GR2-A Hybrid Knowledge-based System Using General Rules

Zhe Ma, **Robert F Harrison**
Department of Automatic Control and Systems Engineering
The University of Sheffield
PO Box 600, Mappin Street, Sheffield S1 4DU, UK
E-mail: z.ma@shef.ac.uk, R.F.Harrison@shef.ac.uk
Telephone: 44-114-2825198 Fax: 44-114-2731729
R. Lee Kennedy
Department of Medicine, The University of Edinburgh

Research Report 561
17 January 1995

Abstract

GR2 is a hybrid knowledge-based system consisting of a Multilayer Perceptron and a rule-base system for hybrid knowledge representations and reasoning.

Knowledge embedded in the trained Multilayer Perceptron (MLP) is extracted in the form of general (production) rules -- a natural format of abstract knowledge representation. The rule extraction method integrates Black-box and Open-box techniques on the MLP, obtaining feature salient and statistical properties of the training pattern set.

The extracted general rules are quantified and selected in a rule validation process. Multiple inference facilities such as categorical reasoning, probabilistic reasoning and exceptional reasoning are performed in GR2.

Experiments have shown that GR2 is a reliable and general model for Knowledge Engineering.

Key Words: Rule Extraction, Hybrid Knowledge-based System, Neural Network, Rule Validation

1. Why are Hybrid Knowledge-based Systems Important?

The combination of Artificial Neural Networks (ANNs) and Rule-based Systems (RBSs) has become a fast growth subject [3-11, 13-19, 21-25]. This is because the two kinds of systems represent knowledge at different levels of abstraction, promising a complement of the systems without compromising their individual strengths. An optimally organised hybrid system, which includes both ANNs as the automatic knowledge acquisition facilities and a RBS as a high level inference engine and user interface, provides some major advantages.

•It overcomes the knowledge acquisition bottleneck problem existing in knowledge engineering where a RBS is used alone, enabling a completely automatic knowledge engineering process.

•It provides an explicit representation of the knowledge encoded in ANNs, in the form of, say, production rules, solving the problem of opaqueness of knowledge representation suffered by most ANNs. This explanatory capability will be beneficial in the following ways. People will more willingly accept ANNs if it can be seen that the knowledge they have acquired concurs with their domain knowledge at hand. ANNs will be capable of knowledge discovery for human study in the application domains. Understanding connectionist model of cognition will be much easier.

•The knowledge encoded in a trained ANN presents a uniform property, whereas knowledge acquired from human experts usually involves direct or indirect conflicts, which are very difficult to detect and to avoid.

•Symbolic knowledge representation provides better approaches to improve further the knowledge acquired by the ANNs. For instance, generality of the rule set can be obtained and adjusted even if the ANNs have been under-trained or over-trained. Accuracy of the rule set representing of the application domain can be optimized through a process known as rule validation. Probabilistic representation and reasoning with rules lead to system robustness in noisy and redundant environments.

The central themes of hybrid system methodology include the following two considerations: (i) the optimal format of the symbolic knowledge representation and (ii) how the implicit subsymbolic knowledge acquired by ANNs can be translated into symbolic knowledge format.

This paper presents a common symbolic knowledge format, general rules, in Section 2, being used in our hybrid knowledge-based system GR2. Section 3 introduces a novel and efficient heuristic method to extract the knowledge from a trained Multilayer Perceptron. The system GR2 is outlined in Section 4, which also includes rule validation and multiple inference functions. Experiments in some artificial and real-world applications are reported in Section 5. The paper ends with a summary in the final section.

2. General Rules -- an Abstract Knowledge Representation

2.1 Definitions

We address binary problems in this paper.

Definition 1 A **Boolean variable** β, when it is instantiated, is represented in two forms:

•subsymbolic form: 1 or 0, indicating instantiations $\beta=1$ or $\beta=0$ respectively. Traditionally, these are also referred to as positive or negative instantiations.

•symbolic form: β or $\sim\beta$, indicating β positively or negatively instantiated respectively. Here β appears as the symbolic name of the variable.

Definition 2 L boolean variables constitute an L dimensional **binary space**, $B^L=\{0,1\}^L$. An element of the binary space is a vector of length L. A vector also has two forms:

- subsymbolic form: <$\overline{b}_1,\overline{b}_2,..,\overline{b}_L$>, where $\overline{b}_i$ is either 1 or 0, corresponding to the *ith* boolean variable instantiated

- symbolic form: <$\overline{\beta}_1,\overline{\beta}_2,..,\overline{\beta}_L$>, where $\overline{\beta}_i$ is either β_i or $\sim\beta_i$, depending on the symbolic form of the instantiation of the *ith* boolean variable

Definition 3 A **binary problem** is a triple: (I, O, F), where

- I is an N dimensional binary space, the input space, comprising N boolean variables;

- O is a M dimensional binary space, the output space, comprising M boolean variables;

- F is a function, which is a set of mappings $\{\overline{\varpi} \rightarrow \overline{\gamma}\}$, where $\overline{\varpi}$ is a vector from I and $\overline{\gamma}$ is an instantiated variable from O. Note, $\overline{\gamma}$ is not a vector.

Given a binary problem, representation of the function F in an effective, abstract format is the central target of an inductive learning system. An artificial neural network encodes the function in a subsymbolic, implicit representation. A rule-based system, on the other hand, describes the function in a symbolic and explicit representation.

