Reliability Engineering and System Safety 94 (2009) 1499–1509

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Inference in hybrid Bayesian networks

Helge Langseth [a,*], Thomas D. Nielsen [b], Rafael Rumí [c], Antonio Salmerón [c]

[a] Department of Information and Computer Sciences, Norwegian University of Science and Technology, Norway
[b] Department of Computer Science, Aalborg University, Denmark
[c] Department of Statistics and Applied Mathematics, University of Almería, Spain

ARTICLE INFO

Article history:
Received 11 September 2008
Received in revised form
10 February 2009
Accepted 11 February 2009
Available online 13 March 2009

Keywords:
Bayesian networks
Reliability
Hybrid models
Inference

ABSTRACT

Since the 1980s, Bayesian networks (BNs) have become increasingly popular for building statistical models of complex systems. This is particularly true for boolean systems, where BNs often prove to be a more efficient modelling framework than traditional reliability techniques (like fault trees and reliability block diagrams). However, limitations in the BNs' calculation engine have prevented BNs from becoming equally popular for domains containing mixtures of both discrete and continuous variables (the so-called *hybrid* domains). In this paper we focus on these difficulties, and summarize some of the last decade's research on inference in hybrid Bayesian networks. The discussions are linked to an example model for estimating human reliability.

1. Introduction

A reliability analyst will often find himself making decisions based on uncertain information. Examples of decisions he may need to take include defining a maintenance strategy or choosing between different system configurations for a safety system. These decisions are typically based on only limited knowledge about the failure mechanisms that are in play and the environment the system will be deployed in. This uncertainty, which can be both aleatory and epistemic, requires the analyst to use a statistical model representing the system in question. This model must be mathematically sound, and at the same time easy to understand for the reliability analyst and his team. To build the models, the analyst can employ different sources of information, e.g., historical data or expert judgement. Since both of these sources of information can have low quality, as well as come with a cost, one would like the modelling framework to use the available information as efficiently as possible. Finally, the model must be encoded such that the quantities we are interested in (e.g., the availability of a system) can be calculated efficiently.

All of these requirements have led to a shift in focus, from traditional frameworks, like fault trees, to more flexible modelling frameworks. One such framework for building statistical models for complex systems is the Bayesian network (BN) framework [1–3]. BNs have gained popularity over the last decade [4], partly because a number of comparisons between BNs and the classical reliability formalisms have shown that BNs have significant advantages [5–10].

BNs consist of a qualitative part, an *acyclic directed graph*, where the nodes mirror stochastic variables and a quantitative part, a set of conditional probability functions (CPFs). An example of the qualitative part of a BN is shown in Fig. 1. This BN models the risk of an explosion in a process system. An explosion (*Explosion*?) might occur if there is a leak (*Leak*?) of chemical substance that is not detected by the gas detection (GD) system. The GD system detects all leaks unless it is in its failed state (*GD Failed*?). The environment (*Environment*?) will influence the probability of a leak as well as the probability of a failure in the GD system. Finally, an explosion may lead to a number of casualties (*Casualties*?).

The graphical structure has an intuitive interpretation as a model of causal influences. Although this interpretation is not necessarily entirely correct, it is helpful when the BN structure is to be elicited from experts. Furthermore, it can also be defended if some additional assumptions are made [11].

BNs originated as a robust and efficient framework for reasoning with uncertain knowledge. The history of BNs in reliability can (at least) be traced back to [12,13]; the first real attempt to use BNs in reliability analysis is probably the work of Almond [13], where he used the GRAPHICAL-BELIEF tool to calculate reliability measures concerning a low pressure coolant injection system for a nuclear reactor (a problem originally addressed by Martz [14]). BNs constitute a modelling framework which is particularly easy to use in interaction with domain experts, also in the reliability field [15]. BNs have found

* Corresponding author.
E-mail addresses: helgel@idi.ntnu.no (H. Langseth), tdn@cs.aau.dk (T.D. Nielsen), rrumi@ual.es (R. Rumí), antonio.salmeron@ual.es (A. Salmerón).

0951-8320/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ress.2009.02.027

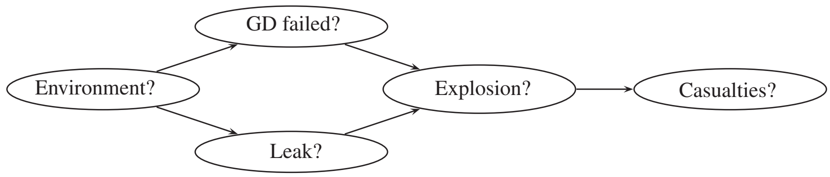

Fig. 1. An example BN describing a gas leak scenario. Only the qualitative part of the BN is shown.

applications in, e.g., fault detection and identification, monitoring, software reliability, troubleshooting systems, and maintenance optimization. Common to these models are that all variables are *discrete*. As we shall see in Section 3, there is a purely technical reason why most BN models fall in this class. However, in Section 4 we introduce a model for human reliability analysis, where both discrete and continuous variables are in the same model. Attempts to handle such models are considered in Section 5 and we conclude in Section 6.

2. Preliminaries

Mathematically, a BN is a compact representation of a joint statistical distribution function. A BN encodes the probability density function governing a set of variables by specifying a set of conditional independence statements together with a set of CPFs.

