A Method for Automatically Eliciting node Weights in a Hierarchical Knowledge-Based Structure for Reasoning with Uncertainty

S. E. Hegazy, C. D. Buckingham
School of Engineering and Applied Science, Aston University, Birmingham, UK.
hegazys@aston.ac.uk c.d.buckingham@aston.ac.uk

***Abstract* - Hierarchical knowledge structures are frequently used within clinical decision support systems as part of the model for generating intelligent advice. The nodes in the hierarchy inevitably have varying influence on the decision-making processes, which needs to be reflected by parameters. If the model has been elicited from human experts, it is not feasible to ask them to estimate the parameters because there will be so many in even moderately-sized structures. This paper describes how the parameters could be obtained from data instead, using only a small number of cases.**

The original method [1] is applied to a particular web-based clinical decision support system called GRiST, which uses its hierarchical knowledge to quantify the risks associated with mental-health problems. The knowledge was elicited from multidisciplinary mental-health practitioners but the tree has several thousand nodes, all requiring an estimation of their relative influence on the assessment process. The method described in the paper shows how they can be obtained from about 200 cases instead. It greatly reduces the experts' elicitation tasks and has the potential for being generalised to similar knowledge-engineering domains where relative weightings of node siblings are part of the parameter space.

Keywords: *Clinical Decision Support Systems; Mental Health; Risk Screening; Hierarchical Knowledge; Decision Trees; Mathematical Modelling***.**

I. INTRODUCTION

Clinical decision support systems (CDSSs) often work in complex domains that require modelling of human expert knowledge [2,3]. The resulting models may possess high numbers of parameters that need to be instantiated, which is extremely time-consuming for the domain experts and may not even be realistically achievable. An important element of human expertise is its hierarchical structuring [2], which leads to equivalent knowledge structures within CDSSs. These structures or trees have many nodes and the influence of each child node on its parent node will vary across the siblings when it comes to processing uncertainty through the tree. Each node will therefore require a parameter to represent its particular influence on the decision making process, which adds up to a very large number of values to be given by the domain experts on whom the CDSS is being modelled. This paper describes a method for inducing the parameters from a small number of cases instead and shows how it has been applied to a particular CDSS in the domain of mental health risk assessment. The method has the potential for being generalised to any tree where siblings of single parent nodes need individual weights to fit the data. The paper will begin by introducing the domain and the specific CDSS.

A. Risk assessment in mental health

Risk screening in the mental health field is a particularly complex procedure but lacks much assistance beyond paper-based tools [4]. At present, actuarial approaches to risk prediction gain favour because of their evidence base, but have a predictive value that remains unsatisfactory. They also tend to rely on isolated factors, not combinations [5], and ignore the individual qualitative and idiosyncratic patient data that support clinical judgements in practice [6]. There is a need for tools based on clinical expertise as well as empirical evidence and this was precisely the motivation for developing the Galatean Risk Screening Tool, GRiST [7, 8]. It is a web-based CDSS that is designed to assist the early detection of multiple risks, including suicide, self-harm, harm to others, self-neglect, and vulnerability amongst people with mental health problems. It is the only risk-assessment tool that uses a computational model of psychological processes to represent structured clinical judgements of multidisciplinary mental-health practitioners [9, 10].

GRiST has successfully elicited the hierarchical knowledge used by expert mental-health practitioners [11] but it generated a tree with over one thousand nodes, each of which has a parameter representing its relative influence on the assessment process. Asking the domain experts to set these parameters was not feasible and an alternative approach was investigated instead.

In essence, GRiST is a weighted decision tree where risk is represented by fuzzy-set membership grades (MGs) [12] that are associated with each node of the tree. Figure 1 shows a small portion of the GRiST tree for suicide risk. The bottom level boxes are the data for a patient assessment (case). These generate a MG at the matching leaf node using a function that depends on some parameters given by the experts for each leaf node (see [9] for more details). The MGs then propagate up the risk hierarchy and eventually to the top level risks, where the MG associated with a risk represents the simulated clinical risk judgement. The relative influence (RI) of each node in the hierarchy is a parameter that decides how much risk is propagated up the tree by a node compared to its siblings [9]. This parameter also needs to be set so that it reflects the expertise of mental-health practitioners. Getting them to do it themselves as part of the knowledge elicitation process is an arduous task when the tree has so many nodes. This makes it unlikely that a large enough set of participants can be obtained to ensure the consensus for each RI is reliable, as opposed to eliciting the leaf node parameters, which are far fewer: 192 for GRiST.

