# Modeling Dependable Systems using Hybrid Bayesian Networks

Martin Neil[†‡], Manesh Tailor[‡], David Marquez[†], Norman Fenton[†‡] and Peter Hearty[†]

† *Department of Computer Science, Queen Mary, University of London*

‡ *Agena Ltd*

## Abstract

*A hybrid Bayesian Network (BN) is one that incorporates both discrete and continuous nodes. In our extensive applications of BNs for system dependability assessment the models are invariably hybrid and the need for efficient and accurate computation is paramount. We apply a new iterative algorithm that efficiently combines dynamic discretisation with robust propagation algorithms on junction tree structures to perform inference in hybrid BNs. We illustrate its use on two example dependability problems: reliability estimation and diagnosis of a faulty sensor in a temporal system. Dynamic discretisation can be used as an alternative to analytical or Monte Carlo methods with high precision and can be applied to a wide range of dependability problems.*

## 1. Introduction

We have used Bayesian Nets (BNs) in a range of real-world applications of system dependability assessment (see for example [1, 2, 3] In such applications it is inevitable that there will be a mixture of discrete and continuous nodes (the resulting BNs are called hybrid). The traditional approach to handling (non-Gaussian) continuous nodes is static: you have to discretise them using some pre-defined range and intervals. However, this approach is unacceptable for critical type 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 hybrid BNs. We use a process of *dynamic discretisation* of the domain of all continuous variables in the BN. The approach is influenced by the work of [4] using 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 [5]. 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 is demonstrated by applying it to two dependability problems:

- Estimating the reliability of a single system
- Diagnosing whether a sensor is faulty from a sequence of observed readings

These problems represent a very simplified version of fragments of the wide range of models we have implemented as part of commercial and research projects. These have been in areas as diverse as data fusion, parameter learning, discrete systems simulation, RAM (Reliability, Availability and Maintainability) evaluation and software defect prediction. The modeling has been made possible because our dynamic discretisation algorithm has recently been implemented in the commercial general-purpose Bayesian Network software tool AgenaRisk [6].

## 2. Background

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 characterized 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

[7]. 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* (*CPD*) associated with each node and its respective 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 [7, 8, 9, 10]. The common feature of these algorithms is that the exact computation of posterior marginals 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 [11].

The present generation of BN software tools attempt to model continuous nodes by numerical approximations using static discretisation as implemented in a number of software tools [12, 13]. Although disctretisation 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 discretisation 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 are necessarily only crude approximations.

Alternatives to discretisation have been suggested by [14, 15], who describe potential approximations using mixtures of truncated exponential (MTE) distributions, [16] who combine MTE approximations with direct sampling (Monte Carlo) methods, and [17] who uses variational methods. There have also been some attempts for approximate inference on hybrid BNs using Markov Chain Monte Carlo (MCMC) approaches [18]. However, constructing dependent samples that mixed well (i.e., that move rapidly throughout the support of the target posterior distribution) remains a complex task.

## 3. Dynamic Discretisation

Let us denote by $\Omega_X$ the state space or range of a continuous valued node $X$ in the BN. The idea of discretisation is to approximate the marginal probability density of $X$ as follows:

1. Split $\Omega_X$ into $n$ mutually exclusive and exhaustive intervals, where

   $\Omega_X = [x_1, x_2], ]x_1, x_2], \ldots, ]x_{n-1}, x_n]$

   when $\Omega_X$ is finite and

   $\Omega_X = ]x_1, x_2], ]x_1, x_2], \ldots, ]x_{n-1}, x_n[$

   when $\Omega_X$ is infinite, and

2. Define a piecewise constant function in each subinterval.

Intervals can either be of uniform or variable width. Discretisation 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.

Our approach to dynamic discretisation searches $\Omega_X$ for the most accurate specification of the high-density regions, given the model and the evidence, calculating a sequence of discretisation intervals in $\Omega_X$ iteratively. At each stage in the iterative process a candidate discretisation, $\Psi_X$, is tested to determine whether the resulting discretised pdf $p(X)$ has approached the true pdf within an acceptable degree of precision. At convergence $\Omega_X$ is approximated by $\Psi_X$.

