Explanation By General Rules Extracted From Trained Multi-Layer Perceptrons

Zhe Ma and Robert F Harrison, The University of Sheffield, Department of Automatic Control and Systems Engineering, Sheffield, S1 3JD, UK
Simon S Cross, The University of Sheffield Medical School, Department of Pathology, Sheffield, S10 2RX, UK

9 November 1996
AC&SE Research Report 650
Key Words: Explanation, Rule Extraction, Hybrid System, Multi-layer Perceptron

Abstract

GR2 is a hybrid knowledge-based system where the knowledge acquired in a trained Multi-layer Perceptron is translated into a symbolic and abstract form, called general rules. This is based on both white-box and black-box criteria. The extracted rules can be used for inference on a case-by-case basis, explaining how a decision is made. The extracted rules possess both qualitative and quantitative properties of the domain knowledge, thus enhancing the reasoning capability of the system.

The methodology for extracting rules from a trained MLP via two heuristics – the Potential Default Set and the Feature Salient Degree – is outlined, and the use of the resulting domain rules in case-by-case explanation is described. A number of examples from synthetic domains is considered and the problem of diagnosing malignancy in breast lesions from observed cytopathological features is presented. Here the case explanations are commented upon by a senior pathologist and favourable agreement is found.

1. Introduction

There are three essential functions for an ideal knowledge-based system, a convenient support for knowledge acquisition, an accurate and reliable reasoning facility for decision making and a user friendly interface to explain what the system has acquired, how it has come to a conclusion and which therefore enables the user to interact with the system. Most successful traditional AI systems, such as expert systems, fulfil functions 2 and 3, whilst most artificial neural networks possess functions 1 and 2. Can a hybrid knowledge-based system capable of symbolic reasoning and neural computation archive all of the three functions above? This paper describes a hybrid AI system GR2 which supports this conjecture. GR2 consists of a Multi-layer Perceptron (MLP) neural network, a rule-based inference system and an interface that translates the knowledge encoded in the trained MLP into general (production) rules. It also addresses how the explanation by extracted rules helps the comprehension of the behaviour of the system.

Neural networks are capable of learning domain knowledge by example (inductively). They, especially the MLP, can classify or reason on a set of training instances more accurately than other machine learning approaches such as decision trees [10,17,23]. Neural networks are widely used and reliable systems in data rich, noisy application environments. It is natural to attempt to include neural networks within knowledge-based system. This nevertheless causes a new problem for IKBS "alleged to be capable of providing the explanations that users need to be able to make their own judgements about the program's recommendations" [6]. Most feedforward neural networks are used as black-box devices. Neither the knowledge encoded in the trained network nor the reasoning (recall) process is easily understood.

Explicit explanation is an important capability for a knowledge-based system. First, rational users, especially experts in the application domain do not simply accept a conclusion given by the system. They instead expect more information about what knowledge the system has acquired, and about how a conclusion is reached by the system. They may require further information to justify or confirm what action they should take.

Second, the explanation function is particularly important in safety critical systems such as air traffic control, or in knowledge-rich disciplines such as medical diagnosis, where human verification is necessary. Without a comprehensible form of knowledge representation, it is impossible for the expert to inspect the knowledge possessed or acquired by the system. Therefore there is no way to answer such questions as "what if" or "how" in an easily understood way.

Third, the assistance of a computerised knowledge-based system is especially useful when human experts are unsure of what should be concluded, based on their knowledge in certain instances, or when the system gives solutions that are different from the expert's expectation. Then the system must be able to explain why it concludes so, and to what extent it can support its conclusion.

In GR2, a conventional MLP is first invoked via the training process to learn the knowledge embedded in the training set into the network in the form of a set of weights. The weights, together with the training set, are used by a rule extraction component to generate a set of general rules. A general rule is a production rule extended with some quantitative descriptors. The general rules with these quantitative descriptors represent both qualitative and quantitative knowledge, and thus convey richer information than either the numerical connections in the MLP or the traditional production rules. This not only makes reasoning more reliable and explanation more relevant, but also enables users directly to monitor the knowledge-based system. In terms of the answer types defined by Nilbert [6], GR2 explains knowledge at both the domain level and at the case level.

The paper is organised as follows. Section 2 presents existing techniques of rule extraction from neural networks. Section 3 introduces a method of rule extraction from a trained MLP, which employs feature evaluation criteria using both "white-box" and the "black-box" approaches. The extracted rules are used in reasoning and explanation on a case-by-case basis in Section 4. Three examples are discussed in Section 5, two are artificial and the other involves a medical decision task. The final section summarises and discusses the explanation capability of GR2.

2. Review of rule extraction techniques

2.1 Rule extraction strategies

There are three strategies for extracting rules from a neural network, according to how the neural network is observed. The first two have been addressed in [22,27]. Here we specifically explore the third type. The three approaches are outlined below.

(i) The "white" or "open"-box strategy, as presented in [19,21,29]. The internal structure of the neural network, including the weights, the hidden units and the output units, is translated into rules directly. This strategy is also called "decompositional" [22]. Briefly, a hidden or output unit is converted into a rule while the weights to the unit are converted into its premises. The unit activation is taken as the consequence of the rule.

(ii) The "black" or "closed"-box approach, called "pedagogical" [22]. The neural network is treated as a black-box in which only the input/output mapping relationship of the trained neural network is considered; the internal structure is [5,7,20]. The input/output mapping is captured as the activation of the output unit changes in response to changes in the activations of the input units. Since enumeration of all possible changes of the input units is combinatorial, some restrictions must be enforced to reduce computational workload.

(iii) The third strategy is a combination of the previous two [18,27]. Although the final target is to extract rules that map the inputs to the outputs of the network, the internal structure of the network is closely analysed for heuristic clues. Thrun [18] develops the Validity Interval Analysis (VIA) as the white-box criterion. A validity interval is defined as the maximal activation range of a unit in the trained network. The validity interval of a unit is constrained by the values and the validity intervals of other units in the networks, either directly or indirectly connected to it, according to the forward or the backward propagation mechanisms in the MLP. This is used to verify the hypothesis that a set of initial values exists at the input/output units. From the black-box viewpoint, Thrun identifies the input/output relationship by a search in the input space from the most general part (i.e. the whole space) to the most specific parts (i.e. the individual instances) gradually, because the VIA can verify the hypotheses of the inputs that are partially (or even not) initialised. This search style not only significantly reduces the necessary number of tests, but also generates non-redundant rules which are quite general.

GR2 [27] shares some common features with Thrun's method. It generates rules including only input/output (instantiated) variables. The analysis of the input/output relationship includes the cascading effects via the internal connections in the network. The rules are quite general. The method is relatively independent of the network and is easy to realise for most feedforward neural networks. The neural network is not constructed and trained with the intention of easing analysis, thus giving general applicability.

In GR2, rules are extracted from training instances by identifying the most relevant subsets of the input variables dependent on two criteria. The white-box criterion we call the Potential Default Set (PDS) that indicates those input variables which possibly do not directly contribute to the output for a given training instance. The black-box criterion we call the Feature Salient Degree (FSD) indicating the degree of influence each input variable has on the output for a given training instance. This is calculated over the whole training data set which has itself been modified by passing it through the MLP. This stage removes any conflicts from the training data. The PDS is a qualitative feature of the input/output relationship, and the FSD is a quantitative descriptor of the input/output relationship.

The extracted rules are called *general rules* for two reasons.

(i) The rules are generated to be as general as possible, representing the domain knowledge at an abstract level.

(ii) A production rule consists of a premise (antecedent) part and a consequence (consequent) part, possessing only qualitative properties. A general rule in addition contains a real number for every rule element. The number attached to a premise is called its contribution factor, representing a causal relationship between the premise and the consequence of the rule. The number attached to the consequence of the rule is the certainty factor [1].

2.2 The Quality of the extracted rules

An essential issue for explaining what knowledge the neural network has acquired is how to assess the quality of the extracted rules. Andrew *et al* point out four criteria:

(a) accuracy;

(b) fidelity;

(c) consistency;

(d) comprehensibility [22].

Here we add another criterion: generality or abstractness, which is closely related to all the other quality criteria.

Generality is a property that distinguishes *knowledge* from *data* [9]. A knowledge representation having good generality must be able to represent qualitative information. Generalisation is a basic step of learning that enables the generalised knowledge to classify unseen examples correctly. Generality is a quality of properly trained neural networks. So it should be a quality of any rules extracted from such neural networks.

2.3 Difficulties in extracting rules

The central task and the complexity of rule extraction is how efficiently to compile rules to represent the *context relevance* among the input variables (attributes) as well as the input/output correspondence.

The KBANN system by Towell and Shavlik [19] requires that unit activations in the networks are near to either 0 or 1. This is quite restrictive in practice. It is commonly the case that the network may become over-trained instead of improved in terms of generality after as backpropagation proceeds, before most non-input units have activations near to either 0 or 1 [8,13,14]. For maximum generality we wish to avoid this restriction in this work.

In addition, the explanatory capacity of a knowledge representation relies on whether or not it can represent the qualitative properties of the domain knowledge, reflecting its quality of generality. From this viewpoint, the extracted rules should omit certain details of the internal structure of the network. If there are networks that have different structures but represent the same domain knowledge, the rule sets extracted from these networks should be the same or very similar. This can be illustrated by some simple networks for the two bit AND and XOR problems. The networks are conventional Multi-layer Perceptrons with one hidden layer, sigmoidal activation functions and the generalised Delta learning rule for the backpropagation training process [2]. Although it is well known that hidden units are not necessary for the network to solve such a simple domain as the two-bit AND problem, we use them for the purpose of demonstration. As mentioned above, the unnecessary hidden unit should not cause a qualitative difference between these and those derived from the network with the minimal structure.

The two bit AND problem is represented in Table1 where A and B are Boolean input variables and C is a Boolean output variable. Column C is the result of the AND operation on the values of A and B. Setting the error tolerance[1] δ =0.5, the MLP (see Figure 1) correctly classifies all instances as presented in the last column.

A	B	C	MLP output
0	0	0	0.0
0	1	0	0.02
1	0	0	0.02
1	1	1	0.94

Table 1. The two-bit AND problem, its truth value table and the outputs of the MLP in Figure 1.

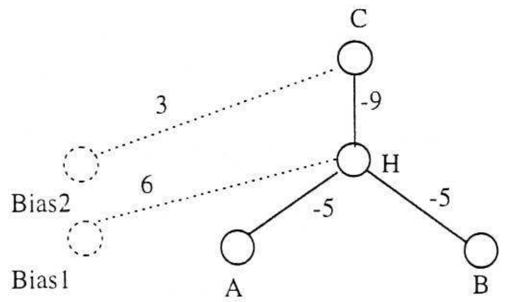

Figure 1. The trained MLP for the two-bit AND problem

The outputs of the MLP can be changed if a part of the network, Bias1 say, is changed. Table 2 records the results as the bias, Bias1, is given different values.

A	B	*C* (Bias1=-1)	*C* (Bias1=-0.7)	*C* (Bias1=4.3)	*C* (Bias1=9.3)	*C* (Bias1=9.9)
0	0	0.64	0.5	0.0	0.0	0.04
0	1	0.95	0.95	0.5	0.0	0.11
1	0	0.95	0.95	0.5	0.0	0.11
1	1	0.95	0.95	0.95	0.5	0.26

Table 2. The outputs of the MLP in Figure 1 as Bias1 is changed

The network only represents the two-bit AND problem when Bias1 is in the range (4.3, 9.3). If Bias1 is in the range (-0.7, 4.3), the network represents the two-bit OR problem. It fails to implement either the AND or the OR problems when Bias1>9.3 or Bias1<-0.7. Ideally, the rules extracted from the network should be uniform corresponding to the domain knowledge for each of these ranges. Furthermore, the rule set should be completely general. For example, from the networks that are the same as the one in

[1] Here "error tolerance" denotes the absolute distance from zero or one within which the MLP output must fall.

Figure 1, except that Bias1 is in the range (4.3, 9.3), the following rule set is the ideal solution:

IF(~A) THEN (~C);

IF(~B) THEN (~C);

IF(A, B) THEN (C);

where the tilde indicates NOT and the comma indicates AND in the rules.

For some white-box rule extraction methods, an intermediate variable H corresponding to the hidden unit is included in the extracted rules. Introducing hidden units as intermediate variables has both a good and a bad effect on the expressive capability; this will be discussed later. The major problem is that the rule set obtained from the white-box methods fail correctly to explain some networks. For example, in setting Bias1 in the range (9.3, 9.9), the rule set may be

IF(~A) THEN (~H);

IF(~B) THEN (~H);

IF(A, B) THEN (H);

IF(~H) THEN (~C);

IF(H) THEN (C);

which is an explanation for the two-bit AND problem. This is incorrect because the network in that situation cannot recognise any two-bit Boolean problem. Again, the rule set for Bias1 in the range (4.3, 4.9) using the white-box method alone may explain the two-bit OR problem, but the network in this case corresponds to the two-bit AND problem instead.

There is a number of reasons for such failures.

(i) The white-box methods usually select the input variables via a linear comparison of the subsets of the weights to a hidden unit with the bias to this unit. This does not match the activation function of the network $F(net) = \frac{1}{1+e^{-net}}$.

(ii) Including the hidden units as atoms in the rules represents them in Boolean form, while the difference between the real values of the hidden units and the Boolean values may give a very different overall representation of the rule set from the network.

(iii) The weights in a layer of the network affect processing in parallel whilst in the symbolic rules this property is only reflected within those rules whose premises correspond to subsets of the weights. Lack of representation of the parallelism among the rules may substantially change the domain knowledge from that represented by the trained MLP.

The representation of qualitative knowledge via white-box methods alone remains a challenge.

The challenge becomes harder still if we require that the rule extraction method generates a set of rules from another network in Figure 2 which is similar to the rule set from the network in Figure 1. The network in Figure 2 is for the two-bit AND problem too, but it contains two hidden units.

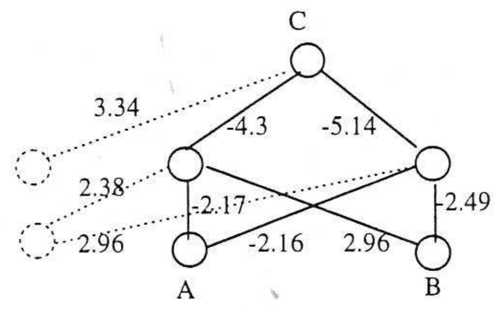

Figure 2. A network for the two-bit AND domain with two hidden units

For the same reasons, the white-box methods find it difficult to handle the following problem. In Figure 3, we display two networks. The bias for a hidden unit is included in the circle for each unit. Figure 3a is the network for the XOR problem. In Figure 3b, every value is correspondingly half of that in Figure 3a. Most white-box methods will extract the same rule sets for the two networks. But the network in Figure 3b can not recognise any Boolean problem, as shown in Table 3, if we still assume the error tolerance δ =0.5.

