

Using Ranked nodes to model qualitative judgements in Bayesian Networks

Norman Fenton, Martin Neil and Jose Gallan Caballero
Queen Mary, University of London

28 Feb, 2007

Abstract

Although Bayesian Nets (BNs) are increasingly being used to solve real world risk problems, their use is still constrained by the difficulty of constructing the node probability tables (NPTs). A key challenge is to construct relevant NPTs using the minimal amount of expert elicitation, recognising that it is rarely cost-effective to elicit *complete* sets of probability values. We describe a simple approach to defining NPTs for a large class of commonly occurring nodes (called ranked nodes). The approach is based on the doubly truncated Normal distribution with a central tendency that is invariably a type of weighted function of the parent nodes. In extensive real-world case studies we have found that this approach is sufficient for generating the NPTs of a very large class of nodes. We describe one such case study for validation purposes. The approach has been fully automated in a commercial tool, called AgenaRisk, and is thus accessible to all types of domain experts. We believe this work represents a useful contribution to BN research and technology since its application makes the difference between being able to build realistic BN models and not.

Keywords: Bayesian networks, node probability tables, ranked nodes, probability elicitation, risk analysis

1. Introduction

In recent years Bayesian Networks (BNs) have become increasingly recognised as a potentially powerful solution to complex risk assessment problems [9]. 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 and directed arcs representing causal or influential relationships between variables [26]. 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. 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.

Our own work in this area has produced solutions to a number of real world, high-stakes problems such as:

- Safety of embedded systems in railway applications [21];
- Military vehicle reliability, [22];
- Risk of mid-air collisions in Air Traffic [23];

- Software defect prediction in consumer electronics products [6],[7],[8],[24];

All of these applications involved building extremely large-scale BN models. As a result of the difficulties we encountered in earlier real-world BN model building (such as in the model for software safety in the nuclear industry [17] we were well aware of the limitations of relying on purely ‘hand-crafted’ approaches in which each variable and each NPT had to be elicited exhaustively with domain experts. By extending the ideas of object-oriented Bayesian Networks [13] we developed a range of methods that could be deployed in practice. For example, in [21] we described a range of techniques that were primarily targeted at the problem of building large-scale BN topologies. The techniques described there have been validated in numerous projects and have been formally incorporated into BN tools such as Hugin [11] and AgenaRisk [1]. Similar methods are described in [18]. However, our previous work, and that of others, has said little about the even harder problem of building NPTs in large-scale BNs, especially for nodes with many states. This paper focuses on one, and only one, especially important part of this problem: how to build large NPTs for a commonly occurring class of nodes called ‘ranked’ nodes (which represent qualitative variables that are abstractions of the underlying continuous quantities).

We begin in Section 2 by outlining the problem and relevant related work. In Section 3 we formalise the notion of ranked nodes along with the conditions under which they occur most commonly in BNs. In Sections 4 and 5 we describe the class of causal weighting functions required to generate the NPTs for these ranked nodes. Our method is based on the representation of NPTs by means of parametric probability functions where the child node’s probability is

defined as a weighted function of the parent node values. The weighted rank node functions specified herein (which turn out to be sufficient for most applications) are:

- Mean Average
- Minimum
- Maximum
- MixMinMax

In Section 6 we describe the other instance where ranked nodes commonly occur, namely as indicator nodes. In Section 7 we describe how ranked nodes are declared in the AgenaRisk software and how the corresponding NPTs are generated. Section 8 describes how experts can use AgenaRisk to build the expressions needed to generate NPTs for ranked nodes quickly and easily. This approach has been validated in a number of recent case studies such as [7],[8],[25]; an illustrative case study example is provided in Section 9 that shows how the approach was used effectively by domain experts (with little statistical expertise) to generate large-scale NPTs and overcome problems with previous manual approaches.

2. The problem and background

Consider the BN fragment shown in Figure 1. Such fragments are very typical of those that frequently occur in the real-world models already cited – this particular one occurred in at least two of the projects referred to in Section 1. They are characterised by the fact that node values are typically measurable only on a subjective scale like {very low, low, medium, high, very high} and only extremely limited statistical data (if any) is available to inform the probabilistic relationship for Y given X_1 and X_2 . Yet, there is significant expert subjective judgement that can be used. .

Assuming each of the nodes has five states (in the many commercial studies we have been involved with experts are rarely satisfied with 3-point scales), the NPT for the node Y has 125 states. This is not an impossible number to elicit exhaustively, but from extensive experience we know that all kinds of inconsistencies arise when experts attempt to do so (some specific issues are described in the case study in Section 8). If the number of states increases to seven (which experts commonly insist on) and/or the node Y has additional parents then exhaustive elicitation becomes infeasible, especially as real-world models invariably involve dozens of fragments like these.

Hence, the problem and challenge is to produce an appropriate NPT for the node Y that makes the most of limited expert elicitation. This problem is certainly not new since it has been addressed in [4],[28],[31] and there have been serious case studies on specific elicitation techniques [15],[16],[20],[29]. Also the Noisy-OR [10] and Noisy-MAX [3] methods are well established as a standard way of encoding expertise in large NPTs . Noisy-OR has the disadvantage that it applies only to Boolean nodes and implicitly ignores the interaction effects between variables. Noisy-MAX, despite the fact that it applies to ranked nodes with many states, does not model the range of relationships we seek here.

There is a large body of literature covering the psychological biases encountered during elicitation and use of probability values. Such biases often arise through inappropriate or misleading question choice and depend on how the problem and question are framed (for more see [12]). In the BN literature there are a few relevant papers that describe experimental results gained from applying different probability elicitation. One is [32] which found that human experts produced better results when Noisy-OR parameters were elicited rather than complete NPTs Also [27] gives a very good overview of a number of different methods that can be used

for elicitation, including probability wheels and the verbal-numerical response scale. The work on verbal-numerical response scales is described in detail in [30] where it was reported that its use markedly improved the efficiency of elicitation and accuracy of results. Size restrictions prevent us from directly addressing the role of elicitation in the whole model building process and the inherent challenges that might be encountered, so we address only one type of probabilistic relationship that one might want to build into a BN. Our approach is complementary to the elicitation methods and for the purpose of quick comparison the differences are: ranked nodes are useful when representing ranked relationships in NPTs involving nodes that are near continuous, Noisy-OR is useful in cases involving Boolean nodes, and verbal-numerical response scale is useful for relationships when nodes are labelled.

3. The nature of Ranked nodes

Ranked nodes represent discrete variables whose states are expressed on an ordinal scale that can be mapped onto a bounded numerical scale that is continuous and monotonically ordered. We can assume that all ranked nodes are defined on an underlying unit interval, $[0-1]$, scale. For a given number of intervals defined, and labelled, on this scale we simply discretise accordingly. For example, for a 5-point scale such as {very low, low, average, high, very high} our interval widths for each state are 0.2. Thus "very low" is associated with the interval $[0 - 0.2)$, "low" is associated with the interval $[0.2 - 0.4)$ etc.

As far as the user is concerned the underlying numeric scale is invisible — the displayed scale is still the labelled one rather than the numeric one, but the latter is used for the purposes of computation and generating the NPT. The crucial thing about ranked nodes is that they can make the BN construction and editing task much simpler than is otherwise possible. In particular,

provided they appear in the appropriate combinations described below, the normally complex task of constructing sensible associated NPTs is drastically simplified.

In the real-world applications described in Section 1, experts typically wanted to complete an NPT using a simple averaging scheme to compute the maximum or minimum value as a guide to defining the “central tendency” of the child node based on a set of causal parent node values. Hence, in [6] (in attempting to construct the NPT for a node like Y) we adopted an approach based on sampling values, that resulted in expert elicitation assertions like the following:

- When X_1 and X_2 are both ‘very high’ the distribution of Y is heavily skewed toward ‘very high’.
- When X_1 and X_2 are both ‘very low’ the distribution of Y is heavily skewed toward ‘very low’.
- When X_1 is ‘very low’ and X_2 is ‘very high’ the distribution of Y is centred below ‘medium’.
- When X_1 is ‘very high’ and X_2 is ‘very low’ the distribution of Y is centred above ‘medium’.