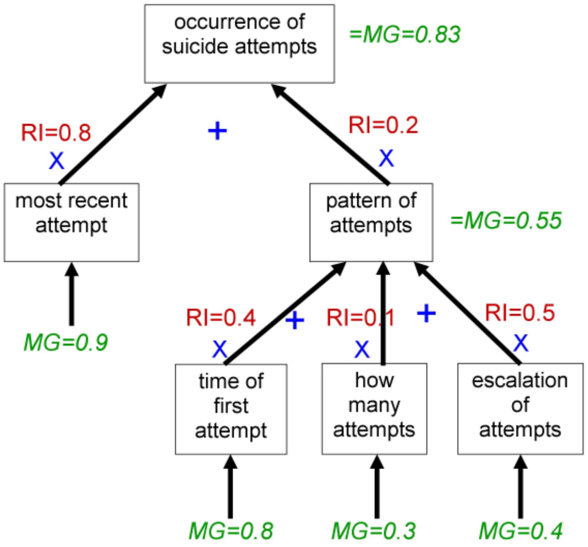

Figure 1: *A portion of the GRiST for suicide risk showing how the relative influences of the nodes moderate the flow of risk. Each node MG is multiplied by its associated RI and summed with the siblings to give the parent MG. Note that the actual values are hypothetical.*

In this paper, we devise an algorithm that induces the RIs from the clinical judgements given by expert mental-health practitioners for patient cases. This will mean the RIs are modelled on the clinicians' own risk judgements because the RIs are set to the exact values required for simulating those judgements. It depends on knowing the MGs at the leaf nodes for a patient's data along with the associated clinical risk judgements, where the risk judgements equate to the MG that GRiST needs to generate at the root node (risk) for that patient. The number of cases required to solve the RIs must be the same as the number of cues in the patient's data set. For GRiST, these judgements are given by clinicians as part of their everyday use of GRiST in practice. Hence the elicitation process has been reduced to providing only the parameters for the 192 leaf nodes. It is important, if not mandatory, for having such an automated system to elicit RIs because the sheer number is likely to mean experts don't do it accurately themselves.

This paper will give some background to the basic problem, after which the method and algorithm will be described. It will conclude with a discussion about how the approach could have generic applicability and be extended.

II. BACKGROUND

The problem we are trying to solve could be represented in a more generalized form, which is a decision tree with weighted inputs. Each input at the leaves contributes to the final decision at the top of the tree, through a weight that determines how much influence the node has compared to its siblings. Every node has these weights applied to its child nodes and, for GRiST, there is an additional constraint that the weights across all the sibling nodes must sum to unity. The task is to find a way of automatically deducing the weights throughout the tree from a minimal set of inputs and outputs.

Most algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible trees. These algorithms generally construct a decision tree, *T*, from a set of training cases [13]. J. Ross Quinlan developed the first algorithm, ID3 [14], and based it on the Concept Learning System (CLS) algorithm [15]. Other methods like CART (Classification and Regression Trees) were introduced for the induction of a tree [16].

Variations on the above methods usually deal with the type of the input variables, the data pool or set properties, or the output type (i.e. continuous or discrete data) [17-19]. Most of these methods attempt to construct the tree without prior knowledge of the desired tree structure. This means, they try to predict the layout of the tree and number of nodes based on the training cases. The trees are then pruned and optimized to the minimum structure that satisfies the classes in the training instances.

Our problem is very different. We aim to model the GRiST decision tree parameters mathematically, since the structure of the tree is known in advance from the psychological model that has been induced from the experts [10, 11]. Hence, we are in control of the structure, don't require pruning and optimization processes, and can use the training sets purely to induce the unknown weights in our model.

III. METHODOLOGY

In this section, we introduce the general structure of the decision tree used by GRiST, which is the same structure that our model will use to calculate the RI values. It is shown in *Figure 2* as follows:

L_{Rn} : denotes the RIs in level n.
M_n : denotes the MGs (Membership Grades) in level n.
M_{xy} : denotes the MG of node y in level x; y=0 to Zjh , where Zjh is the number of children of node number h-1 at level j.
R_{ti} : denotes the RI of node number i at level t on the total MG at level t-1 which equals $M_{(t-1)y}$ where y is the number of the parent node of R_{ti} .

