Ranked nodes: A simple and effective way to model qualitative judgements in large-scale Bayesian Networks

Norman Fenton and Martin Neil

Risk Assessment and Decision Analysis Research Group
Department of Computer Science, Queen Mary, University of London

and

Agena Ltd

21st Feb, 2005

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) for each node. In the absence of hard data, we must rely on domain experts to provide, often subjective, judgements to inform the 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. The approach has been automated and is thus accessible to all types of domain experts, including those with little statistical expertise. The result has been that such individuals have been able to build large-scale realistic BN models that solve important problems. Hence, this work represents a breakthrough in BN research and technology since it can make the difference between being able to build realistic BN models and not.

Keywords: Bayesian networks, node probability tables, ranked nodes, doubly truncated Normal distribution

1. Introduction

In recent years Bayesian Networks (BNs) have become increasingly recognised as a potentially powerful solution to complex risk assessment problems [Heckerman et al 1995]. 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 [Neil et al, 2000];
- Military vehicle reliability, [Neil et al 2001];
- Risk of mid-air collisions in Air Traffic Control [Neil et al 2003a];
- Software defect prediction in consumer electronics products [Fenton et al 2002, Neil et al 2003b];

All of these applications involved building extremely large-scale BN models. By extending the ideas of object-oriented Bayesian Networks [Koller and-Pfeffer 1997] we developed a range of methods that could be deployed in practice. For example, in [Neil et al 2000] 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 [Hugin 2005] and AgenaRisk [Agena 2005]. However, that work said little about the even harder problem of building Node Probability Tables (NPTs) in large-scale BNs. Hence, this paper focuses on one especially important part of this problem: how to build NPTs for the qualitative nodes that occur so frequently when modelling real-world problems. Consider, for example, the BN fragment shown in Figure 1.

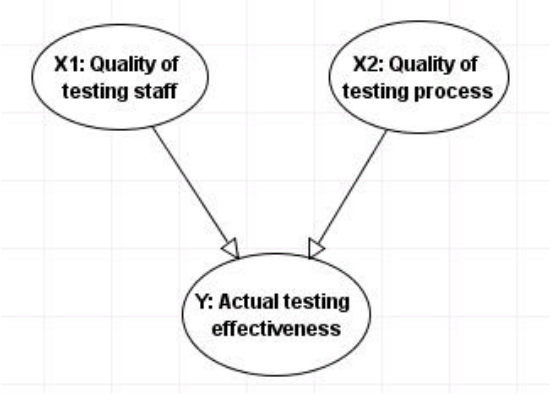

Figure 1 Typical qualitative BN fragment

Such fragments, which are very typical of those that frequently occur in the real-world models already cited, are characterised by the following:

- The node values are typically measurable only on a subjective scale like {very low, low, medium, high, very high}
- 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.

Assuming each of the nodes has five states ranging from very low to very high, 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. If the number of states increases to seven and/or the node Y has an additional parent then exhaustive elicitation becomes infeasible. Moreover, real-world models invariably involve dozens of fragments like these.

Hence, the problem and challenge is to produce an appropriate NPT for nodes like Y that makes the most of limited expert elicitation. This problem is certainly not new. It has, for example, been addressed in [Druzdzel and van der Gaag 1995, Takikawa and D'Ambrosio 1999, Wellman 1990] and there have been serious case studies on specific elicitation techniques [van der Gaal et al 2001, Monti and Carenini 2000, Laskey and Mahoney 1998, 2000]. Also the Noisy-OR is well established as a standard way of encoding expertise in large NPTs [Huang and Henrion 1996], but has the disadvantage that it applies only to Boolean nodes and implicitly ignores the interaction effects between variables. What has been missing is a general, easily accessible approach that could be used directly with domain experts who are neither expert probability theorists nor mathematicians. We were determined that we could produce a method that could be easily explained to such people as well as easily used. The approach presented here has evolved over a number of years from the process of engaging with domain experts in real commercial situations.

