

# Prediction using a Symbolic Based Hybrid System

Richard Dazeley and Byeong-Ho Kang

School of Information Technology and Mathematical Sciences,  
University of Ballarat, Ballarat, Victoria 7353, Australia.  
School of Computing and Information Systems,  
University of Tasmania, Hobart, Tasmania, 7001.  
[r.dazeley@ballarat.edu.au](mailto:r.dazeley@ballarat.edu.au), [bhkang@utas.edu.au](mailto:bhkang@utas.edu.au)

**Abstract.** Knowledge Based Systems (KBS) are highly successful in classification and diagnostics situations; however, they are generally unable to identify specific values for prediction problems. When used for prediction they either use some form of uncertainty reasoning or use a classification style inference where each class is a discrete predictive value instead. This paper applies a hybrid algorithm that allows an expert's knowledge to be adapted to provide continuous values to solve prediction problems. The method applied to prediction in this paper is built on the already established Multiple Classification Ripple-Down Rules (MCRDR) approach and is referred to as Rated MCRDR (RM). The method is published in a parallel paper in this workshop titled *Generalisation with Symbolic Knowledge in Online Classification*. Results indicate a strong propensity to quickly adapt and provide accurate predictions.

**Keywords.** knowledge based systems, knowledge representation, prediction, ripple-down rules

## 1 Introduction

Knowledge Based Systems (KBS) have illustrated the ability to capture complex human knowledge and experience, which they can then apply to classification and diagnosis. However, when applied to prediction problems they either rely on uncertainty modeling, a specialized form of classification or hard coded mathematical functions. When using uncertainty modeling their predicted value is a probability, confidence or measure of membership in a classification rather than a true prediction of a value. Likewise, when using classification techniques they rely on having a classification for each predictive value instead. In this paper, prediction is regarded as the process of providing a single value from a continuous range rather than a membership of a class. This value may be a stock price of a company, relevance rating of a document, or a thrust level on a satellite stabilizer.

The aim of this paper is to introduce a method particularly well suited to the application of human knowledge for the problem of prediction. One significant advantage of directly incorporating incrementally acquired human knowledge is that it potentially can significantly improve the speed of learning. The method in this paper applies human knowledge yet also learns a generalisation of this knowledge that allows for value prediction. This paper is broken into two main sections. The first section will provide a brief introduction in to the methodology applied. The second section will describe the experimental method and give a number of results detailing the systems ability to predict.

## **2 Methodology**

The approach developed in this paper is a hybrid methodology, referred to as Rated MCRDR (RM), combining Multiple Classification Ripple-Down Rules (MCRDR) [1-6] with a function fitting technique, namely an artificial neural network (ANN). This hybridisation was performed in such a way that the function fitting algorithm learns patterns of conclusions found during the inferencing process. The method in this paper has been fully detailed in a parallel paper, also published in these proceedings (see [7]).

Basically, the system discussed in this paper is designed to recognize patterns of rules and classifications for particular cases and to attach a weighting to this observed pattern. Nowhere in the actual knowledge map is this information actually recorded; it is simply derived information from the pattern of rules evaluated in the MCRDR tree. This pattern exists because there is either a conscious or subconscious relationship between these classes in the expert's mind. The ANN can then be applied to learn a range of tasks. In this paper we evaluate the method's ability to learn a continuous value in a prediction environment.

One potential application for such a method could be in intelligent agents such as an email agent. Using knowledge gathered from an expert when they organize their email, combined with details such as their speed in replying, saving, or deleting, to determine a level of importance to the user. This learnt weighting could then be used to determine values of importance for future emails, which could be used to decide whether to inform the user of the email.

## **3 Experiments and Results**

This section's results illustrate how RM compares against a backpropagation neural network. Backpropagation was used as this matched the underlying network used in RM. In this paper RM and the ANN are compared in two environments: generalisation and online prediction. This section consists of a discussion of the experiments performed and the simulated expert and dataset used for the experiments. Secondly, this section will provide results and a discussion illustrating how RM compares against the ANN.