Since we are assuming that each node has an underlying numerical scale in the interval $[0, 1]$ such assertions suggest intuitively that Y is some kind of weighted average function. In fact, experts found it easier to understand and express relationships in such terms. Many so-called "self-assessment" or "scorecard" systems are based around little more than weighted averages of attribute hierarchies. However, such systems are usually implemented in spreadsheet-based programmes that have associated with them a number of problems: difficulty in handling missing

data; problems with assessing credibility of information sources; difficulty in using different scales.

Since all of these problems are readily solved using BNs, the challenge is to provide the appropriate BN implementation that captures the explicit simplicity of the weighted average while also preserving the intuitive properties that the resulting distributions have to satisfy. For example, simply making Y the (exact) weighted average of its parents does not work – since the only uncertainty in the distribution of Y given its parents will be the result of discretisation inaccuracy rather than deliberate modelling. What is especially tricky to model properly are the intuitive beliefs about the causes given certain child observations — i.e. so-called back propagated beliefs. For example, suppose we have observed Y and X_1 and wish to infer the value of X_2 like: If Y is ‘very high’ and X_1 is ‘very low’ then we would be almost certain that X_2 is ‘very high’. If Y is ‘very high’ and X_2 is ‘average’ then we would be confident that X_2 is ‘very high’ but not as confident as in the above case.

Using an interpolated Beta distribution to approximate Y (as in [6]), does *not* preserve these back-propagation beliefs. However, a straightforward solution for defining the NPT for $p(Y | \underline{X})$ (where \underline{X} represents the set of parent variables X_1, X_2, \dots, X_n) in such a way that these various properties *are* all satisfied is provided by the Truncated Normal distribution, which we describe next.

4. Modelling ranked causes using a doubly truncated Normal distribution

Formally, the ranked nodes’ causal structure is characterised by a joint probability distribution with a set of causes, \underline{X} , containing $i = 1, 2, \dots, n$ ranked nodes, X_i , as parents of Y :

$$p(\underline{X}, Y) = p(Y | \underline{X}) \prod_{i=1}^n p(X_i)$$

In general, the node Y is considered to be a *consequence* of two or more *cause* nodes where each of the cause nodes is assumed to be independent when calculating the NPT. The BN in Figure 1 is a very simple example.

We draw an analogy with linear regression where $y_i = \underline{\beta}_x + \varepsilon_i$ with $\varepsilon \sim N(0, \sigma_y^2)$ and where the contribution to the variance of Y is σ_y^2 . The regression analogy is apt since we are attempting to “target” the area of central tendency in Y given different values of X_i and then are adding a fixed amount of uncertainty around this. The only issue we need to resolve is the contribution of each cause to the effect and a clear way to do this is to use the correlation between the cause and the effect as the appropriate measure.

Rather than the Normal distribution commonly assumed in linear regression for ranked causal nodes, we use the doubly truncated Normal distribution (denoted *TNormal* hereafter) as defined, for example, in [2], where all nodes are truncated in the $[0, 1]$ region. Unlike the regular Normal distribution (which must be in the range $-\infty$ to $+\infty$) the TNormal has *finite* end points. We denote the TNormal by $\text{TNormal}(\mu, \sigma^2, 0, 1)$ where μ is the mean and σ^2 is the variance. In the TNormal we start with a regular Normal distribution but ‘ignore’ the probability mass to the left and right of the finite endpoints and then normalise the resulting distribution over the finite range $[0, 1]$. This enables us to model a variety of shapes including a uniform distribution, achieved when the variance $\sigma^2 \rightarrow \infty$, and highly skewed distributions, achieved when $\sigma^2 \rightarrow 0$.

We use a simple weighted sum model to measure the contribution of each X_i to explaining Y as a “credibility weight”, w_i , (it can also be elicited from an expert in this way) expressed as real values, $w_i \geq 0$. The higher the credibility index the greater the correlation between X_i and Y . Thus, in our method the equivalent to the error variance, σ_Y^2 , in the linear regression model is simply the inverse of the sum of the weights:

$$\sigma_Y^2 = \frac{1}{\sum_{i=1}^n w_i}$$

Given that Y lies within $[0, 1]$ we must normalise the regression equation, $E(Y) = \sum_{i=1}^n w_i X_i$, by

dividing with $\sum_{i=1}^n w_i$ thus:

$$p(Y | \underline{X}) = TNormal \left[\frac{\sum_{i=1}^n w_i X_i}{\sum_{i=1}^n w_i}, \frac{1}{\sum_{i=1}^n w_i}, 0, 1 \right]$$

Suppose, for example, that $n = 3$ and that the allocation of weights, w_i , for each X_i 's contribution to explaining Y is in the ratio 2:3:5 with a variance, $\sigma_Y^2 = 0.001$. Then the joint distribution generated will be:

$$\begin{aligned} p(Y | \underline{X}) &= TNormal \left[\frac{200X_1 + 300X_2 + 500X_3}{200 + 300 + 500}, \frac{1}{200 + 300 + 500}, 0, 1 \right] \\ &\equiv p(Y | \underline{X}) = TNormal \left[\frac{2X_1 + 3X_2 + 5X_3}{10}, 0.001, 0, 1 \right] \end{aligned}$$

The resulting distribution, and BN model is shown in Figure 2.

The resulting distribution for $p(Y)$ will not produce summary statistics exactly matching the function because we are using coarse discretisations in arriving at the result. Given this, the mean values will tend to differ within the bin range specified; specifically for five ranks defined on [0-1] the mean value may be out by up to 0.1. Also, the variance values observed will be considerably higher because of the coarse discretisation. However, neither of these are major problems since the aim is to produce a good fit to the expert's distribution rather than a good approximation to a TNormal distribution.

Our approach is only designed to cover unimodal probability distributions. If $p(Y | \underline{X})$ is bi or multi modal the simplest solution involves mapping each of the unranked states in \underline{X} that provide a causally equivalent, and exchangeable, *ranked* response on Y to a single new state in a new ranked node. For example, if we have a model of the effect of workload on human concentration such that

$$p(Y = \text{concentration} | X = \{\text{overload, normal, underload}\})$$

and

$$p(Y | X = \text{overload}) = p(Y | X = \text{underload})$$

then we can simply insert a new node Z whose states are ranked in terms of their effect on Y :

$$p(Y = \text{concentration} | Z = \{\text{normal, abnormal}\})$$

and

$$p(Z = \text{normal} | X = \text{normal}) = 1$$

and

$$p(Z = \text{abnormal} | X = \text{overload OR } X = \text{underload}) = 1.$$

5. Modelling ranked causes using weighted min and max

The weighted average is not the only natural function that could be used as the measure of central tendency in the ranked cause model. Suppose, for example, that in Figure 1, we replace the node “Quality of testing process” with the node “Testing effort” as shown in Figure 3. In this case (which was exactly the scenario in the commercial project described in [8]) we elicited the following information:

- When X_1 and X_2 are both ‘very high’ the distribution of Y is heavily skewed toward ‘very high’.
- When X_1 and X_2 are both ‘very low’ the distribution of Y is heavily skewed toward ‘very low’.
- When X_1 is ‘very low’ and X_2 is ‘very high’ the distribution of Y is centred toward ‘very low’.
- When X_1 is ‘very high’ and X_2 is ‘very low’ the distribution of Y is centred toward ‘low’.

Intuitively, the expert was saying here that for testing to be effective you need not just to have good people, but also to put in the effort. If either the people or the effort is insufficient then the result will be poor. However, really good people can compensate, to a small extent, for lack of effort.