We begin in Section 2 by formalising the notion of ranked nodes along with the conditions under which they occur most commonly in BNs. In Section 3 we describe the simple class of functions required to generate the NPTs for these ranked nodes. In each case we find that a special function (the doubly truncated Normal) whose mean is a type of weighted function of the parent nodes is sufficient. In Section 4 we describe the other instance where ranked nodes commonly occur, namely as indicator nodes. Again the doubly truncated Normal is sufficient for generating the necessary NPTs for this class of ranked nodes. Throughout the paper we explain how the method has been fully implemented in the AgenaRisk software so that we can justify our claim that the approach can (and has) been used effectively by domain experts (with little statistical expertise) to generate large-scale NPTs.

2. Ranked nodes

Consider the following examples of BN nodes:

- "Staff experience": {very low, low, average, high, very high}
- "Requirements complexity": {very complex, complex, average complexity, simple, very simple}
- "Product quality": {abysmal, very poor, poor, average, good, very good, perfect}

These nodes are discrete variables whose states are expressed on an ordinal scale. However, what makes them special is that we can assume that not only is the scale ordinal, but that it can be mapped onto a bounded numerical scale that is continuous and monotonically ordered. Under such assumptions we shall call the node a ***ranked node***.

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

2.1 The Problem

When building an NPT there are typically situations where we want to use a simple averaging scheme as a guide to defining the "central tendency" of the child node based on a set causal parent node values. For example, in [Fenton et al 2002] (in attempting to construct the NPT for a node like Y of Figure 1) we adopted an approach based on sampling values, getting 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 find 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 where, for example, 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_1 is 'average' then we would be confident that X_2 is 'very high' but not as confident as in the above case.
- In general if Y is 'very high' then the lower value that X_1 is the more confident we are that X_2 is 'very high'.

For example, using an interpolated Beta distribution to approximate Y (as in [Fenton et al 2002]), does ***not*** preserve these back-propagation beliefs. However, a straightforward solution for defining the NPT for $p(Y \mid \underline{X})$ in such a way that these various properties ***are*** all satisfied is given in Section 2.2.

2.2 Theory

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

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

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 $y_i = \underline{b}\,\underline{x} + e_i$ with $e \sim N(0, s_Y^2)$ and where the contribution to the variance of Y is s_Y^2. We feel the regression analogy is apt here 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 [Cozman and Krotkov 1997], where all nodes are truncated in the [0, 1] region. Unlike the regular Normal distribution (which must be in the range –infinity to +infinity) the TNormal has *finite* end points. We denote the TNormal by TNormal(μ, s^2, 0, 1) where μ is the mean and s^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 $s^2 \to \infty$, and highly skewed distributions, achieved when $s^2 \to 0$.

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 could even 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, s_Y^2, in the linear regression model is simply the inverse of the sum of the weights:

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

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:

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

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, $s_Y^2 = 0.001$. Then the joint distribution generated will be:

$$p(Y \mid \underline{X}) = TNormal\left[\frac{200X_1 + 300X_2 + 500X_3}{200 + 300 + 500}, \frac{1}{200 + 300 + 500}, 0, 1\right]$$

This is equivalent to:

$$p(Y \mid \underline{X}) = TNormal\left[\frac{2X_1 + 3X_2 + 5X_3}{10}, 0.001, 0, 1\right]$$

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 approach is a means to an end.

It is important to note that the theory and practice must go hand-in-hand. For the purpose of building realistic NPTs that adequately capture expert judgement, the existence of a good theoretical approach

is insufficient. The approach we have described works in practice ***because*** it is supported by a tool, such as AgenaRisk, 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.

It is also worth noting that domain experts can change the scale (from say a 5-point scale to a 7-point scale) 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.

2.3 Example

We use the example shown in Figure 1. In this example we have two cause nodes, $\underline{X} = \{X_1, X_2\}$. The weights associated with the cause nodes are respectively 3, 1 and the variance is 0.01. Hence

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

Using the AgenaRisk software this distribution can either be entered directly as an expression for the node Y, or via a simple wizard.