Definition 4 An **MLP** in this paper is a Multilayer Perceptron network having one layer of hidden units. Completely weighted connections are used for any adjacent unit layers. The input, the hidden and the output layers of units are denoted as $\{I_i\}$, $\{H_h\}$ and $\{O_o\}$ respectively, where the indices i=1..N, h=1..Q, and o=1..M respectively. The two layers of weight connections from the input to the hidden layer and from the hidden to the output layer are $W_1=\{w_{ih}\}$ and $W_2=\{w_{ho}\}$ respectively. An MLP can approximate the function in a binary problem.

Definition 5 A **Sole Pattern,** $P_j=(I, O_o)$, is an input vector from the input space, taken together with one instantiated output variable. When the output variable is instantiated by 1, the sole pattern is positive; otherwise, it is negative.

Sole patterns collectively describe a function of a binary problem, and also the function of the MLP. A set of sole patterns can appear contradictorily in a binary problem with noise, i.e. an input vector may correspond to different values of an output variable. Sole patterns generated by the MLP are uniform, because there is a unique output corresponding to any input vector from an ANN, if it is not further modified. GR2 uses sole patterns generated by the trained MLP, and retains the uniformity of the acquired knowledge.

Definition 6 A **Rule** is constituted with a premise part and a conclusion part, having the form

IF $(\alpha_1, \alpha_2,... \alpha_L)$ THEN (γ)

where $0<L\leq N$, α_js and γ are instantiated boolean variables, in a form either positive or negative (headed with a ~). An α_j, an instantiated input variable, is called a premise, attribute or feature. α_js are conjunctively related. The γ, an instantiated output variable, is a conclusion or consequence. If γ is positive, the rule is a **positive rule**. If the γ is negative, the rule is a **negative rule**. This form of rules is equivalent to the Horn Clause Format.

Definition 7 To a binary problem with N input variables, a rule having N premises is a **special rule**. A special rule is the symbolic representation of a sole pattern. A rule R having L premises, where $0<L\leq N$, is a **general rule**. A general rule represents a set of special rules whose conclusion is the same as that of the general rule, and whose premises are those of the general rule conjunctive with any possible combinations of the absent premises in the general rule. In other words, a general rule includes a set of special rules, or it covers a set of sole patterns. In a general rule, the present premises preserve the essential attributes which are commonly held in the set of sole patterns in its coverage, by which this set is distinguished from other sole patterns. The absent premises in a general rule are not significant and thus may be ignored.

Definition 8 Given two general rules R_1 and R_2, R_1 is **more general** than R_2 if R_1 represents a set of special rules subsuming the set of special rules represented by R_2. Given any two general rules R_i and R_j on the same binary problem, the necessary and sufficient conditions for R_i to be more general than R_j are, (a) R_i and R_j have the same conclusion; AND (b) R_i's premises form a subset of R_j's. Obviously, the fewer premises a general rule possesses, the more general it is. Two general rules are **incomparable** if (a) they have different conclusions (either with a different variable, or with the same variable but instantiated by different values); OR (b) their premises mutually do not subsume.

Definition 9 A **non-redundant set** is a set of general rules, any pair of which are incomparable.

Sole patterns and rules are two different representations of the same objects. GR2 uses both representations for its hybrid components. Knowledge from a trained MLP is translated into a non-redundant set of general rules.

2.2 Generality vs. Accuracy

Accuracy and generality are essential criteria for machine learning. For GR2, accuracy is the ratio of the sole patterns correctly covered or classified by the general rules in the given training set. Generality is the degree of abstraction at which the general rules represent the property. Generality is characterised by the average and the standard deviation of the numbers of the premises in the general rules.

Learning capability is attributed to generality provided that accuracy is at an acceptable level. As shown in Figure 1, an ellipse indicates a rule's coverage in the input space. The Figure 1 illustrates the fact that a few general rules (corresponding to the two large ellipses) can cover the range whereas otherwise many more less general rules (corresponding to the four small ellipses) are required. The importance of this is that the more general the rules are, the more likely they evenly cover both ranges corresponding to the trained and the untrained (i.e. those not used in training the MLP) sole pattern sets, embodying higher learning capability. A less general rule, however, is unlikely to cover any sole pattern beyond the particularly predetermined range of the training set.

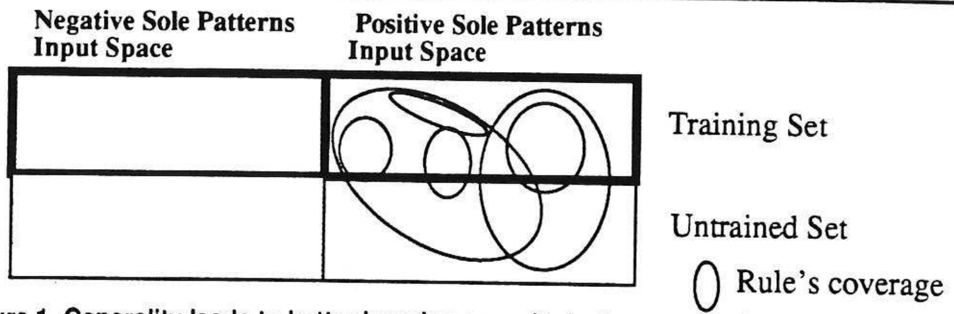

Figure 1. Generality leads to better learning capacity in the untrained set range

However, a rule more general than necessary is more likely to lose the uniformity of its coverage. It may misclassify some sole patterns. As shown in Figure 2, the two large ellipses cover the ranges across the boundary of Negative/Positive Sole Patterns. In other words, accuracy decreases as generality increases. The relationship between generality and accuracy is depicted in Figure 3.