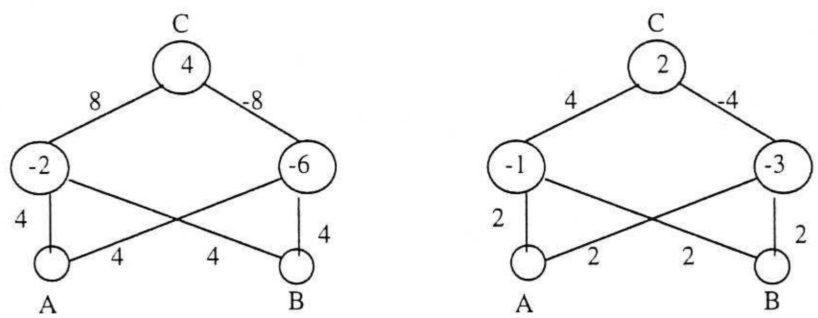

Figure 3. (a) a network for the XOR problem. (b) a network without distinct outputs

A	B	Output by the net in Fig. 3a	Output by the net in Fig. 3b
0	0	0.04	0.25
0	1	0.89	0.46
1	0	0.89	0.46
1	1	0.04	0.25

Table 3. Outputs by the networks in Figure 3

There is value to the white-box approach yet. The white-box methods using intermediate variables are capable of representing the knowledge of the network for some complex domains which may not be able to be directly represented only by the input/output variables.

For most black-box methods, a difficult and important task is to extract the most general rules as possible. This requires a process to find the abstract features of the knowledge domain. Without the abstraction process, understanding of the extracted rules is very difficult, if it is possible at all. For example, an abstract statement for the two-bit AND problem is that "any false value to an input variable is the deciding feature to give a false output value, the other input variable is neglected". For the two-bit OR problem, an abstract statement is that "any true value at an input variable is the deciding feature to give a true output value, the other input variable is neglected". Without a direct description of such abstractness, extracted rules hardly offer any advantage over the original data set. Thrun's rule extraction method [18] using a combined strategy appears not to suffer from the above limitations, neither does GR2.

3. Rule extraction in GR2

GR2 extracts rules that only include the input and output variables. The resultant rules are independent of the internal structure of the neural network. The rules represent an abstraction of the domain knowledge.

We use Boolean variables for the definitions given in this section in order to correspond to the status of the units in the neural networks. A multiple-valued variable can be represented by multiple units in the neural networks, so with multiple Boolean variables, GR2 can handle most discrete problems.

A training instance comprises an input vector $<I_1, I_2, ..., I_N>$ and an output value O. The elements in the input vectors are a set of instantiated input variables. The output value is an instantiated output variable obtained by recalling the MLP which represents the class to which the instance belongs.

The training set is modified in two operations. The first operation is to replace the outputs in the training set by the outputs after recalling the MLP. This presupposes that the training process has attained a satisfactory level of performance. There will thus be no conflict instances in the training set. The second operation is to select distinct instances from the training set into a distinct training set after the first operation. In the rest of the paper, the term "training set" indicates this distinct training set and a "training instance" indicates a distinct training instance.

The central operation of rule extraction in GR2 is the selection of the subset of the input vector from each training instance, presented in Section 3.3. The selection is based on two criteria, being defined in Sections 3.1 and 3.2.

3.1 The Potential Default Set

The criterion derived from internal behaviour is called the Potential Default Set (PDS). Observing the *i*th input unit I_i relating to the output unit O via all hidden units and the weights between them in the network, we can estimate if I_i provides a positive or negative contribution to the activation of the unit O by $L_i = \sum_h w_{ih} h_h w_{ho}$, where the subscript h ranges over all hidden units in the network, h_h is the hth hidden unit activation[2], w_{ih} is the weight between the input unit I_i and the hidden unit h_h, and w_{ho} is the weight between h_h and the output unit O. The activation of h_h and the output unit O are obtained as the input vector of a training instance is fed to the input units of the network and the network is recalled. Now we consider the L_is together with the activations of the input and output units. For example, if the activation of the input unit I_i is 0, and L_i>0, switching I_i from 0 to 1 tends to increase the activation of the output unit O. But if the activation of O is already high, i.e. in the range [δ, 1], where δ is the error tolerance, this change on I_i does not affect the status of O. On the other hand, the value of I_i, either 0 or 1, does not affect the status of O. The input variable corresponding to I_i is possibly redundant in this situation. There are 3 more situations similar to the one above. All the 4 situations are listed in Table 4, among the total 8 possible combining situations of the input unit activation I_i, the output unit activation O, and L_i. The *Potential Default Set* is the set of those input variables corresponding to these 4 situations. In order to make the measure more reliable, we set a threshold in Table 4, $\tau(\tilde{O}) = \left|\log\frac{\tilde{O}}{1-\tilde{O}}\right|$, where $\tilde{O}$ is the output activation but limited in the range [0.0001,0.9999], i.e. $\tilde{O} = \begin{cases} 0.0001 & if\ O < 0.0001 \\ 0.9999 & if\ O > 0.9999 \\ O & otherwise \end{cases}$.

$f(x) = \log\frac{x}{1-x}$ is the inverse function of the activation function in the network $F(x) = \frac{1}{1-e^{-x}}$.

The property of the function $\tau(x)$ is shown in Figure 4. $\tau(x)$ has a minimal value 0 as x=0.5, at which situation the classification result is not clear and is sensitive to the change on the input variable, thus the Potential Default Set depends on the sign of L_i. The closer the output activation to the extreme values, 0 or 1, the more confidently the classification is made, the less influence the change of the input has on the output result, and the higher the threshold $\tau(x)$ is set.

[2] We use I_i, h_h, and O to refer to the unit activations when a calculation is required. They also refer to the units themselves when the network is discussed. The two uses can clearly be distinguished from the context.

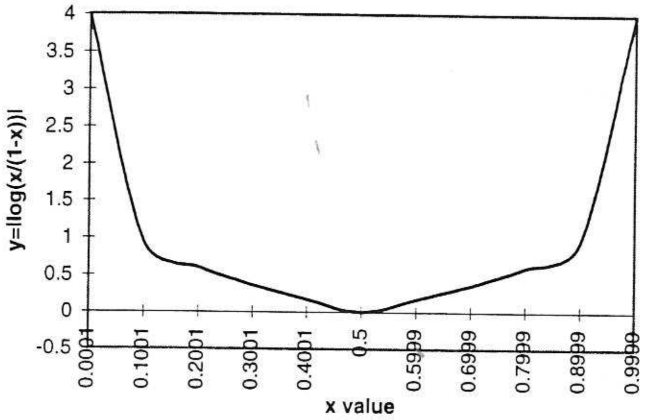

Figure 4. $\tau(x)$ function curve

In brief, an input variable is in the Potential Default Set in the following situations: $L_i \geq \tau(x)$ and I_i and O are of opposite status or $L_i \leq -\tau(x)$ and I_i and O are of the same status. The input variables in the Potential Default Set are possibly not important in generating the output and may therefore be absent from the extracted rules.

I_i	O	L_i
0	[1-δ,1]	$\geq \tau(\tilde{O})$
1	[0, δ]	$\geq \tau(\tilde{O})$
1	[1-δ,1]	$\leq -\tau(\tilde{O})$
0	[0, δ]	$\leq -\tau(\tilde{O})$

Table 4. Situations where I_i belongs to the Potential Default Set

The calculation of the Potential Default Set is suited to the MLP for any number of hidden layers or to the simple continuous Perceptron network without the hidden layer. In the latter case, the weights themselves are the L_is. In the former case, an L_i is the cascade product of the weights and the activations of the hidden units which are linked via the weights in the network.

Taking the network for the two-bit AND problem in Figure 1 as an example, and defining δ=0.1, the weight matrices after training are:

$$W_1 = \begin{bmatrix} -5 \\ -5 \end{bmatrix};\ W_2 = [-9].$$

For the first instance of the training set, the input vector I=<0 0>, and the output by the MLP O=0.00253 that is in the range [0, δ]. The hidden unit activation h_0=1.0 and L=<45 45>. The PDS threshold τ=5.978. The Potential Default Set PDS_1 = { } because $L_i > \tau$ and the input variables and the output variable are at the same status.

For the second instance, input $I=<0\ 1>$, $O=0.02713$, $h_0=0.9933$, $L=<44.7\ 44.7>$, $\tau=3.58$. Only the input B=1 has the opposite status from that of O. Thus $PDS_2=\{B\}$. Similarly for the third instance, $PDS_3=\{A\}$.

For the fourth instance, $I=<1\ 1>$, $O=0.9447$, $h_0=10^{-5}$, $L=<0.00045\ 0.00045>$, $\tau=2.838$. Since I_i and O have the same status and $L_i<\tau$, $PDS_4=\{\}$.

Now deleting the PDS from the full set of input symbols, we obtain the following remainder matrix, where ~ before a symbol indicates that the variable is instantiated with value 0, and – indicates absence of any symbol which belongs to the PDS:

$$\begin{bmatrix} \sim A & \sim B \\ \sim A & - \\ - & \sim B \\ A & B \end{bmatrix}$$

Rows 2 and 3 represent the abstraction property that "any false value to an input variable is the deciding feature to result in the false value to the output, the other input variable is neglected". In the first row, the redundant elements will lead to a redundant rule (i.e. it can be subsumed by the other rules) if we do not consider the other selection criterion, the Feature Salient Degree to be defined in next subsection. The PDS results for the other networks for the two-bit AND problem in last subsection are the same as above.

Similarly the PDS matrix for the two-bit OR problem is

$$\begin{bmatrix} - & - \\ \sim A & - \\ - & \sim B \\ - & - \end{bmatrix}$$

and the remainder matrix is

$$\begin{bmatrix} \sim A & \sim B \\ - & B \\ A & - \\ A & B \end{bmatrix}$$

Rows 2 and 3 represent the abstraction property that "any true value at an input variable is the deciding feature to give the true output value, the other input variable is neglected".

Although the Potential Default Set alone is not sufficient to describe all the properties represented by a network, it partially reflects the statistical and qualitative input/output relationship encoded in the trained network. It is also very simple to calculate.

3.2 The Feature Salient Degree

The criterion obtained from external behaviour is the Feature Salient Degree of an input variable in a training instance, which measures the strength of influence of that input variable on the output variable in the context of the whole training set. For the whole training set the feature salient degree comprises a PxN matrix, where P is the number of the training instances and N the input dimension of the MLP. First we define the matrix *fsd*, whose *ji*th element is given by following equation.

$$fsd_{ji} = \sum_{\{k|(j \neq k, O^j \neq O^k, I_{ji} \neq I_{ki})\}} \frac{1}{|P_j, P_k| 2^{|P_j, P_k|}}$$

and the matrix FSD is the normalisation of the matrix fsd:

$$FSD = \frac{fsd}{\max(fsd)}$$

The explanation of the FSD is given in [25,26].

P_j and P_k denote different instances in the training set and $|P_j, P_k|$ is the hamming distance of their input vectors. μ_j is the number of times the *j*th distinct instance appears in the original training set. If the input vectors are the same, so will the outputs be after modification by the MLP.

A feature salient degree of the *i*th input variable in the *j*th training instance, FSD_{ij}, depends on how many other training instances whose output value and *i*th input value are different3 from those in the *j*th distinct training instance. The FSD_{ij} is also exponentially inversely relatied to the Hamming distance of the input vectors of the training instances, reflecting the context relevance among the input variables.

Taking the two-bit AND problem as an example, we have:

In the first instance (<0 0>, 0.0), for the input variable A=0, only instance 4, (<1 1>, 0.94), is counted since it is the only one in which the value of A and the output value *C* are different from those in instance 1. The hamming distance $|P_1,P_4|$ is 2. Thus the fsd element is $fsd_{11} = \frac{1}{2 \times 2^2} = 0.125$.

For the same reason $fsd_{12}=fsd_{11}$.

For the second instance, (<0 1>, 0.02), only the fourth instance has a different output from it and $|P_1,P_4|=1$. Thus $fsd_{21} = \frac{1}{1 \times 2^1} = 0.5$ for the input variable A. For B=1, $fsd_{22}=0$ because B=1 in the fourth instance as well.

Similarly for the instance (<1 0>,0.02), $fsd_{31} = 0.0$ and $fsd_{32} = 0.5$.

For instance four, (<1 1>, 0.94), the other three instances have an output variable different from that in this instance so that whether they are counted for the *fsd* values for this instance depends on whether their corresponding input variables are different from those in this instance. On the input variable A, this instance differs from instance 1 and 2, and the Hamming distances of the input pairs are 2 and 1 respectively.

Therefore $fsd_{41} = \frac{1}{2 \times 2^2} + \frac{1}{1 \times 2^1} = 0.625$. Similarly on the input variable B, instance 1 and instance 3 differ from this instance thus $fsd_{41} = fsd_{42}$.

The fsd matrix and FSD matrix are respectively

---

[3] Here different output values means they belong to different ranges of [0, δ] and [1–δ, 1].

$$fsd = \begin{bmatrix} 0.125 & 0.125 \\ 0.5 & 0 \\ 0 & 0.5 \\ 0.625 & 0.625 \end{bmatrix} \text{ and } FSD = \begin{bmatrix} .269 & .269 \\ .731 & 0 \\ 0 & .731 \\ 1 & 1 \end{bmatrix}$$

The first row of the *FSD* matrix indicates that both input variables in instance 1 have equal and little importance. In the second instance, the first input variable, ~A, is important whilst the second input variable, B, is of no importance at all. Similarly for the second input variable, ~B, in the third instance. Both input variables in the last instance are important in generating the corresponding rule.

The *FSD* matrix for the network in Figure 2 is the same as above.

For the two-bit OR problem, the *fsd* and *FSD* matrices are

$$fsd = \begin{bmatrix} 0.625 & 0.625 \\ 0 & 0.5 \\ 0.5 & 0 \\ 0.125 & 0.125 \end{bmatrix} \text{ and } FSD = \begin{bmatrix} 1 & 1 \\ 0 & .731 \\ .731 & 0 \\ .269 & .269 \end{bmatrix}$$

The strength of the FSD just indicates the importance of the element in the corresponding rules which matches the designed logic of the domain.

For the XOR problem, to each input variable in an training instance, there corresponds just one other instance whose input and output variables are different, and the Hamming distance of the input vectors involved is 1. Therefore

$$fsd = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \text{ and } FSD = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

3.3 Rule generation

Rules are generated from the training instances using the black-box approach. The central idea is to select the fewest elements in the input vector that can characterise the necessary and sufficient input features to conclude the output result. Here the output must be that of the MLP, so that there will be no conflicts in the training set even if there are some before the MLP is applied. The selected input variables are converted into the premises of the extracted rule, and the outputs are converted into its consequence. The fewer the premises in the rule, the more general the rule. General rules represent the general properties of the input/output relationship encoded in the trained MLP. If the rules are general enough to cover larger ranges than the training set, other instances in the problem space beyond the training set can be classified. This is the generalisation ability the rule extraction process provides.