When we have crisp evidence for the parents, $\underline{X} = \{X_1 = x_1, X_2 = x_2\}$, the prediction of $p(Y)$ has a mean value equal to the weighted average and a variance that reflects our confidence in the result. Figure 2 shows that the result is weighted by the importance of the parent nodes. Since X_2 is less important than X_1 the resulting $p(Y)$ is biased towards the X_1 value.

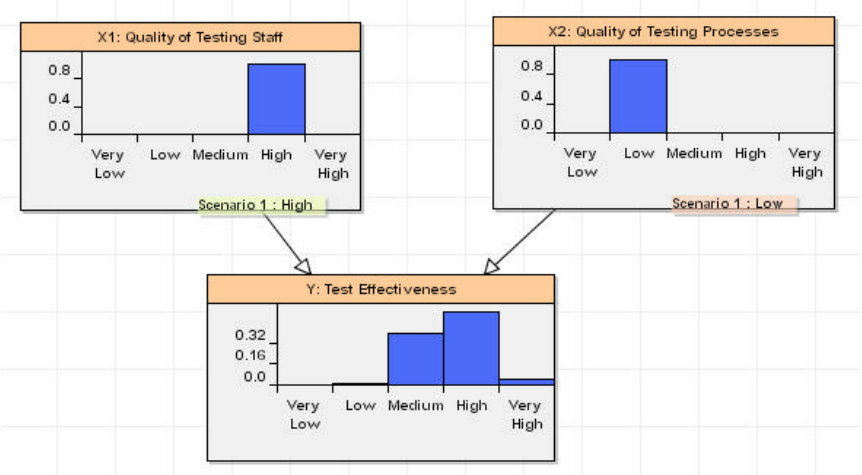

Figure 2 Prediction of $p(Y)$ **given** $\underline{X} = \{X_1 = high, X_2 = medium\}$

The relative importance reflected in the weighting scheme used is also evident when we propagate evidence from effects to causes. Those nodes with higher weights will be identified as the most likely causes of the consequence. This is shown in Figure 3 where we can see that a high value of X_1 is identified as the most likely cause of the high value of Y because of its higher weight.

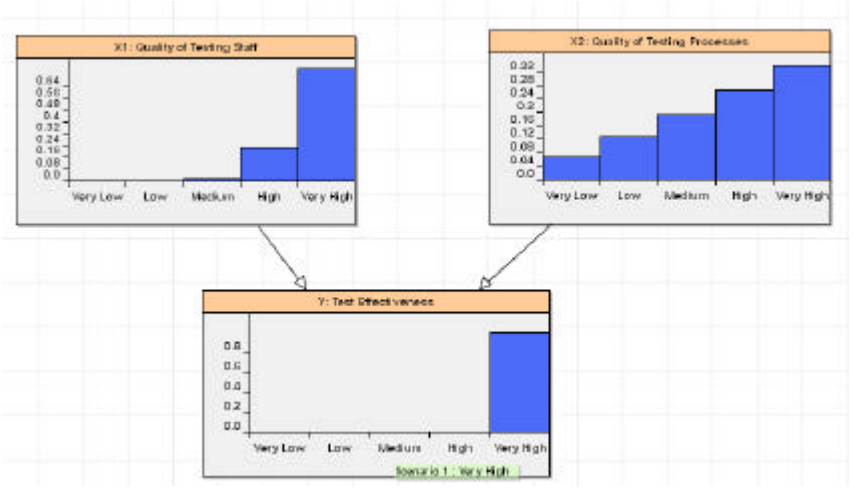

Figure 3 Prediction of $p(\underline{X})$ **given** $Y =$ very high

3. Alternative functions for ranked causes

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 4.