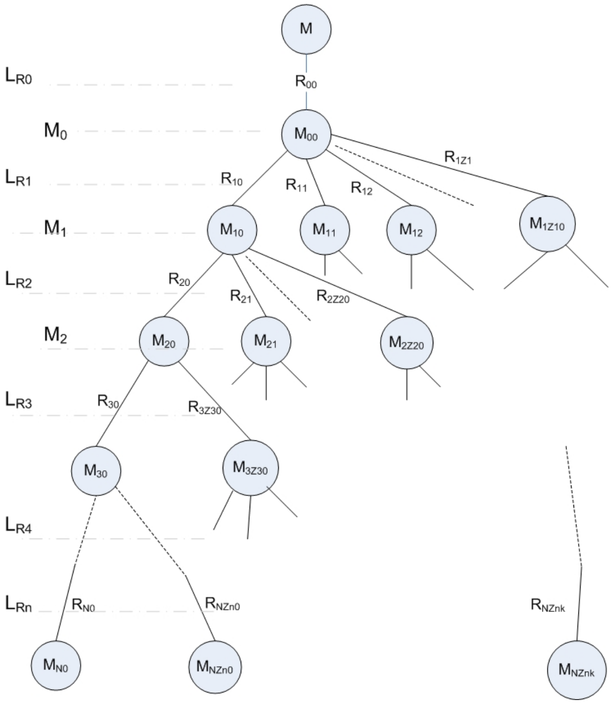

Figure 2: *The General GRiST DSS Tree.*

To find M, the total membership grade of the tree (which represents the overall diagnosis or risk of the patient's mental health [9]), there are several methodologies we could follow. One would be to train the model using known cases and, assuming that leaf MG values and M are given, we could use a neural networks simulation. The problem with neural networks is, though, that we won't be able to represent the internal hierarchical structure of the GRiST tree as given by the experts, which is crucial to the explanation of how risks are generated. We have thus developed a method that maintains the tree structure within a mathematical representation and uses training sets to induce the values of RIs. To model the tree mathematically, we follow the psychological model underlying GRiST [9], which defines how to calculate the overall result, M (the details of the model are not relevant for this paper because our algorithm applies to the RIs only, not the generation of MGs at the leaf nodes). The MG at each node is the summed product of each child node's MG and RI, which then feeds through to the next parent node in the same way, as follows:

$$M = R_{00}\, M_{00}$$

$$R_{00} = 1\text{, thus, } M = M_{00}$$

$$M = R_{00}\ (R_{10}\, M_{10} + R_{11}\, M_{11} + R_{12}\, M_{12} + \ \ldots\ldots\ldots\ + R_{1Z10}\, M_{1Z10}\ )$$

If we expand the calculations for the MGs in the child nodes, we get

$$M = R_{00}\ (R_{10}\ (\ R_{20}\, M_{20} + R_{21}\, M_{21} + R_{22}\, M_{22} + \ldots\ldots\ldots + R_{2Z10}\, M_{2Z20}\ ) + \ldots\ldots \\ R_{1Z10}\ (\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..\ )\ )$$

If we continue this process, until we reach the leaves, the resulting expression will be the sum of the products of all RIs along the path to a leaf node and that leaf node's MG, which creates a certain pattern for the multiplication expression that we will clarify and make use of later.

To illustrate the above, we use a simpler example of a tree with just two levels, as shown in Figure 3, where a to g are used to represent the specific leaf node MGs of M_{20} to M_{27} for clarity.

$$\begin{aligned} M &= R_{00}\ (R_{10}\, M_{10} + R_{11}\, M_{11} + R_{12}\, M_{12}\ ) \\ &= R_{00}\ (R_{10}\ (\ R_{20}\, a + R_{21}\, b\ ) + \ R_{11}\ (\ R_{22}\, c + R_{23}\, d\ ) + R_{12}\ (\ R_{24}\, e + R_{25}\, f + R_{26}\, g) \end{aligned} \tag{1a}$$

Or:

$$M = R_{00} R_{10} R_{20}\, a + R_{00} R_{10} R_{21}\, b + R_{00} R_{11} R_{22}\, c + R_{00} R_{11} R_{23}\, d + R_{00} R_{12} R_{24}\, e + R_{00} R_{12} R_{25}\, f + R_{00} R_{12} R_{26}\, g \tag{1b}$$

Since, a to g are given, the unknowns are all the Rs. The top-level M is also given, because it represents the clinical judgement associated with the case (for different cases, we will use M1, M2, M3, …). We have several of the above equations, one per case, and can regard them as a system of linear simultaneous equations. To solve the R values, we need the same number of cases as there are leaf nodes.

