RULES EXTRACTION WITH AN IMMUNE ALGORITHM

Deqin Yan* and Liping Zhang

Department of Computer Science, Liaoning Normal University, Dalian 116029
**Email:* yandeqin@163.com

ABSTRACT

In this paper, a method of extracting rules with immune algorithms from information systems is proposed. Designing an immune algorithm is based on a sharing mechanism to extract rules. The principle of sharing and competing resources in the sharing mechanism is consistent with the relationship of sharing and rivalry among rules. In order to extract rules efficiently, a new concept of flexible confidence and rule measurement is introduced. Experiments demonstrate that the proposed method is effective.

Keywords: Immune algorithm, Rule extraction, Data mining, Knowledge discovery

1 INTRODUCTION

Classified rule extraction is an important aspect in the data mining. Usually, a rule extraction method is based on sample learning by using some classification method to obtain the classified rules. The techniques used in classification rule methods mainly include decision trees, neural networks, and the genetic algorithm. However, by using the decision tree method, it is difficult to mine large data sets. Sometimes, the meaning of rules produced by the neural network method is hard to be understood. The standard genetic algorithm can possibly have degenerated phenomenon in the evolution process, and it is unable to express relationships of complementing and competing among rules. When dealing with multi-peaks search issues, it is often converged to the point of sub-optimal solution. In order to extract rules better, a method based on an immune algorithm with flexible confidence is proposed in this article. Experimental results have confirmed the validity of the new algorithm.

2 SIGNIFICANCE MEASURE METHOD OF THE RULES

In this article, a rule is expressed in the form of $A \Rightarrow B$, where A is the conjunction of condition attribute values called the rule's *antecedent*, and B is the value of a decision attribute called the *consequent*. If some condition attributes are unimportant to the decision, that is to say, whatever values the condition attribute takes do not affect the decision, these positions are expressed with zeros.

When extracting rules, it is extremely important to select an appropriate rule measure. There are many kinds of rule measure methods previously proposed. Let $||$ represent the cardinal number of the set. In the literature (Khoo & Zhai, 2001) a genetic algorithm was used to extract rules, and the formula to measure rule is $\left(|A \& B|/|A|\right)^2$. The square operator appears to ensure rapid convergence. He et al. (2005) give the formula $\alpha * str(r) + \beta * cer(r) + \gamma * co(r)$, where $str(r)$, $cer(r)$ and $co(r)$ represent the coverage,

confidence and completeness. α, β, γ are three parameters, which are adjustable according to the actual situation.

When evaluating the significance of the rule, we consider the following several factors:

Completeness. The completeness of the rule, i.e. the proportion of the samples satisfied by the rule antecedent A and the consequent B to the samples satisfied by the rule consequents, is given by $|A \& B|/|B|$, denoted by x_1. When the decision class distribution is imbalanced, the rules of the small class frequently are neglected, because they contain few samples. This factor enables the rules of a small class would be extracted. The bigger x_1 is, the more important the rule is to class B.

Coverage. The coverage of the rule, i.e. the proportion of the samples satisfied by the rule antecedent A to the total number of the samples, is given by $|A|/M$, denoted by x_2. It indicates the induction ability of the rule in the sample sets. The bigger x_2 is, the stronger the induction ability is. Such rule is the important one.

The scale of rules. The scale of rules is the number of samples satisfied by the rule antecedent A in the sample space, given by $|A|^* \left(|A|^* \geq |A|\right)$, denoted by N. This factor indicates the predictive ability of the rule. When we extract rules, a rule that has the bigger value of N is desired. The bigger N is, the stronger the predictive ability of the rule is.

Confidence. The confidence of rule refers to accuracy, denoted by imp. The common confidence measure method is given by $|A \& B|/|A|$ denoted by x_3. The bigger x_3 is, the more confident the rule is. When we extract rules, a rule that has the bigger value of x_3 is expected. However, the request of excessively high confidence causes the probability of the small extracted rules to increase. On the other hand, it cannot indicate the ability of accommodating noise when dealing with the data which contains little noise. Therefore, in order to extract rules with a higher summary ability, the confidence of a rule may be reduced properly, especially when high confidence is not essential.