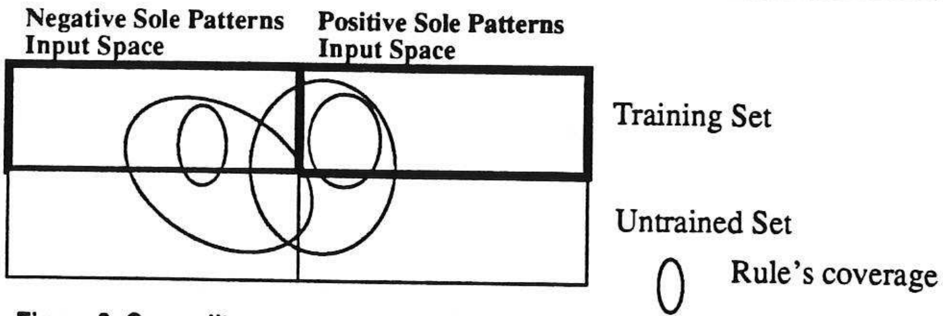

Figure 2. Generality may cause partial misclassification too

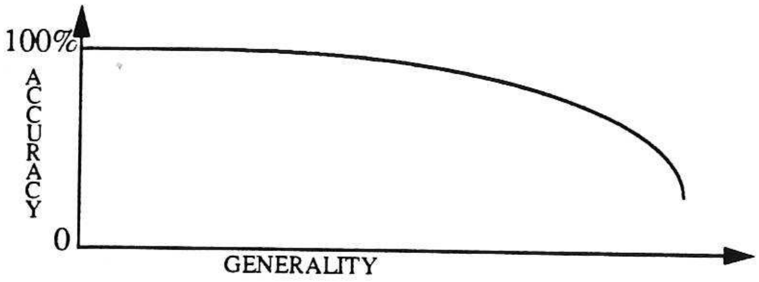

Figure 3. The relationship between generality and accuracy

Optimization on knowledge representation is beneficial to trade off between accuracy and generality. Section 2.3 and 2.4 discuss some attempts at such optimization, by which accuracy will not be harmed much as generality increases to a certain extent.

Control of generality is easy in GR2 by varying a threshold (see later). Obtaining accuracy is however much more complicated. Many factors for this are beyond the scope of this paper. Note, an accuracy of 100% is usually not ideal, and may be impossible in a practical data set which includes some noisy cases.

2.3 Probabilistic Rules

The rule previously mentioned is **categorical** whose coverage is assumed to be uniform. A probabilistic rule is a rule to which the uniform assumption on its coverage is not held. A **Confidence Factor** is the additional component to a probabilistic rule. It counts on the sole patterns correctly classified or misclassified in the rule coverage.

$$cf(R_i)=(cc-ec)/(cc+ec)$$

where cc is the number of sole patterns correctly classified, and ec is the number misclassified. The value range is [-1, 1]. When the confidence factor is 1, the rule is equivalent to a categorical rule. Categorical rules are a special case of probabilistic rules.

The confidence factor provides a better capability for classification under uncertainty. GR2 classifies sole patterns in two consequential procedures:

- **Categorical Reasoning**: If an input vector is covered by a set of categorical rules which have the same conclusion, the conclusion of the rules is the class the input vector belongs to. Otherwise

- **Probabilistic Reasoning**: if the vector is covered by a set of rules which have different conclusions, its class is decided by the conclusion of those rules whose confidence factors, when summed, are more than those of the opposing rules.

2.4 Exceptional Reasoning

GR2 usually generates both positive and negative rules for every output variable. However, as training patterns occasionally appear with features not sufficiently distinct in every aspect, the rules for an output variable are provided by only either a positive or a negative form. GR2 performs Exceptional Reasoning to cope with this situation.

Exceptional Reasoning: check the input vector by the existing rules. If it is covered by any of the rules, the class is decided by the conclusion of the rules. Otherwise, the class is the opposite of the conclusion in the existing rules.

3. How are the General Rules Extracted?

General rules are extracted from a trained MLP by two techniques in two approaches. In the Open-box approach, the weight matrices are explored and a linear statistical property of the MLP is obtained. In the Black-box approach, the MLP is taken as a black-box, and only its Input/Output behaviour is observed for examining the salient individual features from the trained pattern set. Gathering these two sorts of properties, the rule extraction algorithm generates the rules and controls the generality degree of the rule set with a threshold. The details are explained in the following three sections respectively. This method is first introduced by us in [15], and improved in this paper. This method does not require any special change on the ANN and is effective in both information dense cases such as two or more bit AND, OR, and parity problems, and real-world domains of large scale in noisy and redundant circumstances.

3.1 Potential Default Set

The contributive relationship from the input units to the output units of the MLP can be partially observed by the matrix $L = (W_1 W_2)^T$. An element of L, $L_{oi} = \sum_h w_{ih} \cdot w_{ho}$ is the summed link strength between the *ith* input unit I_i and the *oth* output unit O_o.

Analysis of the contributive relationship from the input units to the output units is also based on two foundations: (a) the monontonicity of the sigmoid function the MLP uses for computing the activations of its units; (b) the fact that the activations of the output units always fall in the tolerance range, either in $[0, \delta]$ or in $[1-\delta, 1]$, when the input vectors in the training patterns are fed to the MLP. The δ is the tolerance used in the MLP test (classification) process. Note: point (a) is always true for MLPs; point (b) can always be held too, since δ can be loosely assigned as long as all patterns for test are uniquely classified, rather than being as restrict as Δ, the tolerance for MLP training.