A weighted sum for Y will *not* produce an NPT to satisfy these elicited requirements. Formally, Y 's mean is something like the *minimum* of the parent values, but with a small weighting in

favour of X_1 . The necessary function, which we call the weighted min function, *WMIN*, has the following general form:

$$WMIN = \min_{i=1..n} \left[\frac{w_i X_i + \sum_{i \neq j}^n X_j}{w_i + (n-1)} \right] \text{ where } w_i \geq 0 \text{ and } n \text{ is the number of parent nodes}$$

with a suitable variance σ_y^2 that quantifies our uncertainty about the result, thus giving:

$$p(Y | \underline{X}) = TNormal[WMIN(\underline{X}), \sigma_y^2, 0, 1]$$

The *WMIN* function can be viewed as a generalised version of the normal MIN function. In fact, if all of the weights w_i are large then *WMIN* is close to MIN. At the other extreme, if all the weights $w_i = 1$ then *WMIN* is simply the average of the X_i s. Mixing the magnitude of the weights gives a result between a *MIN* and an *AVERAGE*. In the above example, taking $w_1 = 3$ and $w_2 = 1$ (with a variance $\sigma_y^2 = 0.01$) yields the results shown in Figures 4.

We can also use an analogous *WMAX* function:

$$WMAX = \max_{\forall i=1..n} \left[\frac{w_i X_i + \sum_{i \neq j}^n X_j}{w_i + (n-1)} \right] \text{ where } w_i \geq 0$$

And finally a function *MIXMINMAX* which is a mixture of the classic MIN and MAX functions.

$$MIXMINMAX = \frac{w_{\min} MIN(\underline{X}) + w_{\max} MAX(\underline{X})}{w_{\min} + w_{\max}} \text{ where } w_{\min}, w_{\max} > 0$$

In each case the experts need only supply the parameters to generate the NPT. We found that this set of functions has been sufficient to generate almost all of the ranked node NPTs elicited in

practice. The efficiency savings are considerable: If there are m ranked cause nodes each with n states the expert need only supply $m + 1$ parameter values, compared requires $(m + 1)^n$ values for full elicitation.

It should be noted that ranked nodes can be further partitioned by declaring additional labelled, Boolean or numeric parents that can be used to condition the type of weighted expression one might wish on the child node.

6. Ranked indicators

In addition to their occurrence as described in Section 4, ranked nodes occur frequently as *indicators* of other ranked nodes, such as shown in Figure 6. Here we can see a simple single ranked indicator modelling the relationship between “staff quality” and “staff motivation” and another supplementing the first by adding an additional two indicators: “staff training” and “staff experience”. In this section we describe the notion of indicator nodes formally and explain how to define the necessary NPTs.

Indicator nodes operate in a similar way to “filter” nodes in a Kalman filter. Here we can think of the indicators as providing noisy or imperfect observations and the parent node as the true (but possibly unobservable or not economically measurable) value awaiting estimation [Maybeck 1979]. In a Kalman filter we wish to condition our estimate for the “true” value on the data on hand from each of our “indicator” nodes assuming each indicator is Gaussian distributed.

Formally, the joint distribution for a set, \underline{X} , containing $i = 1, 2, \dots, n$ ranked indicators, X_i , of a single causal parent node, Y is:

$$p(\underline{X}, Y) = p(Y) \prod_{i=1}^n p(X_i | Y)$$

We model the NPT for each indicator node using the doubly truncated TNormal distribution:

$$p(X_i | Y) = TNormal(Y, \sigma_i^2, 0, 1)$$

This assumes that the nodes Y and X_i are on the same scale. The expert simply has to specify the variance parameter, σ_i^2 , whose inverse acts as a “credibility index” — the higher the credibility index the greater the correlation between the indicator and the parent cause node.

Indicator nodes are correlated with each other by virtue of the structure of the Bayesian net. This correlation is desirable given that indicators reflect the true state of the underlying, unknown, cause. Only when the cause itself is instantiated with hard evidence are the indicators uncorrelated. However, given that the causal nodes are usually unobservable (this is after all why we will use an indicator) the indicator nodes are generally not independent in practice.

Unlike in Kalman filters our indicator nodes are bounded on $[0, 1]$ so we cannot use Normal distributions and instead we must use doubly truncated Normal distributions, solved numerically (there is no analytical solution to $\prod_{i=1}^n p(X_i | Y)$ when the indicators are doubly truncated Normal).

Given this, we should not necessarily expect the results achieved using a ranked node formulation to give the same results as the Kalman filter. It is, however, helpful to know where the differences lie.

The general properties and behaviour are similar insofar as our approach very closely approximates a Kalman filter in the region where $\mu_y = 0.5$. However, when an observation is

made on an indicator node near its truncation boundary, $[0, 1]$, its actual variance is less because of the effects of truncation and this lower variance obviously translates into a stronger influence on Y . Note also that as the variance values allocated to indicator nodes get very large, the resulting NPT approximates a conditional uniform distribution. Therefore $\mu_Y \rightarrow 0.5$ and as a result the correlation between the indicators, X , and Y approaches zero. Practically speaking, for 5 and 7 point scaled rank nodes, setting $\sigma_i^2 > 0.1$ indicates a very poor correlation. This also means that the actual mean value induced on Y for an indicator with high variance will not be $\mu_Y \rightarrow x_i$ but rather $\mu_Y \rightarrow 0.5$.

In practice small variance values tend to be selected and this means our rank node solution approximates the analytical Kalman filter nicely. For example, if we had two rank nodes with $\sigma_1^2 = \sigma_2^2 = 0.01$ and $X_1 = 0.05$ and $X_2 = 0.15$ the difference between the analytical, Kalman filter result, Y , and the rank node approximation, \hat{Y} , is: $Y \sim N(0.10, 0.005)$, $\hat{Y} \sim TNormal(0.0909, 0.0037, 0, 1)$. We believe that this level of error is acceptable given the unavoidably crude nature of the rank scales we are using.

Another perspective on the use of indicator nodes is that each can be treated either as a different sub-attribute of the parent node or as a different measure of that sub-attribute from a different source. This second view has proven helpful where there were multiple experts, each with a different credibility, producing different observations. Also, using indicator nodes is simply a form of object classification and traditionally classification is done using naive Bayesian methods where a hidden “unknown” node, Y , is classified from a set \underline{X} containing n ranked indicators or classifiers.

In [21] we described a common idiom called the *measurement idiom* where the credibility of an indicator is itself contingent on some other factor. This is easily modelled in practice by setting up an additional parent node for one or more indicators with parameterised values for σ_i^2 .

Suppose we have three indicators of Y , such as that presented in Figure 7 where $\underline{X} = \{X_1, X_2, X_3\}$. In this example we assume that X_1 is a reasonably accurate indicator of Y , while X_2 is much less so and X_3 even worse. We could capture this information by specifying the variance values as follows $p(X_1 | Y) = TNormal(Y, 0.001, 0, 1)$, $p(X_2 | Y) = TNormal(Y, 0.008, 0, 1)$, and $p(X_3 | Y) = TNormal(Y, 0.02, 0, 1)$. Figure 7 shows the marginal probability distribution on the indicator nodes $\underline{X} = \{X_1, X_2, X_3\}$ given an observation on the parent, $Y = \textit{medium}$. Clearly X_1 is more highly correlated with Y than either X_2 or X_3 .

Figure 8 shows how we can use the indicator nodes to infer the true state of the parent node, Y , from the observations $X_2 = \textit{medium}, X_3 = \textit{low}$. Note also that the unobserved indicators, such as X_1 , are correlated with observed indicators because of the shared parent node, Y . Compare this to Figure 9 where we invert the observation values such that $X_2 = \textit{low}, X_3 = \textit{medium}$ and notice how the distribution on Y is influenced more highly by indicator X_2 in both figures.

If there are m ranked indicator nodes each with n states full elicitation requires n^2m values to be provided by the expert. Each of the rank node functions only requires m parameter values by comparison.

Note that the credibility indices, σ_i^2 , for each indicator can be estimated by simple trial and error or, if the data is available, the parameters estimated using standard Bayesian parameter learning techniques.

