GR2 — A Hybrid Knowledge-based System Using General Rules

Zhe Ma, Robert F Harrison
The University of Sheffield,
Department of Automatic Control and Systems Engineering,
Mappin Street, Sheffield S1 3JD, 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, UK

Abstract

GR2 is a hybrid knowledge-based system consisting of a Multilayer Perceptron (MLP) and a rule-based system for hybrid knowledge representations and reasoning. Knowledge embedded in the trained 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, 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.

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

1 Motivation

The knowledge acquisition bottle-neck is a major obstruction to knowledge engineering. The technology of Artificial Neural Networks (ANNs) provides a helpful approach to get around it. However, the black-box nature of ANNs makes users reluctant to use them. An optimally organised hybrid system, which includes an ANN fulfilling automatic knowledge acquisition and a Rule-based System (RBS) supporting it with a symbolic inference engine and user interface, can overcome those problems and provide richer knowledge representations and reasoning facilities than the ANN.

The central themes of hybrid system methodology include the following two considerations: (i) the optimal format of the symbolic knowledge representation and (ii) the rule extraction method which transfers the subsymbolic knowledge acquired by the ANN into the symbolic knowledge format accurately, abstracdy and efficiently.

This paper presents a common symbolic knowledge format — general rules — in Section 2. These are 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. A binary problem is defined as a triple $<\{0,1\}^N, \{0,1\}^M, F>$, where F is a relationship or a function of the N dimensional inputs to the M dimensional outputs. Knowledge of a binary problem can be represented in three formats in GR2: the training pattern set, the trained MLP and a symbolic form — the general rules.

As we are interested in the relationships between the input bits and each individual output bit, a pattern can be partitioned into M *sole patterns* $<(I_1, I_2, .. , I_N), O_o>$, where I_i (i=1..N) is the *ith* instantiated input bit, and the O_o is the *oth* instantiated output bit.

An *MLP* in this paper has 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, F, in a binary problem.

A *General Rule*, or a rule for short, comprises a premise part and a conclusion part, having the form

$$\text{IF } (\alpha_1, \alpha_2, ... \alpha_L) \text{ THEN } (\gamma)$$

where $1 \le L \le N$, α_is 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*; otherwise, the rule is a *negative rule*. It is notable that both α_j and γ are instantiated binary variables, not only binary values. This form of rules is equivalent to the Horn Clause Format.

A general rule is, syntactically, a production rule. As L<N, however, the corresponding absent premises from a general rule are defined as being insignificant and thus can be ignored. A general rule represents a set of sole patterns whose outputs correspond to the conclusion of the general rule, and whose input vectors correspondingly subsume the premise part of the general rule. The set of sole patterns a rule represents is called its *coverage*. The fewer premises a rule has, the more general it is, and the larger its coverage is.

Comparing the three formats, the sole patterns can appear contradictory, reflecting the noisy property or uncertain origins of the given problem. The trained MLP encodes the knowledge in an implicit format and unifies contradictory training patterns by some statistical treatment. The general rule is superior to the two previous formats: it is abstract and explanatory. In addition, probability is naturally incorporated by rules which embody inherent uncertainty or incomplete knowledge. The generality of the representation in general rules supports multiple inference facilities to be discussed in the following subsections.

2.2 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 necessarily held. A *Confidence Factor* is the additional component of a probabilistic rule. It counts those sole patterns correctly classified or misclassified in the rule coverage:

$$cf(R_i)=(\sigma-\varepsilon)/(\sigma+\varepsilon)$$

where σ is the number of sole patterns correctly classified, and ε is the number misclassified. The value range is [-1, 1]. When the confidence factor is 1, the rule is equivalent to a categorical rule. The confidence factor provides a better capability for classification under uncertainty. GR2 classifies sole patterns in two sequential 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 opposite rules.

2.3 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.

If there are only positive or negative rules to an output variable, 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 Rule Extraction

Rule extraction methods are released in Black-box [Saito and Nakano, 1988] and/or Open-box (White-box) [Fu, 1994; Towell and Shavlik 1991, 1993a, 1993b] approaches. GR2 takes advantage of the synergy of both approaches. In the Open-box approach, the weights of the MLP are explored and a static linear statistical property of the MLP is obtained. In the Black-box approach, the Input/Output behaviour of the MLP is observed for examining the salient individual features in the context of the training 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 subsections respectively. This method is first introduced by us in [Ma *et al.*, 1995], and improved on in this paper. This method does not require any special modification to the MLP and is effective in both information dense cases, such as most artificial binary problems, and information sparse, real-world domains.

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, named *Static Link*, between the *ith* input unit I_i and the *oth* output unit O_o.