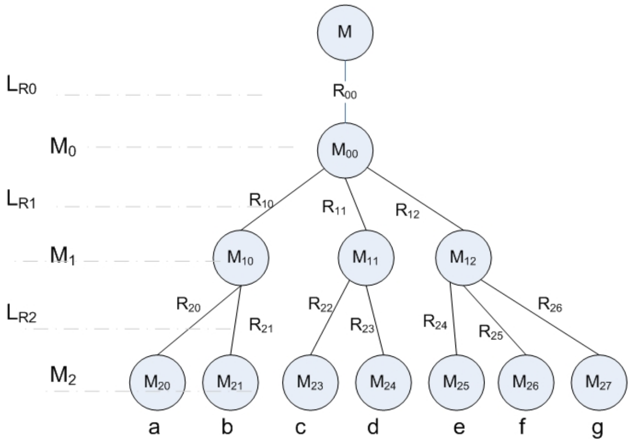

Figure 3: *A simple two level GRiST tree.*

To simplify, we rename RI products along a path as:

$A = R_{00}\,R_{10}\,R_{20}$
$B = R_{00}\,R_{10}\,R_{21}$
$C = R_{00}\,R_{11}\,R_{22}$
$D = R_{00}\,R_{11}\,R_{23}$
$E = R_{00}\,R_{12}\,R_{24}$
$F = R_{00}\,R_{12}\,R_{25}$
$G = R_{00}\,R_{12}\,R_{26}$ (2)

to give seven equations for our example with $R_{00} = 1$ (from the RI properties).

The system can be set up as a set of linear simultaneous equations, as follows:

M1 = a1. A + b1. B + c1. C + d1. D + e1. E + f1. F + g1 .G

M2 = a2 .A + b2. B + c2 .C + d2 . D + e2. E + f2 .F + g2. G

.... and so on ....

M7 = a7. A + b7. B + c7. C + d7. D + e7. E + f7. F + g7 .G

(3)

Solving (3) is straightforward (using matrices), which gives us A to G. But originally, we had eleven unknowns, so to determine RIs, we need an extra four equations in addition to the above seven. For this we use the inherent property of RIs that they must sum to one across all siblings:

$$\sum_{y=0}^{Zxn} R_{xy} = 1 \qquad (4)$$

In our case this gives us:

$R_{10} + R_{11} + R_{12} = 1$ (4b)
$R_{20} + R_{21} = 1$
$R_{22} + R_{23} = 1$
$R_{24} + R_{25} + R_{26} = 1$ (5)

So we have eleven equations and eleven unknowns.
By substitution, we can solve the system exploiting another pattern:

$A / B = (R_{10} . R_{20}) / (R_{10} . R_{21})$

$= R_{20} / R_{21}$

So: $R_{21} = (B / A)\, R_{20}$ (5a)

Substituting in the relevant equation, we get:

$R_{20} + R_{21} = R_{20} + (B / A)\, R_{20} = 1$

Or: $R_{20}\,(1 + (B/A)) = 1$

Or: : $R_{20}\,((A+B) / A) = 1$

Thus: $R_{20} = A / (A+B)$

By continuing in the same manner, we can obtain the rest of the RIs.

$$R_{20} = \frac{A}{(A+B)}$$
$$R_{21} = \frac{B}{(A+B)}$$
$$R_{22} = \frac{C}{(C+D)}$$
$$R_{23} = \frac{D}{(C+D)}$$
$$R_{24} = \frac{E}{(E+F+G)}$$
$$R_{25} = \frac{F}{(E+F+G)}$$
$$R_{26} = \frac{G}{(E+F+G)}$$

(5b)

In other words, each leaf RI can be found as a function of the RI products along the path from each sibling leaf to the root node. These products, A to G, have been solved from the simultaneous equations, so each individual leaf RI can thus be calculated.

IV. THE COMPLETE ALGORITHM

The input to the algorithm would be n vectors of known and diagnosed cases given by experts. In the example for Figure 3, that vector will contain the following:

$$V = (M, a, b, c, d, e, f, g) \qquad (6)$$

where M is the top-level clinical judgement given by the clinician for the patient MGs of a,b, … g (i.e. the leaf-node MGs generated directly from the patient values).