By dynamically discretising the model we achieve more accuracy in the regions that matter and incur less storage space over static discretisations. Moreover, we can adjust the discretisation any time in response to new evidence to achieve greater accuracy.

In outline, dynamic discretisation follows these steps:

1. Convert the BN to a JT and choose an initial discretisation for all continuous variables.

2. Calculate the discretised density of each continuous node given the current discretisation and propagate evidence through the BN.
3. Query the BN to get marginals for each node and split those intervals with highest entropy error in each node.
4. Repeat the process by recalculating the densities and propagating the BN, and then querying to get the marginals and then split intervals with highest entropy error.
5. Continue to iterate until the model converges to an acceptable level of accuracy.

## 4. Estimating System Reliability

Our first example considers the efficacy of the dynamic discretisation approach to estimate the reliability of a continuous use system. Using a noninformative conjugate prior distribution for the unknown failure rate, $\lambda$, we can solve for the posterior distribution, $p(\lambda \mid f,t)$, given $f$ failures in $t$ exposure time as follows:

$$p(\lambda) = Gamma(a,b)$$
$$p(f \mid \lambda,t) = Poisson(\lambda t)$$
$$p(\lambda \mid f,t) \propto p(f \mid \lambda,t)p(\lambda)$$
$$p(\lambda \mid f,t) = Gamma\left(a+f, \left[t+\frac{1}{b}\right]^{-1}\right)$$

With dynamic discretisation we do not need to assume conjugacy; hence the prior distribution for the rate could be any sensible shape and scale (a non-conjugate example, that uses an expert opinion as a prior, is provided with the evaluation version of AgenaRisk). However, to aid comparison we will use a conjugate example using the BN model shown in Figure 1. The failure rate, $\lambda$, is set as an non-informative Gamma prior distribution $p(\lambda) = Gamma(a = 0.001, b = 1000)$ and the Mean Time Between Failure(MTBF) is th reciprocal of the failure rate: $MTBF = 1/\lambda$. In addition to nodes for each of the variables defined above we have added an additional discrete node to test the requirement that the system will survive for more than 600 hours, thus transforming it to a hybrid BN. In Figure 1 the node labels are annotated with their relevant functions.

If we observe five failures in 2000 hours of use the estimated marginal distributions after 50 iterations in AgenaRisk are shown in Figure 2. The mean failure rate estimate, $E(\lambda)$, is 0.00249 and the probability of surviving 600 hours is 0.244. This compares very well with the analytical solution, $E(\lambda \mid f,t) = 0.0025$.

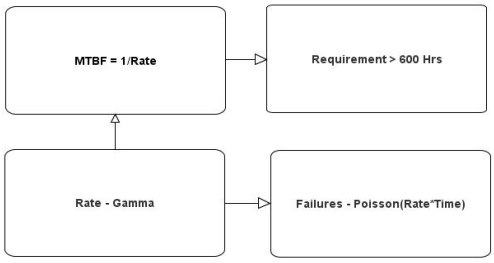


**Figure 1 BN for reliability estimation example**

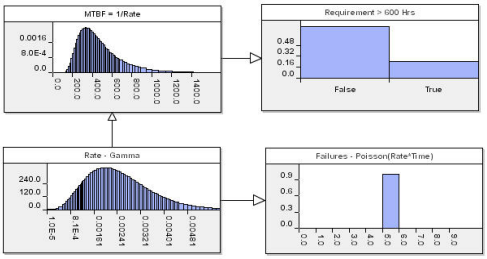


**Figure 2 BN for reliability estimation example with marginal distributions superimposed on the graph**

## 5. Faulty Sensor Diagnosis

Here we wish to estimate the probability of a sensor being in a faulty state at any point in time. We are given a sequence of sensor readings taken from a system whose state is represented by a position and velocity vector $(P_t, V_t)$ where the nodes are hidden and evolve over time. We also assume the system is self-repairing and that these repairs are random (or the fault might be transient).