Once a rule is generated it is tested on the training set in the validation process. The number of the training instances matching the rule is the *matched cover range* of the rule, and the number of instances in conflict with the rule is the *conflict cover range* of the rule[4]. A *certainty factor* is given by the ratio of the matched cover range to the sum of both of the cover ranges. This is also used to select valid rules. There is another quantitative descriptor for each premise of the rule, called the *contribution factor*. It represents the strength of the causal relationship between the input and the output variables.

GR2 selects the subset of the input vector in the light of the two criteria aforementioned. We present a simplified algorithm in this paper for easy understanding, although a more complex one [25,27] gives more control facilities which has benefits for practical and complicated or noisy data sets. Three sets of data are required for generating a rule: the training instances (modified by the MLP), the Feature Salient Degree vector and the Potential Default Set corresponding to the input vector. We have also changed the general control strategy of the extraction algorithm in line with [20], i.e. maintaining a rule set for each class and checking if a training instance has been covered by the rule set in order to decide if the rule extraction operation takes place.

3.3.1 Rule extraction: general control:

1. For each class d, initialise a rule set

$$R_d = \{\}$$

2. For each training instance e_i whose output indicates class c

if e_i is not covered by R_c then

$$R_c = R_c \vee \text{Rule\_extraction}(e_i)$$

3.3.2 Rule extraction: operation (Rule_extraction(e_i))

Selection by FSD: collect all elements of the input vector whose feature salient degrees are no less than T/N into a set θ. If the set is empty, stop, no rule is to be generated. Here N is the dimension of the input vector and T is a FSD threshold to the class of this instance. The FSD threshold T can be separately set for every class of the domain. Adjusting T for each class can change the generality and the accuracy of the extracted rules. T=1 is recommended as the default setting.

Selection by PDS: from the set θ, delete those elements which are in the Potential Default Set and to which FSD $\le \psi/2$. Here ψ is the maximal value in FSD vector.

Construction: the remainder elements in θ are symbolised as the premises, the output value is symbolised as the consequence. To symbolise an input or output value means to give a symbolic name to the variable, and to add a ~ to represent "not" if its value is 0 or in the range of $[0, \delta]$.

Validation: apply the rule on the training set and find two numbers. γ, the *matched cover range* of the rule, and ε, the *conflict cover range* of the rule. The rule is endowed with a

---

[4] A rule matching an instance means the premise part of the rule is correspondingly the subset of the input vector of the instance, and the consequence of the rule is correspondingly the status of the output value of the instance. A rule conflicting with an instance means the premise part of the rule is correspondingly the subset of the input vector of the instance and the consequence of the rule is correspondingly different from the status of the output value. There is no relationship between a rule and an instance where the premise part of the rule is not correspondingly a subset of the input vector of the instance.

certainty degree , $\alpha=\gamma/(\gamma+\varepsilon)$. If α is less than an *accuracy expectation*, say 80%, dispose of this rule.

Generalisation: this is an iterative but *optional* step. It can be taken by the user who has found the previous generated rule set is not general enough. If the rule is quite long compared with the input dimension size, find the input variable whose FSD is minimal in the set θ. If the more general rule obtained by discarding this input variable is more accurate than the previous rule in terms of the cover ranges, the new rule takes place of the previous one. It is suggested that the length of the new rule should not to be less than log(N) according to our experience.

Reserving FSDs: Each premise of the rule is attached with a *contribution factor*, which are the normalised FSDs in the set θ,

$$\beta_i = \frac{FSD_i}{\sum_j FSD_j}.$$

In the rule validation step, the certainty factor is checked against the expected accuracy. The value of the expected accuracy is decided according to the minimal accuracy with which we expect the rule set to work on the training set. If the cover ranges are counted on the training set modified by the output results of the MLP, it is quite right to set the accuracy expectation 100%. However, if the cover ranges are counted on the original training set itself, the training set may contain conflict instances. The accuracy expectation should then be accordingly reduced. Otherwise some useful rules can be missed.

Take the two-bit AND as an example and set the FSD threshold as the default T=1. Here N=2.

For the first instance, (<0 0>, 0.0), no rule is generated because $FSD_{11}= FSD_{12}=0.125$, that is less than 1/N, where 1/N=0.5. Indeed, this instance is less important than the others.

For instance 2, (<0 1>, 0.02), a rule is generated in the Rule Extraction operation.

Selection by FSD: $\theta = \{A\}$ because $FSD_{21}=0.731>1/N$ corresponding to A=0, and $FSD_{22}=0<1/N$ corresponding to B=1.

Selection by PDS: nothing is done in this step.

Construction: IF(~A) THEN C;

Validation: the rule matches instance 1 and 2, thus $\gamma=2$. It conflicts with no instance, thus $\varepsilon=0$. $\alpha=\gamma/(\gamma+\varepsilon)=1$. The rule is valid.

Reserving FSDs: $\beta_1=1.0$ for premise A since it is the only element selected.

The general rule is briefly presented as

IF(~A(1.0)) THEN ~C; (1.0)

For instance 3, similar to instance 2, a rule is generated as

IF(~B(1.0)) THEN ~C; (1.0)

For instance 4, (<1 1>, 0.94), a rule is generated in the steps:

Selection by FSD: $\theta = \{A, B\}$ because $FSD_{41} = FSD_{42} = 1 > 1/N$.

Selection by FSD: nothing is done since neither A nor B belongs to the PDS.

Construction: IF(A, B) THEN C;

Validation: the rule matches only instance 4 and conflicts with no instance. $\gamma=1$ and $\varepsilon=0$. $\alpha=\gamma/(\gamma+\varepsilon)=1$. The rule is valid.

Reserving FSDs: $\beta_1=\beta_2=0.5$ for premise A and B.

IF(A(0.5), B(0.5)) THEN C; (1.0)

This rule indicates that the condition A=1 and B=1 are equally necessary to conclude C=1, consonant with our knowledge of the domain.

In this domain the FSD alone has given sufficient clues to extract the most general rule set. The PDS, however, also correctly indicates the input variables to be eliminated from the general rules at the appropriate stages. GR2 generates the same rule set for both of the networks shown in Figures 1 and 2.

Similarly the rule set for the two-bit OR problem is:

IF(~A(0.5), ~B(0.5)) THEN ~C; (1.0)

IF(B(1.0)) THEN C; (1.0)

IF(A(1.0)) THEN C; (1.0)

The XOR problem is studied as a challenging learning domain for the Perceptron (including the MLP) networks. One hidden unit is required. The domain is easily handled by GR2 however. Since each FSD element is 1, the resultant rules are the symbolic form of the training instances themselves, with all certainty factors 1.0 and all contribution factors equally 0.5. Indeed, the four rules with the full length can not be further generalised, and every premise in the rules are equally necessary.

4. Explaining cases by general rules

Explanation within the field of symbolic AI refers to "an explicit structure which can be used internally for reasoning and learning, and externally for the explanation of results to a user" [22]. This provides important lessons in realisation of the explanation functionality for a knowledge-based system involving neural networks, as well as raising a challenging requirement to such realisation. This section looks at how the general rules can be used in reasoning with and explaning individual instances.

Not as rigid as the format of the traditional production rules which only conveys qualitative information, the rules extracted from the trained MLP can inherit the numerical properties produced and used in the extraction process. The numerical properties afford valuable quantitative information which is difficult to acquire from human experts but is relatively easily understood by them. From the viewpoint of knowledge discovery in some complex domain, the quantitative information can provide further subtle insights of the domain knowledge which might be unknown to or may not yet have been verbalised by the human experts.

Explanation at case level in GR2 is based on the proof trace concept in the reasoning (inference) process [4]. Since the extracted rules only include the input and output variables, rather than intermediate variables, the proof trace is trivial.

GR2's advantage arises from the its use of general rules to represent knowledge. The symbolic part of a rule represents the qualitative knowledge and the numerical part represents the quantitative knowledge and makes for explanation with more relevant information. This can be explained in terms of the cover ranges. A rule with l premises covers 2^{N-l} instances in the domain space, where N is the input dimension. A subspace of the domain space can be a conjunction of the cover ranges of the rules, or a conjunction of the rules for simplicity. If the subspace is a conjunction of the rules for one class, the decision making for an instance in this subspace is easy and the qualitative knowledge plays a major role. On the other hand, if the subspace is a conjunction of the rules for different classes, the quantitative knowledge is more important for decision making.

The representation by general rules also makes reasoning by rules more reliable. For example, if there is a case that could match the rules of different classes, the general rules provide more information for conflict resolution than the traditional production rules. "Failure" is a difficulty for explanation by proof trace, because the proof trace only records those rules being fired [3]. GR2 provides some facilities to handle this situation. First, some cases of "failure" in traditional rule-based systems can be identified and concluded by exceptional reasoning. Second, GR2 can simply give the result from the MLP, without any more information, of course. Finally, there is a more flexible approach which is to invite the user's interaction.

Explanation by rules in GR2 is closely relevant to the process of reasoning by rules, so we introduce the reasoning process first.

4.1 Reasoning with extracted rules

Parallel to the recall process in the MLP, reasoning by rules is the process to decide the conclusion based on the extracted rules when an input vector is given. Reasoning by rules in GR2 uses two operations.

4.1.1 Matching

All rules covering the input vector are collected as voting candidates. Collection is similar to being stored in the working memory in traditional expert systems. These rules are called the *matched rules* to the input vector. There are four situations of rule collection that may be encountered in this operation, which we call matching situations:

1. All matched rules are of one class – *uniform matching*;
2. the matched rules are of different classes – *cross matching*;
3. there is no matched rule but there are no extracted rules of one class at all – *exceptional matching*;
4. there are extracted rules of every class but there is no matched rule to the input vector – *unmatched*.

All other operations of reasoning and explanation in the rest of this section are manipulated according to these four matching situations. Situation 1 is the most common one which is easy to cope with. When situation 2 occurs, there is evidence that the current input vector does not uniformly correspond to one class. Although not occurring very often, this is a difficult situation in which some heuristic or expert judgement is most needed. The third situation – exceptional matching – is needed if no rule is extracted. This is possibly because the training set does not contain sufficient information to generate a reliable classification or because the FSD threshold, T, for the class in the rule extraction process is set too high. Hence the current input vector is assumed to correspond to the absent class in the extracted rule set, even if there are no rules directly matching this input vector. Situation 4 is the only case where the extracted rules are not able to be used directly for reasoning. There are extracted rules for every class but no one covers the input vector. This is because either the input vector is an exception, so that the training set does not represent this case[5], or the extracted rules are not general enough to cover the domain in the vicinity of the input vector. We will discuss methods to deal with this situation in Section 4.3 even though the conclusion can be gained simply from the MLP.

4.1.2 Concluding

This operation decides the class that the input vector corresponds to by application of the matched rules. This is similar to conflict resolution in traditional expert systems.

In situation 1, *direct decision*, the class is that of the matched rule.

In situation 2, a *vote* is held among the classes by the matched rules. The confidence degree of the matched rule is relevant to the vote. We also need to consider other factors such as the lengths and the cover ranges of the rules. A short rule covers a larger range than a longer rule does. It is thus more apt to tolerate exceptional instances in its cover range and less certain to confirm an instance than a longer rule if they are in conflict and both match the input vector. So we derive a value given by the following equation. For the *i*th class, the value is:

$$\sigma_i = \sum_{Rule_j \text{ is of } class_i} \alpha_j \times \frac{\gamma_j}{2^{N-\lambda_j}}$$

where α_j is the certainty factor of the *j*th rule for the *i*th class, γ_j and λ_j are respectively the matched cover range and the length of the rule. The conclusion is the class whose value σ is the maximum.

In situation 3, exceptional reasoning is performed. The input vector corresponds to the class for which there is no extracted rule.

In situation 4, the conclusion is given by the trained MLP.

4.2 Explanation by the factors in the matched rules

Explanation provides three kinds of information for how the conclusion is reached: the certainty degree of the conclusion, the causal relationship between the attributes in the input vector and the consequence, and the matched rules themselves.

4.2.1 Certainty degree

The conclusion of reasoning by a rule is given with a degree of certainty, ϕ.

In situation 1, the certainty degree is related to the certainty factor of the matched rules.

$$\phi = \max_{rule_i}(\alpha_i)$$

[5] This does not mean that the input vector has to appear in the training set in order to be learnt.

where rule_i is a matched rule and α_i is the certainty factor for the rule.

In situation 2, the certainty degree is:

$$\phi = \frac{\sigma_j}{\sum_k \sigma_k}$$

where j is the winning class as the conclusion and the index k ranges over all classes.

In situations 3 and 4, the certainty degree can not be computed from the rules. However, if the MLP is applied, the class conditional accuracy provides the degree of certainty.

4.2.2 Contribution degree

A *contribution degree* is given to an input variable in the input vector. It indicates the contribution strength that an individual input variable has in supporting the conclusion, or a quantitative causal relationship. Contribution degrees for an input vector are not necessarily normalised which means that a part of the input vector may be sufficient to result in the conclusion. They can also be negative, indicating the contribution the input variable makes in denying the conclusion. The contribution degree is defined according to the four matching situations.

In situation 1, the contribution degree of the ith input variable depends on the contribution factors in the matched rules, and the certainty factors of the rules.

$$\omega_i = \max_{\text{matched rule}_j} (\beta_{ji}\alpha_j)$$

where β_{ji} is the contribution factor of the premise in the jth rule corresponding to the ith input variable, which is 0 if the premise does not exist in the rule.

In situation 2, the contribution degree is reduced because the same input variable may support different classes in different matched rules.

$$\omega_i = \max_{\text{matched rule}_k \text{ in the winning class}_i} (\beta_k\alpha_k) - \sum_{j \neq i} \max_{\text{matched rule}_m \text{ in another class}_j} (\beta_k\alpha_k)$$

where the ith class is the winning one and j ranges over the other classes. The contribution degree can be negative, indicating that the ith input variable in the input vector tends to counteract the conclusion.

In situations 3 and 4, no contribution degree can be computed.

4.2.3 Matched rules and cover ranges

The matched rules form the detailed qualitative content of the explanation. The rules are self-explanatory. The cover ranges of the rules indicate the generality of each rule..

RFH

4.3 Simple examples

Reasoning and explanation on the simple two bit domains are trivial. Notice that all possible instances have been included in the training sets without noise, so that reasoning and explaining are performed only at matching situation 1.

For the two bit AND problem, given an input vector <0 0>, there are two matched rules:

> $Rule_1$: IF(~A(0.73)) THEN ~C; (1.0)
>
> $Rule_2$: IF(~B(0.73)) THEN ~C; (1.0)

What is the explanation?

