Using Ranked Nodes

The problem

When building real world risk maps you will often arrive at qualitative fragments such as that shown in Figure 1.

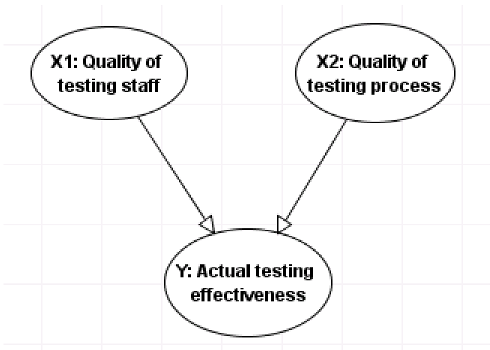

Figure 1 Typical qualitative risk map fragment

Such fragments 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 judgment.

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.

One solution would be get samples of 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’.

If we can assume 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 risk maps, the challenge is to capture the explicit simplicity of the weighted average approach 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'.

The solution: Use ranked nodes with the TNormal distribution

By defining such nodes as *ranked nodes* in AgenaRisk you will be able effortlessly to define NPTs that satisfy the criteria we described. That is because ranked nodes in AgenaRisk have a special distribution, called the Truncated Normal (or TNormal) that you can use to generate the NPTs you need simply by setting a two parameters.

Unlike the regular Normal distribution (which must be in the range –infinity to +infinity) the TNormal has *finite* end points. For ranked nodes these endpoints are 0 and 1 respectively. The TNormal is characterized by two parameters (*mean* and *variance*). This enables us to model a variety of shapes including a uniform distribution, achieved when the variance is very large, and highly skewed distributions, achieved when the variance is close to 0.

Suppose, for example, we have the risk map shown in Figure 1. We can define the NPT of Y to be TNormal with parameters as shown in Figure 2. Here the mean is a weighted average of X1 and X2 (with weights 3 and 1 respectively) and a fixed variance of 0.01.

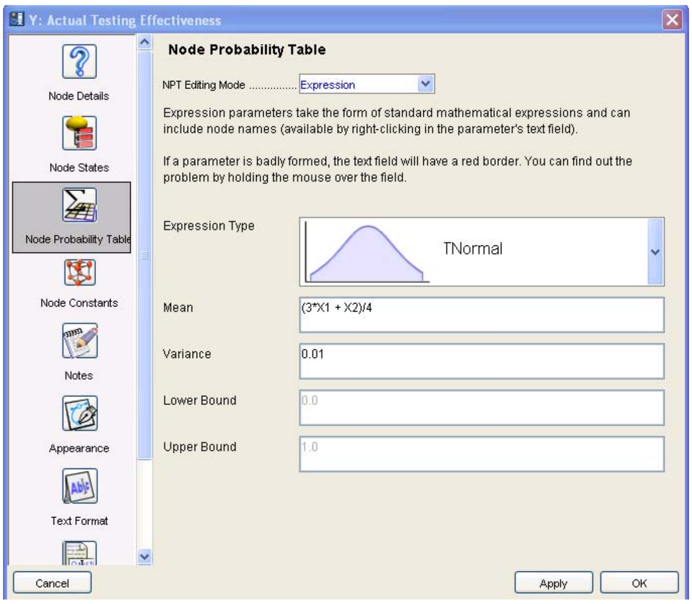

Figure 2 Using the TNormal for ranked nodes

Using the AgenaRisk this distribution can either be entered directly as an expression for the node Y, or via the simple wizard as shown in Figure 3.