### 3.1 Experimental Method

In the prediction domain RM and the ANN must output a single value, which must be as close to the expected value as possible. In the collection of results presented in this paper each test used 10 different randomisations of the dataset. The first, *generalisation* test, divides each dataset into ten equal sized groups. Results are presented where 9/10<sup>ths</sup> of the dataset are used for training and 1/10<sup>th</sup> for testing. The size of the training set is then reduced incrementally in steps of 1/10<sup>th</sup>, down to 1/10<sup>th</sup>. The same 1/10<sup>th</sup> set is always used as the test set. The *online* prediction test investigates how the methods can correctly predict values over time. In this test the entire dataset is broken up into smaller blocks, each 1/50<sup>th</sup> of the original dataset, and passed through the system one group at a time. The system's performance is recorded after each group. The value returned is then compared to the simulated expert's correct value. The absolute difference between these two values (*error*) is then averaged over all the cases in the data segment and logged.

### 3.2 Simulated Expertise

One of the greatest difficulties in KA and KBSs research is how to evaluate the methodologies developed [8]. The method used by the majority of RDR based research has been to build a simulated expert, from which knowledge can be acquired [8]. It is this approach that has been taken in this paper. However, testing RM using simulation has an added difficulty. This is because available datasets do not give both symbolic knowledge and a target value instead of a classification. This could be partially resolved by assigning each classification a value. However, fundamentally this would still be a classification type problem.

The approach taken in this paper was to develop a heuristic based simulated expert, which has two stages in calculating a value for a case based on a set of randomly generated attributes. The first stage uses a randomly generated table of values, representing the level that each attribute,  $a \in A$ , contributes to each class,  $c \in C$ . This classification stage is merely an intermediate step to finding a rating for the case. It is also used during knowledge acquisition for identifying relevant attributes in the difference lists. When creating a new rule, the expert selects the attribute from the difference list that distinguishes the new case from the cornerstone case to the greatest degree. This was achieved by locating the most significant attribute, either positively or negatively, that appeared in the difference list (see example in Table 1).

	a	b	c	d	e	f	g	h	i	j	k	l
C1	0	0	-1	3	0	0	0	0	0	0	-1	3
C2	0	0	0	-2	2	0	0	-2	0	0	1	0
C3	0	-2	1	0	0	0	0	0	0	1	0	-1
C4	-1	3	0	0	0	0	1	0	-1	0	0	0
C5	0	0	0	0	-2	2	-2	0	2	0	0	0
C6	2	0	0	0	0	-2	0	1	0	-2	0	0

**Table 1.** Example of a randomly generated table used by the *non-linear multi-class* simulated expert. Attributes a - l are identified across the top, and the classes C1 - C6 down the left.

Case A = {a, b, c, d}							Case B = {a, c, e, g}						
Attributes	Classifications						Attributes	Classifications					
	1	2	3	4	5	6		1	2	3	4	5	6
a	0	0	0	-1	0	2	a	0	0	0	-1	0	2
b	0	0	-2	3	0	0	c	-1	0	1	0	0	0
c	-1	0	1	0	0	0	e	0	2	0	0	-2	0
d	3	-2	0	0	0	0	g	0	0	0	1	-2	0
Total	2	-2	-1	2	0	2	Total	-1	2	1	0	-4	2
Classified	✓	✗	✗	✓	✗	✓	Classified	✗	✓	✓	✗	✗	✓

**Table 2.** Two example cases being evaluated by the classification component of the simulated expert.

Table 2, gives two example cases each with 4 attributes where the method for calculating the case's appropriate classification can be seen. Each attribute contributes a value for the class. The simulated expert's resulting classification for both of these cases are {1, 4, 6} for case A and {2, 3, 6} for case B.

