Agena's Bayesian Network Technology

Summary

Bayesian Nets (BNs) are an increasingly popular formalism for reasoning and decision-making in problems that involve uncertainty and probabilistic reasoning. Full details about BNs, their benefits, and their applications can be found in [1]. This paper focuses on how Agena has taken BN technology on to a new level that opens up the power of BNs to a much wider range of users and applications. This technology is implemented in the AgenaRisk toolset. Specifically, AgenaRisk incorporates the latest research results and algorithms developed by Fenton and Neil and their team; these developments make it possible to build truly scalable and accessible solutions of much greater accuracy than was previously considered possible.

Background: BNs and their technology

A BN is a directed graph, like the one shown in Figure 1. The nodes represent variables (that may or may not be observable and are normally uncertain) and the arcs represent causal/influential relationships between variables. Associated with each node is a Node Probability Table (NPT), which expresses the conditional probability of each state of the node given each combination of values for the node parents.

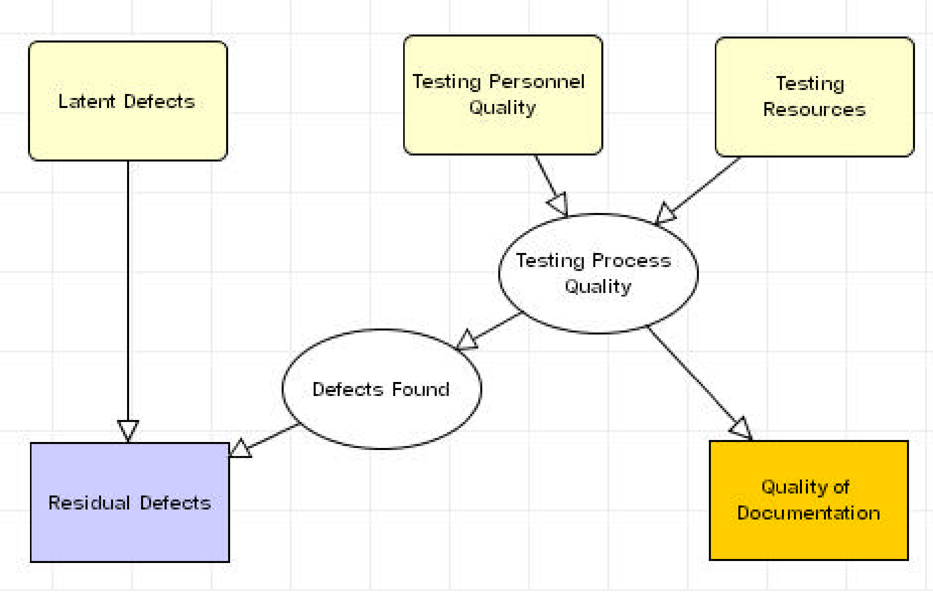

Figure 1 - BN representing a fragment of the software development process

With BNs we enter observations for any node to update the probability distributions of all the unknown variables. This process, called *propagation*, uses probability calculus and Bayes theorem. These techniques are hundreds of years old. Yet, although Bayesian probability has long been regarded by many as the best means of modelling uncertainty, it was not until the late 1980s that BNs could be effectively used on realistic problems. This is because Bayesian propagation is known to be computationally intractable (the computations grow exponentially with the number of nodes of the BN). In 1988 Lauritzen and Spiegelhalter [6] made a major breakthrough by producing a propagation algorithm that is efficient for many classes of very large BNs. This algorithm, and variations of it, were later implemented in tools such as Hugin and Netica. Hence, such tools made it possible to execute BNs efficiently and led to an explosion of interest in applying BNs to a wide range of problem domains. These tools are excellent for BN and statistical experts who are either working with pre-defined models (that can be large) or who are building small-scale models. However, they do not address the following three fundamental barriers that have acted as an impediment to more wide-scale BN adoption:

1. Difficulty (for both experts and non-experts) of *building* (as opposed to executing) and maintaining large-scale models necessary to solve real-world problems.
2. Loss of accuracy in large models especially those involving numeric variables.
3. Difficulty for non-experts (who are the target audience) of using large-scale models for their intended use in decision support.