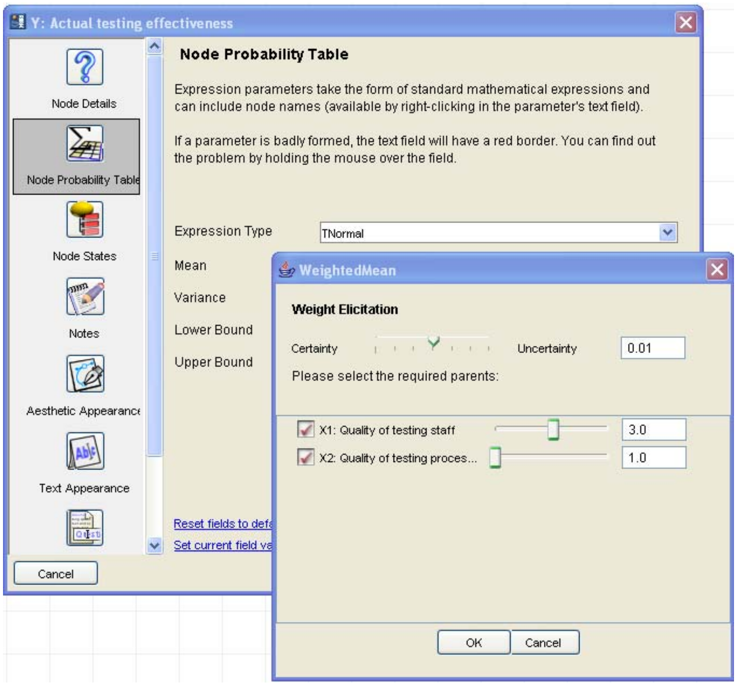

Figure 3 Wizard for defining the weighted mean NPT

When we have evidence for the parents, $\underline{X} = \{X_1 = High, X_2 = Low\}$, the prediction of child node Y has a mean value equal to the weighted average and a variance that reflects our confidence in the result. Figure 4 shows that the result is weighted by the importance of the parent nodes. Since X_2 is less important than X_1 the result on child risk node Y is biased towards the X_1 value.

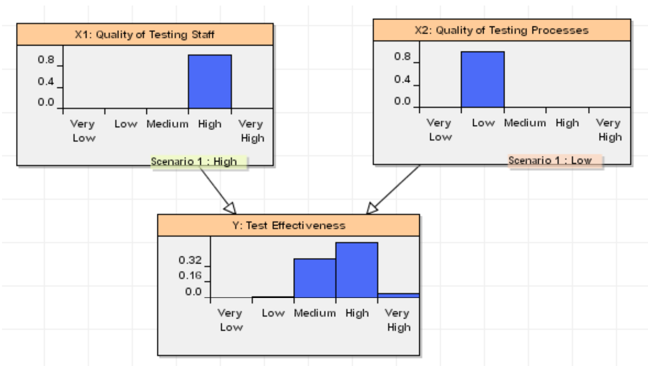

Figure 4 Prediction of $p(Y)$ **given** $\underline{X} = \{X_1 = high, X_2 = medium\}$

The relative importance reflected in the weighting scheme used is also evident when we calculate causes given evidence about effects. Those nodes with higher weights will be identified as the most likely causes of the consequence. This is shown in Figure 5 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.

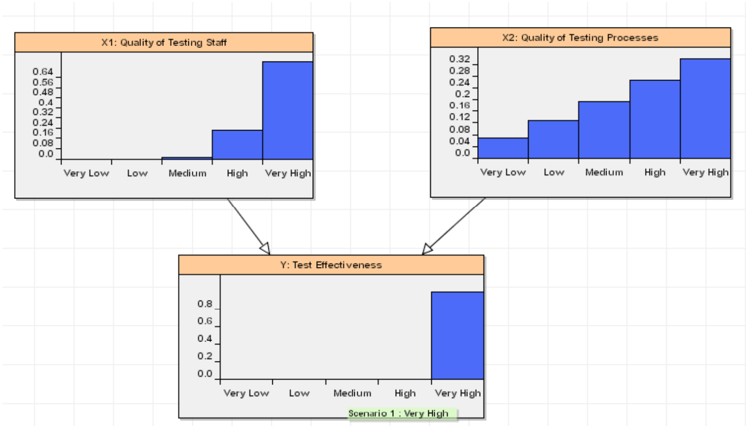

Figure 5 Prediction of $p(\underline{X})$ **given** $Y = \text{Very High}$

Note, however, that if we now observe that X_1 is in fact only medium, the very high value of Y is explained away by out belief in a very high value of X_2 (shown in Figure 6). It is this kind of expected back-propagation that could not be achieved with other functional approaches.