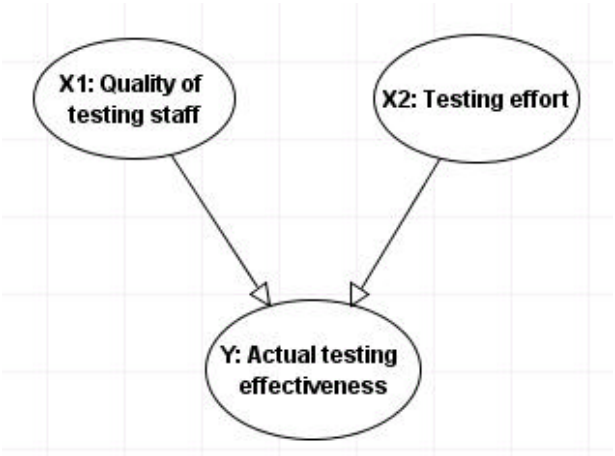

Figure 4 Revised BN fragment

In this case we elicit 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 is 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 are insufficient then 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 s_Y^2 that quantifies our uncertainty about the result, thus giving:

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

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 the normal 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 $s_Y^2 = 0.01$) yields the results shown in Figure 5 and Figure 6.

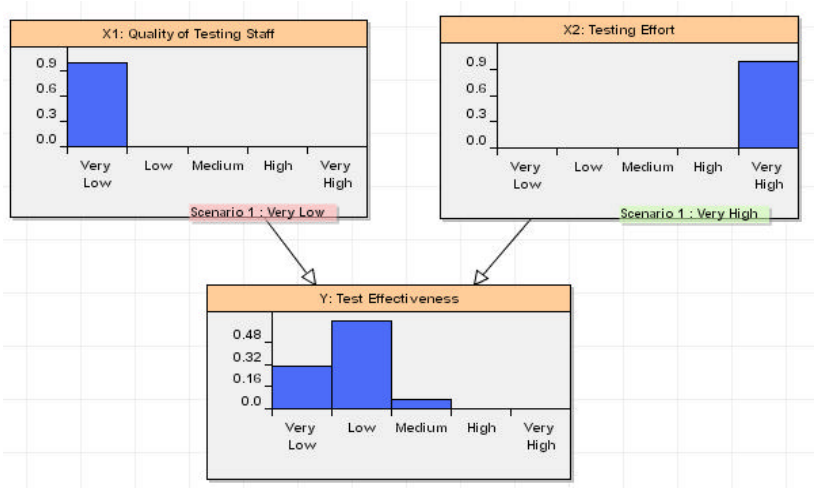

Figure 5 WMIN function for Y. Quality of Testing Staff = "very low" , Testing Effort = "very high"

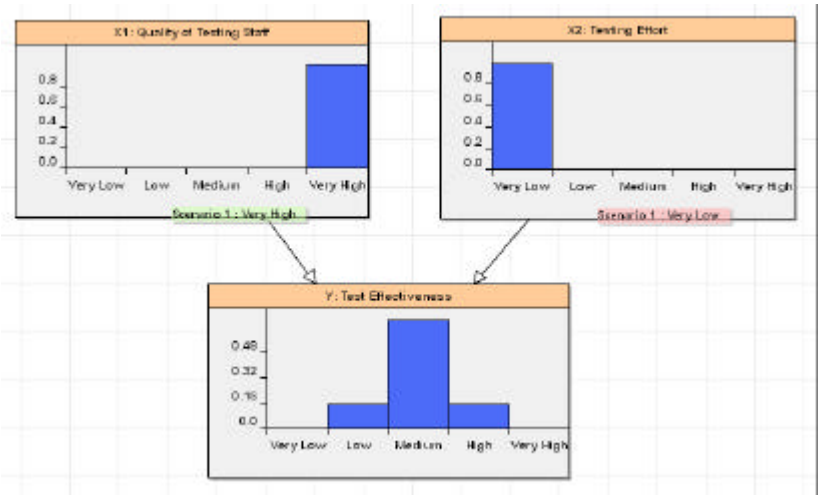

Figure 6 WMIN function for Y. Quality of Testing Staff = "very high", Testing Effort = "very low"

Again it is important to stress that constructing the necessary NPT requires experts only to go through two simple steps involving the use of a Wizard in AgenaRisk to select the weighted expression and then use slider bars to select the weights and apply the variance value.

We can also use an analogous *WMAX* function:

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

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

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

In each case the user need only supply the parameters to generate the NPT. We have found that this set of functions is sufficient to generate almost any ranked node NPT in practice.

4. Ranked Indicators

In addition to their occurrence as described in Section 2, ranked nodes occur frequently as *indicators* of other ranked nodes, such as shown in Figure 7. 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.

