure constitute the set of possible attributes which a BN node can contain. These include name, conditional probability table (CPT), state names (for discreet variables, the list of values which the variable the node represents can take), levels (for continuous variables, the ranges of values which the variable can take).

When a BNnode instance of a domain concept is created, these attributes are derived from properties of that concept in the domain ontology (see section 3.2). By defining inheritance relations between concepts of interest in the domain ontology and the BNnode concept, it is possible to automate the creation of BN nodes, their attributes and the arcs that connect them, as set out in the sections that follow. The inheritance relation expresses that a class of the domain model is to be included as a node in the behaviour model. This combined ontology is enriched with facts which describe how a domain can be represented as a Bayesian Network.
The combined domain and BN ontology can be further enriched to constrain BN creation. In addition to the basic BNNode concept, the BN ontology may contain additional BN concepts which are more specific either to the BN application or to characteristics of the domain ontology. Figure 3 illustrates a simple ontology for the telecommunications network management application for which this approach has been implemented. The root concept of the BN ontology remains the BNNode. The domain ontology in this figure consists of the concepts domain:SubConceptOfNoInterest and domain:SubConceptOfInterest and their parent concept domain:Concept1. The domain:SubConceptOfInterest concept inherits both from the domain ontology and the BN ontology and only this concept node will be included in an output BN for this domain. In this figure, however, between the root node and a conceptOfInterest node, there are two additional, intermediate concepts: BehaviourModelNode and bmConcept1Node. The BehaviourModelNode concept represents the characteristics of BN nodes required for a particular application, in this case the network management application. This separation between pure BN and BN for an application allows the original generic BNNode ontology to be re-used for other applications which require a BN component by defining a different ApplicationNode concept. The Concept1Node concept defines characteristics of Concept1 instances which should be treated in a particular way. The ontology can define a hierarchy of more specific BNNode classes for any domain concepts which should be included in the output BN, if these concepts would benefit from additional processing. This additional level is not a requirement of this approach. However, the structure enables tailored processing of domain ontology concepts in the generation of the Bayesian Network, for example, setting ranges for continuous variables or default probability values.

---

Footnote: 2 See section 4 and Baliosian and Devitt [2006] for an outline of the application architecture and the BN “BehaviourModel” component function.
3.2. Specifying the Attributes of the Bayesian Network Node

The properties, specified in the BNnode concept, of the BN nodes created at the previous step must be derived from the combined BNnode and domain ontology. These properties include name, stateNames, kind of bayesian network node. This is done using the constraints offered by ontology restrictions. The domain concepts specify restrictions on their properties and these are used to populate the BNnode properties of newly created BN nodes. In particular, the hasValue restriction specifies the values which a property can assume. For example, the hasStateNames BNnode property, which all sub-concepts of BNnode inherit, can be constrained in the sub-concept class to a specific value of the domain ontology concept property using the hasValue restriction.

(1) 

\[
\text{<Restriction>}
\begin{align*}
\text{<onProperty hasStateNames/>} \\
\text{<hasValue "Present, Absent, Recent"/>}
\end{align*}
\text{</Restriction>}
\]

Other restrictions are used to control correct node notation. For example, the requirement that a node must have exactly one name is expressed by a cardinality restriction hasName property = 1. Figure 4 shows the restrictions on a sample EventNode for the telecommunications ontology, expressing for particular node properties what values this property can take for this class and its subclasses. The ontology inheritance structure allows some restrictions to be specified at a very generic level (e.g. notational restrictions) and others at a lower level in the ontology (e.g. value specifications), thereby maximising the generaliz-ability of the ontology. For each BNnode instance created, the ontology reasoner gets all restrictions and the BNnode properties are generated from this.

3.3. Finding Parent Nodes

In building the Bayesian network from the behaviour model ontology, the ontology reasoner is used over the hierarchy and its restrictions to create nodes in network and to populate the properties of those nodes. As noted above, an instance is created for each BehaviourModelNode subclass, in the domain model. At the same time, the algorithm also generates a node in the Bayesian Network representation with appropriate property values. To create arcs between these nodes, the algorithm relies on rules. These rules are specific to the application domain and define which ontology properties or relations between concepts correspond to arcs in the Bayesian Network. For example, in the medical domain, a disease may present as one or more symptoms, a single rule can express this causal relation from disease to symptoms for all sub-classes of disease and the symptoms associated with them. The rules are then used to generate arcs in
Section 4.3 details the rules used to generate BN arcs for the network management application. This rule-based approach over ontology classes provides a means of specifying generic BN relationships which are then generated automatically when the nodes are initialised. Finally, the reasoner is used to check that the generated Bayesian Network is valid by checking all BN node instances and the domain ontology for consistency.