Observation is isolated only to 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, where the δ is the tolerance used in the MLP test (classification) stage, switching I_i will possibly not impact the classification result represented by O_o. I_i may be ignorable in this circumstance. Similarly, if I_i is 1 and O_o is in the range $[0, \delta]$, switching I_i may not change the O_o's status either. I_i may therefore be ignorable. The situations are reversed as $L_{oi}<0$. These are summarised in Table 1.

Table 1: Contingencies when I_i may be ignorable

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

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 the *oth* row from the matrix L, $L_o=\{L_{oi}\}$, we define two sets (Note: there is an overlap as $L_{oi}=0$):

$$Z_1=\{I_i \mid L_{oi} \geq 0\} \qquad N_1=\{I_i \mid L_{oi} \leq 0\}$$

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

$$Z_0=\{I_i \mid Ii \in [1-\delta, 1]\} \qquad N_0=\{I_i \mid Ii \in [0, \delta]\}$$

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

$$(Z_0 \cap N_1) \cup (N_0 \cap Z_1) \quad \text{if } O_o=1 \text{ or } O_o \in [1-\delta, 1]$$
$$(Z_0 \cap Z_1) \cup (N_0 \cap N_1) \quad \text{if } O_o=0 \text{ or } O_o \in [0, \delta]$$

The elements of the PDS are the candidates possibly absent from the rules extracted from a sole pattern $<I, O_o>$.

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 within 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. Note: point (a) is true for most well known MLPs; point (b) can be always satisfied too, since δ can be loosely assigned as long as all patterns for test are uniquely classified by the MLP, rather than being as restricted as Δ — the tolerance for MLP *training*.

The PDS represents a statistical property of the trained MLP. It averages to half the size of the input vectors, from empirical observations. Hence the dimensionality of the test space on the input values is 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 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

$$fsd_{ji} = \sum_{\{k \mid (j \neq k,\ o_o^j \neq o_o^k,\ I_{ji} \neq I_{ki})\}} e^{-|P_j, P_k|}$$

where I_{ji} and I_{ki} are the *ith* input values respectively in the 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. $|P_j, P_k|$ is the hamming distance between the input vectors 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, whose output variable and the *ith* input variable instantiated by different values from those in P_j. $e^{-|P_j, P_k|}$ 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 conveyed by the input units in the context of the training set. It represents the correlation of the changes on the input variables and an output variable, estimating the possibility of a change of the output status when the input variables are switched.

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

3.3 Rule Extraction with PDS and FSD

There is a parameter, the FSD threshold, τ, used to control the generality of the extracted rule set. τ is used to decide if a set of input bits is preserved for forming premises in an extracted rule. It should be within the range (0,1]. A default value 0.4 is recommended for τ.

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.

Step 1. Compute PDS and FSD. (FSD is built up once for all sole patterns regarding the same output variable O_o).

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

Step 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 $\theta_k=\{I_i \mid I_i \notin PDS, FSD_{ji}<\tau, \sum_i FSD_{ji} \geq \tau/N^{1/2}$

Step 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 consequence of the rule. 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 [Ma *et al.*, 1995], where a comparison with other relevant work was addressed too. In fact, the computation is mostly consumed in P calls to 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 FSD threshold τ. The higher the τ, the more general the rule set is, and vice versa.

4 GR2 System Architecture

The GR2 system is depicted in Figure 1. The first component, an MLP has 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 here. The Rule Validation process includes several functions:

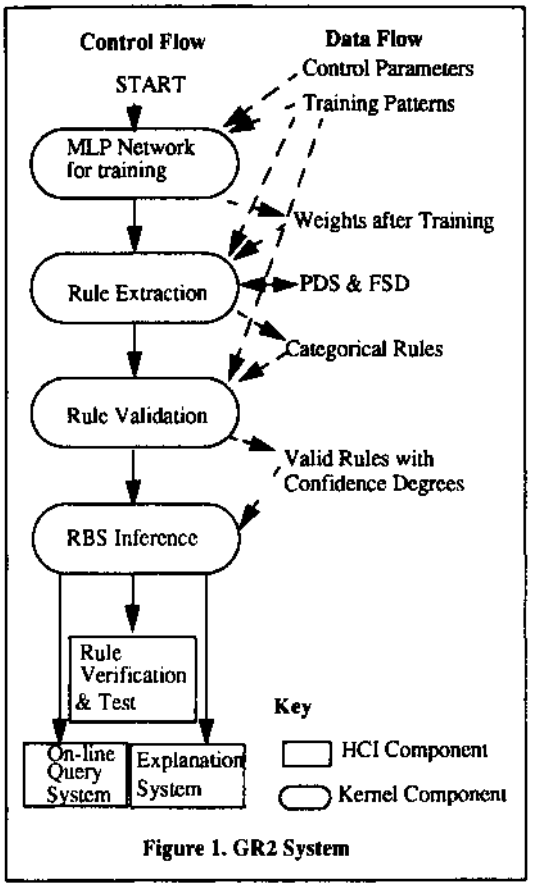


Figure 1. GR2 System

•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 directly mapping from the input space to the output space, no intermediate variables being involved. The inference process is

* If rule A is more special than rule B, A and B must have the same consequence. The set of premises in A subsumes the set in B.