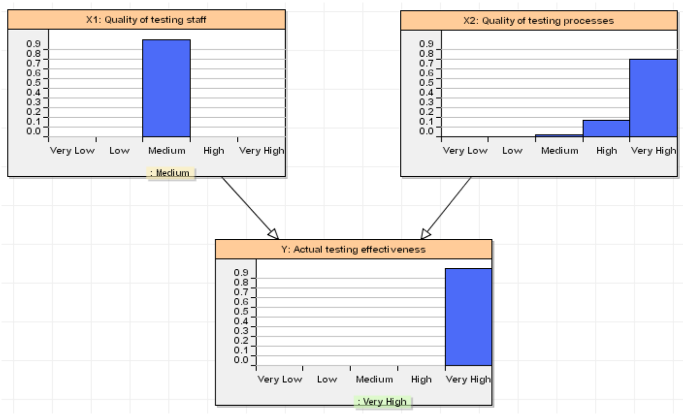

Figure 6 Explaining away

Alternative weighted functions for ranked nodes

The weighted average is not the only natural function that could be used as the mean of the TNormal ranked node NPTs. Suppose, for example, that in Figure 1, we replace the node "Quality of testing process" with the node "Testing effort" as shown in Figure 7.

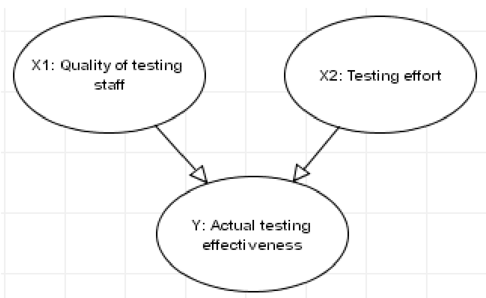

Figure 7 Revised BN fragment

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 v that quantifies our uncertainty about the result., and thus

$$p(Y \mid \underline{X}) = TNormal[WMIN(X_i), v]$$

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=8 and w_2=2 (with a variance v=0.01) yields the results shown in

Figure 8.