7. Creating ranked nodes using the AgenaRisk software

For the purpose of building realistic NPTs that adequately capture expert judgement, the existence of a good theoretical approach is insufficient. Good tool support is also needed, and successful use of ranked nodes must be supported by a reliable tool that:

- Enables domain experts without any statistical knowledge to quickly and easily generate distributions
- Provides instant visual feedback to check that the NPT is working as expected.

The AgenaRisk software [1] satisfies these requirements and implements the approach described in Sections 4-6. Constructing the necessary NPT requires experts only to go through the following simple steps in AgenaRisk (supported by the Dialog shown in Figure 10):

1. Select the Node Probability Table property for a given node and declare that the NPT is defined by an Expression. The TNormal distribution is automatically selected.
2. Either type in the full weighted expression or access the Dialog by a simple right mouse click as shown in Figure 10.
3. Complete the appropriate weights via the dialog presented by selecting the parent nodes, using a slider bar to define the weight values and the “certainty” or variance value. The user can also overwrite these by simply entering in values for all necessary parameters.

It is also worth noting that users can change the scale (from say a 5-point scale to a 7-point scale as was required for a number of the nodes in our commercial case studies) with the click of a single button and without having to redefine the weighted function. They can also, if they wish, edit individual NPT entries by hand in rare cases where certain combinations of parent values result in a probability value not properly captured by the generic function.

8. Case study validation

The approach described in the paper to constructing large NPTs for ranked nodes was used extensively in the real world cases described in [7],[8],[23],[25] as well as in many other commercially confidential applications. Here we will highlight the difficulties, the process and the results (including comparison with manually constructed NPTs) on one specific case study. The goal of this case study, which was undertaken with a multinational telecoms company, was to produce a BN model for the purpose of reliability evaluation of electronic components. The company's domain experts constructed a model comprising 50 nodes in total that allowed the model's users to perform a qualitative adjustment of reliability given information about the component manufacturer's development and test processes. The experts involved in the study were professional engineers who had some statistical training in Six Sigma concepts but were not practising statisticians.

This case study reports "research in action" and, as such, does not give a full and rigorous validation of rank nodes versus other elicitation techniques. Nor does it, because of confidentiality issues, reproduce the validation results achieved with other commercial research partners in practice. We would therefore like to encourage other researchers to experiment with, test the approach and publish the resulting data.

Here we will focus on a single fragment of the overall model that is the same as Figure 1. In this example we have two cause nodes:

$$\underline{X} = \{X_1 : \text{Quality of Testing Staff}, X_2 : \text{Quality of Testing Process}\}$$

and we are interested in using these to estimate $p(Y | X_1, X_2)$.

The first attempt to estimate the NPT for $p(Y | X_1, X_2)$ relied wholly on manual methods whereby each of the 125 cells in the NPT was discussed and a value entered. The resulting NPT is shown in Figure 11 (the number of decimal places is the result of the tool's automatic normalisation process that ensures all column probabilities sum to one, irrespective of how the user's chose to enter the values). During this process it became apparent that one of the parent nodes was much more important than the other in terms of its effects on the child and this was "kept in mind" when the NPT was produced. Some impediments to producing this NPT (consistent with evidence from the psychology literature, such as [12]) are worth mentioning:

- The experts were continually backtracking between previously estimated values and current values because cases were felt to be similar and so the NPT values, and in particular whole columns in the NPT, could be reused.
- It was very difficult to apply the weighting heuristic the experts wished to apply given the very large number of values being considered.
- In an attempt to maintain consistency previously elicited parts of the NPT were revisited and amended. Frequently this led to degradation rather than improvement and consequential rework, which itself was error prone.

Once the NPT was completed the experts could examine the sensitivity of results by running the model. Their expectation was that the resulting marginal distribution for Y would be monotonic and smooth given supplied test values for the parent nodes, X_1, X_2 and that progressive increasing values of the parents would have a commensurate increase in the child value. Our test clearly shows that the rank order of results is not obtained:

$$\text{Mode}(Y | X_1 = \text{Medium}, X_2 = \text{Very low}) = \text{Medium}$$

$$\text{Mode}(Y | X_1 = \text{Medium}, X_2 = \text{Low}) = \text{High}$$

$$\text{Mode}(Y | X_1 = \text{Medium}, X_2 = \text{Medium}) = \text{Medium}$$

$$\text{Mode}(Y | X_1 = \text{Medium}, X_2 = \text{High}) = \text{Medium}$$

$$\text{Mode}(Y | X_1 = \text{Medium}, X_2 = \text{Very High}) = \text{Medium}$$

Likewise, when a value for $Y = \text{Very High}$ was instantiated in the model it was expected, and hoped, the response on node X_1, X_2 , by back propagation, would result in marginal distributions of monotonic character. Figure 12 shows that monotonicity is not achieved on the distribution of node X_2 .

Given these problems ranked nodes were used as an alternative to manual derivation of the NPT. As a precursor to the definition of parameter values and weights in the ranked nodes a simple “truth” table was used to determine what type of ranked node function would be best for each BN fragment. This simply involved taking combinations of values at the extreme of each parent node state range, such as “Very High”, “Very Low” etc, and asking the experts to estimate the mean response of the child node conditioned on these values. An example of part of such a table is shown in Table 1.

This helped reveal a possible heuristic that could then be used to generate and test the generated NPT in AgenaRisk. For example, by mapping the scale ‘very low’, ..., ‘very high’ to a 0-1 scale the table often quickly revealed that the mean response of the child node was a simple weighted sum of the parents (such as in Table 1, where the weighting was heavily in favour of ‘Quality of testing staff’). This weighted sum would then be used as the mean of the TNormal distribution, with experts happy to try different variance values until they were satisfied with the results. Once the experts became familiar with the approach they often identified the appropriate expression and selected parameter values without the use of the “truth” table. Note that our use of a truth table is not unique to rank nodes and is only one way of helping experts’ double-check their thinking. It is also worth noting that the way we used a truth table is similar to the way [van der Gaag 2002] aligned different verbal-numerical response scales for a single NPT along side each other for quick comparison by the expert.

The weights the experts associated with the cause nodes were respectively 3, 1, for a weighted mean and the variance is 0.01. Hence, the rank node expression is:

$$p(Y | \underline{X}) = TNormal \left[\frac{3X_1 + X_2}{4}, 0.01, 0, 1 \right]$$

Empirical validation of this rank node function is difficult in practice given the qualitative reasoning being captured and the fact that observable data is predicted by a set of BN fragments working together. In any case our aim in the case study was simply to determine whether the ranked node solution offered practical advantages and insights over the manual approach. In terms of reducing the numerical load on the experts the approach has clear benefits, as has been

pointed out. Table 2 demonstrates the effort savings achieved in building the NPTs in two BN models (in both these cases the NPTs were originally elicited manually).

However, whether or not the rank node formulation is better is reliant on the judgement of the expert, in that it should capture the judgement he wishes to articulate. This is not easy to formalise statistically since the original manual table is in no sense “correct” and in any case, for pragmatic reasons, many experts, not being statisticians, may wish to construct the model to represent their judgments without validating every fragment contained therein.

We do however use a simple scoring aid for experts to compare different NPTs that they might wish to generate using different rules and schemes. This simply performs a pairwise comparison for each hypothetical marginal distribution created by each NPT approach (“model”), whether it be manual or done by ranked nodes (or indeed by some other scheme, such as by multiple experts). There are more obviously sophisticated schemes for performing sensitivity analysis, such as [14], but given our intended audience is primarily non-statisticians we prefer a more qualitative aid, which has the aim of identifying difference and magnitude of difference only.