The algorithm we propose can be divided into two steps: solving for the multipliers of each leaf MG, which are the products of the RIs along the path from the leaf to the root (i.e. A to G), and then solving for the individual RIs themselves.

Step 1: Solving for Multipliers

The first step will be solving n simultaneous linear equations, where n is the total number of leaves of the GRiST tree (seven, a to g, in Figure 3):

$$M1 = a1.A + b1.B + c1.C + d1.D + e1.E + f1.F + g1.G$$
$$M2 = a2.A + b2.B + c2.C + d2.D + e2.E + f2.F + g2.G$$
$$\dots\dots\dots\dots\dots\dots\dots\dots$$
$$\dots\dots\dots\dots$$
$$M7 = a7.A + b7.B + c7.C + d7.D + e7.E + f7.F + g7.G \qquad (7)$$

Or in matrix form:

$$\begin{pmatrix} M1 \\ M2 \\ M3 \\ M4 \\ M5 \\ M6 \\ M7 \end{pmatrix} = \begin{pmatrix} a1 & b1 & c1 & d1 & e1 & f1 & g1 \\ a2 & b2 & c2 & d2 & e2 & f2 & g2 \\ a3 & b3 & c3 & d3 & e3 & f3 & g3 \\ a4 & b4 & c4 & d4 & e4 & f4 & g4 \\ a5 & b5 & c5 & d5 & e5 & f5 & g5 \\ a6 & b6 & c6 & d6 & e6 & f6 & g6 \\ a7 & b7 & c7 & d7 & e7 & f7 & g7 \end{pmatrix} \times \begin{pmatrix} A \\ B \\ C \\ D \\ E \\ F \\ G \end{pmatrix} \qquad (8)$$

Equation 8 can be solved using Gaussian Elimination so we now know the values of A to G, which is given by the solution, S:

$$S = (A, B, C, D, E, F, G) \qquad (9)$$

Step 2: Solving for individual RIs

To find each RI, we look at a general leaf node and its children (see *Figure 4)*.

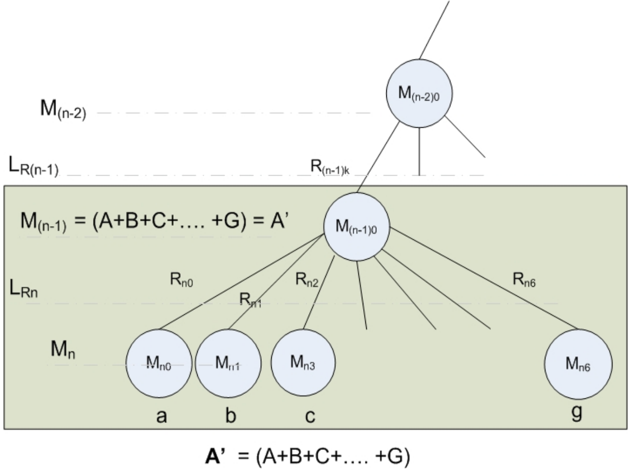

Figure 4: *A general leaf node with seven children.*

The challenge is to devise a systematic way for deriving the solution. Let us take a slice of matrix S, and call it S' for simplicity; it only contains entries for leaf nodes that are siblings and that therefore share the same ancestral path of RIs, which is $R_{10} \dots R_{(n-1)0}$ in our example.

$$S' = \begin{pmatrix} R_{10} \dots\dots R_{(n-1)0}\ R_{n0} \\ R_{10} \dots\dots R_{(n-1)0}\ R_{n1} \\ R_{10} \dots\dots R_{(n-1)0}\ R_{n2} \\ R_{10} \dots\dots R_{(n-1)0}\ R_{n3} \\ R_{10} \dots\dots R_{(n-1)0}\ R_{n4} \\ R_{10} \dots\dots R_{(n-1)0}\ R_{n5} \\ R_{10} \dots\dots R_{(n-1)0}\ R_{n6} \end{pmatrix} \qquad (10)$$

From the GRiST model [9], we know that:

$$R_{n0} + R_{n1} + R_{n2} + R_{n3} + R_{n4} + R_{n5} + R_{n6} = 1 \qquad (11)$$

We will convert Equation 10 into a function of only one variable, e.g. R_{n0}. To do this we use S', where each of the rows are represented by a symbol, A to G, for the RI product along the path.