Generally, there are two methods in dealing with confidence of a rule. The first method is using primitive confidence as one factor of rules measure (Khoo & Zhai, 2001; He, 2005). A rule with low confidence could still participate in the competition. Although its ability within the competition is weakened, the effect is slight. This obviously does not conform to the original intention of rule extraction. The second method is limiting the confidence of the rule by using the threshold Tr $(0 \leq Tr \leq 1)$. Rules with the confidence lower than Tr are not allowed to compete, namely taking a bi-value function

$$imp = \begin{cases} 1 & x_3 \geq Tr \\ 0 & x_3 < Tr \end{cases}.$$

It is too strict, however, when processing the boundary with bi-value function. If there are two rules $r1$ and $r2$,

and their scale are such that $N_{r1} >> N_{r2}$, the values of the coverage and the completeness are similar. Suppose their confidences are $x_3(r1) = Tr - k$, $x_3(r2) = Tr + k$, where k is a small number. According to the bi-value function, we get $imp_{r1} = 0$, $imp_{r2} = 1$. Thus, the rule $r1$ will lose its competitive ability completely. In fact, the rule $r1$ is not too bad. By inspecting whole rule space, one can see that near any threshold, the primitive confidences of the rules are all extremely close. It is not reasonable to use the bi-value function. Therefore, it is very important to measure the confidence of rules and determine the significance of the rules with high confidence and to process the confidence flexibly. Processing the rules with slightly low confidence, not only decreases their significance, but also allow them to retain some competitive ability.

Here we introduce a new flexible confidence function:

$$imp = (\tanh((x_3 - Tr) \times a) + 1) / 2,$$

where, a is called the gradient factor. The bigger the value of a, the more obviously the value of imp changes nearby Tr.

By adjusting the value of a, values of imp can be improved. The span of the values takes the threshold Tr as the center point. The interval of the span increases as $\boldsymbol{a}$ is reduced. This span is called a flexible confidence interval. Taking a map from the interval to $(0,1)$, if confidence of a rule is within the interval, then we can get a value in $(0,1)$, and this value is taken as new flexible confidence. A rule for which primitive confidence is very high $(>> Tr)$ has the new flexible confidence still very high, approaching to 1. A rule for which primitive confidence is very low $(<< Tr)$ has the new flexible confidence still very low, approaching to 0. A rule for which primitive confidence is near the threshold has the new flexible confidence distributed over the entire interval $(0,1)$. In the precondition of approximately satisfying the demand of a threshold, regarding the confidence threshold as a non- compulsory factor becomes a part of the rule significant measure formula.

From the discussion at above, we give the formula to measure the significance of the rule as:

$$f(x) = (x_1 + x_2) \times N \times imp \qquad (1)$$

3 SIMILARITY

Similarity refers to the similar degree of rules. In this paper, similarity is used to determine the degree of similarity of antibodies in the immune memory system. When we extract rules, the rules with a large degree of similarity are controlled, and the rules covered by others are inhibited. The similarity function is defined as: $sim = \max(s(i, j))$, where, $s(i, j)$ is the similarity between the rule i and the rule j (the rule j is in the

immune memory system). Suppose n is the number of the condition attributes, we have:

$$s(i,j) = \left(\sum_{k=1}^{n} f_k \right) / n .$$

$rule(i,k)$ is the value of the rule i on the condition attribute k. f_k is the similarity between rule i and rule j on condition attribute k.

$$f_k = \begin{cases} 1 & rule(i,k) = rule(j,k) \\ 0 & rule(i,k) \neq rule(j,k) \quad and \quad rule(i,k) \times rule(j,k) \neq 0 \\ -1/a_k & rule(i,k) = 0 \quad and \quad rule(j,k) \neq 0 \\ 1/a_k & rule(i,k) \neq 0 \quad and \quad rule(j,k) = 0 \end{cases} .$$

where, a_k is the number of values on the condition attribute k. If two rules have the same value on a condition attribute, we consider that they are completely similar on this attribute. If their values are different, and neither of them is 0 (any other value), they are completely dissimilar on this attribute. On a condition attribute, if the value of rule i is 0 and the value of rule j is not 0, they are similar with probability of $1/a_k$. Because the value of rule i is 0 (i.e. rule i contains rule j on this attribute), the similarity decreases. In the contrary situation, it increases. The similarity of the rule is the average similarity of other rule on all the condition attributes. The similarity between a rule and the immune system is defined as the maximum value of similarities of the rule to other rules in the immune system.