To fully push the system's abilities, the rating calculated by the simulated expert needs to generate a non-linear value across the possible classifications. The implementation used for prediction generates an energy space across the level of class activations, giving an energy dimensionality the same as the number of classes possible. Each case is then plotted on to the energy space in order to retrieve the case's value. First, the strength of each classification found is calculated. As previously discussed a case was regarded as being a member of a class if its attribute value was greater than 0. However, no consideration was made to what was the degree of membership. In this expert the degree of the case's membership is calculated as a percentage,  $p$ , of membership using Equation 2.

$$p = t^a / t^m \quad (2)$$

This is simply the actual calculated total,  $t^a$ , divided by the maximum possible total,  $t^m$ , for that particular class. Extending the example from Table 2 for case A, classification C1, the total 2 is divided by the best possible degree of membership 6, from Table 1, thereby, giving a percentage,  $p$ , membership of 33%. This calculation is performed for each class. Each class then has a randomly selected point of highest value, or centre,  $c$ , which is subtracted from the percentage and squared, Equation 3. This provides a value which can be regarded as a distance measure,  $d$ , from the centre. This distance measure can be *stretched* or *squeezed*, widening or contracting the energy patterns around a centre, by the inclusion of a width modifier,  $w$ .

$$d = w ( p - c )^2 \quad (3)$$

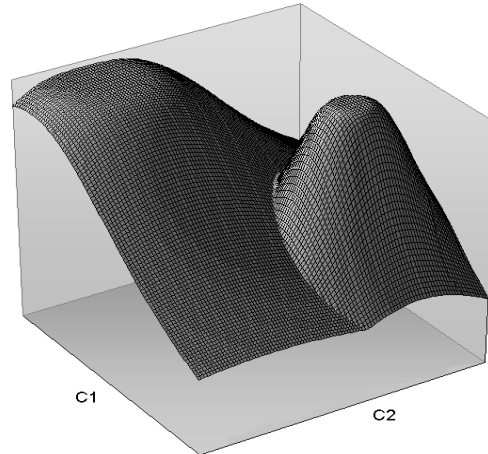
The classes' centres are combined to represent the point of highest activation for the expert, referred to as a *peak*. Therefore, if the square root of the sum of distances is taken then the distance from this combined centre can be found. This distance can then be used to calculate a lesser value for the case's actual rating. Therefore, as a case moves away from a *peak* its *value* decreases. Any function can be used to calculate the degree of reduction in relation to distance. In this paper a Gaussian

function was used. Equation 4 gives the combined function for calculating a value for each possible peak,  $v^p$ , where  $n$  is the number of classes in the dataset.

$$v^p = \frac{1}{1 + e^{-0.5 \left( \frac{1}{\sqrt{\sum_{j=0}^n \left( \frac{t_j^a}{t_j^m} \right) w_j - c_j}} \right)^2}} - 0.5 \quad (4)$$

Finally, it is possible to have multiple peaks in the energy space. In such a situation each class has a centre for each peak. Each peak is then calculated in the same fashion as above, resulting in a number of values, one for each peak. The expert then simply selects the highest value as the case's actual rating. This rating method is best understood by looking at a three dimensional representation shown in Fig 6.

The third dimension, shown by the height, illustrates the value at any particular point in the energy space. This figure shows a dataset with only two possible classes, C1 and C2, and two peaks. A three class dataset cannot be represented pictorially. The advantage of this approach is that it generates an energy pattern that is nonlinear. At no location can a straight line be drawn where values are all identical.