$B/A = R_{n1} / R_{n0}$

$$R_{n1} = (B/A) . R_{n0}$$

$C/A = R_{n2} / R_{n0}$

$$R_{n2} = (C/A) . R_{n0}$$

…. and so on ….

$G/A = R_{n6} / R_{n0}$

$R_{n6} = (G/A) . R_{n0}$

Substituting in Equation 11:

$R_{n0} + (B/A) . R_{n0} + (C/A) . R_{n0} + (D/A) . R_{n0} + (E/A) . R_{n0} + (F/A) . R_{n0} + (G/A) . R_{n0} = 1$

Factoring out R_{n0}:

We get

$$R_{n0} = \left(\frac{A}{A+B+C+D+E+F+G} \right)$$

Solving in the same way, we obtain:

$$R_{n1} = \left(\frac{B}{A+B+C+D+E+F+G} \right)$$

… and so on to …

$$R_{n6} = \left(\frac{G}{A+B+C+D+E+F+G} \right) \qquad (12a)$$

Hence the general rule in the algorithm, to find a certain RI in the leaf nodes is:

$$RI_j = \left(\frac{S'(j)}{\sum_{j=1}^{k} S'(j)} \right) \qquad (12b)$$

Where j is the leaf node MG (in our example, a,b,c, …), k is the total number of siblings, and S' is the product of all RIs along the path from the specified leaf node to the root node..

Step 3: Shrinking the tree

Having found the RIs of the leaf node (see Figure 4), we can now calculate the MG for the parent node, $M_{(n-1)0}$, which can then become a leaf itself. We can do this for all the parent nodes that have leaf nodes as children and, by converting them into leaves themselves once their MG has been calculated, the tree is shrunk.

Summary of the generalised algorithm

So far, the explanation has used specific trees to illustrate it. We can now generalize the algorithm as follows.

Inputs:

$V1 = (M1, M_{n01}, M_{n11}, \ldots\ldots\ldots\ldots, M_{nk1})$

To:

$$Vk = (Mk, M_{n0k}, M_{n1k}, \ldots\ldots\ldots\ldots, M_{nkk}) \qquad (16)$$

Where:
M1 to Mk : are the k different cases outcomes.
M_{n0y} : is the input MG at the leaf on the nth level (lowest level) of the GRiST tree of the y^{th} input vector (Vy).
We need k vectors to solve the resulting k simultaneous equations where k = the number of leaf nodes of the GRiST tree = the number of cases required.

Outputs:

RI values, representing the node weightings for every node in the tree.

Procedure:

Step one:

Solve the following simultaneous equations:

$$\begin{pmatrix} M1 \\ M2 \\ M3 \\ .... \\ ... \\ .... \\ Mk \end{pmatrix} = \begin{pmatrix} M_{n01} & M_{n11} & M_{n21} & ... & ... & .. & M_{nk1} \\ M_{n02} & M_{n12} & M_{n22} & ... & .... & ... & M_{nk2} \\ M_{n03} & M_{n13} & M_{n23} & .... & .... & ... & M_{nk3} \\ ... & .... & .... & .... & ... & ... & ... \\ ... & .... & ... & ... & ... & ... & ... \\ .... & ... & ... & ... & ... & ... & .... \\ M_{n0k} & M_{n1k} & M_{n2k} & ... & ... & ... & M_{nkk} \end{pmatrix} \times \begin{pmatrix} A1 \\ A2 \\ A3 \\ ... \\ ... \\ ... \\ Ak \end{pmatrix} \qquad (17)$$

The above matrix is kXk in dimension.
The solution yields vector A1 to Ak.

Step two:

We use S' to denote a sub tree of each node at level (n-1), where n is the deepest level of the tree where all nodes are leaf nodes.
Hence we have: S'1 to S'h where h is the number of nodes at level (n-1) in the GRiST tree.
For each subtree, S'j, we solve to find its RIs.

$$RI_{nr} = \left(\frac{S'j(r)}{\sum_{r} S'j(r)} \right) \qquad (18)$$

Where *r* represents the children (and thus leaf nodes) of parent node *j*, with r going from 1 to the number of leaves of node j at level (n - 1); j = 0 to h.