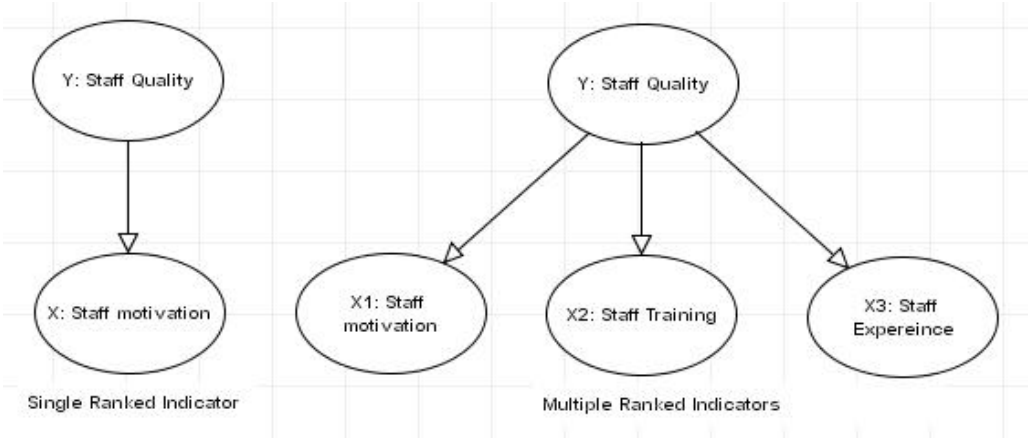

Figure 7 Ranked indicator examples

4.1 Theory

Indicator nodes operate in a similar way to "filter" nodes in a Kalman filter. In object radar tracking applications a Kalman filter operates by filtering out the noise from noisy radar observations in order to reveal the true position of the object in space. Here we can think of the indicators as providing noisy or imperfect observations and the parent node as the true, perhaps unobservable or that cannot be measured directly or economically, value awaiting estimation [Maybeck 1979]. In a Kalman filter we wish to condition our estimate for the "true" value on the data to hand from each of our "indicator" nodes assuming each indicator is Gaussian distributed.

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

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

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

$$p(X_i \mid Y) = TNormal(Y, s_i^{\,2}, 0, 1)$$

This assumes, of course, that the nodes Y and X_i are on the same scale. The expert simply has to specify the variance parameter, $s_i^{\,2}$, whose inverse acts as a "credibility index" — the higher the credibility index the greater the correlation between the indicator and the parent cause node.

Clearly, indicator nodes are correlated with each other by virtue of the diverging d-connected structure. 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.

Before showing some examples of how ranked indicators work it is instructive to discuss a small example Kalman filter and compare it with ranked indicators. In general a Kalman filter uses multiple observations to estimate the hidden node. Given two observations $X_1 = x_1$ and $X_2 = x_2$, each with variance s_1^2 and s_2^2 we can estimate the unknown parent, Y, as $N(m, s^2)$ with parameters:

$$m_Y = \left[\frac{s_2^2}{s_1^2 + s_2^2}\right] x_1 + \left[\frac{s_1^2}{s_1^2 + s_2^2}\right] x_2$$

$$\frac{1}{s_Y^2} = \frac{1}{s_1^2} + \frac{1}{s_2^2}$$

Figure 8 shows a Kalman filter constructed in AgenaRisk with two indicator nodes, X_1 and X_2, each with variance $s_1^2 = 500$ and $s_2^2 = 1000$ respectively. The blue coloured histogram shows $p(Y \mid X_1 = 500)$. The green coloured histogram shows $p(Y \mid X_2 = 700)$ and the red coloured histogram shows $p(Y \mid X_1 = 500, X_2 = 700)$. Notice that under $p(Y \mid X_1 = 500, X_2 = 700)$ the estimates for Y are $N(566,333)$. Thus the quality of the estimate of Y is better than the quality of each of the observations, X_1 and X_2. In this way a series of observations that are in themselves inaccurate can, when taken together, give accurate estimates of the true value.