First, the certainty degree ϕ=max(1.0, 1.0)=1.0. Second, the contribution degrees to the input variables A=0 and B=0 are 1.0, indicating that either of the input variables is sufficient to conclude C=0. The explanation is

The input vector 1 is <0 0>; the result is 0,

with certainty degree 1

The contribution degrees of the conditions to the conclusion:

A=0: 1.0

B=0: 1.0

There are 2 rules matched:

IF (~B(1.0)) THEN (~C); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

IF (~A(1.0)) THEN (~C); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

The conclusion is C=0 by direct reasoning.

The input vector 2 is <0 1>; the result is 0,

with certainty degree 1

The contribution degree of the conditions to the conclusion:

A=0: 1.0

There is 1 rule matched:

IF (~A(1.0)) THEN (~C); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

The conclusion is C=0 by direct reasoning.

Similarly for the case when the input vector is <1 0>.

When the input vector is <1 1>, the explanation is:

The input vector 4 is <1 1>; the result is1,

with certainty degree 1

The contribution degrees of the conditions to the conclusion:

A=1: 0.5

B=1: 0.5

There is 1 rule matched:

IF (A(0.5), B(0.5)) THEN (C); (1.0)

Matched Cover Range=1; Conflict Cover Range=0

The conclusion is C=1 by direct reasoning.

For the XOR problem, the explanations for the four instances are similar. We give only one here:

> The input vector 2 is <0 1>; the result is 0,
>
> with a certainty degree 1
>
> The contribution degrees of the conditions to the conclusion:
>
> A=0: 0.5
>
> B=1: 0.5
>
> There is 1 rule matched:
>
> IF (A(0.50), ~B(0.50)) THEN (~C); (1.0)
>
> Matched Cover Range=1; Conflict Cover Range=0
>
> The conclusion is C=0 by direct reasoning.

The contribution degree of 0.5 for the two attributes indicates that to decide the conclusion, both of the input variables are equally necessary.

We will see examples in Section 5 where explanation is more complex when instances are absent from the training set, are in conflict or are unmatched by any rule.

4.4 How explanation enhances the reasoning capability

An additional advantage of explanation-based learning is that it is relatively easy to enable the user to participate in the reasoning process. Explanation provides users with knowledge in the form of symbolic rules. They are thus able to inspect the rules, to judge if the rules agree with knowledge of the domain previously acquired in some other way, and to modify the rules if necessary and the user is qualified to do so.

Modification of the rules is straightforward. Users can change the certainty factors and contribution factors, add or remove premises, or simply delete the rule as desired. They can also input rules either according to their own knowledge or for the purpose of experimentation. In addition, GR2 provides a function in the rule extraction process that allows users to construct a rule in consultation with the given PDS and FSD vectors, to verifies the rule, and finally to confirm or discard the new rule.

So far, reasoning by rules is based on exact matching. To explain an input vector, only those rules exactly covering the input vector are considerated. This restriction causes the reasoning to fail in some cases, though these are very rare when an optimally extracted rule set has been acquired. Why does this failure happen? It may be because the rules are not properly general since many premises with low contribution factors may be present. Another reason is that the input vector may be an outlier in the training set from which the rules are generated. GR2 provides utilities to recover such failures.

The first utility is a process of *rule generalisation*. Given an extracted rule, the following process is executed in an iteration:

1. From a rule r_i, remove the premise with the minimal *FSD*. If there is more than one premise with minimal *FSD*, remove any belonging to the PDS. A generalised rule $\hat{r}_i$ is generated.
2. If the new rule $\hat{r}_i$ is better than the old rule r_i, in terms of the certainty factor and the matched cover range, replace r_i with $\hat{r}_i$ and go to step 1). Otherwise go to step 3).

3. The rule $\hat{r}_i$ is the final generalised rule. Terminate the process.

The second utility is called *loose matching*. In the first operation of the reasoning process in subsection 4.1, "matching", set a threshold for the contribution factors of those premises that match in the input vector. Let's call it the "Contribution Threshold". When a part of the premises of a rule has a sum of the contribution factors exceeding the Contribution Threshold (in verification), the rule is accepted as matched and is collected. Since the loose matching checks the rules by a part of their premises the unmatched situation will be avoided depending on how low the Contribution Threshold is set.

5. Examples

5.1 "Go-to-beach?" example

We have constructed a simple example, called Go-to-beach?, to illustrate our method. The scenario is to decide whether or not to go to the beach. In the example, there are four binary attributes under consideration:

Attribute A: *weekday*: is it a weekday or not?

Attribute B: *hot*: is it hot or not?

Attribute C: *dry*: is it dry or not?

Attribute D: *windy*: is it windy or not?

The conclusion is represented as:

E: go to the beach; ~E: do not go to the beach.

The decision to go to the beach is made only under the following conditions.

a) If it is not a weekday (the weekend), *and* it is any of the following: not hot, dry and windy (e.g. for sailing), hot and not windy (e.g. for swimming).

b) If it is a weekday, and it is at least hot and dry.

The complete training set (truth table) is listed in Table 5.

Case #	weekday	hot	dry	windy	go to beach
1	0	0	0	0	0
2	0	0	0	1	0
3	0	0	1	0	0
4	0	0	1	1	1
5	0	1	0	0	1
6	0	1	0	1	0
7	0	1	1	0	1
8	0	1	1	1	0
9	1	0	0	0	0
10	1	0	0	1	0
11	1	0	1	0	0
12	1	0	1	1	0
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	1
16	1	1	1	1	1

Table 5 The "Go-to-beach?" truth table

This scenario is designed to demonstrate some difficulty for a common MLP, characterised with one layer of hidden units, generalised delta learning rule, and sigmoidal activation functions. If the number of hidden units is two or fewer, the MLP cannot learn the data with 100% accuracy. When the number of hidden units increases to 3 or more, this problem is solved.

The weights and biases in the trained network are shown in Table 6 when the MLP has only two hidden units,. This network cannot recognise instance 8 and its inability to do so is explained during rule extraction.

	hidden$_1$	**hidden$_2$**
input$_1$	0.96	1.72
input$_2$	4.43	-1.36
input$_3$	-2.34	-3.72
input$_4$	4.65	2.30
Bias	0.18	-0.99

	output
hidden$_1$	5.63
hidden$_2$	-13.3
Bias	-2.96

Table 6. The weights of the MLP with 2 hidden units, for the go-to-beach domain

As the input dimension N=4, the threshold to select FSD values is 1/N=0.25.

There are 8 rules generated. We omit the details in this example but highlight the case that the reasoning by rules gives the wrong solution for instance 8, as does the MLP. The rules are:

IF (~Hot(0.62), ~Dry(0.38)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~Hot(0.60), ~Windy(0.40)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~Dry(0.60), Windy(0.40)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (Weekday(0.33), ~Hot(0.67)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (Weekday(0.45), ~Dry(0.55)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~Weekday(0.32), Hot(0.38), ~Windy(0.30)) THEN (Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

IF (Hot(0.47), Dry(0.53)) THEN (Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

Given the condition <0 1 1 1> at case 8, i.e. ~weekday, hot, dry and windy, the MLP output is 0.76, but the definition in the training set is E=0, i.e. ~ Go-to-beach. The explanation is:

The input vector 8 is <0 1 1 1>, the defined output is 0,

The result of reasoning by rules is E=1

with certainty degree 1.0

The contribution degrees of the conditions to the conclusion:

weekday=0:	0.0
hot=1:	0.47
dry=1:	0.53
windy=1:	0.0

There is 1 rule matched:

IF (Hot(0.47), Dry(0.53)) THEN (Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Go-to-beach by direct reasoning.

The conclusion by the rules is in conflict with the domain definition because the matched rule is over generalised according to the original domain definition.

Extending the hidden layer with three units, the MLP can correctly classify all 16 possible instances. The trained MLP weights are given in Table 7:

	$hidden_1$	**$hidden_2$**	**$hidden_3$**
$input_1$	-1.09	-5.59	6.16
$input_2$	-5.82	-5.90	-6.27
$input_3$	1.10	-0.02	-6.57
$input_4$	5.39	6.09	-0.85
Bias	-3.61	2.80	3.01

	output
$hidden_1$	-24.6
$hidden_2$	88.0
$hidden_3$	99.4
Bias	6.16

Table 7. The weights of the MLP with 3 hidden units, for the go-to-beach domain

This gives a total of 10 extracted rules which correctly recognise all of the input vectors. Thes are given below.

********** Rules for positive class ***********

IF (~weekday(0.26), ~hot(0.20), dry(0.30), windy(0.24)) THEN (Go-to-beach); (1.0)

Matched Cover Range=1; Conflict Cover Range=0

IF (~weekday(0.30), hot(0.36), ~windy(0.34)) THEN (Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

IF (hot(0.40), dry(0.36), ~windy(0.24)) THEN (Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

IF (weekday(0.27), hot(0.35), dry(0.38)) THEN (Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

********** Rules for negative class ***********

IF (~hot(0.63), ~dry(0.37)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~hot(0.62), ~windy(0.38)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~dry(0.45), windy(0.55)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~weekday(0.33), hot(0.24), windy(0.42)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

IF (weekday(0.32), ~hot(0.68)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (weekday(0.44), ~dry(0.56)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The detailed case-bycase explanations are given in Appendix A. Note that for this deterministic domain reasoning by rules yields an overall predictive accuracy of 1.

5.2 The monk's problem data sets

The three Monk's problems constitute a benchmark domain for inductive machine learning algorithms [11]. They are discrete classification problems with six different attributes to identify an artificial robot. The six attributes are

x_1: head_shape {round, square, octagon}

x_2: body_shape {round, square, octagon}

x_3 is_smiling {yes, no}

x_4: holding {sword, balloon, flag}

x_5: jacket_color {red, yellow, green, blue}

x_6: has_tie {yes, no}

We examine problem M_1 and M_3 and omit M2 because of the space limit in this paper.

5.2.1 Monk 1

Problem M_1: (head_shape = body_shape) or (jacket_color = red).

There are 124 training instances randomly selected from 432 possible examples. There are no misclassifications in this data set.

Networks are constructed as presented in Chapter 9 of [11], see Table 7. There are 17 input units, each corresponding to one of the 17 (boolean) input feature values, and one hidden layer with 3 hidden units.

The following observation becomes necessary in improving the quality of the rule set. The input vectors of the training data are coded under a local constraint in order to represent a multiple valued domain by boolean valued input units in the MLP. An input variable (x1 to x6) corresponds to a set of input units depending on the number of its possible values. One of the input units to each input variable must be 1 and the rest must be 0. This local constraint is represented in GR2 by input groups. In the rule extraction process, only the input feature with value 1 can be selected from each group for converting into a premise of the rule. This is a general modification on the rule extraction process to cope with domains whose input variables have multiple values. Of course, boolean input variables are a special case of the variable with multiple values. GR2 allows a boolean input represented either by one input unit, or by two units with exclusive values. Note: N for setting the FSD threshold is the number of input variables, six in this example, instead of the input vector dimension.

	hidden1	**hidden2**	**hidden3**
x1=1	0.379863	3.38416	3.97957
x1=2	-3.36186	0.120335	-1.36079
x1=3	1.59203	-3.66981	-4.54251
x2=1	0.152452	3.11559	4.29571
x2=2	1.70372	-3.55563	-5.28836
x2=3	-3.28578	0.170422	-0.724256
x3=1	-0.679289	-0.300242	-0.879989
x3=2	-0.289235	-0.141164	-0.591719
x4=1	-0.460906	-0.457638	-0.478595
x4=2	-0.04827	0.0186045	-0.40257
x4=3	-0.484881	0.0805534	-0.780363
x5=1	-4.25606	-6.21256	4.23643
x5=2	0.840806	1.7855	-1.91755
x5=3	0.857625	2.15388	-1.66635
x5=4	1.38552	1.57324	-2.4776
x6=1	-0.452624	-0.104307	-0.664879
x6=2	-0.474712	-0.23297	-1.16858
bias	-1.029948	-0.55633402	-1.6094799

	output
hidden1	-5.20181
hidden2	-7.06326
hidden3	9.29557
bias	3.0946786

Table 8. MLP weights for the M_1 problem

There is a special feature in the definition of the M_1 problem: only the positive property is specified. If any negative rule is generated, until they cover all possible space for the negative class, there will be mis-classified instances in the test (whole) set. To prevent GR2 from extracting negative rules, we simply raise the FSD threshold T for the negative class to a large value, say 100, so that no rule of the negative class will be generated. Here only a few representative instances are illustrated in Appendix B. The overall accuracy on the M_1 problem is 100%.

5.2.2 Monk 3

Problem M3: (holding a sword and jacket_color is green) or (body_shape is not octagon and jacket_color is not blue), simplified as (x4=1 and x5=3) or (x2≠3 and x5≠4).

From 432 examples, 122 were selected randomly, and among them there were 6 misclassifications.

The MLP network has one hidden layer including 3 hidden units.

	hidden 1	hidden 2	hidden 3
x1=1	0.0059099235	0.9472996	-0.73383844
x1=2	1.8466698	-0.26342666	-0.67896807
x1=3	-1.4575262	-0.2670728	2.1996262
x2=1	-1.8898789	-0.79776365	1.6834232
x2=2	-2.7031727	-1.9993198	-0.96162158
x2=3	4.9484158	3.1095428	0.27180243
x3=1	1.4532402	-1.1256549	-0.73032016
x3=2	-0.63673681	1.2663549	1.969153
x4=1	0.79512382	-1.225943	3.2781074
x4=2	0.40265474	0.76956165	-1.0391884
x4=3	-0.40681547	0.68001312	-1.189911
x5=1	-2.8933601	-0.64486891	1.6797435
x5=2	-1.8767953	-1.052377	0.12152703
x5=3	0.717704	-3.5005422	0.53808266
x5=4	4.6651802	5.0797367	-1.4538313
x6=1	1.1653591	0.11742916	0.23057088
x6=2	-0.30767989	0.12740196	0.64650488
bias	0.7550745	0.025773972	1.1010749

	output
hidden 1	-6.9178467
hidden 2	-5.5448179
hidden 3	4.7419658
bias	3.675133

Table 9. MLP weights for the M_3 problem

The MLP network classifies the training set to an accuracy 98.36%, with the accuracy on the negative class of 98.39%, and on the positive class of 98.36%. The classification accuracy on the complete set of 432 examples is 97.17% with the 94.3% on the negative and class and 100% on the positive.

The FSD thresholds for the negative class is set to 100 and that for the positive class is the default value of 1.0 for the same reasoning given for the M_1 domain. We will only show those rule extraction situations for the positive class as follows. As there is noise in the training set, the expected accuracy in the rule validation operation is set to 0.75.