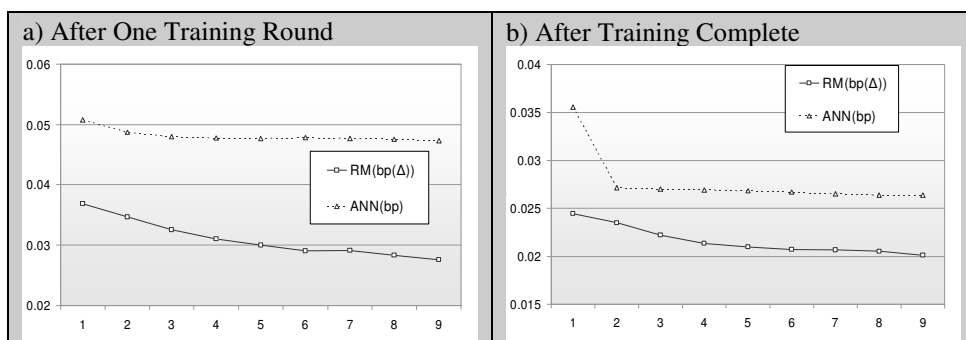
**Fig. 6.** Example of a possible energy pattern used in the Multi-Class-Prediction simulated expert. This would be used for a dataset with two possible classifications. This energy pattern contains two randomly located peaks.

### 3.3 Dataset

The method was tested using a randomly generated group of attributes that could be classified and rated by the above simulated expert. For instance, the environment setup in this paper allows for 12 possible attributes. In the tests carried out in this paper each case selected 6 attributes, giving a possible 924 different cases. Therefore, in each 1/10<sup>th</sup> group there are 92 cases and 18 cases in each 1/50<sup>th</sup> group.

### 3.4 Prediction Generalisation

The ability of a method to generalise is measured by how well it can correctly rate cases during testing that it did not see during training. The value returned by RM and the ANN is then compared to the simulated expert's correct value. The absolute difference between these two values (error) is then averaged over all the cases in the data segment and logged. The results shown in Fig 7 show they each performed. Each point on the charts is the average error for the test data segment averaged over ten randomised runs of the experiment, for each of the nine tests. To reduce the complexity of the charts shown, error bars have been omitted.

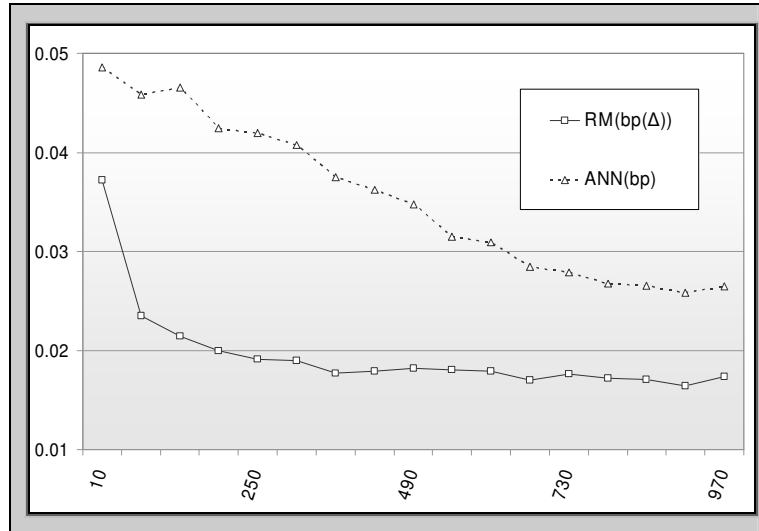


**Fig. 7.** (a) – (b) Two charts comparing how RM and ANN. Chart a) shows how the methods compare after only one viewing of the training set. Chart b) shows how the methods compare after training was completed. The x-axis shows how many tenths of the dataset were used for training. All results used the last tenth for testing. The y-axis shows the average error.

These results show that the RM hybrid system has done exceptionally well both initially as well as after training is complete when generalising. Additionally, it can be observed that the neural network was unable to significantly improve with more training data. This problem is caused by the network having consistently fallen into local minimum, a problem common to neural networks especially in prediction domains. RM is less likely to encounter this learning problem as the knowledge base provides an extra boost, similar to a momentum factor, which propels it over any local minima and closer to the true solution. Therefore, not only does RM introduce KBSs into potential applications in the prediction domain, as well as, allow for greater generalisation similar to an ANN, but it also helps solve the local minima problem.