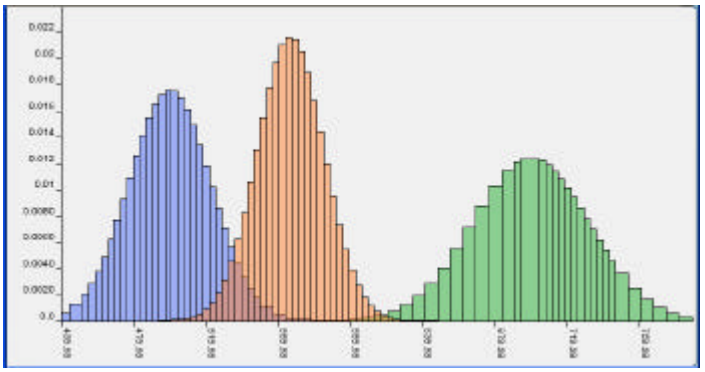

Figure 8: Kalman filter results for the three scenarios

Ranked indicators use a similar philosophy to that used in a Kalman filter. The difference is that the indicator nodes are bounded on [0, 1] and rather than use, analytically solved, Normal distributions we use doubly truncated Normal distributions solved numerically (there is no analytical solution to

$\prod_{i=1}^{n} p(X_i \mid Y)$ when the indicators are doubly truncated Normal). Given this, we should not necessarily expect the results achieved using a rank 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 that our approach very closely approximates a Kalman filter in the region where $m_Y = 0.5$. 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 $m_Y \to 0.5$ and as a result the correlation between the indicators, X, and the Y approaches zero. Practically speaking, for 5 and 7 point scaled rank nodes, setting $s_i^2 > 0.1$ 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 $m_Y \to x_i$ but rather $m_Y \to 0.5$.

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 $s_1^2 = s_2^2 = 0.01$ 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 is helpful where we have 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 [Neil et al 2000] 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 indictors with parameterised values for s_i^2.

4.2 Example

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:

$$p(X_1 \mid Y) = TNormal(Y, 0.001, 0, 1),\ p(X_2 \mid Y) = TNormal(Y, 0.008, 0, 1),\ p(X_3 \mid Y) = TNormal(Y, 0.02, 0, 1)$$

Figure 9 shows how the marginal probability distribution on the indicator nodes $\underline{X} = \{X_1, X_2, X_3\}$ given an observation on the parent, $Y = medium$. Clearly X_1 is more highly correlated with Y than either X_2 or X_3.

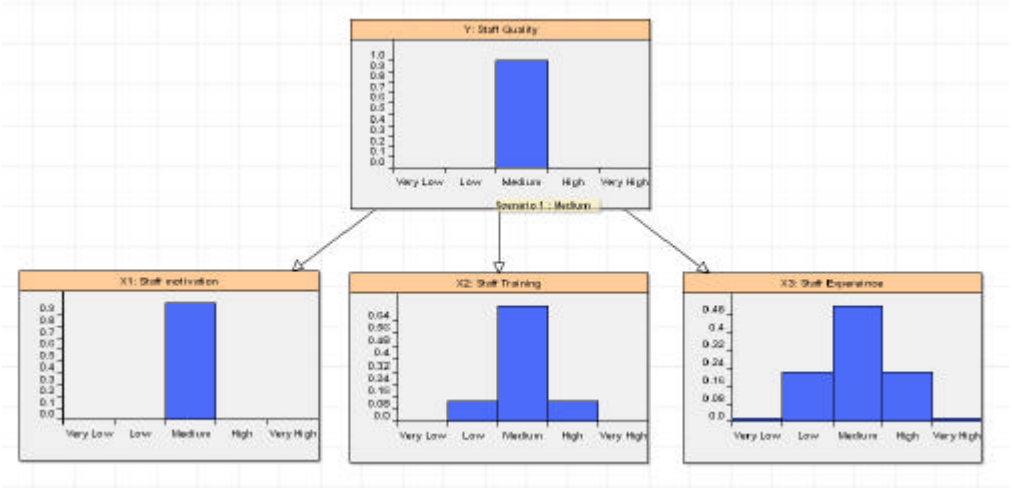

Figure 9 Marginal distributions for indicators, $p(X_i)$, given causal node $Y = medium$

Figure 10 shows how we can use the indicator nodes to infer the true state of the parent node, Y, from the observations $X_2 = medium, X_3 = low$. 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 11

where we invert the observation values such that $X_2 = low, X_3 = medium$ and notice how the distribution on Y is influenced more highly by indicator X_2 in both figures.

