Engineering Applications of Artificial Intelligence 25 (2012) 698–704

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Availability modelling of repairable systems using Bayesian networks

Martin Neil [a,b,*], David Marquez [a]

[a] Department of Computer Science, Queen Mary University of London, London E1 4NS, UK
[b] Agena Ltd., 32 Hatton Garden, London EC1N 8DL, UK

ARTICLE INFO

Article history:
Received 19 November 2009
Received in revised form
11 May 2010
Accepted 8 June 2010
Available online 3 July 2010

Keywords:
Bayesian networks
Bayesian software
Systems dependability
Dynamic discretization
Renewable systems
Systems availability

ABSTRACT

We present a hybrid Bayesian network (HBN) framework to model the availability of renewable systems. We use an approximate inference algorithm for HBNs that involves dynamically discretizing the domain of all continuous variables and use this to obtain accurate approximations for the renewal or repair time distributions for a system. We show how we can use HBNs to model corrective repair time, logistics delay times and scheduled maintenance time distributions and combine these with time-to-failure distributions to derive system availability. Example models are presented and are accompanied by detailed descriptions of how repair (renewal) distributions might be modelled using HBNs.

1. Introduction

Complex systems are required to be dependable in use and one important aspect of a system's dependability is availability. Availability is intrinsically uncertain and is typically defined and measured as the probability of the system being available for use at a given point in time. A system might be unavailable if it has failed and is awaiting repair or the system has failed and is undergoing repair before re-entering service. Over a given time period, a system might therefore be available or unavailable depending not only on the system's reliability but also on how well the support organisation might affect the rate of repair and the duration of such repairs (renewals). Additionally systems may also undergo preventative maintenance usually on a scheduled basis, and we might extend our analysis to consider the modes of failure, the subsystem failure rates, maintenance regimes and different methods of logistical support. Maintenance (renewal time) and reliability (failure time) are stochastic variables and it therefore makes sense to model these using appropriate statistical inference techniques. We could then predict future behaviour and make decisions about the acceptability of the availability one might expect to experience in a given system. For an overview of availability theory, concepts and models see Stapelberg (2009).

We have used Bayesian networks (BNs) in a range of real-world applications of system dependability assessment; see for example Neil et al. (2001, 2003, 2008). In such applications, it is inevitable that there will be a mixture of discrete and continuous nodes (the resulting BNs are called hybrid (HBNs)). The traditional approach to handling (non-Gaussian) continuous nodes is static: you have to discretize the continuous domains using some pre-defined range and intervals. However, this approach is unacceptable for critical systems, where there is a demand for reasonable accuracy. To overcome this problem, we have developed a new and powerful approximate algorithm for performing inference in HBNs. We use a process of dynamic discretization of the domain of all continuous variables in the HBN. The approach to discretizing the domain is influenced by the work of Kozlov and Koller (1997) using an entropy error as the basis for approximation. We differ from their approach by integrating an iterative approximation scheme within existing BN software architectures, such as in junction tree (JT) propagation (Jensen et al., 1990). Thus, rather than support separate data structures and a new propagation algorithm, we use the data structures commonly used in JT algorithms.

The power and flexibility of the approach are demonstrated here by applying it to estimate the availability of repairable systems represented by a series of models each designed to model distinct stages in the renewal process: logistics delays, repairs and scheduled maintenance. Traditionally, modelling these events has relied on Monte Carlo simulation, involving many repeated simulation runs. In contrast to the simulation approach, we show how our HBN algorithms can be used to represent repair and support processes and the durations involved, under any

* Corresponding author at: Department of Computer Science, Queen Mary University of London, London E1 4NS, UK. Tel.: +44 020 7882 5200.
E-mail address: martin@dcs.qmul.ac.uk (M. Neil).

0952-1976/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2010.06.003

assumptions for renewal time distributions (lognormal, exponential, etc.). The modelling has been made possible using the commercial general-purpose Bayesian network software tool AgenaRisk, Agena Ltd. (2010).

2. Bayesian networks

BNs have been widely used to represent full probability models in a compact and intuitive way. In the BN framework, the independence structure in a joint distribution is characterised by a directed acyclic graph, with nodes representing random variables (which can be discrete or continuous, and may or may not be observable), and directed arcs representing causal or influential relationship between variables (Pearl, 1993). The conditional independence assertions about the variables, represented by the lack of arcs, reduce significantly the complexity of inference and allow the underlying joint probability distribution to be decomposed as a product of local conditional probability distributions (CPDs) associated with each node and its respective parents. If the variables are discrete, the CPDs can be represented as node probability tables (NPTs), which list the probability that the child node takes on each of its different values for each combination of values of its parents. Since a BN encodes all relevant qualitative and quantitative information contained in a full probability model, it is an excellent tool for many types of probabilistic inference, where we need to compute the posterior probability distribution of some variables of interest (unknown parameters and unobserved data) conditioned on some other variables that have been observed.