The research of Fenton and Neil and their team since 1998 has directly addressed these problems [2-5]. The resulting methods and algorithms are implemented in AgenaRisk. The next three sections explain how we have addressed each of the challenges in turn.

How AgenaRisk makes it easier to build large-scale BNs

The overall challenge here set by Agena was to make it possible for users with minimal statistical knowledge to build and edit large-scale realistic models for a range of application domains. Underpinning our approach is the notion of an *extended BN*, which can be constructed in several different ways from *BN objects.* At the lowest level of decomposition a *BN object* is just a normal BN. Extended BNs can be built from BN objects using object-oriented and structured methods, as well as recursion, and dynamic modelling. The core propagation algorithm in AgenaRisk is a novel variation of the algorithm in [6] that works efficiently for extended BNs. Users of AgenaRisk do not need to know anything about the algorithm of course, but they need to be able to easily build extended BNs using the different possible methods. AgenaRisk makes this easy by various intuitive graphical techniques (underpinned by a novel process modelling notation) that, at the press of a button, lead to large models being constructed from smaller components.

AgenaRisk also provides a range of predefined BN templates in different application domains, including templates for

.

- Project Risk (such as complex Cost Benefit Analysis, Risk Registers, Stochastic PERT, Fleet Management and Life Costing, Logistics Modelling
- Systems Risk (such as Quality of Service, System Reliability, Maintainability and Availability, Fault Tree Analysis, Safety Analysis
- Operational risk (especially in the financial sector)
- Software project risk (including trade-off and defect prediction models)

Previous-generation BN tools enabled users to build BNs (our BN objects) from scratch using a graphical interface. AgenaRisk provides this functionality, and a radically improved approach for the hardest part, namely the task of defining the NPTs. Look at the node "Testing Quality" in Figure 1. Suppose this node has 5 states ranging from very low to very high. Suppose each of its two parent nodes has similar state values. Then the NPT for this node is a table with 125 cells corresponding to a probability value for each possible combination of parent and child values. In traditional tools, users are expected to fill in each cell by hand based on expertly elicited values– a hard enough task for a simple node like this and impossible for more complex nodes. As a result of extensive empirical research in various application domains Agena has identified a small number of commonly occurring functions that adequately capture expert judgement for many patterns of nodes. For example, an especially simple common function is based on the Truncated Normal distribution whose mean is a weighted sum of the parents and who variance is the extent of uncertainty around the mean. For a node like Testing Quality the necessary parameters of this distribution can be elicited in AgenaRisk in seconds and the full NPT generated instantly. The user simply uses slider bars to express the relative parent weights and their uncertainty, and can see the impact of different selections immediately for validation. AgenaRisk implements a range of similar functions and makes it easy for users to access them without having to understand the underlying statistics.

Central to AgenaRisk's revolutionary approach to generating complex NPTs with minimal user effort is the rich node typing system it has introduced. In addition to the usual set of node types AgenaRisk has implemented the notion of ranked nodes (which are crucial in most applications) and different types of numeric nodes. Associated with each node type is an extensive set of probability distributions (many of which are unique to AgenaRisk). In most situations the NPT for a node of a given type can be defined simply by selecting an appropriate distribution whose parameters are the node's parents. AgenaRisk provides a range of intuitive graphical methods for selecting and defining the functions. The net results of these techniques is that it is possible, using AgenaRisk, to build large, sophisticated models with fully populated NPTs in a matter of hours. Previously, building such models (if possible at all) would take months.