Observation is isolated only upon units I_i and O_o. If L_{oi}>0, O_o tends to increase its activation as I_i switches from 0 to 1, and to decrease as I_i switches from 1 to 0. However, if O_o's activation is in the range $[1-\delta, 1]$ and I_i is 0, switching I_i will not impact the classification result reflected by O_o. I_i is ignorable in this circumstance. Similarly, if I_i is 1, switching I_i may not change the result as O_o is in the range $[0, \delta]$, I_i is therefore ignorable. The situations are reversed as L_{oi}<0. The situations where the I_i is ignorable are summarised in Table 1.

Table 1: Situations where I_i may be ignorable

L_{oi}	I_i	O_o
> 0	0	$[1-\delta, 1]$

Table 1: Situations where I_i may be ignorable

L_{oi}	I_i	O_o
> 0	1	$[0, \delta]$
< 0	1	$[1-\delta, 1]$
< 0	0	$[0, \delta]$

Let α_i (i=1..N) denotes the *ith* input variable. The **Potential Default Set** is defined to identify the subset of an input vector which is possibly not influential on the classification result of a particular output value, based on the foregoing analysis.

Given *oth* row from matrix L, $L_o=\{L_{oi}\}$, we define two sets:

$Z_1=\{\alpha_i | L_{oi} \geq 0\}$ $\qquad$ $N_1=\{\alpha_i | L_{oi} < 0\}$

Given an input vector $I=\{I_i\}$, we define another two sets:

$Z_0=\{\alpha_i | I_i=1\}$ $\qquad$ $N_0=\{\alpha_i | I_i=0\}$

A **Potential Default Set (PDS)** of the input vector I, with respect to the output variable O_o,

$(Z_0 \cap N_1) \cup (N_0 \cap Z_1)$ $\quad$ if O_o=1 or in $[1-\delta, 1]$

$(Z_0 \cap Z_1) \cup (N_0 \cap N_1)$ $\quad$ if O_o=0 or in $[0, \delta]$

The elements of the PDS are the candidates possibly absent from the rules extracted from a sole pattern (I, O_o).

The PDS represents a statistical property of the trained MLP. It is on average half the size of the input vectors from empirical observations. Hence the dimensionality of the test space on the input values may be reduced by up to half. However, PDS has the linear limitation.

3.2 Feature Salient Degree

Concerning all sole patterns $\{P_j\}$ with respect to the an output variable O_o, the Feature Salient Degree (FSD) is a matrix

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

where max(X) is the value of the maximal element of the matrix X. The fsd is a matrix whose *jith* element is

$$\mathrm{fsd}_{ji} = \sum_{\{k|\left(j \neq k,\, o_o^j \neq o_o^k,\, I_{ji} \neq I_{ki}\right)\}} e^{-\mathrm{dist}\,(P_j,\, P_k)}$$

where I_{ji} and I_{ki} are the *ith* input values respectively in sole patterns P_j and P_k; O_o^j and O_o^k are the output values involved in P_j and P_k. The function dist(P_j,P_k) is the Euclidean distance between the input vectors in P_j and P_k. The definition of fsd_{ji} tells: for the *ith* instantiated input variable of pattern P_j, the summation counts for those P_ks, which include both the output variable and the *ith* input variable instantiated by different values from those in P_j. $e^{-\mathrm{dist}\,(P_j,\, P_k)}$ indicates that the fewer different input values the pair of patterns P_j and P_k have, the greater effect P_k gives to fsd_{ji}.

The FSD is a measure of the amount of information carried by the input units in the context of the training set. It represents the correlation of the changes on an instantiated input variable and an instantiated output variable, estimating the possibility of a change of the output value when the input variable is switched.

The MLP is used as a black-box in computation of the output values of the sole patterns.

3.3 Rule Extraction with PDS and FSD

There is a key parameter, FSD threshold T, as the control of generality of the rule set extracted. T is used to decide if (i) an individual input bit, (ii) a set of input bits is preserved for forming premise(s) in the extracted rules. It should be within the FSD range [0,1]. We recommend a rational range [0.3,1], and a default value 0.6 for T.

General rules are extracted from a sole pattern $P_j=(I, O_o)$, where I is an input vector $<I_1,I_2,..,I_N>$, in the following steps. Remember: I_is are instantiated variables, not only values.

1. Compute PDS and $FSD_j=\{FSD_{ji}\}$ for P_j (FSD_j is given as the matrix FSD is built).

2. Generate a set $\psi=\{I_i \mid FSD_{ji} \geq T\}$

3. Generate a set of "smallest subsets" $\Theta=\{\theta_k \mid \exists\theta_l \in \Theta: \theta_l \neq \theta_k \Rightarrow (\theta_k \not\subset \theta_l , \theta_l \not\subset \theta_k)\}$, which says that all the elements θ_ks are mutually exclusive, where $\theta_k=\{I_i \mid I_i \notin PDS, FSD_{ji}<T, \sum_i FSD_{ji} \geq \min(\Delta, T/N^{1/2})\}$, Δ is the tolerance in the MLP training.

4. Construct general rules by all pairs $(\psi \cup \theta_k, O_o)$. The former, $\psi \cup \theta_k$, a set of instantiated input variables, are symbolized into premises. The latter, one instantiated output variable, is symbolized into the conclusion. The word "symbolize" means: if a variable is instantiated by 1, it presents by its corresponding symbol in the rule. If it is by 0, the symbol is headed with a ~, the sign for negation.

The algorithm takes computation of $O(N^2 \times M \times P^2)$, where N is the input vector size, M is the output vector size, and P is the number of the training patterns (not of the sole patterns). Details of this are given in [15]. In fact, the computation is most consumed in P times of recall of the MLP for the output values of the sole patterns, and secondarily most used in step 3, looking for the subsets. The generality of the rule set is decided by the expression $\min(\Delta,T/N^{1/2})$, where only the FSD threshold T is changeable. The higher the T, the more general the rule set is, and vice versa.