For instance, suppose there are one decision attribute and three condition attributes in the decision table. Every condition attribute has two values: 1 and 2. If there are three rules in the immunity memory system $isr1:010 \to 1$, $isr2:111 \to 1$, $isr3:201 \to 1$, we can calculate the similarity of rule $ra:110 \to 1$ and $rb:220 \to 1$:

$$\begin{aligned}
&s(ra,isr1) = (1/2+1+1)/3 = 5/6 \\
&s(ra,isr2) = (1+1-1/2)/3 = 3/6 \\
&s(ra,isr3) = (0+1/2-1/2)/3 = 0/6 \\
&sim_{ra} = \max(5/6,3/6,0/6) = 5/6 \\
&s(rb,isr1) = (1/2+0+1)/3 = 3/6 \\
&s(rb,isr2) = (0+0-1/2)/3 = -1/6 \\
&s(rb,isr3) = (1+1/2-1/2)/3 = 2/6 \\
&sim_{rb} = \max(3/6,-1/6,2/6) = 3/6
\end{aligned}$$

In this paper, the affinity of an antibody is adjusted by the similarity. The rules that are similar with the immune system are controlled at a specified level.

4 METHOD OF EXTRACTING RULES BASED ON THE IMMUNE ALGORITHM

The immunity algorithm is inspired by the principles and mechanisms of the biological immune system. In the

immune process, an antigen invades the immune system. Under stimulation by the antigen, the candidate solution, the antibody, is produced. The matching degree of antibody to antigen is called the affinity. In the evolution and competition process of antibodies, an antibody with high-affinity is remembered as part of the immune memory system.

In this paper we describe an immunity algorithm we designed with the abilities of study, recognition, memory, and self- balance that can extract rules efficiently. The algorithm is based on the following immune system:
In our immune system, the learning process is finished by the reaction of antibody to antigen. In the training stage, each sample is regarded as an antigen. All antibodies are deposited in a memory pool, and the number of antibodies is limited to a specified scale. In the evolution process, antibodies with lesser affinity and higher concentration are replaced by those with better affinity and lower concentrations in the memory pool. Concentration control is necessary to keep the diversity of the antibodies. Extracting the feature of an antigen produces a vaccine, and antibodies are vaccinated by partly changing their structure for presenting the same feature. The vaccination operation promotes the efficiency of searching for a better match. In our method, the vaccination operation modifies the condition attribute of rules in order to including sample rules. The antibodies are produced randomly and are vaccinated by mutation. More high quality antibodies are propagated by clonal selection, and then the propagated antibodies are processed with mutation.

The algorithm procedure is as follows:
Initialization: set initial values of threshold Tr and gradient factor a
While (not all samples are learned or learning be finished)
For each sample:
If (sample is recognized) continue learning next sample
else
Initiate primitive antibody population
Vaccination
Calculation affinity with formula (1)
Let affinity=affinity×similarity
While (less than the pre-given number of immune evolution generations)
Propagation and mutation
Select mutated antibody $\boldsymbol{x}1$ that is the beat and has not appeared in memory of immune systems, and $\boldsymbol{x}2$ that is the worst antibody in seed population
If (the affinity of $\boldsymbol{x}1$ > the affinity of $\boldsymbol{x}2$) replaced $\boldsymbol{x}1$ with $\boldsymbol{x}2$
endif
endwhile
Ending evolution process, looking for new antibody which has not appeared in memory pool
endif
endwhile

5 EXPERIMENTAL RESULTS AND ANALYSIS

The algorithm is implemented with three kinds of data: non-noisy consistency data MONK1, noisy data MONK3,

and uniform distributed decision attribute data set, "lenses."

For dataset MONK1, the training data set is MONK1.train. The sum of the samples is M=124, the sum of the condition attributes is C=6, and the sum of the decision attributes is D=1. There are two decision classes. Let the number of elements in the seed population be P=20, the number of elements in memory pool be N=M (supposing the pool is big enough), the number of elements in the propagation pool be Q=100, and the number of the maximum evolution generation be GG=50. We adopt the flexible confidence function:

$$imp(x) = (\tanh((x - 0.85) \times 25) + 1) / 2 .$$

By learning in one pass through, the correct ratios of the extracting rule both for training data and testing data are 100%.