•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.

There are three user interface components being implemented, which are not addressed here owing to the limited space.

5 Examples

The GR2 has been successfully applied to many typical artificial binary-valued problems, such as the two or more bit AND, OR and parity problems, and to 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 of the rule set are useful.

In Section 5.1, a four bit parity problem with an incomplete training set is presented. One of the real-world domains is discussed in Section 5.2. More examples have been demonstrated in [Ma *et al,* 1995].

5.1 Incomplete four Bit Parity Problem

Learning capability is assessed by the accuracy of recognition of the patterns not included in the training set. This section shows how GR2 tackles this situation.

Given a training set with four binary inputs, named A B C D, and one binary output, named E, it includes 11 patterns instead of the complete set of 16 patterns. The included patterns in the training set are assigned as a part of a four bit parity problem. All possible patterns are listed in Table 2 and the shaded columns are excluded from the training set.

Table 2: Patterns of Incomplete 4 Bit Domain (shaded patterns are not in the training set)

Label	P1	P2	P4	P7	P8	P9	11	12	14	15	16
A	0	1	1	0	1	0	0	1	1	0	1
B	0	0	1	1	1	0	1	1	0	1	1
C	0	0	0	1	1	0	0	0	1	1	1
D	0	0	0	0	0	1	1	1	1	1	1
E	0	1	0	1	1	1	0	1	1	1	0

After training, the MLP (size 4:3:1), classifies all 16 input vectors. The conclusions are rounded into integers, including those patterns absent from the training set. All classification results are also shown in Table2, where the training patterns are exactly recognised as designed and the untrained input vectors are classified too in the shaded columns.

After the rule extraction, the General Rules extracted from this trained MLP are

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

IF (~A, C, D) THEN (E).

The rule validation shows that all the rules are valid and the confidence degree for each rule is 1. Because each rule correctly covered some training patterns and there are no training patterns in conflict with it.

At rule test stage, the rules are applied on the patterns absent from the training set:

For input <0 1 0 0> in pattern 3, 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 pattern 5, 1 rule covers it,
IF (~B, C) THEN (E);
For <1 0 1 0> in pattern 6, 3 rules cover it,
IF (A, C, ~D) THEN (E); IF (~B, C) THEN (E);
IF (A, ~B) THEN (E);
For <1 0 0 1> in pattern 10, 3 rules cover it,
IF (A, ~B) THEN (E); IF (~B, D) THEN (E);
IF (A, ~C, D) THEN (E);
For <0 0 1 1> in pattern 13, 3 rules cover it,
IF (~B, D) THEN (E); IF (~B, C) THEN (E);
IF (~A, C, D) THEN (E);

The rules are uniform at each case. All conclusions are the same as given by the MLP as expected.

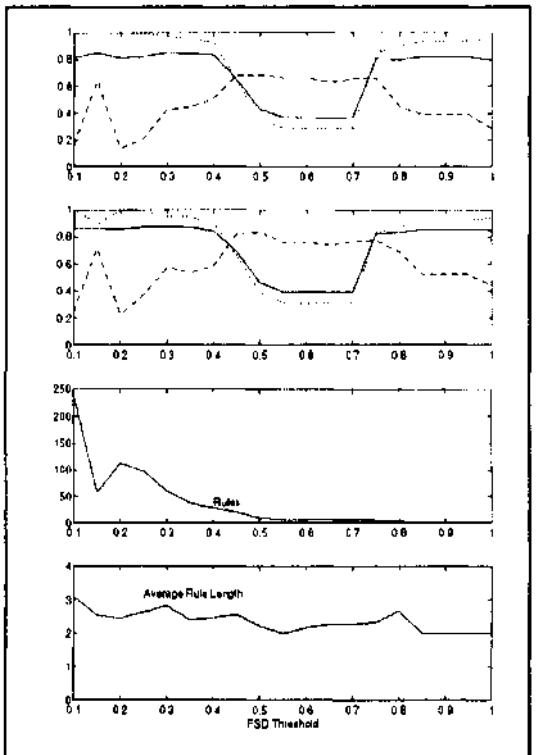
5.2 Diagnosis of Acute Myocardial Infarction (Heart Attack)

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 [Adams *et al.,* 1993b; Stark and Vacek, 1987]. 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 [Adams *et al.,* 1993a]. 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 [Kennedy *et al.,* 1994]. 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 A=0.15. Because the positive and negative patterns are unevenly distributed in the data set, 192 and 778 respectively, random divisions of the training set and test set may result in very different outcomes. The 970 patient records were divided into two data sets, 500 randomly selected as the training set, and the remaining 470 as the test set.