4. GR2 System Architecture

GR2 system is depicted in Figure 4. The first component, an MLP is in a common architecture defined in Section 2.1. After training, the MLP will not be changed at all. The second component for Rule Extraction executes the algorithm described in Section 3.3. Categorical rules are generated.

The third component is for Rule Validation. Rule validation is a process to determine if the rules perform at an acceptable accuracy rate over the training pattern set. We also include rule-base maintenance in it. The Rule Validation process includes several functions:

- Computation of the confidence factors for rules by checking rules in the training pattern set (Section 2.3);
- Elimination of those rules whose confidence factor < Δ, the training tolerance;
- Prevention of redundancy by deleting rules more special than any other rules;
- Generalising rules by combination of similar rules if possible.

The fourth component, Inference by Rules, is the inference engine classifying input vectors by rules. The inference is simple because the rules are direct mapping from the input space to the output space, no intermediate variables being involved. The inference process is

- Do exceptional reasoning if necessary. Otherwise, do reasoning by direct matching. In both cases, the following two steps are executed:
- If the input vector is covered by categorical rules, do categorical reasoning. Otherwise,

- Do probabilistic reasoning.

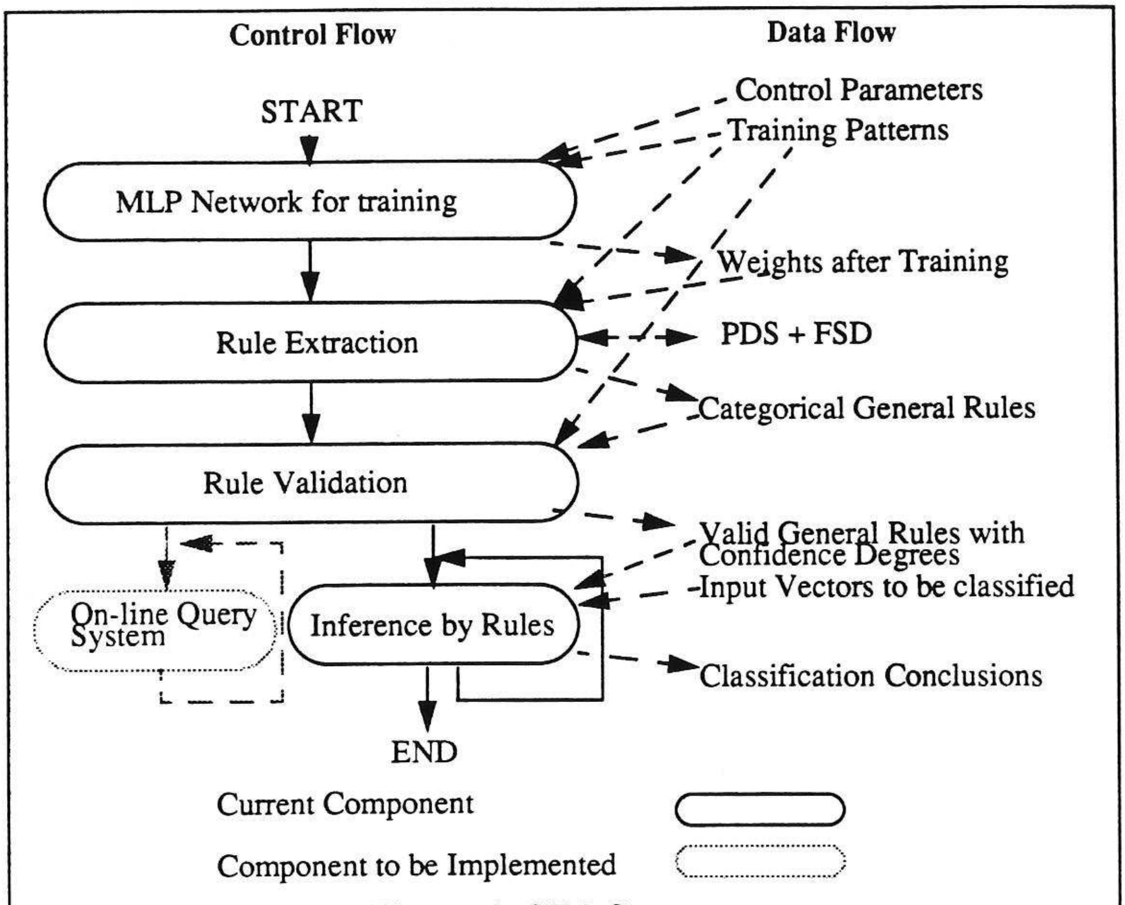

Figure 4. GR2 System

5. Examples

GR2 has been successfully experimented on many typical artificial binary problems and two real-world medical problems. The artificial problems are always information dense. Categorical reasoning is sufficient in those situations. Real-world domains are usually information sparse, where probabilistic reasoning and the tradeoff between generality and accuracy are useful.

In Section 5.1, simple logic domains are used to demonstrate the rule extraction process. A four bit parity with an incomplete training set is presented in Section 5.2. One of the real-world domains is discussed in Section 5.3. The other has been presented in [15].

5.1 Two Bit AND and XOR

5.1.1 Two Bit AND

Two bit AND is represented as:

Table 2: Representations of Two Bit AND

Subsymbolic Form			Symbolic Form
A	B	C	
0	0	0	IF(~A, ~B) THEN (~C)
0	1	0	IF(~A, B) THEN (~C)
1	0	0	IF(A, ~B) THEN (~C)
1	1	1	IF(A, B) THEN (C)

where A, B are one bit inputs, and C is one bit output. Although it is well known that hidden units are not necessary to the simple domain, we use them for the purpose of demonstration.