Step three:

Once the RIs have been found at a particular level, the tree can be shrunk by a level by making the parent nodes the new leaf nodes with their MGs calculated by:

$$M_{(n-1)h} = \sum_{r} S'j(MGj) \qquad (19)$$

Once the new shadow MGs are found for the new level, we can go to step two and repeat step two and three for the new tree. This process is continued n times (for an n-level tree). At the end, we will have determined all the RIs in the tree.

Case study:

This part of the paper demonstrates the effectiveness of the algorithms using a case study with arbitrary numbers. We will use our algorithm to calculate the RI values in the tree shown in Figure 5. The tree has six leaves (A to F), hence we need six training cases. The following matrix sets up the synthetic data in the format of Equation 17:

$$\begin{pmatrix} 0.3 \\ 0.4 \\ 0.9 \\ 0.7 \\ 0.8 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.1 & 0.3 & 0.4 & 0.2 & 0.3 & 0.6 \\ 0.1 & 0.2 & 0.5 & 0.4 & 0.6 & 0.2 \\ 0.2 & 0.1 & 0.3 & 0.7 & 0.8 & 0.7 \\ 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.3 \\ 0.2 & 0.3 & 0.4 & 0.3 & 0.9 & 0.2 \\ 0.3 & 0.1 & 0.6 & 0.5 & 0.5 & 0.4 \end{pmatrix} \times \begin{pmatrix} A \\ B \\ C \\ D \\ E \\ F \end{pmatrix} \qquad (20)$$

Using Gaussian Elimination, to solve the above matrix for the unknowns, we obtain:

A = -0.44	D = 0.44
B = 0.92	E = 0.964
C = -1.067	F = 0.196

Note that we use 3 decimal points approximation for simplicity (rounding).

.

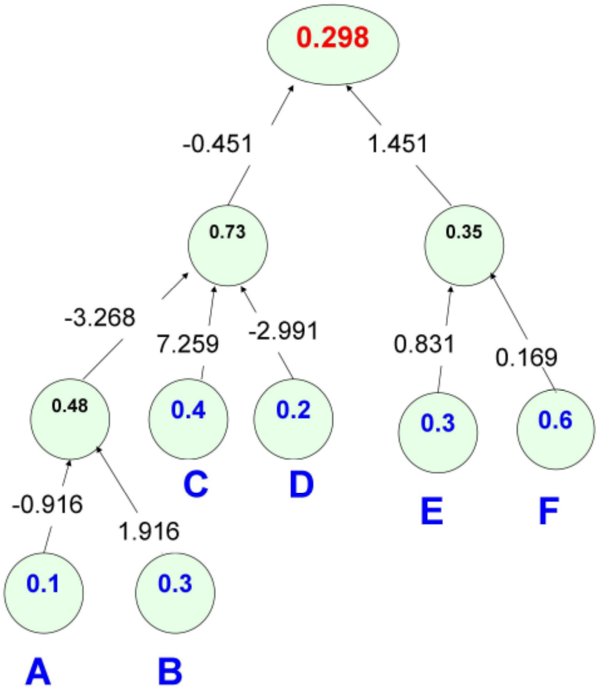

Figure 5: *A sample decision sub-tree.*

Using Equation 18 and the propagation technique in Equation 19, we obtain all the RI values as in Figure 5.

To verify the model, we use the first training case (first line in Equation 20) as an input (on Figure 5, it is the number printed inside each leaf node, A to F). Propagating through the decision tree using the new RI values, we finally reach a decision (M = 0.298, inside the top node). This is almost the same as the desired output in the original test case, in Equation 20 (i.e. 0.3). The error is due to approximation and using only three decimal points precision.

The case study shows that solutions may require negative RI values, which is only a problem if the semantics of the knowledge domain demand positive values. For the GRiST domain, and probably many other knowledge-based systems, the concept of negative RIs is not psychological meaningful, although semantically it can be explained in terms of a bigger span between the RIs of the siblings and those could then be mapped to normalized values. It is possible that real-world data, where clinicians have provided risk assessments for a given set of patient values, will have inherent constraints that mean the RIs willl not be negative. However, it remains a possibility that limiting RIs to positive values would mean a solution cannot be found. In the next section, we will discuss an extension to the method that will circumvent this problem.