How AgenaRisk provides much greater accuracy

All previous generation BN tools required users to define the states of any numeric node as a sequence of pre-defined intervals. So, for example, instead of just specifying that the "number of faults" node ranges from 0 to 1000, you would have to break up 0 to 1000 into a manageable number of intervals. The more intervals you define, the more accuracy you can achieve, but at a heavy cost of computational complexity (as a rule of thumb it is not advisable to allow any node to have more than 200 states). This process (called discretisation) is not only time-consuming, but is made worse by the fact that you do not necessarily know in advance which ranges require the finer intervals. It follows that where a BN contains numerical nodes having a potentially large range, results are necessarily only crude approximations. AgenaRisk solves this critical problem by providing so-called dynamic discretisation, enabling maximal accuracy with no need for user intervention or set-up.

Dynamic discretisation has been something of a 'holy grail' for BN researchers. Agena's revolutionary algorithm is the only successful efficient implementation and hence represents a dramatic step-change for BN accuracy.

The rich and novel node typing system in AgenaRisk also provides a number of other areas where accuracy (and performance) are improved. For example:

- Using ranked nodes it is possible to support efficient BNs involving nodes with many parents (where all the nodes have many states). In previous generation tools such fragments are not possible because of computational intractability – designers were forced to introduce synthetic, intermediate nodes to ensure that no node had more than 2 or 3 parents.
- There are a number of algorithms handling 'intelligent' input of numerical values
- Improved methods for handling 'soft evidence'. When people talk about 'entering observations' into a BN model it generally means that a particular state of a variable is set to 'true' (i.e. probability equal to one) while all other states for that variable are set to false (i.e. probability equal to zero). In many situations, however, it seems more appropriate to enter so-called 'soft evidence' in which more than one state has a non-zero probability value (for example, if 3 out of 5 domain experts assert that the "Quality of documentation" is "very good" and 2 out of 5 assert that it is "good"). One of the major weaknesses of previous BN tools was their inability (or inadequacy) in dealing with soft evidence. We have implemented algorithms and an appropriate interface to enable soft evidence to be entered easily for any variable. We have also applied the same methods to handle evidence on numeric variables.

How AgenaRisk provides usable solutions for a wide range of end-users

In addition to being a powerful BN development environment, AgenaRisk is an application generator. Here we distinguish between two types of AgenaRisk users. On the one hand are the development/modelling users who build or modify BN models. On the other hand are the end-users who might range from data entry clerks through to senior managers who require full decision-support, sensitivity analysis and reporting functionality. Although the underlying model accessed in a particular application may be identical, different types of end-users require completely different interfaces and functionality. Previous generation BN tools provided, at best, an API that enabled programmers to build end-user systems wrapped around the model. Developing such systems required not just expert programmers but also BN experts. AgenaRisk enables non-programmers to tailor and generate attractive questionnaire-based decision support systems wrapped around a BN model in seconds. The extended BN architecture is again key to this process because it incorporates the notion of *questions* associated with each node. Development users have full control over the contents and display of questions (and their answers). They also have full control over what information is displayed (including how much, if any of the underlying graph model they want to display to the end user).

Crucial outputs of any BN are the probability distributions for each node. Previous generation BN tools assumed each node had a small number of discrete states and therefore simply plotted its histogram. This approach is inadequate for numeric nodes (especially using dynamic discretisation). AgenaRisk therefore incorporates a rich set of graphing functions, with users able to configure the outputs in many different ways. The graphing functions include intelligent analysis and layout of numeric variables that is well beyond the functionality of a tool such as MS Excel.

To further support usability, AgenaRisk supports the notion of multiple scenarios. In a BN model a scenario is simply a set of observations on the model. To make using BNs more intuitive users often need to directly compare predictions for different observations on the same model. For example, we may want to compare the predicted the number of residual defects under different testing 'scenarios' or we may want to compare changes in a variable over time. AgenaRisk provides powerful support for multiple scenarios, both in the questionnaire interface and the graphing.