For notational convenience, we consider the variables $\{X_1,\ldots,X_n\}$ when we make general definitions about BNs in the following, and we use the corresponding lower-case letters when referring to instantiations of these variables. Now, we call the nodes with outgoing edges pointing into a specific node the *parents* of that node, and say that X_j is a *descendant* of X_i if and only if there exists a directed path from X_i to X_j in the graph. In Fig. 1, *Leak*? and *GD Failed*? are the parents of *Explosion*?, written $\mathrm{pa}(\textit{Explosion?}) = \{\textit{Leak?}, \textit{GD Failed?}\}$ for short. Furthermore, $\mathrm{pa}(\textit{Casualties?}) = \{\textit{Explosion?}\}$. Since there are no directed paths from *Casualties*? to any of the other nodes, the descendants of *Casualties*? are given by the empty set and, accordingly, its non-descendants are $\{\textit{Environment?}, \textit{GD Failed?}, \textit{Leak?}, \textit{Explosion?}\}$. The edges of the graph represent the assertion that a variable is conditionally independent of its non-descendants in the graph given its parents in the same graph. The graph shown in Fig. 1 does for instance assert that for all distributions compatible with it, we have that $\{\textit{Casualties?}\}$ is conditionally independent of $\{\textit{Environment?}, \textit{GD fails?}, \textit{Leak?}\}$ when conditioned on $\{\textit{Explosion?}\}$.

When it comes to the quantitative part, each variable is described by the CPF of that variable *given its parents* in the graph, i.e., the collection of CPFs $\{f(x_i|\mathrm{pa}(x_i))\}_{i=1}^n$ is required. The underlying assumptions of conditional independence encoded in the graph allow us to calculate the joint probability function as

$$f(x_1,\ldots,x_n) = \prod_{i=1}^{n} f(x_i|\mathrm{pa}(x_i)), \tag{1}$$

i.e., that the joint probability distribution can be completely expressed as the product of a collection of local probability distributions. This is in fact the main point when working with BNs: assume that a distribution function $f(x_1,\ldots,x_n)$ factorizes according to Eq. (1). This defines the parent set of each X_i, which in turn defines the graph, and from the graph we can read off the conditional independence statements encoded in the model. As we have seen, this also works the other way around, as the graph defines that the joint distribution *must* factorize according to Eq. (1). Thus, the graphical representation is bridging the gap between the (high level) conditional independence statements we want to encode in the model and the (low level) constraints this enforces on the joint distribution function. After having established the full joint distribution over $\{X_1,\ldots,X_n\}$ (using Eq. (1)), any marginal distribution $f(x_i,x_j,x_k)$, as well as any conditional distribution $f(x_i,x_j|x_k,x_\ell)$, can in principle be calculated using extremely efficient algorithms. These will be considered in Section 3.

We will use $\mathbf{X} = \{X_1,\ldots,X_n\}$ to denote the set of variables in the BN. If we want to make explicit that some variables are observed, we use $\mathbf{E} \subset \mathbf{X}$ to denote the set of observed variables, and $\mathbf{e}$ will be used for the observed value of $\mathbf{E}$. We will use Ω_{X_i} to denote the possible values a variable X_i can take. If X_i is discrete, then Ω_{X_i} is a countable set of values, whereas when X_i is continuous, $\Omega_{X_i} \subseteq \mathbb{R}$. For $\mathbf{X} = \{X_1,\ldots,X_n\}$ we have $\Omega_{\mathbf{X}} = \times_{i=1}^n \Omega_{X_i}$.

3. Inference

In this paper we will only consider a special type of inference, namely the case of updating the marginal distributions of some variables of interest given that the values of some other variables are known, e.g., to compute the conditional density of $X_i \in \mathbf{X}\backslash\mathbf{E}$ given the observation $\mathbf{E} = \mathbf{e}$, denoted $f(x_i|\mathbf{e})$. Observe that

$$f(x_i|\mathbf{e}) = \frac{f(x_i,\mathbf{e})}{f(\mathbf{e})},$$

and since the denominator $f(\mathbf{e})$ does not depend on x_i, the inference task is therefore equivalent to obtaining $f(x_i,\mathbf{e})$ and normalizing afterwards. A brute force algorithm for carrying out this type of inference could be as follows:

(1) Obtain the joint distribution $f(x_1,\ldots,x_n)$ using Eq. (1).
(2) Restrict $f(x_1,\ldots,x_n)$ to the value $\mathbf{e}$ of the observed variables $\mathbf{E}$, thereby obtaining $f(x_1,\ldots,x_n,\mathbf{e})$.
(3) Compute $f(x_i,\mathbf{e})$ from $f(x_1,\ldots,x_n,\mathbf{e})$ by marginalizing out every variable except X_i.

A problem with this naïve approach is that the joint distribution is usually unmanageably large. For instance, assume a simple case in which we deal with 10 discrete variables that have three states each. Specifying the joint distribution for those variables would be equivalent to defining a table with $3^{10} - 1 = 59\,048$ probability values, i.e., the size of the distribution grows exponentially with the number of variables. For instance, for 11 variables, the size of the corresponding table would increase to $3^{11} - 1 = 177\,146$, and so on. Models used in reliability domains commonly consist of hundreds or thousands of variables, and this naïve inference approach is simply not able to handle problems of this size.

The inference problem can be simplified by taking advantage of the factorization of the joint distribution encoded by the structure of the BN, which supports the design of efficient algorithms for this task. For instance, consider the network shown in Fig. 2, which is structurally equivalent to the model shown in Fig. 1; the variables are labelled $X_1,\ldots,X_5$ for notational convenience.

For this example, assume that we are interested in X_5, that all variables are discrete, and that $\mathbf{E} = \emptyset$. By starting from the joint