V. CONCLUSION AND FUTURE WORK

In this paper, we have addressed the problem of eliciting parameters in the GRiST tree structure [10, 11]. These parameters can then be used to analyze new cases and provide advice for mental health practitioners. The techniques presented here are extending our ARRIVE [1] algorithm, and provide a robust mathematical calculation of the Relative Influence (RI) values in the GRiST tree [9] that that are crucial for enabling risk quantifications to be generated. Similar approaches could be relevant to many intelligent knowledge-based systems based on human expertise where the knowledge is in a hierarchical structure and the nodes have varying influence on the decision making processes. For GRiST, the RIs represent varying weights of sibling nodes on their parents and were normalised so that the total weighting across siblings was unity for all nodes.

At present, the method is intended to initialise the node weightings from a fixed number of cases equal to the number of leaf nodes in the tree, where the risk judgements have been given by expert clinicians for the set of patient data associated with those leaf nodes. It would be better, though, if the weightings could be incrementally updated as new cases are classified and future work will explore techniques for accomplishing this. It means the RIs would be a more representative consensus for the clinicians, having been induced from an ever-increasing data set. The resulting weights would thus be best estimates from the data and would enable constraints on the range of allowable values to be set without jeopardising the generation of solutions. The method described in this paper could create the initial weights that would then be updated as new cases arrive.

Other aspects of future work include analyzing the sensitivity of the algorithms to variations in patient data as well as the impact of missing data and noise in the learning data sets. An interesting problem is how to determine ways of quantifying error margins and confidence in the risk judgements based on the constitution of patient data sets.

REFERENCES

[1] S. E. Hegazy, C. D. Buckingham (2008). ARRIVE: An Algorithm for Robust Relative Values Elicitation, *Proceedings of the International Conference on Computing in the Global Information Technology*, July 2008, 91-96.

[2] Buchanan, B.G., Davis, R., & Feigenbaum, E.A. (2006). Expert Systems: A perspective from computer science. In: K.A. Ericsson, N. Charness, P. Feltovich and R. Hoffman, Editors, *The Cambridge handbook of expertise and expert performance*, Cambridge University Press, New York (2006), pp. 87-103

[3] Berner, E.S. (2006). *Clinical Decision Support Systems: Theory and Practice* (Ed.), 2nd Ed, Springer: New York.

[4] Hawley, C. J., Littlechild, B., Sivakumaran, T., Sender, H., Gale, T. M. & Wilson, K. J. (2006). Structure and content of risk assessment proformas in mental healthcare. *Journal of Mental Health, 15*, 437-448.

[5] Skegg, K. (2005). Self-harm. Lancet, 366, 1471-1483.

[6] Holdsworth, N., & Dodgson, G. (2003). Could a new Mental Health Act distort clinical judgement? a Bayesian justification of naturalistic reasoning about risk. *Journal of Mental Health*, 12(5), 451-462.

[7] Buckingham, C.D. (2007). Improving mental health risk assessment using web-based decision support. *Health Care Risk Report*, 13(3), 17-18.

[8] GRiST [www.galassify.org/grist] GRiST [updated December 2008; cited 2009 Jan 5th]. Available from www.grist.galassify.org/grist

[9] Buckingham, C.D. (2002). Psychological cue use and implications for a clinical decision support system. *Medical Informatics and the Internet in Medicine, **27**(4)*, 237-251

[10] Buckingham, C. D., Adams, A.E. & Mace, C. (2008). Cues and knowledge structures used by mental-health professionals when making risk assessments. *Journal of Mental Health*, 17(3), 299-314.

[11] Buckingham, C.D., Ahmed, A., & Adams, A.E. (2007). Using XML and XSLT for flexible elicitation of mental-health risk knowledge. *Medical Informatics and the Internet in Medicine*, 32(1), 65-81.

[12] Zadeh LA. Fuzzy sets. Information Control 1965;8:338-53.

[13] Ivan Bratko, Dorian Šuc Learning qualitative models Published in *AI Magazine*, 2003, vol. 24, no. 4, pp. 107-119, © AAAI Press

[14] J. R. Quinlan (1975). *Machine Learning*, vol. 1.

[15] J.R. Quinlan (1986). Induction of Decision Trees, Machine Learning, (1), 81-106

[16] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees.* Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software

[17] Quinlan. (1992). *C4.5: Programs for Machine Learning*, Morgan Kaufmann

[18] C. Z. Janikow, “Exemplar learning in fuzzy decision trees,” *Proc. FUZZIEEE*, pp. 1500–1505, 1996

[19] Harris Drucker and Corinna Cortes. Boosting decision trees. In *Advances in Neural Information Processing Systems 8*, pages 479–485, 1996.