Using an MLP architecture with 2 input units, 2 hidden units, and 1 output unit, with one bias to each of the hidden units and to the output unit, the weight matrices after training are:

$$w_1 = \begin{bmatrix} -0.294 & -0.351 \\ -0.005 & 0.826 \end{bmatrix} \qquad w_2 = \begin{bmatrix} -3.42 \\ 1.194 \end{bmatrix}$$

Now we calculate the PDSs. Since both elements in L

$$L = (w_1 w_2)^T = \begin{bmatrix} 0.585 & 1.002 \end{bmatrix}$$ are positive,

$Z_1=\{A, B\}$ $N_1=\{\}$

For the first sole pattern (<0, 0>, 0),

$Z_0=\{\}$ $N_0=\{A, B\}$

As the output bit $O_1=0$, the PDS for pattern 1 is

$$PDS_1 = (Z_0 \cap Z_1) \cup (N_0 \cap N_1) = \{\}$$

For the second pattern (<0,1>, 0),

$Z_0=\{B\}$ $N_0=\{A\}$ $PDS_2=\{B\}$

For the third pattern (<1,0>,0),

$Z_0=\{A\}$ $N_0=\{B\}$ $PDS_3=\{A\}$

For the fourth pattern (<1,1>, 1),

$Z_0=\{A, B\}$ $N_0=\{\}$

As the output bit $O_1=1$,

$$PDS_4=(Z_0 \cap N_1) \cup (N_0 \cap Z_1) = \{A,B\}$$

Delete the PDSs from the full set of input symbols, we have the following *remainder* matrix, where an *o* means absence:

$$\begin{bmatrix} A & B \\ A & o \\ o & B \\ o & o \end{bmatrix}$$

The FSD matrix is:

$$\begin{bmatrix} 0.269 & 0.269 \\ 0.731 & 0 \\ 0 & 0.731 \\ 1 & 1 \end{bmatrix}$$

Set FSD threshold $T=0.6$. There is no rule generated from the first pattern since the FSD_{11} and FSD_{12} are below T, even though both A and B are not in the PDS.

From pattern 2, (<0,1>,0), A=0 remains. A rule is generated,

IF (~A) THEN (~C)

From pattern 3, (<1,0>,0), B=0 remains, generating a rule

IF (~B) THEN (~C)

From pattern 4, (<1,1>,1), even if both input bits are in the PDS$_4$, the FSD$_4$ =[1 1] is however greater than the threshold *T*. Both input variables are hence retained for premises, generating a rule

IF (A,B) THEN (C)

Similarly to AND, the two bit OR was generated by rules

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

5.1.2 XOR

The XOR problem is

Table 3: XOR

Subsymbolic Form			Symbolic Form
A	B	C	
0	0	0	IF(~A, ~B) THEN (~C)
0	1	1	IF(~A, B) THEN (C)
1	0	1	IF(A, ~B) THEN (C)
1	1	0	IF(A, B) THEN (~C)

The FSD for XOR is $\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$

Setting any threshold $T<1$, all elements of the FSD are greater than the threshold. All input variables are retained for premises. Therefore the extracted rules are the same as those in the symbolic form. Parity problems are always given FSDs of unity in GR2, resulting in a full list of the original symbolic form as the rules extracted, as they should be.

5.2 Incomplete Training Set for a four Bit Problem

Learning capability is assessed by the accuracy of recognition on the patterns without the training set. This Section shows how GR2 tackles this situation.

Given a training set with four bit inputs, named A B C D, and one bit output, named E, it includes 11 patterns instead of 16 patterns in the complete set. The included patterns in the training set are assigned as a part of four bit parity problem. The patterns are listed in Table 4 except for those shaded columns.

Table 4: Patterns of Incomplete 4 Bit Domain (shaded patterns are not in the training set)

Label	Pat1	Pat2	Pat3	Pat4	Pat5	Pat6	Pat7	Pat8	Pat9	Pat10	Pat11	Pat12	SP13	SP14	SP15	SP16
A	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
C	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
D	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
E	0	1	0	0	1	1	0	1	1	1	0	1	1	1	1	0

After training, the MLP, sized of 4:3:1 to the input/hidden/output layers, classified all 16 inputs as shown in Table 4. The conclusions were rounded into integers, including those patterns absent in the training set. The General Rules extracted from this trained MLP were

IF (~A, ~C, ~D) THEN (~E);
IF (~A, B, ~C) THEN (~E);
IF (A, C, ~D) THEN (E);
IF (~B, C) THEN (E);
IF (B, ~C, ~D) THEN (~E);
IF (A, B, C, D) THEN (~E);
IF (~B, D) THEN (E);
IF (~A, C, D) THEN (E);
IF (~A, B, ~D) THEN (~E);
IF (A, ~B) THEN (E);
IF (A, ~C, D) THEN (E);

All the rules were valid and the confidence degree for each rule was 1, because each rule correctly covered some training patterns and there were no training patterns conflict to it.

Now apply the rules to the untrained patterns:

For input <0 1 0 0> in Pat3, there are 4 rules covering it, concluding ~E:
IF(~A, ~C, ~D)THEN(~E); IF(B, ~C, ~D)THEN(~E);
IF(~A, B, ~D)THEN (~E); IF(~A, B, ~C)THEN(~E);

For <0 0 1 0> in Pat5, 1 rule covers it, concluding E:
IF (~B, C) THEN (E);

For <1 0 1 0> in Pat6, 3 rules cover it, concluding E:
IF (A, C, ~D) THEN (E); IF (~B, C) THEN (E); IF (A, ~B) THEN (E);

For <1 0 0 1> in Pat10, 3 rules cover it, concluding E:
IF (A, ~B) THEN (E); IF (~B, D) THEN (E); IF (A, ~C, D) THEN (E);

For <0 0 1 1> in Pat13, 3 rules cover it, concluding E:
IF (~B, D) THEN (E); IF (~B, C) THEN (E); IF (~A, C, D) THEN (E);

The rules were uniformly used in inference at every case. All conclusions were the same as given by the MLP as desired.