Our aid involves pairwise comparisons of each candidate model by calculating the distance, Z , between the predicted state given by one model, Y_1 , and the estimated state provided by another, Y_2 , where both are conditioned on the same causal nodes, $p(Y_1 | X_1, X_2)$, $p(Y_2 | X_1, X_2)$:

$$Z = Y_1 - Y_2 \text{ where the ranked state values for } Y_1 \text{ and } Y_2 \text{ are replaced by integer values } 1, 2, 3, \dots$$

This score can be easily embedded in the BN itself as a child node of the candidate models. Figure 13 shows the resulting distribution of $p(Z)$ for our case study example (the model used

here is the manually generated model, that best matches expert’s expectations, used to benchmark the goodness of the model constructed using the proposed method). Notice that the difference centres are zero but there is a slight positive bias, thus showing that the manual NPT, Y_2 , and the ranked NPT, Y_1 , are relatively close matches of each other — on average. Clearly if there were large differences between the models a non symmetric marginal distribution for $p(Z)$ would highlight it instantly (a lack of symmetry would indicate a systematic bias in one direction). This approach can be extended to become a formal hypothesis test involving the null hypothesis, $H_o : Y_1 = Y_2$.

To determine whether there are case-by-case differences between the models the expert can instantiate the causal factors, in this case assigning values to X_1, X_2 , and examining the effects on $p(Z)$. This is easily done in AgenaRisk.

Additional means of validation would involve carrying out sensitivity analysis on the parent nodes and their effects on the child nodes. At present we follow this approach in an ad-hoc manner, as do others, but recognise this as an area ripe for improvement.

9. Conclusions

One of the most important challenges in building effective BN models to solve real-world risk assessment problems is that of constructing the NPTs. Because of the need to involve busy domain experts (who do not necessarily understand probability theory in detail) we have to construct NPTs using the minimal amount of expert elicitation, recognising that it is rarely cost-effective or feasible to elicit *complete* sets of probability values. We have identified a large class of BN nodes (the *ranked* nodes) for which we have provided a semi-automated method of NPT

construction. There is obviously a trade-off between the benefits a general method, like ours, can provide and the costs of developing a bespoke modelling approach for each and every specific situation. In the many real applications we have developed we have found bespoke modelling to be too costly and demanding to be feasible. Our general approach offers a marked improvement over current practice and has proven to be acceptable to practitioners.

The approach presented here has evolved over a number of years from the process of engaging with domain experts in real commercial situations. We have found that this approach makes the difference between being able to build realistic BN models and not. The BN solutions to real-world problems described in [6],[22],[23] used early versions of the approach described in this paper. Moreover, the work in those projects was crucial in informing the automated version of the method that has recently been implemented completely in the AgenaRisk software [1]. An earlier prototype of the automated version was used extensively to build the models described in [7],[8] and has been validated by partners such as Philips, Israel Aircraft Industries, and QinetiQ in that project. Validation was on two levels. On the first level the domain experts we worked with, who were not statisticians, were able to build and tailor serious models that captured their beliefs well. On the second level, our research partners reported that the models produced predictions and decision support insights that were demonstrably better than the results from methods that they had previously used. Also, since then the approach has been used in a number of application areas such as for operational risk assessment [25] and for augmenting reliability prediction methods for electronic components (which was used as a validation case study in this paper). The results show that the elicitation burden is much reduced by using rank nodes by simply eliciting a small number of parameters from experts. This does not, however, mean to say that using rank nodes guarantees better results in all cases and this is why we supplement the

approach with extensions to cope with multimodality and conditioned switching behaviour. Likewise, we use a simple scoring approach to compare and highlight the differences between NPTs generated by different approaches.

We believe that future work in this area should concentrate on three challenges: Eliciting NPTs for complex temporal models involving evolving processes such as rapidly changing design processes; “Expert mediated” semi-automatic learning of parameters from data; and comparing data mining methods against BNs derived from expert opinions.

Acknowledgments

This report is based in part on work undertaken on the following funded research projects: SCULLY (EPSRC Project GR/N00258), SIMP (EPSRC Systems Integration Initiative Programme Project GR/N39234), SCORE (EPSRC Project Critical Systems Programme GR/R24197/01), MODIST (EC Framework 5 Project IST-2000-28749) and eXdecide (EPSRC project EP/C005406/1). We also acknowledge the contributions of Patrick Cates, Simon Forey, Peter Hearty, David Marquez, Manesh Tailor, and William Marsh. We are also grateful to the referees for their comments and helpful suggestions.

References

- [1] Agena Ltd, “AgenaRisk Software”, www.agenarisk.com, 2007.
- [2] F. Cozman and E. Krotkov, “Truncated Gaussians as Tolerance Sets”, *Technical Report CMU-RI-TRI*, Robotics Institute, Carnegie Mellon University, 1997.
- [3] F.J. Díez, “Parameter adjustment in Bayes networks: the generalized noisy or-gate”. *Proc. Ninth Conference on Uncertainty in Artificial Intelligence*, D. Heckerman and A. Mamdani, eds, pp. 99-105, Washington D.C, 1993.
- [4] M.K. Druzdzel and L.C. van der Gaag, "Elicitation of probabilities for belief networks: combining qualitative and quantitative information", *Proc 11th Ann Conf on Uncertainty in Artificial Intelligence (UAI-95)*, pp. 141-148, Montreal, Quebec, Canada, August, 1995.

- [5] M.K. Druzdzel and L.C. van der Gaag, "Building Probabilistic Networks: Where Do the Numbers Come From?", *IEEE Transactions on Knowledge and Data Engineering*, vol. 12 no. 4, pp. 481-486, 2000.
- [6] N.E. Fenton, P. Krause, M. Neil, "Software Measurement: Uncertainty and Causal Modelling", *IEEE Software* vol. 10, no. 4, pp. 116-122, 2002.
- [7] N.E. Fenton, W. Marsh, M. Neil, P. Cates, S. Forey, M. Taylor, "Making Resource Decisions for Software Projects", *Proc 26th International Conference on Software Engineering (ICSE2004)*, May 2004, Edinburgh, United Kingdom, IEEE Computer Society, ISBN 0-7695-2163-0, pp. 397-406, 2004.
- [8] N.E. Fenton, M. Neil, P. Hearty, D. Marquez, W. Marsh, P. Krause, R. Mishra, "Predicting Software Defects in Varying Development Lifecycles using Bayesian Nets", *Information & Software Technology*, vol. 49, no. 1, pp. 32-43.
- [9] D. Heckerman, A. Mamdani, M. Wellman, "Real-world applications of Bayesian networks", *Comm ACM*, vol. 38, no. 3, pp. 25-26, 1995.
- [10] K. Huang and M. Henrion "Efficient Search-Based Inference for Noisy-OR Belief Networks", *Twelfth Conference on Uncertainty in Artificial Intelligence*, Portland, OR, pp. 325-331. 1996.
- [11] Hugin A/S, "Hugin Software", www.hugin.com, 2007
- [12] D. Kahneman P. Slovic, and Tversky, *A. Judgment Under Uncertainty: Heuristics and Biases*, Cambridge, UK: Cambridge University Press, 1982.
- [13] D. Koller and A. Pfeffer, "Object-Oriented Bayesian Networks", *Proceedings of the 13th Annual Conference on Uncertainty in AI (UAI)*, Providence, Rhode Island, August 1997, pp. 302--313, 1997.
- [14] Laskey K. "Sensitivity Analysis for Probability Assessments in Bayesian Networks." *IEEE Transactions on Systems, Man, and Cybernetics* vol. 25, no. 6, June 1995, pp. 901-909.
- [15] K.B. Laskey and S. Mahoney, "Network fragments: representing knowledge for constructing probabilistic model networks", *13 Annual Conference on Uncertainty in AI*, <http://site.gmu.edu/~klaskey/lectures.html>, 1998.
- [16] K.B. Laskey, S.M. Mahoney, "Network engineering for agile belief network models", *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 4, 487-498, 2000.
- [17] B. Littlewood, L. Strigini, D. Wright, N.E. Fenton, M. Neil, "Bayesian Belief Networks for Safety Assessment of Computer-based Systems", *System Performance Evaluation Methodologies and Applications*, E. Gelenbe (ed:), CRC Press, Boca Raton ISBN 0-8493-2357-6, pp. 349-364, 2000
- [18] S. Mahoney and K. Laskey "Network Engineering for Complex Belief Networks." *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence* (1996), pp. 389-396.
- [19] P. S Maybeck, *Stochastic models, estimation and control*. Volume 1, Academic Press, New York, 1979.
- [20] S. Monti, G Carenini, "Dealing with the expert inconsistency in probability elicitation", *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 4, pp. 499-508, 2000.