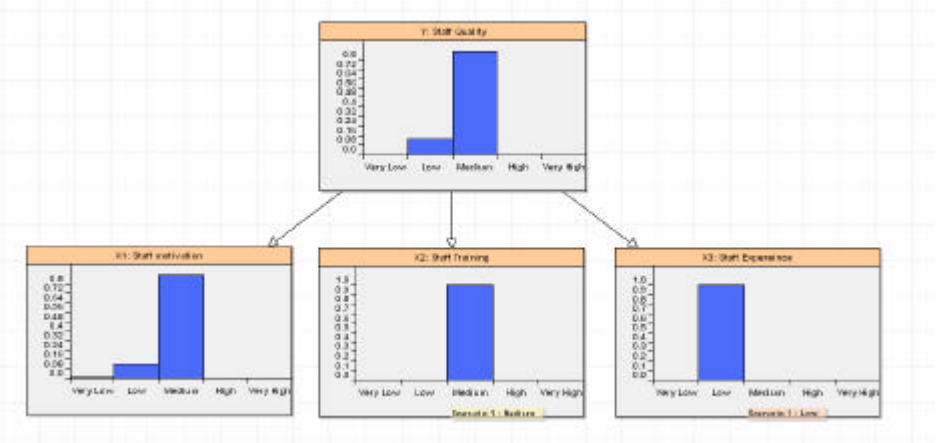

Figure 10 Inferring $p(Y)$ from $X_2 = medium$ and $X_2 = low$ observations

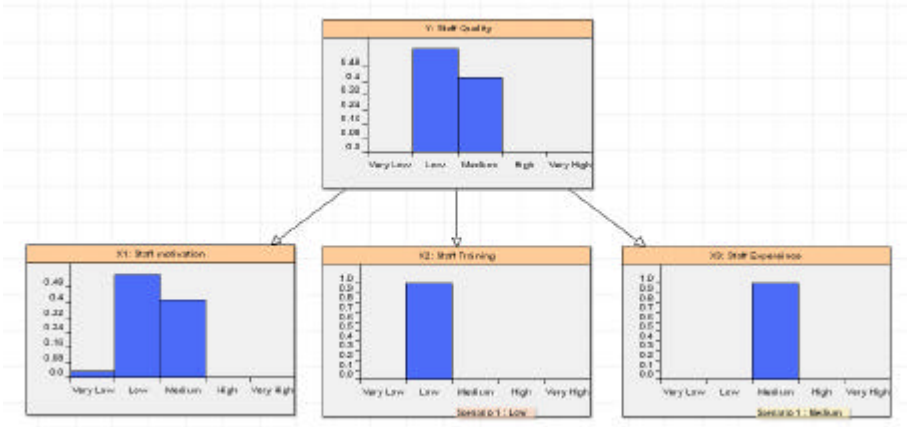

Figure 11 Inferring $p(Y)$ from $X_2 = low$ and $X_2 = medium$ observations

5. 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 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 often makes the difference between being able to build realistic BN models and not. The BN solutions to real-world problems described in [Fenton et al 2002, Neil et al 2001, 2003] all 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 tool. An earlier prototype of the automated version was used extensively to build the models described in [Fenton et al 2004] 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 domain experts, who were not statisticians, were able to build and tailor serious models that captured their beliefs well. On the second level, the models produced predictions and decision support insights that were demonstrably better than previous methods had produced.

Acknowledgements

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), and MODIST (EC Framework 5 Project IST-2000-28749).