The extraction and explanation is illustrated in Appendix B. Applying the rule set on the training set, we have the following results: accuracy for the positive class of 90%; accuracy for the negative class of 96.8%; overall accuracy 93.4%. On the complete test set, we have: accuracy for the positive class of 87.7%; accuracy for the negative class of 100% and overall accuracy of 93.5%.

5.3 Breast cancer data set

The application task described in this section is the diagnosis of cancer from fine needle aspirates of the breast. We first provide a brief synopsis of research detailed [24,28].

Breast cancer is a common disease affecting around 22 000 women yearly in England and Wales and is the commonest cause of death in the 35-55 year age group of the same population [16]. The primary method of diagnosis is through microscopic examination by a pathologist of cytology slides derived from fine needle aspiration of breast lesions. The acquisition of the necessary diagnostic expertise for this task is a relatively slow process. (A trainee pathologist in the UK requires at least five years study and experience before being allowed to sit the final professional pathology examinations for membership of the Royal College of Pathologists.) There is thus scope for a decision-making tool for this domain, both to assist in training junior pathologists and to improve their performance.

There are three performance indicators widely used in medical decision making. Sensitivity: defined as the ratio of the number of correct positive diagnoses to the number of positive outcomes. Specificity: defined as the ratio of the number of correct negative diagnoses to the number of negative outcomes. Accuracy: defined as the ratio of the number of correct diagnoses to the total number of patients. In this domain, taking malignancy as a "positive" outcome, specificity must be high (approaching 100%), to avoid unnecessary surgery being carried out. This is because a patient diagnosed as having a malignant tumour will go straight for surgery, whereas a benign diagnosis will be referred to the surgeon and the patient will be reviewed. Thus some false negatives are acceptable whilst false positives are not.

The data used in this study consisted of 413 patient records each comprising ten binary-valued features recorded from observation of breast tissue samples by an expert pathologist (of Consultant status with 10 years experience in the field). The samples were taken from patients referred to the Royal Hallamshire Hospital, Sheffield, UK with symptomatic breast lesions between 1989-1993. The distribution of categories within the data was fairly even – 53% of cases were malignant, 47% benign. Although the data-set was claimed to have predictive value for the diagnosis task [12,15], there are only 92 distinct input vectors in the set and 12 of them correspond to conflicting conclusions appearing in many places. Thirty eight of the 92 distinct input vectors occur more than once in the data-set. Among the conflicting occurrences, 17 patterns belong to the minority class while most patterns having the same input vectors do not. An additional data set was also employed with 50 records blinded to the true diagnosis.

The ten data features used in the study are all claimed to have diagnostic value for the task. We give the definitions of each feature, together with the abbreviations by which they will be referred to throughout the remainder of this section in Appendix C.

Two hundred and thirteen records are randomly selected as the training set and the remaining 200 records as the test set. An MLP having 3 hidden units is found to perform best in this domain. The classification results of the MLP are shown in Table 10.

Data set	**On the training set (%)**			**On the test set (%)**		
Performance	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy
By MLP	95.96	95.61	95.79	93.75	95.19	94.47

Table 10. MLP classification results on the Breast Cancer data set.

Extracting rules using the default FSD threshold of 1.0, we have the result:

Data set	**On the training set (%)**			**On the test set (%)**		
Performance	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy
By rules	91.23	91.92	91.55	87.5	89.58	88.5

Table 11. Classification results on the Breast Cancer data set using the extracted rules

The rules are shown in Appendix D where ~Malignant indicates the conclusion "benign".

Observing the individual rules, we find they are all acceptably accurate in terms of the cover ranges. They are mostly, however, fairly long, with the average length (the number of the premises) 4.692, nearly half of the input dimension. The rules are therefore not general enough. When the rules are applied to the test set (200 patterns total, 66 distinct patterns) for classification, 7 distinct patterns are not covered by any rules, which causes a reduction of the accuracy.

Now we take the generalisation step (step 5 in Rule_extraction(e)), the performance is improved thus:

Data set	**On the training set (%)**			**On the test set (%)**		
Performance	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy
By rules	95.61	93.94	94.84	95.19	90.62	93

Table 12. Classification results on the Breast Cancer data set using the generalised extracted rules

The generalised rule set is:

IF (~ICL(0.22), ~3D(0.12), ~Foamy(0.13), ~Size(0.52)) THEN (~Malignant); (0.9677)

Matched Cover Range=60; Conflict Cover Range=2

IF (~3D(0.16), ~Naked(0.24), ~Foamy(0.32), ~Size(0.28)) THEN (~Malignant); (0.9000)

Matched Cover Range=54; Conflict Cover Range=6

IF (~ICL(0.29), ~3D(0.11), ~Nucleoli(0.29), ~Size(0.30)) THEN (~Malignant); (0.9674)

Matched Cover Range=89; Conflict Cover Range=3

Matched Cover Range=0; Conflict Cover Range=1

IF (~ICL(0.29), ~Nucleoli(0.40), ~Size(0.31)) THEN (~Malignant); (0.9479)

Matched Cover Range=91; Conflict Cover Range=5

IF (~ICL(0.23), ~3D(0.19), ~Naked(0.16), ~Size(0.43)) THEN (~Malignant); (0.9651)

Matched Cover Range=83; Conflict Cover Range=3

IF (ICL(0.88), ~Apocrine(0.12)) THEN (Malignant); (0.9848)

Matched Cover Range=65; Conflict Cover Range=1

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

IF (Nucleoli(0.85), ~Apocrine(0.15)) THEN (Malignant); (0.9286)

Matched Cover Range=91; Conflict Cover Range=7

IF (Size(0.89), ~Apocrine(0.11)) THEN (Malignant); (0.9697)

Matched Cover Range=96; Conflict Cover Range=3

The total number of rules is nine, and the average length is three. Applying these rules to the test set, all patterns are covered by some rules.

Most rules were confirmed as being correct by the third author who is a consultant pathologist having more than 10 years experience in the field. For example, the last rule tells us that Size (increased nuclear size) is usually only found in malignant cells with the exception of benign epithelial cells showing apocrine change.

Next, we used the latter rule set to classify and explain the additional clinical records. In the 49 records, there are only 17 distinct cases. GR2 explains them as shown in

Appendix D. We attach a pathologist's comment to each case according to the clinical diagnoses and analyses.

6. Conclusion

GR2 realises its explanation functionality both at the *domain* level via the process of rule extraction from a trained neural network, and at the *case* level via reasoning from extracted rules.

Rule extraction is an important technology in the field of neural networks, and there are still some difficulties to overcome. In GR2 both white-box and black-box criteria are used to facilitate the rule extraction process. The resultant general rules represent both qualitative and quantitative knowledge.

At case level, GR2 provides information regarding the questions of "how" and "to what extent". It also provides causal knowledge that reflects the relative importance of the elements in the given input vectors. Special facilities are provided to explain cases in situations such as conflict matching, non-matching and the so-called "failures" of traditional rule-based systems.

The explanation functions in GR2 are at a preliminary stage compared with those of fully-fledged expert systems. For instance, there is neither a real dialogue component nor a natural language interpreter for "social" interaction between users and the computer.

Acknowledgement

This research work has been supported by the Science and Engineering Research Council, UK, Grant Number GR/J29916.

7. References

[1] Adams, JB, A probability model of medical reasoning and the MYCIN model. *Mathematical Bioscience*, **32,** 177-186, 1976.

[2] Rumelhart, D, Hinton, G, and Williams, R, Learning representations by back-propagating errors. *Nature*, **323,** 533-536, 1986.

[3] Rousset, MC and Safar, B, Negative and positive explanations in expert systems. *Applied Artificial Intelligence*, **1,** 25-28, 1987.

[4] DeJong, G, An introduction to explanation-based learning. In *Exploring artificial intelligence*, Shrobe, HE, Ed. Morgan Kaufmann, 1988.

[5] Saito, K and Nakano, R, Medical diagnostic expert system based on PDP model. 255-262, 1988, Proceedings of the IEEE International Conference on Neural Networks. San Diego, CA.

[6] Nilbert, N, Explanation and dialogue. *The Knowledge Engineering Review*, **4,** 235-147, 1989.

[7] Saito, K and Nakano, R, Rule extraction from facts and neural networks. 379-382, 1990, Proceedings of the International Neural Network Conference.

[8] Drucker, H and LeCun, Y, Double backpropagation increasing generalisation performance. 145-150, 1991, Proceedings of the International Joint Conference on Neural Networks. Seattle.

[9] Rich and Knight, Artificial Intelligence. McGraw Hill Inc, 1991.

[10] Shavlik, JW, Mooney, RJ, and Towell, GG, Symbolic and neural learning algorithms: an experimental comparison. *Machine Learning*, **6,** 111-143, 1991.

[11] Thrun, SB, Bala, J, Bloedorn, E, Bratko, I, Cestnlk, B, and Cheng, J, The Monk's problem: a performance comparison of different learning algorithms. Research Report: CMU-CE-91-197, 1991, Carnegie Mellon University, Pittsburg, USA.

[12] Trott, PA, Aspiration cytodiagnosis of the breast. *Diagnostic Oncology*, **1,** 79-87, 1991.

[13] Wada, Y and Kawato, M, Estimation of generalisation capapbility by combination of new information criterion and cross validation. 1-6, 1991, Proceedings of the International Joint Conference on Neural Networks. Seattle, USA.

[14] Whitley, D and Karunanthi, N, Generalisation in feedforward neural networks. 77-82, 1991, Proceedings of the International Joint Conference on Neural Networks. Seattle, USA.

[15] Koss, LG, Diagnostic cytology and its histopathologic basis. 1992.

[16] Underwood, JCE, Tumours: benign and malignant. In *General and Systematic Pathology*, Underwood, JCE, Ed. Edinburgh, Churchill Livingstone, 223-246, 1992.

[17] Quinlan, JR, C4.5: Programs for Machine Learning. Morgan Kauffman, 1993.

[18] Thrun, SB, Extracting provably correct rules from artificial neural networks. Research Report: IAI-TR-93-5, 1993, Institut fur Informatik, Bonn.

[19] Towell, GG and Shavlik, JW, Extracting refined rules from knowledge-based neural networks. *Machine Learning*, **13,** 71-101, 1993.

[20] Craven, M and Shavlik, JW, Using sampling and queries to extract rules from trained neural networks. 1994, Proceedings of the Eleventh International Conference on Machine Learning. San Francisco, CA.

[21] Fu, LM, Neural networks in computer intelligence. McGraw-Hill, 1994.

[22] Andrews, R, Diederich, J, and Tickle, AB, A survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, **8,** 373-389, 1995.

[23] Dietterich, TG, Hild, H, and Bakiri, GA, A comparison of ID3 and backpropagation for English text-to-speech mapping. *Machine Learning*, **18,** 51-80, 1995.

[24] Downs, J, Harrison, RF, and Cross, SS, A neural network decision support tool for the diagnosis of breast cancer. In *Hybrid Problems, Hybrid Solutions*, Hallam, J, Ed. Amsterdam, IOS Press, 51-60, 1995.

[25] Harrison, RF, Ma, Z, and Kennedy, RL, A general method for the discovery and use of rules induced by feedforward artificial neural networks. Research Report: AC&SE RR 607, 1995, The University of Sheffield, Sheffield, UK.

[26] Ma, Z and Harrison, RF, An artificial expert: knowledge acquisition via neural networks. 178-181, 1995, Proceedings of the IEEE International Conference on Neural Information Processing. Beijing.

[27] Ma, Z and Harrison, RF, GR2: a hybrid knowledge-based system using general rules. 488-493, 1995, Proceedings of the International Joint Conference on Artificial Intelligence. Montreal.

[28] Downs, J, Harrison, RF, Kennedy, RL, and Cross, SS, Application of the fuzzy ARTMAP neural network model to medical pattern classification tasks. *Artificial Intelligence in Medicine*, **8,** 403-428, 1996.

[29] Setiono, R, Extracting rules from pruned neural networks for breast cancer diagnosis. *Artificial Intelligence in Medicine*, **8,** 37-51, 1996.

Appendix A: Case-by-case explanations for the "go-to-beach?" example

The input vector is <0 0 0 0>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

hot=0: 0.6278

dry=0: 0.3722

windy=0: 0.3776

There are 2 rules matched:

IF (~hot(0.62), ~windy(0.38)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~hot(0.63), ~dry(0.37)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/**************************************/

The input vector 2 is <0 0 0 1>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

hot=0: 0.6278

dry=0: 0.4518

windy=1: 0.5482

There are 2 rules matched:

IF (~dry(0.45), windy(0.55)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~hot(0.63), ~dry(0.37)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/**************************************/

The input vector 3 is <0 0 1 0>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

hot=0: 0.6224

windy=0: 0.3776

There is 1 rule matched:

IF (~hot(0.62), ~windy(0.38)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/***************************************/

The input vector 4 is <0 0 1 1>. The defined output is 1.

The result of reasoning by rules is 1

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=0: 0.2585

hot=0: 0.1952

dry=1: 0.3048

windy=1: 0.2415

There is 1 rule matched:

IF (~weekday(0.26), ~hot(0.20), dry(0.30), windy(0.24)) THEN (Go-to-beach); (1.0)

Matched Cover Range=1; Conflict Cover Range=0

The conclusion is Go-to-beach by direct reasoning.

/***************************************/

The input vector 5 is <0 1 0 0>. The defined output is 1.

The result of reasoning by rules is 1

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=0: 0.3038

hot=1: 0.3581

windy=0: 0.3381

There is 1 rule matched:

IF (~weekday(0.30), hot(0.36), ~windy(0.34)) THEN (Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

The conclusion is Go-to-beach by direct reasoning.

/***************************************/

The input vector 6 is <0 1 0 1>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=0: 0.3333

hot=1: 0.2437

dry=0: 0.4518

windy=1: 0.5482

There are 2 rules matched:

IF (~weekday(0.33), hot(0.24), windy(0.42)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

IF (~dry(0.45), windy(0.55)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/***************************************/

The input vector 7 is <0 1 1 0>. The defined output is 1.

The result of reasoning by rules is 1

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=0: 0.3038

hot=1: 0.4014

dry=1: 0.3629

windy=0: 0.3381

There are 2 rules matched:

IF (hot(0.40), dry(0.36), ~windy(0.24)) THEN (Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

IF (~weekday(0.30), hot(0.36), ~windy(0.34)) THEN (Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

The conclusion is Go-to-beach by direct reasoning.