3.4. Conditional Probability Table (CPT) Estimation

As noted in section 2, estimating CPTs for Bayesian Networks is and has been a field of intensive research for several decades. The approach set out in this paper does not delve into this area. Indeed the network management application for which this approach was designed exploits existing parameter learning algorithms and the Bayesian Network CPTs are learnt incrementally and on-line from a live feed of network event data. However, the knowledge resource of a domain ontology can be exploited to estimate initial probability distributions for some concepts that lend themselves to this interpretation. Some relations between parent and child nodes can be assigned an initial probability value based on the nature of the concepts involved. For example, a deterministic relation where the value of the parent entails the value of the child variable can be encoded directly in the CPT of the child variable. Like the arc construction method, this can be encoded as a rule in the ontology.

4. Application to Telecommunications Network Management domain

The work described here was conducted as part of a larger project to develop a self-adapting auto-configuration management functionality for network management systems in the mobile telecommunications domain. The area of adaptive, autonomous networks is currently the subject of intense research activity, not only in the telecommunications domain, for a number of reasons. The current explosion in size, complexity and heterogeneity of networks which has been driven by recent advances in wired and wireless networking technology is set to continue into the future with networks growing at an exponential rate. At the same time, network operators are struggling in a highly competitive market where they must keep their running costs to a minimum. More autonomy of networks, network devices and network management systems presents a means of resolving this conflict between the ever-increasing demands of running large, complex and heterogeneous networks and the ever-decreasing OPEX (operational expenditure) budgets of network operators. One crucial obstacle to increased autonomy which must be overcome is that today network management systems have only partial models of the networks they manage and these models are semantically empty. Currently, the Management Information Base (MIB) records the attributes of devices and the state of the network but it does not explicitly represent constraints that hold for individual managed elements and even less between elements of the MIB. Recent approaches propose using ontologies to capture network information models [de Vergara et al., 2003] and a key approach in this project is to use ontologies to annotate, constrain and make machine-readable the O&M descriptions of the network in order to enable automation of O&M activities, as in [Cleary and Danev, 2005]. However, a system which allows automation of management decisions and tasks could be prey to instability. A viable autonomous management solution must include a feedback loop so that it can observe and deal with the consequences of its activities. This can be provided using the on-line machine learning capabilities offered by a Bayesian Network to monitor the effects of O&M activities (human, semi-automated or fully automated) and feed the derived knowledge back into the management system, as described in Baliosian and Devitt [2006]. The ontology is modelled using OWL and uses the OWL-DL reasoner with additional JENA rules. The Bayesian Network software is the Netica API. The construction algorithm was implemented in Java.
4.1. Domain Ontology

The domain ontology model for a single network device stores the current configuration of the device, its relationships with other objects in the network and constraints on its possible configuration imposed by the hardware and software deployed on the network element. It also stores the workflows associated with configuration tasks, i.e., the sequence of actions affecting a network element that need to be completed in order to fulfil a given task. In addition, it models the performance and fault metrics associated with that node (i.e. alarm types, performance counters and KPIs) and any associations between these (e.g. KPI equations, alarm triggers). A subsection of this ontology which was modelled in OWL is shown in figure 5. This ontology subsection is focused on the Service concept and other concepts connected with it. The relations between these concepts are expressed by directed links (blue arcs with arrows) representing object properties of concepts. Links lead from property domain to property range concepts. The primary goal of this model was to facilitate automation of management tasks. However, it has also been exploited to provide an accurate BN representation of the domain to be used for monitoring effects of these management tasks.