We can model this as a Switching Kalman Filter Model (SKFM) and typically would have to use Monte Carlo or more complex methods to find a solution. In AgenaRisk we model this as a Dynamic BN (DBN) as follows.

The first element in the DBN is a double Kalman Filter to model the dynamical elements of the system. This comprises:

- *Observation model* $O_t$ to filter sensor noise from observations:

$$p(O_t \mid P_t) = Normal(P_t, \sigma^2)$$

- *Transition model* of two difference equations for $P_t$ and $V_t$:

$$P_t = P_{t-1} + V_{t-1},\ V_t = V_{t-1}$$

- *Initial conditions* for each of the system variables:

$$V_0 = N(0, \theta_1), P_0 = N(0, \theta_2)$$

All system and observation variables are integers to represent the use of a one-dimensional grid to track the system position.

To model the fact that the sensor is self-repairing we require a transition model for the sensor, $S_t$:

$$p(S_t = OK \mid S_{t-1} = OK) = 0.99$$
$$p(S_t = Faulty \mid S_{t-1} = OK) = 0.01$$
$$p(S_t = Faulty \mid S_{t-1} = Faulty) = 0.9$$
$$p(S_t = OK \mid S_{t-1} = Faulty) = 0.1$$

Sensor reliability is modelled in the observation model by conditioning the variance parameter, $\sigma^2$, on the sensor's state:

$$p(O_t \mid P_t, S_t = OK) = Normal(P_t, \sigma^2 = 10)$$
$$p(O_t \mid P_t, S_t = \neg OK) = Normal(P_t, \sigma^2 = 1000)$$

The DBN graph for the complete model over seven time periods is shown in Figure 3:

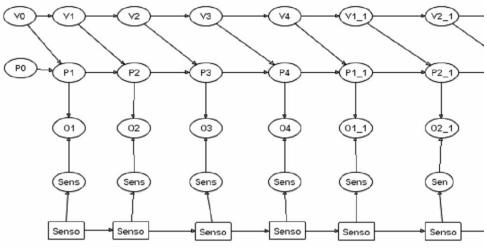


**Figure 3 DBN graph for the SKFM for sensor reliability for seven time periods**

We use dynamic discretisation to solve the model over 25 iterations for a set of simulated actual and observed positions. At each time period we estimate the probability of the sensor being in a faulty state, represented by $S_t$. The results are shown in Figure 4 along with the corresponding time series plot of $A_t$ (simulated actual), $O_t$ (observed) and $p(S_t = OK)$ (shown as %OK) .

Notice that the initial observations lead to a rapid increase in the probability that the sensor is faulty. However, after time period 8 the observed position better tracks the actual position and as a consequence the probability that it has repaired itself also increases.

| time | At | Ot | %OK |
|---|---|---|---|
| 0 | 9 | 10 | 99 |
| 1 | 12 | 20 | 96 |
| 2 | 19 | 17 | 92 |
| 3 | 22 | 30 | 74 |
| 4 | 24 | 20 | 50 |
| 5 | 25 | 20 | 35 |
| 6 | 31 | 10 | 4 |
| 7 | 34 | 25 | 4 |
| 8 | 43 | 40 | 24 |
| 9 | 45 | 47 | 29 |
| 10 | 59 | 55 | 31 |

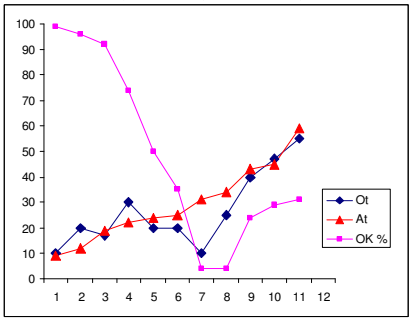


**Figure 4 Table and time series plot of system position and probability of sensor failure**

## 6. Concluding Remarks

We have provided an overview of a new approximate inference algorithm designed for a general class of hybrid BNs. This dynamic discretisation algorithm (implemented in the AgenaRisk software) finally frees BN modellers from the burden (and inaccuracies) associated with having to statically discretise continuous nodes. We have highlighted how this approach enables us to estimate reliability of a simple system. The results compare very favourably with analytical methods. It is a simple leap from this example to considerably more complex