/***************************************/

The input vector 8 is <0 1 1 1>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=0: 0.3333

hot=1: 0.2437

windy=1: 0.423

There is 1 rule matched:

IF (~weekday(0.33), hot(0.24), windy(0.42)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/***************************************/

The input vector 9 is <1 0 0 0>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=1: 0.4426

hot=0: 0.6823

dry=0: 0.5574

windy=0: 0.3776

There are 4 rules matched:

IF (weekday(0.44), ~dry(0.56)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (weekday(0.32), ~hot(0.68)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~hot(0.62), ~windy(0.38)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~hot(0.63), ~dry(0.37)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/**************************************/

The input vector 10 is <1 0 0 1>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=1: 0.4426

hot=0: 0.6823

dry=0: 0.5574

windy=1: 0.5482

There are 4 rules matched:

IF (weekday(0.44), ~dry(0.56)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (weekday(0.32), ~hot(0.68)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~dry(0.45), windy(0.55)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~hot(0.63), ~dry(0.37)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/**************************************/

The input vector 11 is <1 0 1 0>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=1: 0.3177

hot=0: 0.6823

windy=0: 0.3776

There are 2 rules matched:

IF (weekday(0.32), ~hot(0.68)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~hot(0.62), ~windy(0.38)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/**************************************/

The input vector 12 is <1 0 1 1>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=1: 0.3177

hot=0: 0.6823

There is 1 rule matched:

IF (weekday(0.32), ~hot(0.68)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/**************************************/

The input vector 13 is <1 1 0 0>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=1: 0.4426

dry=0: 0.5574

There is 1 rule matched:

IF (weekday(0.44), ~dry(0.56)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/**************************************/

The input vector 14 is <1 1 0 1>. The defined output is 0.

The result of reasoning by rules is 0

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=1: 0.4426

dry=0: 0.5574

windy=1: 0.5482

There are 2 rules matched:

IF (weekday(0.44), ~dry(0.56)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

IF (~dry(0.45), windy(0.55)) THEN (~Go-to-beach); (1.0)

Matched Cover Range=4; Conflict Cover Range=0

The conclusion is Not Go-to-beach by direct reasoning.

/**************************************/

The input vector 15 is <1 1 1 0>. The defined output is 1.

The result of reasoning by rules is 1

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=1: 0.2678

hot=1: 0.4014

dry=1: 0.3845

windy=0: 0.2358

There are 2 rules matched:

IF (weekday(0.27), hot(0.35), dry(0.38)) THEN (Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

IF (hot(0.40), dry(0.36), ~windy(0.24)) THEN (Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

The conclusion is Go-to-beach by direct reasoning.

/**************************************/

The input vector 16 is <1 1 1 1>. The defined output is 1.

The result of reasoning by rules is 1

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

weekday=1: 0.2678

hot=1: 0.3476

dry=1: 0.3845

There is 1 rule matched:

IF (weekday(0.27), hot(0.35), dry(0.38)) THEN (Go-to-beach); (1.0)

Matched Cover Range=2; Conflict Cover Range=0

The conclusion is Go-to-beach by direct reasoning.

/**************************************/

Appendix B: Case-by-case explanations for the Monk's problem

Monk 1

pattern[0]: 1 0 0 1 0 0 1 0 1 0 0 0 0 1 0 1 0, 0.9589;

Non-PDS: 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1;

FSD: 0.38 0.37 0.0 0.46 0.33 0.13 0.05 0.05 0.16 0.1 0.06 0 0.08 0.14 0.06 0.06 0.06;

IF (x1=1(0.38), x2=1(0.46)) THEN (monk); (1.0)

Matched Cover Range=9; Conflict Cover Range=0

Valid rule

Patterns 1-8 are covered by the extracted rules, so that they are passed.

pattern[9]: 1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 1, 1;

Non-PDS: 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0;

FSD: 0.05 0.0 0.04 0.01 0.04 0.03 0.07 0.07 0.15 0.09 0.06 0.56 0.09 0.09 0.37 0.09 0.09;

IF (x5=1(0.56)) THEN (monk);   (1.0)

Matched Cover Range=29; Conflict Cover Range=0

Valid rule.

There are some patterns of the negative class, from which no rules are extracted owing to the high FSD threshold previously set.

pattern[61]: 0 1 0 0 1 0 1 0 1 0 0 0 1 0 0 1 0, 0.9581

Non-PDS: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1,

0.3765 0.3894 0.01285 0.07001 0.1312 0.06118 0.03134 0.03134 0.09155 0.03697 0.05458 0 0.1273 0.0976 0.02969 0.0376 0.0376,

IF (x1=2(0.39)) THEN (monk);   (0.5238)

Matched Cover Range=22; Conflict Cover Range=20

Invalid rule discarded.

pattern[67]: 0 1 0 0 1 0 1 0 0 0 1 0 1 0 0 0 1, 0.9508;

Non-PDS: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1;

0.1732 0.2316 0.05837 0.1321 0.2194 0.08728 0.104 0.104 0.03209 0.06307 0.09516 0 0.1325 0.02897 0.1035 0.061 0.061;

IF (x1=2(0.23), x2=2(0.22)) THEN (monk); (1.0000);

Matched Cover Range=15; Conflict Cover Range=0;

Valid rule

pattern[109]: 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 1, 0.9566

Non-PDS: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1,

FSD: 0.15 0.06 0.21 0.03 0.33 0.36 0.04 0.04 0.16 0.06 0.09 0 0.03 0.02 0.05 0.055 0.055,

IF (x1=3(0.21), x2=3(0.36)) THEN (monk); (1.0)

Matched Cover Range=17; Conflict Cover Range=0

EXPLANATION for Monk1

The input vector 1 is <1 0 0 1 0 0 1 0 1 0 0 0 0 1 0 1 0>. The defined output is 1.

The result of reasoning by rules is 1

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

x1=1: 0.4482

x2=1: 0.5518

There is 1 rule matched:

IF (x1=1(0.45), x2=1(0.55)) THEN (monk); (1.0)

Matched Cover Range=9; Conflict Cover Range=0

The conclusion is monk by direct reasoning.

/***************************************/

The input vector 9 is <1 0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 1>. The defined output is 1.

The result of reasoning by rules is 1

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

x1=1: 0.4482

x2=1: 0.5518

x5=1: 1

There are 2 rules matched:

IF (x5=1(1.00)) THEN (monk); (1.0)

Matched Cover Range=29; Conflict Cover Range=0

IF (x1=1(0.45), x2=1(0.55)) THEN (monk); (1.0)

Matched Cover Range=9; Conflict Cover Range=0

The conclusion is monk by direct reasoning.

/***************************************/

The input vector 10 is <1 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0 1>. The defined output is 1.

The result of reasoning by rules is 1

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

x5=1: 1

There is 1 rule matched:

IF (x5=1(1.00)) THEN (monk); (1.0)

Matched Cover Range=29; Conflict Cover Range=0

The conclusion is monk by direct reasoning.

/***************************************/

The input vector 11 is <1 0 0 0 1 0 1 0 1 0 0 0 1 0 0 1 0>. The defined output is 0.

0

with a certainty degree 0.

There is no rule directly matching this input vector. However,

the exceptional reasoning enables since there is no extracted rules on negative class

The conclusion is Not monk

/**************************************/

The input vector 67 is <0 1 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1>. The defined output is 1.

The result of reasoning by rules is 1

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

x1=2: 0.6578

x2=2: 0.3422

x5=1: 1

There are 2 rules matched:

IF (x1=2(0.66), x2=2(0.34)) THEN (monk); (1.0)

Matched Cover Range=15; Conflict Cover Range=0

IF (x5=1(1.00)) THEN (monk); (1.0)

Matched Cover Range=29; Conflict Cover Range=

/**************************************/

The input vector 113 is <0 0 1 0 0 1 1 0 0 0 1 1 0 0 0 0 1>. The defined output is 1.

The result of reasoning by rules is 1

with a certainty degree 1

The contribution degrees of the conditions to the conclusion:

x1=3: 0.3607

x2=3: 0.6393

x5=1: 1

There are 2 rules matched:

IF (x1=3(0.36), x2=3(0.64)) THEN (monk); (1.0)

Matched Cover Range=17; Conflict Cover Range=0

IF (x5=1(1.00)) THEN (monk); (1.0)

Matched Cover Range=29; Conflict Cover Range=0

The conclusion is monk by direct reasoning.

/**************************************/

Monk 3

pattern[0]: 1 0 0 1 0 0 1 0 1 0 0 1 0 0 0 0 1, 0.9993
Non-PDS: 0 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 0,
FSD: 0.08407 0.06208 0.022 0.1093 0.01466 0.09464 0.1183 0.1183 0.08087 0.06394 0.01693 0.2288 0.008722 0.009452 0.2106 0.07673 0.07673,
IF (x5=1(0.23)) THEN (monk); (0.6875)
Matched Cover Range=22; Conflict Cover Range=10
Invalid rule.
pattern[1]: 1 0 0 1 0 0 1 0 1 0 0 0 1 0 0 1 0, 0.9819
Non-PDS: 0 1 0 1 0 1 0 0 1 1 1 0 1 1 1 0 0,

FSD: 0.08991 0.06872 0.02119 0.4474 0.02416 0.4232 0.06449 0.06449 0.0709 0.03746 0.03344 0.0141 0.4582 0.0134 0.4307 0.02527 0.02527,
IF (x2=1(0.45), x5=2(0.46)) THEN (monk); (1.0000)
Matched Cover Range=9; Conflict Cover Range=0
Valid rule
pattern[3]: 1 0 0 1 0 0 1 0 1 0 0 0 0 1 0 1 0, 0.8561
Non-PDS: 0 1 0 1 0 1 0 0 1 1 1 0 0 0 1 0 0,
FSD: 0.0847 0.06872 0.01598 0.2273 0.05925 0.1681 0.03011 0.03011 0.1405 0.06852 0.07202 0.0141 0.04785 0.4926 0.4307 0.03002 0.03002,
IF (x2=1(0.23), x5=3(0.49)) THEN (monk); (0.8333)
Matched Cover Range=5; Conflict Cover Range=1
Valid rule
pattern[5]: 1 0 0 1 0 0 1 0 0 1 0 1 0 0 0 1 0, 0.986
Non-PDS: 0 1 0 1 1 1 0 0 0 0 1 1 1 1 1 0 1,
FSD: 0.09063 0.07275 0.01788 0.2226 0.0527 0.1699 0.06521 0.06521 0.06715 0.1351 0.0679 0.2785 0.01868 0.0545 0.2053 0.06448 0.06448,
0.09063 0.07275 0.01788 0.2226 0.0527 0.1699 0.06521 0.06521 0.06715 0.1351 0.0679 0.2785 0.01868 0.0545 0.2053 0.06448 0.06448,
IF (x2=1(0.22), x5=1(0.28)) THEN (monk); (1.0000)
Matched Cover Range=12; Conflict Cover Range=0
Valid rule
pattern[17]: 1 0 0 0 1 0 1 0 1 0 0 0 0 1 0 1 0, 0.8696
Non-PDS: 0 1 0 0 1 1 0 0 1 1 1 0 0 0 1 0 0,
FSD: 0.1237 0.07402 0.04973 0.04931 0.2174 0.1681 0.02537 0.02537 0.4587 0.3165 0.1422 0.0141 0.05251 0.2624 0.1958 0.02867 0.02867,
IF (x2=2(0.22), x4=1(0.46), x5=3(0.26)) THEN (monk); (1.0000)
Matched Cover Range=4; Conflict Cover Range=0
Valid rule
pattern[18]: 1 0 0 0 1 0 1 0 0 1 0 0 1 0 0 1 0, 0.8305
Non-PDS: 0 1 1 1 0 1 0 1 1 0 1 1 0 1 1 0 1,
FSD: 0.3762 0.3591 0.01716 0.02023 0.1941 0.1739 0.0207 0.0207 0.0658 0.1356 0.0698 0.01267 0.4455 0.3144 0.1183 0.0259 0.0259,
IF (x1=1(0.38), x2=2(0.19), x5=2(0.45)) THEN (monk); (1.0000)
Matched Cover Range=8; Conflict Cover Range=0
Valid rule
pattern[21]: 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 1 0, 0.9855
Non-PDS: 0 1 1 1 0 1 0 1 1 1 0 0 1 1 1 0 1,
FSD: 0.157 0.08674 0.07028 0.01422 0.4549 0.4407 0.03809 0.03809 0.03269 0.0599 0.09259 0.5054 0.01923 0.1315 0.3547 0.01745 0.01745,
IF (x2=2(0.45), x5=1(0.51)) THEN (monk); (1.0000)
Matched Cover Range=10; Conflict Cover Range=0
Valid rule
pattern[25]: 1 0 0 0 1 0 1 0 0 0 1 0 0 1 0 0 1, 0.8796
Non-PDS: 0 1 1 1 0 1 0 1 1 1 0 1 1 0 1 1 0,
FSD: 0.4189 0.3957 0.02326 0.01548 0.1399 0.1244 0.02554 0.02554 0.01735 0.1024 0.1198 0.00746 0.008808 0.1312 0.1149 0.1808 0.1808,
IF (x1=1(0.42), x6=2(0.18)) THEN (monk); (0.5909)
Matched Cover Range=13; Conflict Cover Range=9
pattern[65]: 0 1 0 0 1 0 1 0 0 0 1 0 1 0 0 0 1, 0.9378
Non-PDS: 1 0 1 1 0 1 0 1 1 1 0 1 0 1 1 1 0,
FSD: 0.01162 0.06934 0.05772 0.01888 0.148 0.1291 0.02894 0.02894 0.05521 0.09714 0.1524 0.003973 0.7275 0.3484 0.3752 0.1265 0.1265,
IF (x5=2(0.73)) THEN (monk); (0.6774)
Matched Cover Range=21; Conflict Cover Range=10
pattern[69]: 0 1 0 0 1 0 0 1 1 0 0 0 1 0 0 0 1, 0.9994
Non-PDS: 1 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 0,
FSD: 0.06591 0.08316 0.01725 0.0143 0.413 0.3987 0.07339 0.07339 0.06095 0.02885 0.03209 0.05306 0.193 0.01008 0.1298 0.03625 0.03625,