![Fig. 5. Part of Telecommunications Network Management Domain Ontology](image.png)

4.2. Behaviour Model Ontology

The BehaviourModelNode is the root class for any node to be included in the Bayesian Network. Below this node, there is a hierarchy of more specific node classes for each node type to be included (KPI, Performance Parameter, Service, and Event) to allow custom processing of the different node types. A part of the combined Bayesian Network and Domain ontology for the application domain is shown in figure 6 where is-a links represents the inheritance hierarchy. The ontology defines two classes, ServiceNode and EventNode, as subclasses of BehaviourModelNode to describe properties of the service and event domain concepts which are specific to a Bayesian Network representation. For example, all service nodes in the Bayesian network share the same state names. This data and other shared property values are recorded using hasValue restrictions on the corresponding ServiceNode properties. Likewise, all event nodes share the same state names, different from service nodes state names. The EventNode subclass contains this information in the form of hasValue restrictions. Every service and event which is of interest to the Bayesian Network inherits from the ServiceNode subclass and the EventNode subclass respectively.
4.3. Rules for BN arc construction

In order to generate arcs automatically from the domain ontology we use rules in the Jena rule language. Rules specify how to create arcs from relations between domain concepts. The reasoner infers hasParent and hasDelayParent relations from inter-concept relations such as those represented by the blue arcs in figure 6. After rule inference, the hasParent relations appear in the behaviour model ontology as shown in figure 7. Example 2 shows a sample rule for generating arcs between the Event and Service concepts.

\[
\text{Service-Event arc rule:} \\
(\text{?s type Service}) \quad \text{// if there is a service} \\
(\text{?e type Event}) \quad \text{// and an event} \\
(\text{?s ?p ?e}) \quad \text{// that is in relation with this service} \\
\rightarrow \quad \text{// then} \\
(\text{?e hasParent ?s}) \quad \text{// the event has the service as a parent} 
\]

Likewise, each KPI has its relevant performance parameters defined in the ontology as properties of the KPI concept. A generic rule for all KPIs generates Bayesian Network arcs to each KPI from their associated performance parameters. If there is no relation defined between classes in the domain ontology, it is also possible to define rules that explicitly specify arc creation. This final model is checked for consistency and recreated as a Bayesian Network, such as the BN shown in figure 1, using the Netica API.

5. Conclusions and Future Work

This paper has outlined an approach to building a Bayesian Network from an ontology model of a given domain. Bayesian Networks are notoriously difficult to hand-code and structure learning algorithms, while useful, can have significant drawbacks. The use of a domain ontology coupled with the capabilities of an inference engine can automate the BN building task, reducing the knowledge bottleneck of expert knowledge to BN structure, while accurately representing the domain of interest. The approach was implemented in the context of an adaptive, self-configuring network management system in the telecommunications domain. In this system, the ontology model has the dual function of knowledge repository and automation facilitator and the generated BN serves to monitor effects of management activity and forms part of a feedback loop for self-configuration decisions and tasks.
This approach opens up several avenues for future work, the first of which is an evaluation of the current system. However, the evaluation of BN structures is a non-trivial task and estimation of the success of this ontology-based approach would require both a subjective and an objective evaluation. The subjective evaluation must compare how the task is perceived by the ontology or BN builders to assess whether there has been any saving in the time and effort of domain experts. The objective evaluation should assess the quality of the generated structure by performing a comparison of the ontology-built structure and other data-learnt models on the basis of a selected metric, such as predictive accuracy for an expert–annotated test data set.

Other technical extensions are also planned. To date, the implemented algorithm does not specify any values for the BN conditional probability tables. In future implementations, we aim to specify CPT priors on the basis of properties of the ontology model. For example, the service workflows which are composed of events imply that the service is active if at least one of its events is present, this could be encoded in the event CPT. Similarly, the triggering of services by KPI violations can be encoded in the service CPT as a deterministic relationship $p_{service} = 1$ when $KPI \geq threshold$. Another more complex direction for future research involves modification of the ontology-built structure by supplementing additional arcs or removing superfluous ones on the basis of learnt data. This is an area ripe for research as existing methodologies entail learning an entirely new structure from data using the original structure as a prior in the learning process. This research direction should also provide interesting insights into the primacy of
expert knowledge, in the form of ontologies, over information learnt from data as the degree and kinds of modification required are an indicator of the (in)accuracies of the expert model.

References