examples, say involving families of systems modelled hierarchically or using censored data, where dynamic discretisation could provide alternative and perhaps better solutions to those provided by other approximate methods such as MCMC. This is amply demonstrated by the second example, which is an order of magnitude more complex than the first. Here we modelled a Kalman Filter Model (KFM) and then extended it to include conditioning discrete variables which themselves evolved over time, to produce a Switching Kalman Filter Model (SKFM). Typical solutions to this involve a variety of complex algorithms not found in discrete BN packages, yet a solution is easily and quickly produced using dynamic discretisation.

## References


[1] Neil M., Fenton N., Forey S., and Harris R. 2001. Using Bayesian Belief Networks to Predict the Reliability of Military Vehicles, IEE Computing and Control Engineering J 12(1), 11-20

[2] Neil M., Malcolm B., and Shaw R. 2003a. Modelling an Air Traffic Control Environment Using Bayesian Belief Networks, 21st International System Safety Conference, Ottawa, Ontario, Canada.

[3] Fenton NE, Marsh W, Neil M, Cates P, Forey S, Tailor T, "Making Resource Decisions for Software Projects", 26th Intl Conf on Software Engineering (ICSE 2004), IEEE Comp Society 2004, ISBN 0-7695-2163-0, pp. 397-406.

[4] Kozlov A.V., and Koller D. 1997. Nonuniform dynamic discretization in hybrid networks, in D. Geiger and P.P. Shenoy (eds.), Uncertainty in Artificial Intelligence, 13: 314–325.

[5] Jensen F., Lauritzen S.L., and Olesen K. 1990. Bayesian updating in recursive graphical models by local computations, Computational Statistics Quarterly, 4: 260-282

[6] Agena Ltd. 2005, www.AgenaRisk.com

[7] Pearl J. 1993. Graphical models, causality, and intervention, Statistical Science, vol 8, no. 3, pp. 266-273

[8] Lauritzen S.L., and Speigelhalter D.J. 1988. Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems (with discussion), Journal of the Royal Statistical Society Series B, Vol. 50, No 2, pp.157-224.

[9] Shenoy P., and Shafer G. 1990. Axioms for probability and belief-function propagation, Readings in uncertain reasoning, Morgan Kaufmann Publishers Inc, p.p. 575 - 610 .

[10] Lauritzen S.L., and Jensen F. 2001. Stable local computation with conditional Gaussian Distributions, Statistics and Computing, 11, 191–203.

[11] Huang C., and Darwiche A. 1996. Inference in belief networks: A procedural guide. Int. J. Approx. Reasoning 15(3): 225-263

[12] Hugin. 2005. www.hugin.com

[13] Netica. 2005. www.norsys.com

[14] Moral S., Rumı R., and Salmeron A. 2001. Mixtures of truncated exponentials in hybrid Bayesian networks, in P. Besnard and S. Benferhart (eds.), Symbolic and Quantitative Approaches to Reasoning under Uncertainty, Lecture Notes in AI, 2143, 156–167.

[15] Cobb B., and Shenoy P. 2005a. Inference in Hybrid Bayesian Networks with Mixtures of Truncated Exponentials, University of Kansas School of Business, working paper 294.

[16] Koller D., Lerner U., and Angelov D. 1999. A general algorithm for approximate inference and its applications to Hybrid Bayes Nets, in K.B Laskey and H. Prade (eds.), Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, pp. 324–333.

[17] Murphy, K. 1999. A variational approximation for Bayesian networks with discrete and continuous latent variables, in Laskey and H. Prade (eds.), Uncertainty in AI, 15, 467-475.

[18] Shacter R., and Peot M. 1989. Simulation approaches to general probabilistic inference on belief networks. In Proceedings of the 5th Annual Conference on Uncertainty in AI (UAI), pages 221-230.