- [21] M. Neil, N.E. Fenton, L. Nielsen, "Building Large-scale Bayesian Networks", *The Knowledge Engineering Review*, vol. 15, no. 3, pp. 257-284, 2000.
- [22] M. Neil, N.E. Fenton, S. Forey and R. Harris, "Using Bayesian Belief Networks to Predict the Reliability of Military Vehicles", *IEE Computing and Control Engineering Journal*, vol. 12, no. 1, pp. 11-20, 2001.
- [23] M. Neil, B. Malcolm and R. Shaw "Modelling an Air Traffic Control Environment Using Bayesian Belief Networks". *21st International System Safety Conference*, August 4 - 8, 2003, Ottawa, Ontario, Canada.
- [24] M. Neil., Krause P., Fenton N. Software Quality Prediction Using Bayesian Networks in Software Engineering with Computational Intelligence, (edited by Khoshgoftaar T. M). The Kluwer International Series in Engineering and Computer Science, Volume 731, 2003.
- [25] M. Neil, N.E. Fenton, M. Taylor, "Using Bayesian Networks to model Expected and Unexpected Operational Losses", *Risk Analysis: An International Journal*, vol. 25, no. 4, pp. 963-972, 2005.
- [26] J. Pearl, "Graphical models, causality, and intervention", *Statistical Science*, vol. 8, no. 3, pp. 266-273, 1993.
- [27] S. Renooij, "Probability elicitation for belief networks: issues to consider". *The Knowledge Engineering Review*, vol. 16, no. 3, pp. 255-269, 2000.
- [28] M. Takikawa and B. D'Ambrosio, "Multiplicative Factorization of Noisy-Max", *Proceedings of the Uncertainty in AI conference*, 1999.
- [29] L.C. van der Gaag, S. Renooij, C.L.M. Witteveen, B.M.P. Aleman, B.G. Taal, "Probabilities for a Probabilistic Network: A Case-study in Oesophageal Carcinoma", University of Utrecht, UU-CS-2001-01, January, 2001.
- [30] L.C. van der Gaag, S. Renooija, C.L.M. Wittemana, B.M.P.Alemanb, B.G. Taal. "Probabilities for a probabilistic network: a case study in oesophageal cancer", *Artif Intell Med*, vol. 25, no. 2, pp.123-48, 2002
- [31] M.P Wellman., "Fundamental concepts of qualitative probabilistic networks", *Artificial Intelligence*, vol. 44, no. 3, pp. 257-303, 1990.
- [32] A. Zagorecki and M. Druzdzal. "An Empirical Study of Probability Elicitation under Noisy-OR Assumption", *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS.2004)*, V. Barr & Z. Markov (eds), pp. 880-885, Menlo Park, CA: AAAI Press, 2004.

List of Figures and Tables

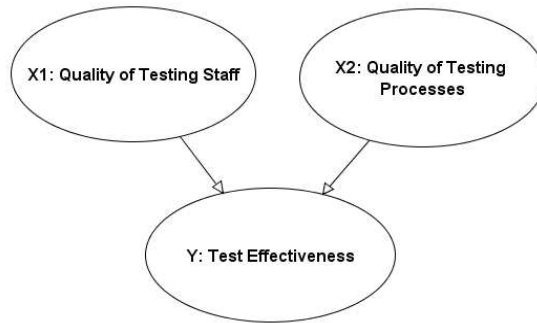


Figure 1: Typical qualitative BN fragment

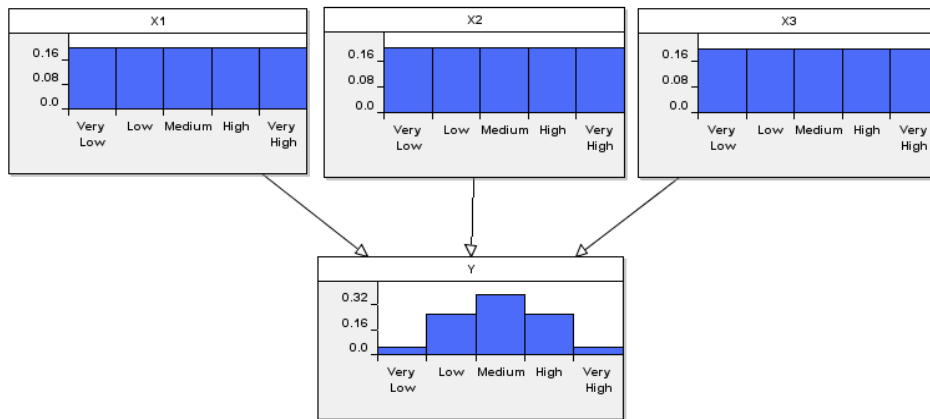


Figure 2: WMEAN function for Y given X_1, X_2, X_3

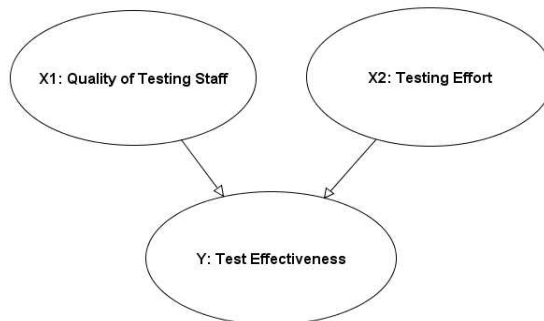


Figure 3: Revised BN fragment

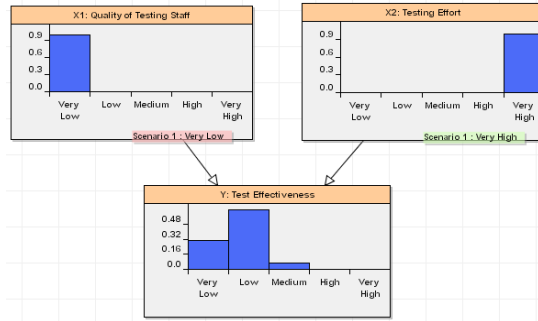


Figure 4: WMIN function for Y. Quality of Testing Staff = “Very Low”, with $w_1 = 3$, Testing Effort = “Very High”, with $w_2 = 1$

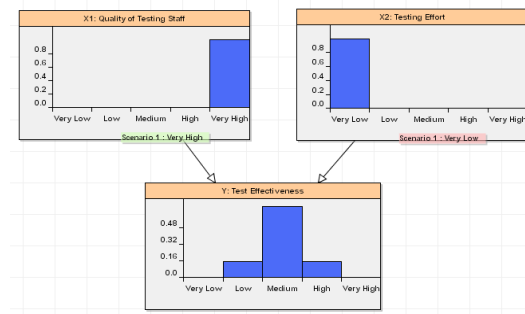


Figure 5: WMIN function for Y. Quality of Testing Staff = “Very High””, with $w_1 = 3$, Testing Effort = “Very Low””, with $w_2 = 1$

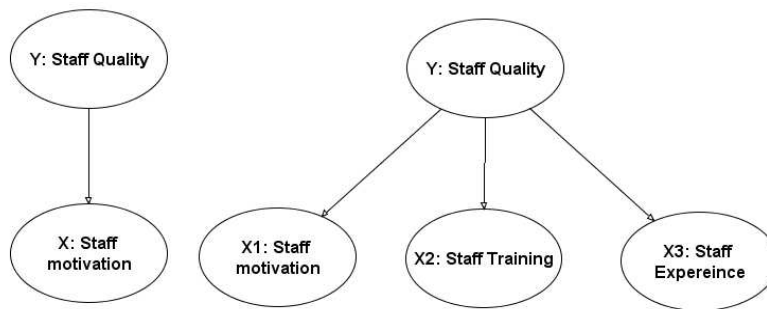


Figure 6: Ranked indicator examples (single indicator on LHS; multiple indicator on RHS)