5.3 Diagnosis of Acute Myocardial Infarction (Heart Attack)

The early identification of patients with acute ischaemic heart disease remains a great challenge in emergency medicine. The ECG only shows diagnostic changes in about half of acute myocardial infarction (AMI) patients at presentation [2, 20]. None of the available biochemical tests becomes positive until at least three hours after symptoms begin, making such measurements of limited use for the early triage of patients with suspected AMI [1]. The early diagnosis of AMI, therefore, relies on an analysis of clinical features along with ECG data. An MLP has been shown to be a good method for combining clinical and electrocardiographic data into a decision aid for the early diagnosis of AMI [12]. The data used in this study were derived from consecutive patients attending the Accident and Emergency Department of the Royal Infirmary, Edinburgh, Scotland, with non-traumatic chest pain as the major symptom. The relevant clinical and ECG data were entered onto a purpose-designed proforma at, or soon after, the patient's presentation. The study included both patients who were admitted and those who were discharged. 970 patients were recruited during the study period (September to December 1993). The final diagnosis for these patients was assigned independently by a Consultant Physician, a Research Nurse and a Cardiology Registrar. This diagnosis made use of follow-up ECGs, cardiac enzyme studies and other investigations as well as clinical history obtained from review of the patient's notes.

The input data items for the MLP were all derived from data available at the time of the patient's presentation. In all, 35 items were used, coded as 37 binary inputs. For the purposes of this application, the final diagnoses were collapsed into two classes termed "AMI" (Q wave AMI and non-Q wave AMI) and "not-AMI" (all other diagnoses). AMI cases were assigned as positive diagnoses, not-AMI cases as negative diagnoses. The MLP was constructed with 37:13:1 as the sizes of the input:hidden:output layers respectively. The error tolerance was Δ=0.1. The 970 patient records were divided into two data sets, 670 randomly selected as the training set, and the remaining 300 as the test set.

The experimental criteria are:

FSD Threshold is selected in the range [0.4, 1].

Total Rules and *Valid Rules* respectively indicate the numbers of total and valid rules.

Ave Premises is the average number of the premises of the valid rules.

Std Deviation of Premises is the standard deviation of the numbers of premises of the valid rules.

There are also three performance criteria on the training set and the test set respectively, being used in the medical community.

Sensitivity is defined as the ratio of the number of correct positive diagnoses to the number of positive outputs. This is most important as the disease is life-threatening.

Specificity is defined as the ratio of the number of correct negative diagnoses to the number of negative outputs. This is important as treatment is expensive and can be risky.

Accuracy is defined as the ratio of the number of correct diagnoses to the total number.

The statistical results are listed in Table 5

Table 5: GR2 Experiment Results on AMI Records

FSD Threshold	0.85	0.7	0.65	0.6	0.55	0.5	0.4
Total Rules	51	66	51	65	96	112	176
Valid Rules	18	31	20	31	60	68	124
Ave Premises	2.33	2.81	2.55	2.61	2.83	2.93	2.94
Std Deviation of Premises	0.235	1.03	0.682	0.445	0.65	0.905	0.623
Sensitivity on Training Set (%)	87	75.6	75.9	68.3	61	60.2	36.6
Specificity on Training Set(%)	87.7	69.5	71.1	85	94.2	95.4	99.1
Accuracy on Training Set(%)	87.5	70.6	71.9	82	88.1	89	87.6
Sensitivity on Test Set(%)	90	72.5	69.4	65.5	59.5	55.1	37.7
Specificity on Test Set(%)	82.7	70.6	75.4	84	93.1	94.4	99.6
Accuracy on Test Set(%)	84	71	74	80	85.3	85.3	85.3

The extracted rules are not given in this paper because of the space limit.

The best case was at FSD Threshold=0.85, where exceptional reasoning was performed. All extracted rules were negative.

The rule extraction processes took between 6 and 46 seconds on Sun Sparc10; Rule Validation processes took 4 - 11.66s; and Rule Reasoning processes on all the test set took 0.33 to 7.7s.

These results are comparable to those experimented on C4.5 [20], shown in Table 6.

Table 6: C4.5 Experiment Results on AMI Records

	By Decision Tree	By Extracted Rules
Sensitivity on Training Set (%)	84	97.6
Specificity on Training Set(%)	97	61
Accuracy on Training Set(%)	94	67.3
Sensitivity on Test Set(%)	73	95.7
Specificity on Test Set(%)	96	66.2
Accuracy on Test Set(%)	92	73

6. Conclusion and Further Work

The general rule is a format representing only important data features, ignoring superfluous ones. This representation of knowledge provides the capabilities of generalisation, simplicity and efficiency in knowledge engineering. It is feasible for probabilistic representation and multiple inference utilization, providing systematic robustness.

GR2 extracts knowledge from an MLP in the form of general rules via an Open-box method for obtaining the linear statistical property, and a Black-box method for collecting individual feature salient properties. Generality of the rule set extracted is easily adjustable by varying the threshold of the feature salient degree.

We intend to expand GR2 with more functions such as on-line knowledge acquisition and explanation. The former guides users by giving queries sensitive to dynamic context, archiving time-labour efficiency. The latter provides a quantitative premise-conclusion causal relationship, which will be valuable information to system optimization in applications.