IF (x2=2(0.41), x5=2(0.19)) THEN (monk); (0.9231)
Matched Cover Range=12; Conflict Cover Range=1
Valid rule
pattern[71]: 0 1 0 0 1 0 0 1 0 1 0 0 0 1 0 1 0, 0.807
Non-PDS: 1 0 0 0 1 1 1 1 0 0 1 0 0 0 1 0 0,
FSD: 0.05873 0.07472 0.01598 0.04868 0.4093 0.3606 0.155 0.155 0.01285 0.1259 0.113 0.0141 0.05377 0.1798 0.1119 0.03738 0.03738,
IF (x2=2(0.41), x5=3(0.18)) THEN (monk); (0.7500)
Matched Cover Range=9; Conflict Cover Range=3
pattern[74]: 0 1 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0, 0.7531
Non-PDS: 1 0 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0,
FSD: 0.1615 0.2332 0.07171 0.05729 0.06921 0.1265 0.1269 0.1269 0.6141 0.1906 0.4235 0.07441 0.1129 0.392 0.2047 0.08646 0.08646,
IF (x1=2(0.23), x4=1(0.61), x5=3(0.39)) THEN (monk); (1.0000)
Matched Cover Range=2; Conflict Cover Range=0
Valid rule
pattern[100]: 0 0 1 0 1 0 1 0 0 1 0 0 0 1 0 1 0, 0.9673
Non-PDS: 1 1 1 0 1 1 0 0 0 0 1 0 0 0 1 0 0,
FSD: 0.3218 0.1177 0.4395 0.01144 0.09531 0.08387 0.02402 0.02402 0.05029 0.1395 0.08926 0.0133 0.05377 0.1543 0.0872 0.04044 0.04044,
IF (x1=3(0.44)) THEN (monk); (0.5000)
Matched Cover Range=17; Conflict Cover Range=17
Invalid rule
pattern[108]: 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 0 1, 0.7688
Non-PDS: 1 1 1 0 0 0 0 0 1 1 1 0 0 0 1 1 1,
FSD: 0.07983 0.08461 0.1644 0.02006 0.01941 0.03947 0.1713 0.1713 0.3149 0.1878 0.1271 0.06445 0.06848 0.2705 0.1376 0.09862 0.09862,
IF (x3=1(0.17), x4=1(0.31), x5=3(0.27)) THEN (monk); (0.8333)
Matched Cover Range=5; Conflict Cover Range=1
Valid rule
IF (x2=1(0.49), x5=2(0.51)) THEN (monk); (1.0000)
Matched Cover Range=9; Conflict Cover Range=0
Valid rule
IF (x2=1(0.32), x5=3(0.68)) THEN (monk); (0.8333)
Matched Cover Range=5; Conflict Cover Range=1
Valid rule
IF (x2=1(0.44), x5=1(0.56)) THEN (monk); (1.0000)
Matched Cover Range=12; Conflict Cover Range=0
Valid rule
IF (x2=2(0.23), x4=1(0.49), x5=3(0.28)) THEN (monk); (1.0000)
Matched Cover Range=4; Conflict Cover Range=0
Valid rule
IF (x2=2(0.47), x5=1(0.53)) THEN (monk); (1.0000)
Matched Cover Range=10; Conflict Cover Range=0
Valid rule
IF (x2=2(0.68), x5=2(0.32)) THEN (monk); (0.9231)
Matched Cover Range=12; Conflict Cover Range=1
Valid rule
IF (x1=2(0.19), x4=1(0.50), x5=3(0.32)) THEN (monk); (1.0000)
Matched Cover Range=2; Conflict Cover Range=0
Valid rule
IF (x3=1(0.23), x4=1(0.42), x5=3(0.36)) THEN (monk); (0.8333)
Matched Cover Range=5; Conflict Cover Range=1
Valid rule
A part of explanations to Monk3:
The input vector 1 is <1 0 0 1 0 0 1 0 1 0 0 1 0 0 0 1 0>. The defined output is 1.
with a certainty degree 0.9677.
There is no rule directly matching this input vector. However,

the exceptional reasoning enables since there is no extracted rules
on positive class
The conclusion is monk
/***************************************/
The input vector 7 is <1 0 0 1 0 0 1 0 1 0 0 0 0 0 1 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x5=4: 1
There is 1 rule matched:
IF (x5=4(1.00)) THEN (~monk); (1.0)
Matched Cover Range=31; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/***************************************/
The input vector 97 is <1 0 0 0 0 1 1 0 1 0 0 1 0 0 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=1: 0.1605
x2=3: 0.8395
x5=1: 0.4252
There are 2 rules matched:
IF (x2=3(0.57), x5=1(0.43)) THEN (~monk); (1.0)
Matched Cover Range=10; Conflict Cover Range=0
IF (x1=1(0.16), x2=3(0.84)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/***************************************/
The input vector 99 is <1 0 0 0 0 1 1 0 1 0 0 0 1 0 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=1: 0.1605
x2=3: 0.8395
x5=2: 0.4191
There are 2 rules matched:
IF (x2=3(0.58), x5=2(0.42)) THEN (~monk); (1.0)
Matched Cover Range=9; Conflict Cover Range=0
IF (x1=1(0.16), x2=3(0.84)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/***************************************/
The input vector 101 is <1 0 0 0 0 1 1 0 1 0 0 0 0 1 0 1 0>. The defined output is 1.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=1: 0.1605
x2=3: 0.8395
There is 1 rule matched:
IF (x1=1(0.16), x2=3(0.84)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
This result is different from the demain definition.
/***************************************/
The input vector 103 is <1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:

x1=1: 0.1605
x2=3: 0.8395
x5=4: 1

There are 2 rules matched:
IF (x1=1(0.16), x2=3(0.84)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
IF (x5=4(1.00)) THEN (~monk); (1.0)
Matched Cover Range=31; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 105 is <1 0 0 0 0 1 1 0 0 1 0 1 0 0 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:

x1=1: 0.1605
x2=3: 0.8395
x4=2: 0.2142
x5=1: 0.4252

There are 3 rules matched:
IF (x2=3(0.57), x5=1(0.43)) THEN (~monk); (1.0)
Matched Cover Range=10; Conflict Cover Range=0
IF (x2=3(0.79), x4=2(0.21)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
IF (x1=1(0.16), x2=3(0.84)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 109 is <1 0 0 0 0 1 1 0 0 1 0 0 0 1 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:

x1=1: 0.1605
x2=3: 0.8395
x4=2: 0.2142

There are 2 rules matched:
IF (x2=3(0.79), x4=2(0.21)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
IF (x1=1(0.16), x2=3(0.84)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 111 is <1 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:

x1=1: 0.1605
x2=3: 0.8395
x4=2: 0.2142
x5=4: 1

There are 3 rules matched:
IF (x2=3(0.79), x4=2(0.21)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
IF (x1=1(0.16), x2=3(0.84)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
IF (x5=4(1.00)) THEN (~monk); (1.0)
Matched Cover Range=31; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/

The input vector 113 is <1 0 0 0 0 1 1 0 0 0 1 1 0 0 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=1: 0.1605
x2=3: 0.8395
x4=3: 0.3355
x5=1: 0.4252
There are 3 rules matched:
IF (x2=3(0.57), x5=1(0.43)) THEN (~monk); (1.0)
Matched Cover Range=10; Conflict Cover Range=0
IF (x2=3(0.66), x4=3(0.34)) THEN (~monk); (1.0)
Matched Cover Range=14; Conflict Cover Range=0
IF (x1=1(0.16), x2=3(0.84)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 117 is <1 0 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=1: 0.1605
x2=3: 0.8395
x4=3: 0.3355
There are 2 rules matched:
IF (x2=3(0.66), x4=3(0.34)) THEN (~monk); (1.0)
Matched Cover Range=14; Conflict Cover Range=0
IF (x1=1(0.16), x2=3(0.84)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 155 is <0 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0>. The defined output is 1.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=2: 0.4081
x3=1: 0.09282
x4=2: 0.1531
x5=2: 0.2043
x6=1: 0.1417
There is 1 rule matched:
IF (x1=2(0.41), x3=1(0.09), x4=2(0.15), x5=2(0.20), x6=1(0.14)) THEN (~monk); (1.0)
Matched Cover Range=2; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
This result is different from the demain definition.
/**************************************/
The input vector 165 is <0 1 0 1 0 0 1 0 0 0 1 0 0 1 0 1 0>. The defined output is 1.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=2: 0.2256
x4=3: 0.2073
x5=3: 0.398
x6=1: 0.1691
There is 1 rule matched:
IF (x1=2(0.23), x4=3(0.21), x5=3(0.40), x6=1(0.17)) THEN (~monk); (1.0)
Matched Cover Range=2; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.

This result is different from the demain definition
/*************************************/
The input vector 213 is <0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 1 0>. The defined output is 1.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=2: 0.252
x2=2: 0.1825
x3=1: 0.06042
x4=3: 0.2156
x5=3: 0.398
x6=1: 0.1691
There are 2 rules matched:
IF (x1=2(0.25), x2=2(0.18), x3=1(0.06), x4=3(0.22), x5=3(0.29)) THEN (~monk); (1.0)
Matched Cover Range=2; Conflict Cover Range=0
IF (x1=2(0.23), x4=3(0.21), x5=3(0.40), x6=1(0.17)) THEN (~monk); (1.0)
Matched Cover Range=2; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
This result is different from the demain definition.
/*************************************/
The input vector 214 is <0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 0 1>. The defined output is 1.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=2: 0.252
x2=2: 0.1825
x3=1: 0.06042
x4=3: 0.2156
x5=3: 0.2895
There is 1 rule matched:
IF (x1=2(0.25), x2=2(0.18), x3=1(0.06), x4=3(0.22), x5=3(0.29)) THEN (~monk); (1.0)
Matched Cover Range=2; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
This result is different from the domain definition.
/*************************************/
The input vector 241 is <0 1 0 0 0 1 1 0 1 0 0 1 0 0 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x2=3: 0.5748
x5=1: 0.4252
There is 1 rule matched:
IF (x2=3(0.57), x5=1(0.43)) THEN (~monk); (1.0)
Matched Cover Range=10; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/*************************************/
The input vector 249 is <0 1 0 0 0 1 1 0 0 1 0 1 0 0 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x2=3: 0.7858
x4=2: 0.2142
x5=1: 0.4252
There are 2 rules matched:
IF (x2=3(0.57), x5=1(0.43)) THEN (~monk); (1.0)
Matched Cover Range=10; Conflict Cover Range=0
IF (x2=3(0.79), x4=2(0.21)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0

The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 251 is <0 1 0 0 0 1 1 0 0 1 0 0 1 0 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=2: 0.4081
x2=3: 0.7858
x3=1: 0.09282
x4=2: 0.2142
x5=2: 0.4191
x6=1: 0.1417
There are 3 rules matched:
IF (x2=3(0.58), x5=2(0.42)) THEN (~monk); (1.0)
Matched Cover Range=9; Conflict Cover Range=0
IF (x2=3(0.79), x4=2(0.21)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
IF (x1=2(0.41), x3=1(0.09), x4=2(0.15), x5=2(0.20), x6=1(0.14)) THEN (~monk); (1.0)
Matched Cover Range=2; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 258 is <0 1 0 0 0 1 1 0 0 0 1 1 0 0 0 0 1>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x2=3: 0.6645
x4=3: 0.3355
x5=1: 0.4252
There are 2 rules matched:
IF (x2=3(0.57), x5=1(0.43)) THEN (~monk); (1.0)
Matched Cover Range=10; Conflict Cover Range=0
IF (x2=3(0.66), x4=3(0.34)) THEN (~monk); (1.0)
Matched Cover Range=14; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 261 is <0 1 0 0 0 1 1 0 0 0 1 0 0 1 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x1=2: 0.2256
x2=3: 0.6645
x4=3: 0.3355
x5=3: 0.398
x6=1: 0.1691
There are 2 rules matched:
IF (x1=2(0.23), x4=3(0.21), x5=3(0.40), x6=1(0.17)) THEN (~monk); (1.0)
Matched Cover Range=2; Conflict Cover Range=0
IF (x2=3(0.66), x4=3(0.34)) THEN (~monk); (1.0)
Matched Cover Range=14; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 275 is <0 1 0 0 0 1 0 1 0 1 0 0 1 0 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x2=3: 0.7858
x4=2: 0.2142
x5=2: 0.4191

There are 2 rules matched:
IF (x2=3(0.58), x5=2(0.42)) THEN (~monk); (1.0)
Matched Cover Range=9; Conflict Cover Range=0
IF (x2=3(0.79), x4=2(0.21)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 277 is <0 1 0 0 0 1 0 1 0 1 0 0 0 1 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x2=3: 0.7858
x4=2: 0.2142
There is 1 rule matched:
IF (x2=3(0.79), x4=2(0.21)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 279 is <0 1 0 0 0 1 0 1 0 1 0 0 0 0 1 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x2=3: 0.7858
x4=2: 0.2142
x5=4: 1
There are 2 rules matched:
IF (x2=3(0.79), x4=2(0.21)) THEN (~monk); (1.0)
Matched Cover Range=13; Conflict Cover Range=0
IF (x5=4(1.00)) THEN (~monk); (1.0)
Matched Cover Range=31; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 388 is <0 0 1 0 0 1 1 0 1 0 0 0 1 0 0 0 1>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x2=3: 0.5809
x5=2: 0.4191
There is 1 rule matched:
IF (x2=3(0.58), x5=2(0.42)) THEN (~monk); (1.0)
Matched Cover Range=9; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 403 is <0 0 1 0 0 1 1 0 0 0 1 0 1 0 0 1 0>. The defined output is 0.
The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x2=3: 0.6645
x4=3: 0.3355
x5=2: 0.4191
There are 2 rules matched:
IF (x2=3(0.58), x5=2(0.42)) THEN (~monk); (1.0)
Matched Cover Range=9; Conflict Cover Range=0
IF (x2=3(0.66), x4=3(0.34)) THEN (~monk); (1.0)
Matched Cover Range=14; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.
/**************************************/
The input vector 407 is <0 0 1 0 0 1 1 0 0 0 1 0 0 0 1 1 0>. The defined output is 0.

The result of reasoning by rules is 0
with a certainty degree 1
The contribution degrees of the conditions to the conclusion:
x2=3: 0.6645
x4=3: 0.3355
x5=4: 1
There are 2 rules matched:
IF (x2=3(0.66), x4=3(0.34)) THEN (~monk); (1.0)
Matched Cover Range=14; Conflict Cover Range=0
IF (x5=4(1.00)) THEN (~monk); (1.0)
Matched Cover Range=31; Conflict Cover Range=0
The conclusion is Not monk by direct reasoning.

Appendix C: FNAB feature definitions

DYS: True if majority of epithelial cells are dyhesive, false if majority of epithelial cells are in cohesive groups.

ICL: True if intracytoplasmic lumina are present, false if absent.

3D: True if some clusters of epithelial cells are not flat (more than two nuclei thick) and this is not due to artefactual folding, false if all clusters of epithelial cells are flat.

NAKED: True if bipolar "naked" nuclei in background, false if absent.

FOAMY: True if "foamy" macrophages present in background, false if absent.

NUCLEOLI: True if more than three easily visible nucleoli in some epithelial cells, false if three or fewer easily visible nucleoli in epithelial cells.

PLEOMORPH: True if some epithelial cell nuclei with diameters twice that of other epithelial cell nuclei, false if no epithelial cell nuclei twice the diameter of other epithelial cell nuclei.

SIZE: True if some epithelial cells with nuclear diameters at least twice that of lymphocyte nuclei, false if all epithelial cell nuclei with nuclear diameters less than twice that of lymphocyte nuclei.

NECROTIC: True if necrotic epithelial cells present, false if absent.

APOCRINE: True if apocrine change present in all epithelial cells, false if not present in all epithelial cells.