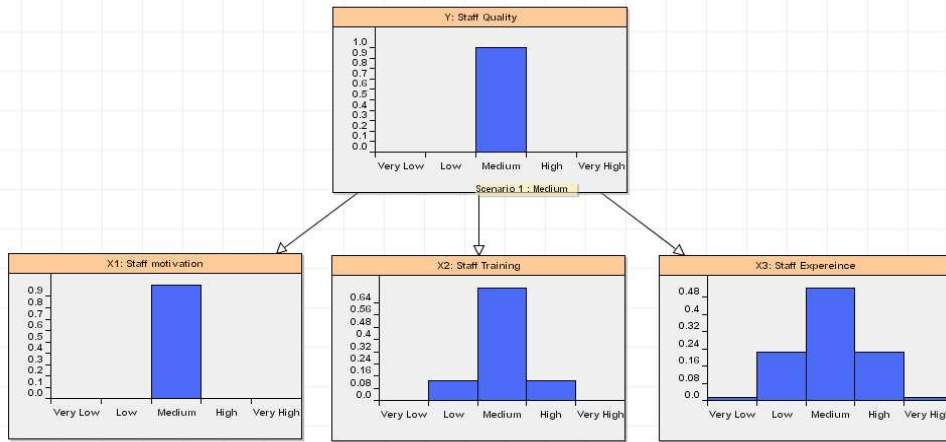


Figure 7: Marginal distributions for indicators, $p(X_i)$, given causal node $Y = \text{medium}$

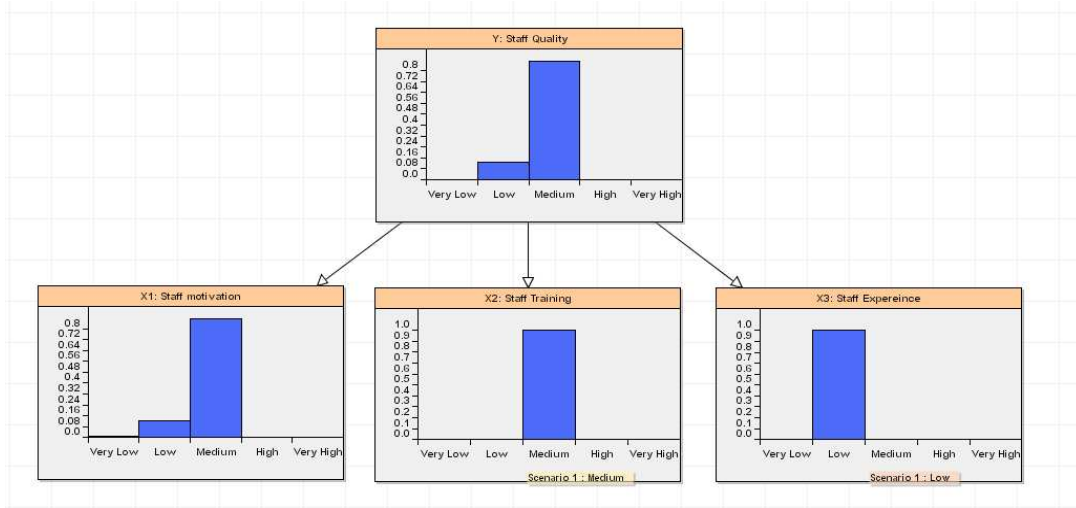


Figure 8: Inferring $p(Y)$ from $X_2 = \text{medium}$ and $X_3 = \text{low}$ observations

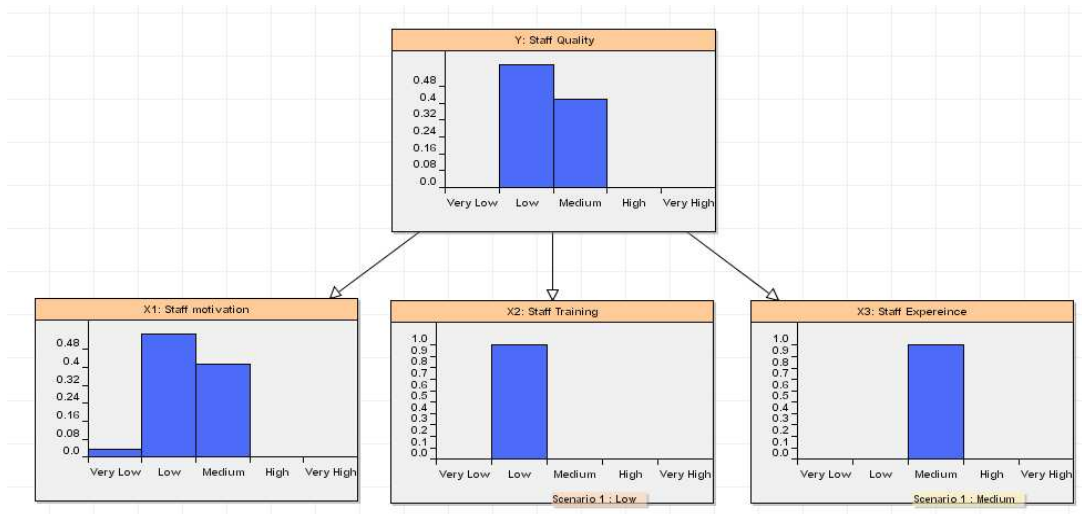


Figure 9: Inferring $p(Y)$ from $X_2 = low$ and $X_3 = medium$ observations

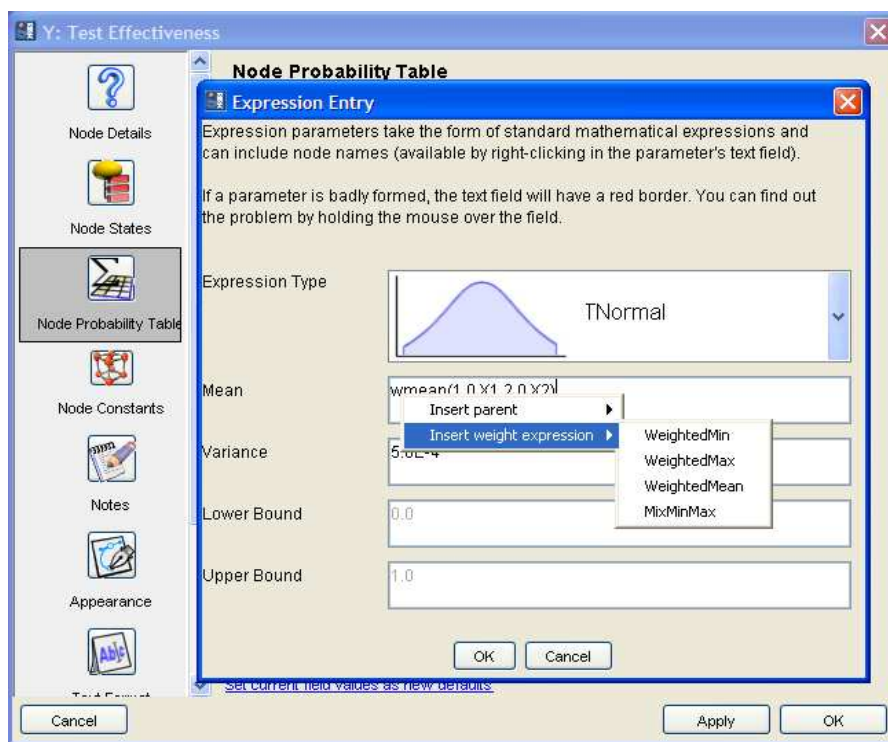


Figure 10: Declaring a rank weight expression for a node in AgenaRisk

X2: Quality of Testing Proces...	Very Low				
X1: Quality of Testing Staff	Very Low	Low	Medium	High	Very High
Very Low	0.9395708	0.047125354	0.009345794	9.4250706E-4	9.4250706E-4
Low	0.049451094	0.8953817	0.046728972	0.009425071	0.009425071
Medium	0.009890218	0.047125354	0.88785046	0.047125354	0.047125354
High	9.890218E-4	0.009425071	0.046728972	0.8953817	0.8953817
Very High	9.890219E-5	9.4250706E-4	0.009345794	0.047125354	0.047125354