A range of robust and efficient propagation algorithms has been developed for exact inference on Bayesian networks with discrete variables (Pearl, 1988; Shenoy and Shafer, 1990; Jensen et al., 1990). The common feature of these algorithms is that the exact computation of posterior marginal distributions is performed through a series of local computations over a secondary structure, a tree of clusters, enabling calculation of the marginal without computing the joint distribution. See also Huang and Darwiche (1996).

The present generation of BN software tools attempt to model continuous nodes by numerical approximations using static discretization as implemented in a number of software tools (e.g., Hugin, 2005; Netica, 2005). Although discretization allows approximate inference in a hybrid BN without limitations on relationships among continuous and discrete variables, current software implementations require users to define a uniform discretization of the states of any numeric node (whether it is continuous or discrete) as a sequence of pre-defined intervals, which remain static throughout all subsequent stages of Bayesian inference regardless of any new conditioning evidence. The more intervals you define, the more accuracy you can achieve, but at a heavy cost of computational complexity. This is made worse by the fact that you do not necessarily know in advance, where the posterior marginal distribution will lie on the continuum for all nodes and which ranges require the finer intervals. It follows that where a model contains numerical nodes having a potentially large range, results run the risk of being only crude approximations.

Alternatives to discretization have been suggested by Moral et al. (2001) and Cobb and Shenoy (2006), who describe potential approximations using mixtures of truncated exponential (MTE) distributions, and by Murphy (1999) who uses variational methods. There have also been some attempts for approximate inference on hybrid BNs using Markov Chain Monte Carlo (MCMC) approaches (Shacter and Peot, 1989). However, constructing dependent samples that mixed well (i.e., that moves rapidly throughout the support of the target posterior distribution) remains a complex task.

3. Dynamic discretization

3.1. Inference and discretization

In this paper, inference is carried out using a standard BN propagation algorithm (Lauritzen and Speigelhalter, 1988; Jensen et al., 1990). Unfortunately, for hybrid BNs containing mixtures of discrete and continuous nodes with non-Gaussian distributions, exact inference becomes computationally intractable. The traditional approach to handling (non-Gaussian) continuous nodes is static discretization. This requires the user to define a uniform discretization of the domains of all continuous nodes, using some pre-defined range and intervals. The discretization remains static throughout all subsequent stages of exact inference performed on the resulting discrete BN. The more intervals you define, the more accuracy you can achieve, but at a heavy cost of computational complexity. The level of accuracy of this approach is also constrained by the feasibility of identifying the high-density regions for each variable in the model, and this needs to be done in advance of any inference taking place. This is cumbersome, error prone and, where a model contains numerical nodes having a potentially large range, results are necessarily only crude approximations.

Let X be a continuous random node in the BN. The range of X is denoted by Ω_X, and the probability density function (PDF) of X, with support Ω_X, is denoted by f_X. The idea of discretization is to approximate f_X as follows:

- partition Ω_X into a set of intervals $\Psi_X=\{w_j\}$, and
- define a locally constant function $\tilde{f}_X$ on the partitioning intervals.

Discretization operates in much the same way when X takes integer values, but in this paper we will focus on the case where X is continuous. As Kozlov and Koller (1997), we use an upper bound of the Kullback–Leibler (KL) metric between two density functions f and g:

$$D(f\|g)=\int_S f(x)\log\frac{f(x)}{g(x)}\,dx$$

as an estimate of the relative entropy error induced by the discretized function. Under the KL metric, the optimal value for the discretized function $\tilde{f}$ is given by the mean of the function f in each of the intervals of the discretized domain. The main task reduces then to finding an optimal discretization set $\Psi_X=\{\omega_j\}$.

Our approach to dynamic discretization searches Ω_X for the most accurate specification of the high-density regions given the model and the evidence, calculating a sequence of discretization intervals in Ω_X iteratively. At each stage in the iterative process, a candidate discretization, Ψ_X, is tested to determine whether the relative entropy error of the resulting discretized probability density $\tilde{f}_X$ is below a given threshold, defined according to the tradeoff between the acceptable degree of precision and computation time.

By dynamically discretizing the model, we achieve more accuracy in the regions that matter and incur less storage space over static discretizations. Moreover we can adjust the discretization any time in response to new evidence to achieve greater accuracy. A detailed description of the dynamic discretization algorithm is given in Neil et al. (2007). In an outline, dynamic discretization follows these steps:

1. Convert the BN to a junction tree (JT) and choose an initial discretization for all continuous variables.
2. Calculate the NPT of each node given the current discretization.