Acknowledgement

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

References

[1] J. E. Adams, D. R. Abendschein and A. S. Jaffe. *Biochemical Markers of Myocardial Injury. Is MB Creatine Kinase the Choice for the 1990s?* Circulation, 88, 750–63. (1993)

[2] J. E. Adams, R. Trent and J. Rawles. *Earliest Electrocardiographic Evidence of Myocardial Infarction: Implications for Thrombolytic Treatment.* British Medical Journal, 307, 409–13. (1993)

[3] L. Bochereau, P. Bourgine. *Rule extraction and validity Domain on a Multilayer Neural Network.* International Joint Conference on Neural Networks, 1990 VolI pp97-100

[4] G. A. Carperter and Ah-Hwee Tan. *Rule Extraction', Fuzzy ARTMAP, and Medical Databases.* Proceedings of the World Congress on Neural Networks, Volume I, 501–506.

[5] M. W. Craven, J. W. Shevlik. *Understanding Neural Networks via Rule Extraction and Pruning.* Proceedings of the 1993 Connectionist Models Summer School, Edited by M Mpzer, P Smolensky, D Touretsky, J Elmanm, A Weigen

[6] M. W. Craven and J. W. Shevlik. *Learning Symbolic Rules Using Artificial Neural Networks.* Proceedings of the Tenth International Conference in Machine Learning, Edited by P. E. Utgoff, Morgan Kaufmann, San Mateo, 1993

[7] J. Diederich. *An Explanation Component for a Connectionist inference System.* EJCAI 90, pp222-227

[8] J. Downs, R. F. Harrison, S. S. Cross. *A Neural Network Decision-Support Tool for the Diagnosis of Breast Cancer.* The University of Sheffield, Department of Automatic Control and Systems Engineering, Research Report No 548, 1994

[9] L. M. Fu. *Knowledge-based connectionism for revising domain theories.* IEEE Transactions on Systems, Man, and Cybernetics, 23(1), pp173-182

[10] L.M.Fu. *Neural Networks in Computer Intelligence.* 1994, McGraw-Hill

[11] C. M. Higgines, R. M. Goodman. *Learning Fuzzy Rule-Based Neural Networks for Control.* Advances in Neural Information Process Systems, 1988

[12] R. L. Kennedy, R. F. Harrison and S. J. Marshall. *A comparison of Logistic Regression and Artificial Neural Network Models for the Early Diagnosis of Acute Myocardial Infarction.* The University of Sheffield, Department of Automatic Control and Systems Engineering, Research Report No. 539, 13 Oct. 1994

[13] D. C. Kuncicky, S. I. Hruska, and R. C. Lacher. *Hybrid Systems: the Equivalence of Rule-Based Expert System and Artificial Neural Network Inference.* International Journal of Expert Systems Research and Applications, Vol: 4, Issue: 3, p. 281-97, 1991

[14] D. A. Linkens and J. Nie. *Rule Extraction for BNN Neural Network-Based Fuzzy Control System by Self-learning.* Proceedings of International Conference in Artificial Neural Networks, 1992

[15] Z. Ma, R. F. Harrison, R. L. Kennedy. *A Heuristic for General Rule Extraction from a Multilayer Perceptron.* The University of Sheffield, Department of Automatic Control and Systems Engineering, Research Report No 549, 1994; also accepted by AISB-95, to appear

[16] J. J. Mahoney, R. J. Mooney. *Combining Neural and symbolic Learning to Revise probabilistic Rule base.* EJCAI 93, pp107-114

[17] C. McMillan, M. C. Mozer, P. Smolenskey. *Rule Induction through Integrated Symbolic and Subsymbolic Processing.* Proceedings of Advances in Neural Information Processing Systems, pp969-976, Morgan Kaufmann

[18] S. Ridella, G. Speroni, P. Trebino, R. Zunino. *Class-Entropy Minimisation Networks for domain Analysis and Rule Extraction.* Neural computing and Applications, 1994 Springer-Verlag London Limited, pp 40-52

[19] K. Saito, R Nakano. *Rule Extraction from Facts and Neural Networks.* International Neural Network Conference 1990, VolI PP379-382

[20] J.R.Quinlan, *C4.5 Programs for Machine Learning.* 1993, Morgan Kaufmann

[21] M. E. Stark, J. L. Vacek, *The Initial Electrocardiogram During Admission for Myocardial Infarction.*

Use as a Predictor of Clinical Course and Facility Utilization, Archives of Internal Medicine, 147, pp843–6, 1987.

[22] H. Tirri. *Implementing Expert System Rule Conditions by Neural Networks*. New Generation Computing, Vol. 10 No. 1, 1991

[23] G. Towell, J. W. Shavlik, M. O. Noordewier. *Refinement of Approximately correct Domain Theories by Knowledge-based Neural Networks*. 8th National Conference on AI, Boston, MA, pp861-866. AAAI Press

[24] G. Towell, J. W. Shavlik. *Extracting refined rules from knowledge-based neural networks*. Machine Learning, vol.13, no. 1,p.71-101

[25] G. Towell, J. W. Shavlik. *Interpretation of Artificial Neural Networks: Mapping Knowledge-Based Neural Networks into Rules*. EJCAI 93, pp977-984

[26] G. Towell, J. W. Shavlik. *Using Symbolic Learning to Improve Knowledge-Based Neural Networks*. AAAI 93

[27] P. A. Trott, (1991) *Aspiration Cytodiagnosis of the Breast*, Diagnostic Oncology, vol 1, pp79-87