X2: Quality of Testing Proces...	Low				
X1: Quality of Testing Staff	Very Low	Low	Medium	High	Very High
Very Low	0.047125354	0.009345794	9.4250706E-4	9.4250706E-4	9.4250706E-4
Low	0.8953817	0.046728972	0.009425071	0.009425071	0.009425071
Medium	0.047125354	0.88785046	0.047125354	0.047125354	0.047125354
High	0.009425071	0.046728972	0.8953817	0.8953817	0.09425071
Very High	9.4250706E-4	0.009345794	0.047125354	0.047125354	0.84825635

X2: Quality of Testing Proces...	Medium				
X1: Quality of Testing Staff	Very Low	Low	Medium	High	Very High
Very Low	0.047125354	0.009794319	9.4250706E-4	9.4250706E-4	9.4250706E-4
Low	0.8953817	0.93046033	0.047125354	0.009425071	0.009425071
Medium	0.047125354	0.048971597	0.8953817	0.047125354	0.047125354
High	0.009425071	0.009794319	0.047125354	0.8953817	0.09425071
Very High	9.4250706E-4	9.794319E-4	0.009425071	0.047125354	0.84825635

X2: Quality of Testing Proces...	High				
X1: Quality of Testing Staff	Very Low	Low	Medium	High	Very High
Very Low	0.047125354	0.009794319	9.4250706E-4	9.4250706E-4	9.890219E-5
Low	0.8953817	0.048971597	0.047125354	0.009425071	9.890218E-4
Medium	0.047125354	0.93046033	0.8953817	0.047125354	0.009890218
High	0.009425071	0.009794319	0.047125354	0.8953817	0.049451094
Very High	9.4250706E-4	9.794319E-4	0.009425071	0.047125354	0.9395708

X2: Quality of Testing Proces...	Very High				
X1: Quality of Testing Staff	Very Low	Low	Medium	High	Very High
Very Low	0.047125354	0.009794319	9.4250706E-4	9.4250706E-4	9.890219E-5
Low	0.8953817	0.048971597	0.047125354	0.009425071	9.890218E-4
Medium	0.047125354	0.93046033	0.8953817	0.047125354	0.009890218
High	0.009425071	0.009794319	0.047125354	0.8953817	0.049451094
Very High	9.4250706E-4	9.794319E-4	0.009425071	0.047125354	0.9395708

Figure 11: Manually declared table for $p(Y | X_1, X_2)$ (values normalised)

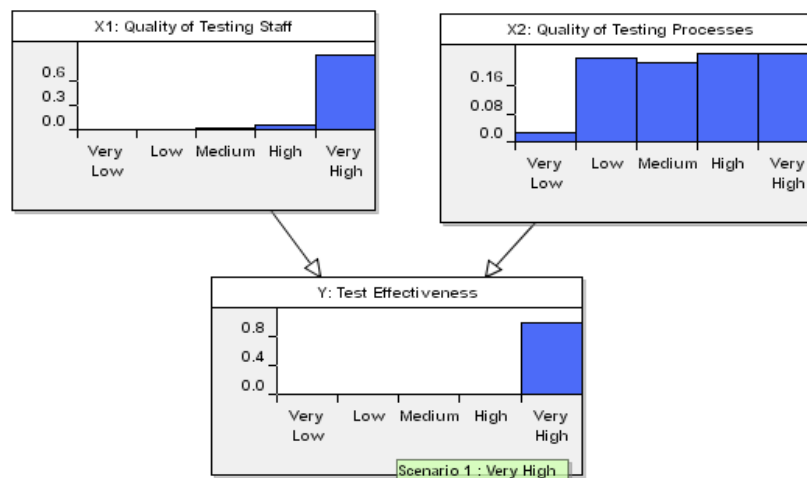


Figure 12: Results from sensitivity testing of the manual NPT

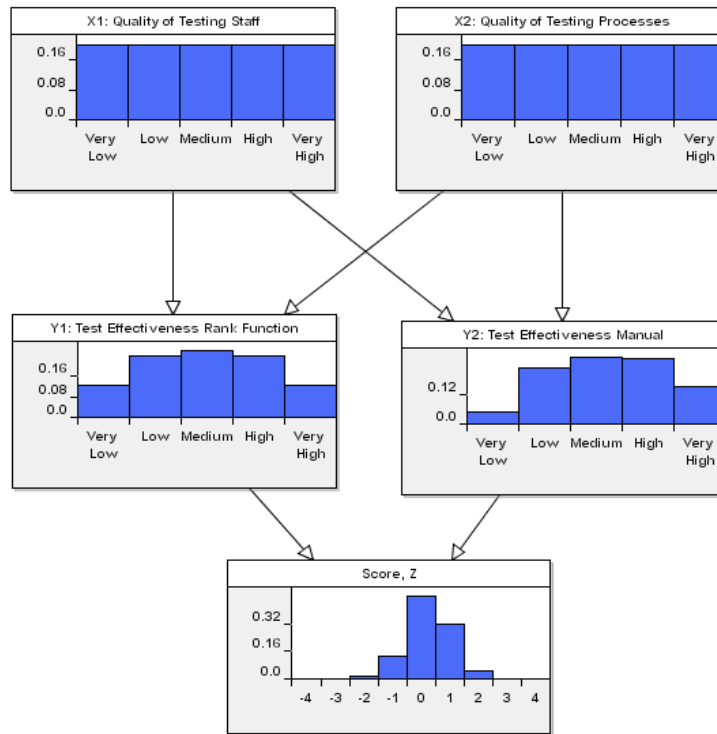


Figure 13: Results from sensitivity testing of the manual NPT

Table 1 Part of 'truth table' used to help elicit ranked node function

Quality of testing staff	Quality of testing processes	Test effectiveness
very high	very high	very high
very low	very low	very low
very low	very high	low
very high	very low	high

Table 2: Effort savings (in two models) for new approach to eliciting NPTs

	BN Model: Safety assessment (28 nodes) [17]	BN Model: Software defect prediction (31 nodes) [6]
Size of most complex NPT	A node with 5 parents had an NPT with 324 entries. The size of the NPT was already artificially reduced by making two of the parent nodes a 2-point rather than 3-point scale. Moreover, ideally a 5-point scale would have been preferred for all nodes.	14 nodes each having an NPT with 125 entries. This was because each node was on a 5-point scale and, because of previous experience, all nodes were limited to two parents.
Total effort to elicit most complex NPT manually	72 hours (6 experts in sessions totalling 12 hours). This effort only elicited 81 entries manually; the rest were completed using interpolation techniques.	24 hours (3 experts for a full day)
Total effort to elicit same NPT using new approach	6 hours (experts need only agree on 7 parameters – the 5 parent weights, the variance, and the function type). Note also that with this approach a full 5-point scale for all nodes is possible without any extra effort.	1.5 hours (3 experts for 30 minutes to agree 4 parameters).
Effort saved for single NPT	66 hours.	22.5 hours
Potential effort saved in building full model	143 hours (total of 28 hours compared with 171 hours) There were 11 other less complex NPTs that required an average of 9 hours to build compared with 2 hours using the new approach.	Minimum of 315 hours (total of 21 hours compared with 336 hours) There were 14 similar nodes. The saving is a minimum because with new approach would have avoided introducing 5 ‘dummy’ nodes simply to ensure no more than 2 parents for each node.
Percentage saving of effort on manual approach	84%	93% (minimum)