### 3.5 Prediction Online

The process of RM being able to predict an accurate value in an online environment could potentially allow the use of RM in a number of environments that have previously been problematic. For instance, KBSs in *information filtering* (IF) have difficulties due to their problems in prediction, while neural networks are far too slow. RM allows for the inclusion of expert knowledge with the associated speed but also provides a means of value prediction. Fig 8 shows a comparison between RM



**Fig. 8.** This chart compares how RM and an ANN, perform in an online environment. The x-axis shows the amount of 1/50<sup>th</sup> data segments that have been seen. The y-axis shows the average error over the last 10 data segments, also averaged over 10 trials.

and an ANN in an online environment. Here it can once again observed that RM has performed outstandingly well from the outset and was able to maintain this performance. This fast initial learning can be vital in many applications as it is what users usually expect.

#### 4. Conclusion

This paper presented a hybrid algorithm that allows an expert's knowledge to be adapted to prediction problems. The method developed builds on the already established Multiple Classification Ripple-Down Rules (MCRDR) approach and was referred to as Rated MCRDR (RM). RM retains a symbolic core while using a connection based approach to learn a prediction value.

This method has been applied to a prediction domain where results indicate a strong propensity to quickly adapt and generalize, providing accurate predictions. RM's ability to perform well can be put down to two features of the system. First, is that the flattening out of the dimensionality of the problem domain by the MCRDR component allows the system to learn a problem that is mostly linear even if the original problem domain was non-linear. This allows the network component to learn significantly faster. Second, the network gets an additional boost through the *single-step-Δ-initialisation rule*, allowing the network to start closer to the correct solution when knowledge is added. A prediction method, such as this, that relies on symbolic knowledge for rapid learning, is particularly useful in a number of domains such as information filtering, prudence analysis and anomaly detection.

## Acknowledgements

The majority of this paper is based on research carried out while affiliated with the Smart Internet Technology Cooperative Research Centre (SITCRC) Bay 8, Suite 9/G12 Australian Technology Park Eveleigh NSW 1430 and the School of Computing, University of Tasmania, Locked Bag 100, Hobart, Tasmania.

## References

1. Compton, P., and Jansen, R., (1988) Knowledge in Context: a strategy for expert system maintenance, *Second Australian Joint Artificial Intelligence Conference (AI88)*, Vol. 1, pp. 292-306.
2. Compton, P., Edwards, G., Kang, B., (1991) Ripple Down Rules: Possibilities and Limitations, *6th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW91)*, Vol. 1, SRDG publications, Canada, pp. 6.1-6.18.
3. Compton, P., Kang, B., Preston, P., (1993) Knowledge Acquisition Without Knowledge Analysis, *European Knowledge Acquisition Workshop (EKAW93)*, Vol. 1, Springer, pp. 277-299.
4. Kang, B., (1996) Validating Knowledge Acquisition: Multiple Classification Ripple Down Rules (PhD thesis),
5. Kang, B.H., Compton, P., and Preston, P., (1995) Multiple Classification Ripple Down Rules: Evaluation and Possibilities, *The 9th Knowledge Acquisition for Knowledge Based Systems Workshop*, SRDG Publications, Department of Computer Science, University of Calgary, Banff, Canada,
6. Preston, P., Compton, P., Edwards, G., (1996) An Implementation of Multiple Classification Ripple Down Rules, *Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop*, SRDG Publications, Department of Computer Science, University of Calgary, Calgary, Canada UNSW, Banff, Canada.
7. Dazeley, R. P. and Kang, B. H (2008) Generalisation with Symbolic Knowledge in Online Classification, *The 2008 Pacific Rim Knowledge Acquisition Workshop (PKAW 2008)*, Springer (in Press).
8. Compton, P., (2000) Simulating Expertise, *Proceedings of the 6th Pacific Knowledge Acquisition Workshop*, Sydney, Australia, pp. 51-70.