At-a-glance AgenaRisk Feature set and architecture

- Uses a variety of Bayesian probabilistic modeling technologies
- Model graph editor / viewer:
  - Create cause-effect relations between risks
  - Full control of font, colour and shape properties
  - Ability to cut and paste model objects
- Probability table editor:
  - Specify statistical distributions using Monte Carlo simulator
  - Declare arithmetic & logical relations;
  - Input and output data with Excel spreadsheets and text files
- Questionnaire editor / viewer:
  - Add questionnaire headings to organise data collection
  - Add questions and enter data
  - View AgenaRisk spreadsheet to access many scenarios at once
- Scenario manager:
  - Save data as actual or hypothetical scenarios
  - Database and file system support
  - User authentication, audit tracking and status reporting
- Model navigator:
  - View enterprise or project as hierarchy;
  - Use navigator to control your applications using standard risk models.
- Statistics and graphs:
  - Marginal and cumulative probability graphs
  - Summary statistics
  - Histograms; bar, area and line charts.
- Reporting
  - Note taking via text editor
  - HTML reporting and graphs output as JPEGs
  - Input and output data via Excel spreadsheets and text files

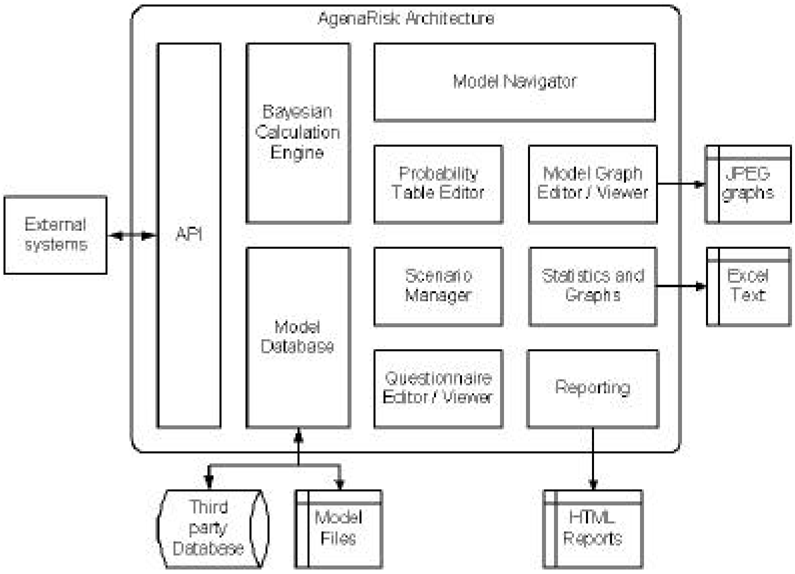

Figure 2 AgenaRisk architecture

References

1. Fenton NE, and Neil M, "Combining Evidence in Risk Analysis using Bayesian Networks", Agena White Paper Agena White Paper W0704/01, www.agena.co.uk, 2004
2. Fenton NE, Krause P, Neil M, "Software Measurement: Uncertainty and Causal Modelling", IEEE Software 10(4), 116-122, 2002
3. Fenton NE, Marsh W, Neil M, Cates P, Forey S, Tailor T, 'Making Resource Decisions for Software Projects', 26th International Conference on Software Engineering (ICSE 2004), May 2004, Edinburgh, United Kingdom. IEEE Computer Society 2004, ISBN 0-7695-2163-0, pp. 397-406
4. Neil M, Fenton N, Forey S and Harris R, "Using Bayesian Belief Networks to Predict the Reliability of Military Vehicles", IEE Computing and Control Engineering J 12(1), 11-20, 2001
5. Neil M, Fenton NE, Nielsen L, "Building large-scale Bayesian Networks", The Knowledge Engineering Review, 15(3), 257-284, 2000.
6. Lauritzen SL and Spiegelhalter DJ, "Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems (with discussion)". J. R. Statis. Soc. Series B, 50, No 2, pp.157-224, 1988.

Agena Ltd
32-33 Hatton Garden
London EC1N 8DL
Tel: +44 (0)20 7404 9722
Fax: +44 (0)20 7404 9723

norman@agena.co.uk