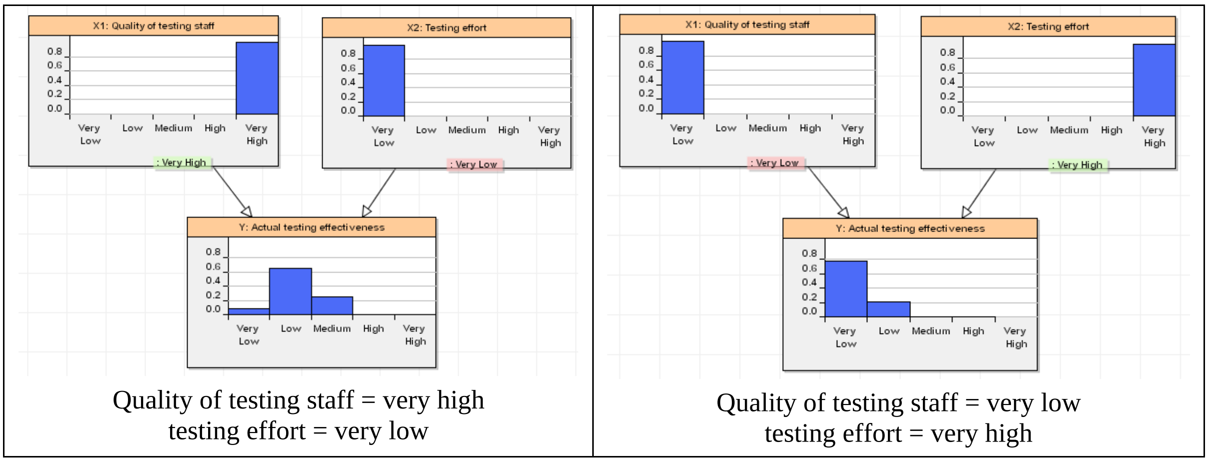

Figure 8 WMIN function for Y

Again it is important to stress that constructing the necessary NPT requires experts only to go through two simple steps. Firstly, select the expression type WeightedMin from the Mean combo box as shown in Figure 10:

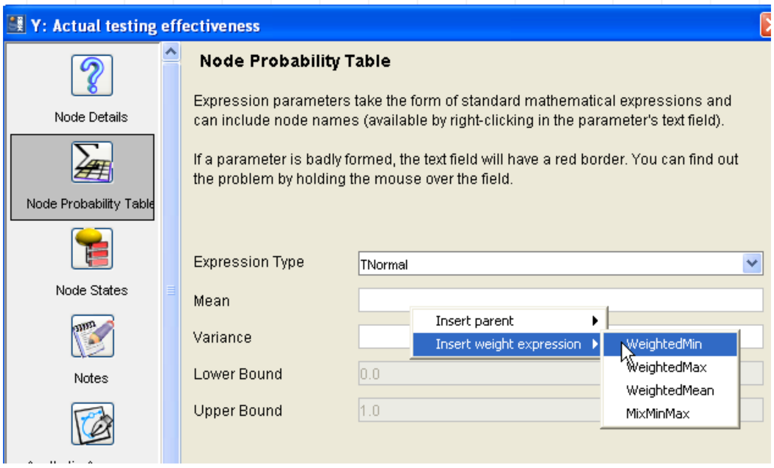

Figure 9 Choosing a weighted expression

Secondly, assign the weights of each parent and the variance (either manually or by dragging the slider bars) as shown in Figure 12:

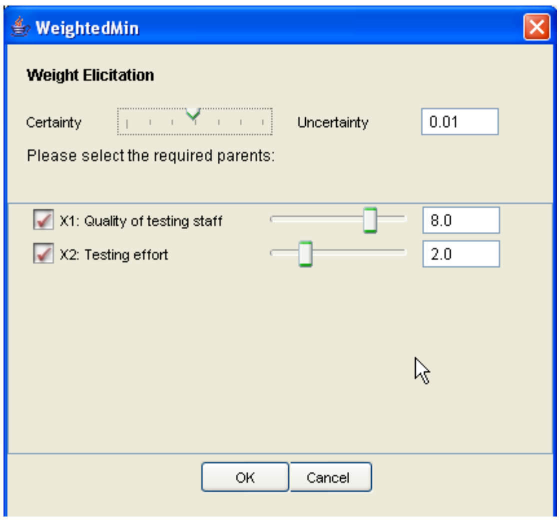

Figure 10 Specifying parent weights and variance

This results in Y's NPT being defined as a TNormal with mean wmin(8.0, X1, 2.0, X2) and variance 0.01.

We also need 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 two.

$$\text{MIXMINMAX} = \frac{w_{\min} MIN(X,Y,Z,...) + w_{\max} MAX(X,Y,Z,...)}{w_{\min} + w_{\max}}$$

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.

Note 1: In the above examples the variance was a constant, but in many situations the variance will be dependent on the parents. For example, when both parents are vary different values it might be reasonable to expect the child variance to increase. In that case it might be sensible to define the variance as a multiple of something like ABS(X1-X2).

Note 2: Because the TNormal for ranked nodes is always in the range [0,1] any variance above, say, 0.5 would be considered very high (you should try it out on a simple weighted sum example). You may need to experiment with the variance to get it just right.