Appendix D: Rules for the breast cancer example

Extracted rules

IF (~ICL(0.16), ~3D(0.09), ~Naked(0.06), ~Foamy(0.10), ~Nucleoli(0.22), ~Size(0.38)) THEN (~Malignant); (0.9630)

Matched Cover Range=52; Conflict Cover Range=2

IF (~3D(0.10), ~Naked(0.16), ~Foamy(0.21), ~Nucleoli(0.19), ~Pleo(0.16), ~Size(0.18)) THEN (~Malignant); (0.8929)

Matched Cover Range=50; Conflict Cover Range=6

IF (~ICL(0.14), ~3D(0.12), ~Naked(0.16), ~Foamy(0.15), ~Pleo(0.16), ~Size(0.27)) THEN (~Malignant); (0.9630)

Matched Cover Range=52; Conflict Cover Range=2

IF (~ICL(0.25), ~3D(0.12), ~Nucleoli(0.38), ~Pleo(0.09), ~Size(0.17)) THEN (~Malignant); (0.9651)

Matched Cover Range=83; Conflict Cover Range=3

IF (~ICL(0.18), ~3D(0.09), ~Naked(0.05), ~Nucleoli(0.21), Pleo(0.16), ~Size(0.31)) THEN (~Malignant); (1.0000)

Matched Cover Range=5; Conflict Cover Range=0

IF (~ICL, ~3D, Naked, Foamy, ~Nucleoli, Pleo, ~Size) THEN (~Malignant); (0.7600)

Matched Cover Range=1; Conflict Cover Range=0

IF (ICL(0.57), Foamy(0.29), ~Necrotic(0.07), ~Apocrine(0.07)) THEN (Malignant); (0.9375)

Matched Cover Range=15; Conflict Cover Range=1

IF (~Foamy(0.16), Pleo(0.28), Size(0.56)) THEN (Malignant); (0.9808)

Matched Cover Range=51; Conflict Cover Range=1

IF (ICL(0.53), Naked(0.38), ~Apocrine(0.09)) THEN (Malignant); (1.0000)

Matched Cover Range=7; Conflict Cover Range=0

IF (3D(0.31), Foamy(0.19), Nucleoli(0.43), ~Necrotic(0.08)) THEN (Malignant); (1.0000)

Matched Cover Range=13; Conflict Cover Range=0

IF (ICL(0.36), 3D(0.29), Nucleoli(0.35)) THEN (Malignant); (1.0000)

Matched Cover Range=28; Conflict Cover Range=0

IF (Nucleoli(0.37), Size(0.63)) THEN (Malignant); (0.9775)

Matched Cover Range=87; Conflict Cover Range=2

Explanation

/*************************************/

The input vector 1 is <0 0 0 0 1 0 0 0 0 0>. MLP results in 0.0219.

The result of reasoning by the rules is 0

with a confidence degree 0.9651

The contribution degrees of the conditions to the conclusion are:

ICL=0 : 0.2752

3D=0: 0.1823

Naked=0: 0.1499

Nucleoli=0: 0.3808

Size=0: 0.4132

There are 2 rules matched:

IF (~ICL(0.23), ~3D(0.19), ~Naked(0.16), ~Size(0.43)) THEN (~Malignant); (0.9651)

Matched Cover Range=83; Conflict Cover Range=3

IF (~ICL(0.29), ~Nucleoli(0.40), ~Size(0.31)) THEN (~Malignant); (0.9479)

Matched Cover Range=91; Conflict Cover Range=5

The conclusion is *Not Malignant* by direct reasoning.

Comment: right conclusion.

/*************************************/

The input vector 2 is <0 0 0 0 0 0 0 0 0 0>. MLP results in 0.0173.

The result of reasoning by the rules is 0

with a confidence degree 0.9677

The contribution degrees of the conditions to the conclusion:

ICL=0: 0.2752

3D=0: 0.1823

Naked=0: 0.2156

Foamy=0: 0.2898

Nucleoli=0: 0.3808

Size=0: 0.5074

There are 4 rules matched:

IF (~ICL(0.23), ~3D(0.19), ~Naked(0.16), ~Size(0.43)) THEN (~Malignant); (0.9651)

Matched Cover Range=83; Conflict Cover Range=3

IF (~ICL(0.29), ~Nucleoli(0.40), ~Size(0.31)) THEN (~Malignant); (0.9479)

Matched Cover Range=91; Conflict Cover Range=5

IF (~3D(0.16), ~Naked(0.24), ~Foamy(0.32), ~Size(0.28)) THEN (~Malignant); (0.9000)

Matched Cover Range=54; Conflict Cover Range=6

IF (~ICL(0.22), ~3D(0.12), ~Foamy(0.13), ~Size(0.52)) THEN (~Malignant); (0.9677)

Matched Cover Range=60; Conflict Cover Range=2

The conclusion is Not Malignant by direct reasoning.

Comment: right conclusion.

/*************************************/

The input vector 3 is <0 0 0 1 0 0 0 0 0 0>. MLP results in 0.0604.

The result of reasoning by the rules is 0

with a confidence degree 0.9677

The contribution degrees of the conditions to the conclusion:

ICL=0: 0.2752

3D=0: 0.1149

Foamy=0: 0.1285

Nucleoli=0: 0.3808

Size=0: 0.5074

There are 2 rules matched:

IF (~ICL(0.29), ~Nucleoli(0.40), ~Size(0.31)) THEN (~Malignant); (0.9479)

Matched Cover Range=91; Conflict Cover Range=5

IF (~ICL(0.22), ~3D(0.12), ~Foamy(0.13), ~Size(0.52)) THEN (~Malignant); (0.9677)

Matched Cover Range=60; Conflict Cover Range=2

The conclusion is Not Malignant by direct reasoning.

Comment: right conclusion.

/**************************************/

The input vector 6 is <0 0 0 0 0 0 1 1 0 1>. MLP results in 0.4181.

The result of reasoning by the rules is 1

with a confidence degree 0.9894

The contribution degrees of the conditions to the conclusion:

Pleo=1: 0.3796

Size=1: 0.6098

There is 1 rule matched:

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

The conclusion is Malignant by direct reasoning.

Note: the MLP result is 0!

Comment: right conclusion.

/**************************************/

The input vector 9 is <0 0 0 1 0 1 0 0 0 0>. MLP results in 0.650895.

0

with a confidence degree 0.3301

The contribution degrees of the conditions to the conclusion:

ICL=0: 0.2171

3D=0: 0.1149

Foamy=0: 0.1285

Nucleoli=1: -0.7904

Size=0: 0.5074

Apocrine=0: -0.1382

There are matched rules in different classes:

>>> In class 0 there are 1 rule matched:

IF (~ICL(0.22), ~3D(0.12), ~Foamy(0.13), ~Size(0.52)) THEN (~Malignant); (0.9677)

Matched Cover Range=60; Conflict Cover Range=2

>>> In class 1 there are 1 rule matched:

IF (Nucleoli(0.85), ~Apocrine(0.15)) THEN (Malignant); (0.9286)

Matched Cover Range=91; Conflict Cover Range=7

The conclusion is Not Malignant by a vote.

Comment: the output of the MLP is 0.65, although week, but different from this conclusion.

/***************************************/

The input vector 14 is <0 0 0 0 0 1 1 1 0 0>. MLP results in 0.981960.

The result of reasoning by the rules is 1

with a confidence degree 0.9894

The contribution degrees of the conditions to the conclusion:

Nucleoli=1: 0.7904

Pleo=1: 0.3796

Size=1: 0.8612

Apocrine=0: 0.1382

There are 3 rules matched:

IF (Size(0.89), ~Apocrine(0.11)) THEN (Malignant); (0.9697)

Matched Cover Range=96; Conflict Cover Range=3

IF (Nucleoli(0.85), ~Apocrine(0.15)) THEN (Malignant); (0.9286)

Matched Cover Range=91; Conflict Cover Range=7

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

The conclusion is Malignant by direct reasoning.

Comment: right conclusion.

/***************************************/

The input vector 15 is <0 0 0 0 0 1 1 1 1 0>. MLP results in 0.9226.

The result of reasoning by the rules is 1

with a confidence degree 0.9894

The contribution degrees of the conditions to the conclusion:

Nucleoli=1: 0.7904

Pleo=1: 0.3796

Size=1: 0.8612

Apocrine=0: 0.1382

There are 3 rules matched:

IF (Size(0.89), ~Apocrine(0.11)) THEN (Malignant); (0.9697)

Matched Cover Range=96; Conflict Cover Range=3

IF (Nucleoli(0.85), ~Apocrine(0.15)) THEN (Malignant); (0.9286)

Matched Cover Range=91; Conflict Cover Range=7

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

The conclusion is Malignant by direct reasoning.

Comment: right conclusion.

/**************************************/

The input vector 16 is <0 0 0 1 1 1 0 0 0 0>. MLP results in 0.7183.

The result of reasoning by the rules is 1

with a confidence degree 0.9286

The contribution degrees of the conditions to the conclusion:

Nucleoli=1: 0.7904

Apocrine=0: 0.1382

There is 1 rule matched:

IF (Nucleoli(0.85), ~Apocrine(0.15)) THEN (Malignant); (0.9286)

Matched Cover Range=91; Conflict Cover Range=7

The conclusion is Malignant by direct reasoning.

Comment: right conclusion, also right contribution fctors.

/**************************************/

The input vector 18 is <0 0 0 1 1 0 0 0 0 0>. MLP results in 0.0953.

The result of reasoning by the rules is 0

with a confidence degree 0.9479

The contribution degrees of the conditions to the conclusion:

ICL=0: 0.2752

Nucleoli=0: 0.3808

Size=0: 0.292

There is 1 rule matched:

IF (~ICL(0.29), ~Nucleoli(0.40), ~Size(0.31)) THEN (~Malignant); (0.9479)

Matched Cover Range=91; Conflict Cover Range=5

The conclusion is Not Malignant by direct reasoning.

Comment: right conclusion.

/**************************************/

The input vector 21 is <0 0 1 0 1 0 0 0 0 0>. MLP results in 0.1463.

The result of reasoning by the rules is 0

with a confidence degree 0.9479

The contribution degrees of the conditions to the conclusion:

ICL=0: 0.2752

Nucleoli=0: 0.3808

Size=0: 0.292

There is 1 rule matched:

IF (~ICL(0.29), ~Nucleoli(0.40), ~Size(0.31)) THEN (~Malignant); (0.9479)

Matched Cover Range=91; Conflict Cover Range=5

The conclusion is Not Malignant by direct reasoning.

Comment: right conclusion.

/************************************/

The input vector 24 is <0 0 0 0 1 0 1 1 0 0>. MLP results in 0.9834.

The result of reasoning by the rules is 1

with a confidence degree 0.9894

The contribution degrees of the conditions to the conclusion:

Pleo=1: 0.3796

Size=1: 0.8612

Apocrine=0: 0.1085

There are 2 rules matched:

IF (Size(0.89), ~Apocrine(0.11)) THEN (Malignant); (0.9697)

Matched Cover Range=96; Conflict Cover Range=3

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

The conclusion is Malignant by direct reasoning.

Comment: right conclusion.

/************************************/

The input vector 25 is <1 0 0 0 0 1 1 1 0 0>. MLP results in 0.9938.

The result of reasoning by the rules is 1

with a confidence degree 0.9894

The contribution degrees of the conditions to the conclusion:

Nucleoli=1: 0.7904

Pleo=1: 0.3796

Size=1: 0.8612

Apocrine=0: 0.1382

There are 3 rules matched:

IF (Size(0.89), ~Apocrine(0.11)) THEN (Malignant); (0.9697)

Matched Cover Range=96; Conflict Cover Range=3

IF (Nucleoli(0.85), ~Apocrine(0.15)) THEN (Malignant); (0.9286)

Matched Cover Range=91; Conflict Cover Range=7

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

The conclusion is Malignant by direct reasoning.

Comment: right conclusion.

/************************************/

The input vector 31 is <0 0 0 1 1 0 1 1 0 1>. MLP results in 0.9725.

The result of reasoning by the rules is 1

with a confidence degree 0.9894

The contribution degrees of the conditions to the conclusion:

Pleo=1: 0.3796

Size=1: 0.6098

There is 1 rule matched:

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

The conclusion is Malignant by direct reasoning.

Comment: right conclusion.

/*************************************/

The input vector 36 is <0 0 0 0 1 0 1 1 0 1>. MLP results in 0.7561.

The result of reasoning by the rules is 1

with a confidence degree 0.9894

The contribution degrees of the conditions to the conclusion:

Pleo=1: 0.3796

Size=1: 0.6098

There is 1 rule matched:

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

The conclusion is Malignant by direct reasoning.

Comment: right conclusion.

/*************************************/

The input vector 40 is <0 0 0 0 1 1 1 1 0 0>. MLP results in 0.993.

The result of reasoning by the rules is 1

with a confidence degree 0.9894

The contribution degrees of the conditions to the conclusion:

Nucleoli=1: 0.7904

Pleo=1: 0.3796

Size=1: 0.8612

Apocrine=0: 0.1382

There are 3 rules matched:

IF (Size(0.89), ~Apocrine(0.11)) THEN (Malignant); (0.9697)

Matched Cover Range=96; Conflict Cover Range=3

IF (Nucleoli(0.85), ~Apocrine(0.15)) THEN (Malignant); (0.9286)

Matched Cover Range=91; Conflict Cover Range=7

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

The conclusion is Malignant by direct reasoning.

Comment: right conclusion.

/*************************************/

The input vector 41 is <1 1 0 0 1 0 1 1 0 0>. MLP results in 0.9968.

The result of reasoning by the rules is 1

with a confidence degree 0.9894

The contribution degrees of the conditions to the conclusion:

ICL=1: 0.8715

Pleo=1: 0.3796

Size=1: 0.8612

Apocrine=0: 0.1133

There are 3 rules matched:

IF (Size(0.89), ~Apocrine(0.11)) THEN (Malignant); (0.9697)

Matched Cover Range=96; Conflict Cover Range=3

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

IF (ICL(0.88), ~Apocrine(0.12)) THEN (Malignant);(0.9848)

Matched Cover Range=65; Conflict Cover Range=1

The conclusion is Malignant by direct reasoning.

Comment: right conclusion.

/**************************************/

The input vector 43 is <1 0 0 0 0 0 1 1 0 0>. MLP results in 0.9878.

The result of reasoning by the rules is 1

with a confidence degree 0.9894

The contribution degrees of the conditions to the conclusion:

Pleo=1: 0.3796

Size=1: 0.8612

Apocrine=0: 0.1085

There are 2 rules matched:

IF (Size(0.89), ~Apocrine(0.11)) THEN (Malignant); (0.9697)

Matched Cover Range=96; Conflict Cover Range=3

IF (Pleo(0.38), Size(0.62)) THEN (Malignant); (0.9894)

Matched Cover Range=93; Conflict Cover Range=1

The conclusion is Malignant by direct reasoning.

Comment: right conclusion.

/**************************************/