There are three performance criteria on the data set, being used in the medical community. *Sensitivity* is defined as the ratio of the number of correct positive diagnoses to the number of positive outcomes. 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 outcomes. 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.

Figure 2 displays the performance of GR2 on this domain



Diag. a and b. the results by the extracted rules at different τ values on the training set and the test set respectively
— accuracy - - sensitivity ··· specificity
Diag. c. the number of the extracted rules at different τ values
Diag. d. the average rule length at different τ values

Figure 2. GR2 performance on AMI diagnosis data

as the FSD threshold x changes in the range [0.1,1]. It is important that trading-off the values of sensitivity and specificity for different medical requirements is easy to do by simply changing x.The extracted rules are not given in this paper because of the space limit. The results on the training set (Diag. a) and the test set (Diag. b) are similar. The specificity and accuracy curves are closely correlated because the majority cases in both data sets are negative. The accuracy and specificity are at relatively high levels as t is in the range [0.1, 0.4]. They decline for x in the range [0.45, 0.7] which leads to fewer rules extracted. For x>0.7, the specificity and accuracy subsequently increase because the extracted rules appear solely in positive form and exceptional reasoning is in force. The number of the rules (Diag. c) is generally reduced as x increases. But the average length of the rules (Diag.d) does not change much.

The rule extraction processes took between 6 and 36 seconds on Sun Sparc 10; Rule Validation processes took 4 - 11.66 seconds; and Rule Reasoning processes on all the test set took 0.33 to 7.7 seconds.

Table 3 compares part of the experiment results on different platforms such as MLP, GR2 and C4.5 [Quinlan, 1993], in which we believe that the poor outcome at the sensitivity by C4.5 extracted rules may be caused by our unfamiliarity with the use of C4.5 at present.

Table 3: Experiments on AMI Diagnosis Records

(%)	MLP	GR2 τ=0.4	GR2 τ=0.75	C4.5 Dec Tree	C4.5 Rules
Sen on Tra	100	53.3	67.5	68.5	25.7
Spc on Tra	100	92.4	88.2	98.3	96.7
Acc on Tra	100	86.3	82.0	92	81.4
Sen on Test	55.9	57.6	79.4	61.6	20.2
Spc on Test	91.5	93.3	84.2	94.6	95.8
Acc on Test	84.5	87.0	83.1	88.9	81.9

In the first column, Sen denotes Sensitivity, Spe denotes Specificity, Acc denotes Accuracy, Tra denotes Training Set, and Test denotes Test Set.

6 Conclusion and Further Work

The general rule is a format representing only important 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 extracted rule set is easily adjustable by varying the threshold of the feature salient degree.

We are expanding GR2 with more functions such as online knowledge acquisition and explanation. The former guides users by giving queries sensitive to dynamic context, achieving 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

[Adams *et al.*, 1993a] 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)

[Adams *et al.*, 1993b] 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)

[Bochereau and Bourgine, 1990] L. Bochereau, P. Bourgine. *Rule extraction and validity Domain on a Multilayer Neural Network.* International Joint Conference on Neural Networks, 1990 Vol1 pp97-100

[Fu, 1994] L.M.Fu. *Neural Networks in Computer Intelligence.* Chapter 14, 1994, McGraw-Hill

[Kennedy *et al.*, 1994] 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

[Ma *et al.*, 1995] Z. Ma, R. F. Harrison, R. L. Kennedy. *A Heuristic for General Rule Extraction from a Multilayer Perceptron.* Hybrid Problems, Hybrid solutions, IOS Press, Edited by J. Hallam, 1995, pp133-144

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

[Saito and Nakano, 1988] K. Saito, R Nakano. *Medical Diagnostic Expert System Based on PDP Model.* International Conference on Neural Networks, IEEE press, San Diego CA, 1988, pp255-262

[Stark and Vacek, 1987] 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.

[Towell and Shavlik, 1991] G. Towell, J. W. Shavlik. *Extracting refined rules from knowledge-based neural networks.* Machine Learning, vol.13, no. 1,p.71-101

[Towell and Shavlik, 1993a] G. Towell, J. W. Shavlik. *Interpretation of Artificial Neural Networks: Mapping Knowledge-Based Neural Networks into Rules.* IJCAI 93, pp977-984

[Towell and Shavlik, 1993b] G. Towell, J. W. Shavlik. *Using Symbolic Learning to Improve Knowledge-Based Neural Networks.* AAAI 93