References

[Agena 2005] Agena Ltd, "AgenaRisk Software Package", www.agenarisk.com, 2005

[Cozman and Krotkov 1997] Cozman F. and Krotkov E. "Truncated Gaussians as Tolerance Sets". Technical Report CMU-RI-TRI, Robotics Institute, Carnegie Mellon University, 1997.

[Druzdzel and van der Gaag 1995] Druzdzel MK and van der Gaag LC, "Elicitation of probabilities for belief networks: combining qualitative and quantitiative information", Proc 11th Ann Conf on Uncertainty in Artifical Intelligence (UAI-95), 141-148, Montreal, Quebec, Canada, August, 1995.

[Druzdzel MK and van der Gaag 2000] Druzdzel MK and van der Gaag LC, "`Building Probabilistic Networks: Where Do the Numbers Come From?'", IEEE Transactions on Knowledge and Data Engineering, 12(4):481-486, 2000.

[Fenton et al 2002] Fenton N, Krause P, Neil M, "Probabilistic Modelling for Software Quality Control", Journal of Applied Non-Classical Logics 12(2), 173-188, 2002

[Fenton et al 2004] Fenton NE, Marsh W, Neil M, Cates P, Forey S, Tailor M, "Making Resource Decisions for Software Projects", 26th International Conference on Software Engineering (ICSE2004) May 2004, Edinburgh, United Kingdom. IEEE Computer Society, ISBN 0-7695-2163-0, pp. 397-406, 2004.

[Heckerman et al 1995] Heckerman D, Mamdani A, Wellman M, "Real-world applications of Bayesian networks", Comm ACM, 38(3), 25-26, 1995.

[Huang and Henrion 1996] Huang K & Henrion M. "Efficient Search-Based Inference for Noisy-OR BeliefNetworks", Twelfth Conference on Uncertainty in Artificial Intelligence, Portland, OR, 325-331. 1996.

[Hugin 2005] Hugin A/S, "Hugin Software", www.hugin.com, 2005

[Koller and Pfeffer 1997] Koller D and Pfeffer A, "Object-Oriented Bayesian Networks", Proceedings of the 13th Annual Conference on Uncertainty in AI (UAI), Providence, Rhode Island, August 1997, pages 302--313, 1997.

[Laskey and Mahoney 1998] Laskey KB and Mahoney S, "Network fragments: representing knowledge for constructing probabilistic model networks", 13 Annual Conference on Uncertainty in AI, http://site.gmu.edu/~klaskey/lectures.html, 1998.

[Laskey and Mahoney 2000] Laskey, KB. Mahoney, S.M, "Network engineering for agile belief network models", IEEE Transactions on Knowledge and Data Engineering, 12 (4), 487-498, 2000.

[Maybeck 1979] Maybeck P. S. Stochastic models, estimation and control. Volume 1. Academic Press, New York, 1979.

[Monti and Carenini 2000] Monti, S. Carenini, G, "Dealing with the expert inconsistency in probability elicitation", IEEE Transactions on Knowledge and Data Engineering, 12(4), 499-508, 2000.

[Neil et al 2000] Neil M, Fenton NE, Nielsen L, "Building Large-scale Bayesian Networks", The Knowledge Engineering Review, 15(3), 257-284, 2000.

[Neil et al 2001] 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.

[Neil et al 2003a] Neil M. Malcolm B. and Shaw R. "Modelling an Air Traffic Control Environment Using Bayesian Belief Networks". 21st International System Safety Conference, August 4 - 8, 2003, Ottawa, Ontario, Canada.

[Neil et al 2003b] Neil M., 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.

[Takikawa and D'Ambrosio 1999] Takikawa M. and D'Ambrosio B. Multiplicative Factorization of Noisy-Max. Proceedings of the Uncertainty in AI conference ,1999.

[van der Gaag et al 2001] van der Gaag LC, Renooij S, Witteveen CLM, Aleman BMP, Taal BG, "Probabilities for a Probabilistic Network: A Case-study in Oesophageal Carcinoma", University of Utrecht, UU-CS-2001-01, January, 2001.

[Wellman 1990] Wellman M.P., "Fundamental concepts of qualitative probabilistic networks", Artificial Intelligence, 44(3), 257-303, 1990.