Dataset MONK3 is with 5% noisy data, and the training data set is MONK3.train. The sum of the samples is M=122, the sum of the condition attributes is C=6, and the sum of the decision attributes is D=1. Let the number of elements in seed population be P=20, the number of elements in the memory pool be N=M (suppose the pool is big enough), the number of elements in the propagation pool be Q=100, and the number of the maximum evolution generation be GG=50. We adopt flexible confidence function with different values of threshold Tr and gradient factor a to extract rules. The contrasting results with flexible confidence function and bi-value function are shown in Figure 1. In the figure, the highest position of the line indicates the maximum recognition ratio. The results show that our algorithm does a good job in extracting correct rules for this kind of data.

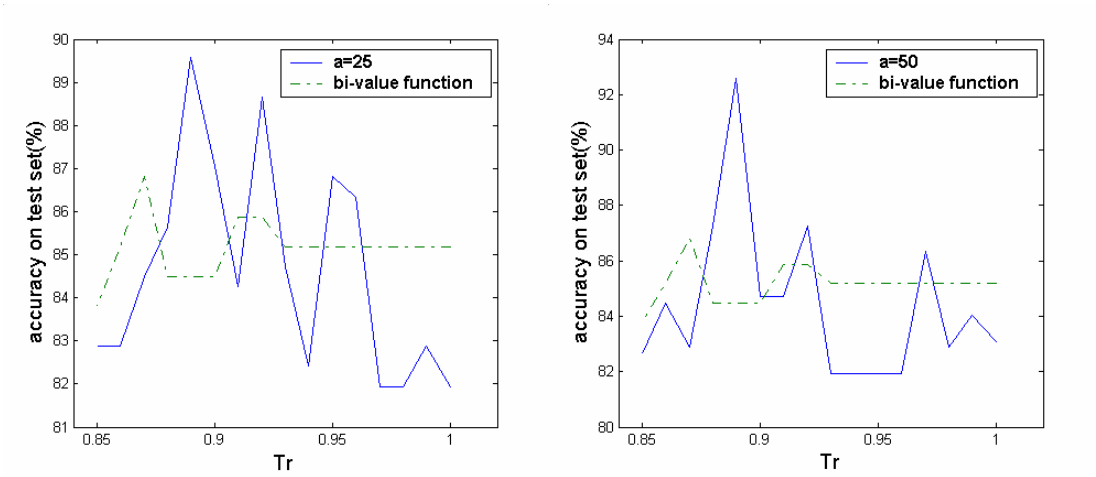

Figure 1. Accuracy of rule extraction for data MONK3

Lenses data are complete, non-conflicting, and non-noisy. There are 24 samples, four condition attributes, and one decision attribute with three decision classes (percentages of the 3 classes in decision attribute are 16.67%, 20.83%, and 62.5% respectively). With threshold $Tr = 0.95$ and gradient factor $a = 25$, the correct ratio of extracting rule is 100%.

6 CONCLUSION

The artificial immune system is an imitation of the vertebrate immune system; the evolution algorithm deduced by this inspiration provides both ideas and techniques to solve problems. In this paper, we describe our design of an immune algorithm that includes the concept of flexible confidence. The proposed method obtains good results not only from data without conflict or noise but also from noise data and data with the imbalance of the class distribution.

7 REFERENCES

De Castro, L. & Von Zuben, F. (2002) Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation 6(3)*: 239-251.

Forrest, S., Javornik, B., Smith, R., & Perelson, A. (1993) Using genetic algorithms to explore pattern recognition in the immune system. *Evolutionary Computation, 3(1):* 191-211.

Freitas, A. (1999) On rule interestingness measure. *Knowledge-Based System 12*: 309-315.

Gao, W. (2004) Fast immunized evolutionary programming. *Proceedings of the Congress on Evolutionary Computation 1:* 666-670.

Ghosh, A. & Nath, B. (2004) Multi-objective rule mining using genetic algorithms. *Information Sciences 163*: 123-133.

He, M. et al. (2005) Genetic algorithm based multi-knowledge extraction method for rough set. *Mini-Micro System (In Chinese) 26*: 651-654.

Khoo, L-P. & Zhai, L-Y. (2001) Aprototype genetic algorithm-enhanced rough set-based rule induction system. *Computers in Industry, 46*: 95-106.

Palmes, P., Hayasaka, T., & Usui, S. (2005). Mutation-Based genetic neural network. *IEEE Transaction on Neural Networks, 16(3):* 587-600.

Timmis, J., Edmonds, C., & Kelsey, J. (2004) Assessing the performance of two immune inspired algorithms and a hybrid genetic algorithm for function optimization. *Proceedings of the Congress on Evolutionary Computation 1*: 1044-1051.