A General Classification Algorithm Based on Immune Network

Xiaohui Huang[1] Wenjian Luo[1] Xufa Wang[1] Jiying Wang[2]

[1] Department of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China

[2]Department of Electronic Science and Technology, University of Science and Technology of China, Hefei, Anhui, 230026, China

Abstract

This paper proposes a classification algorithm based on immune network theory, named after GCA-IN. Based on immune network, the algorithm also employs clone, mutation and bacterin inoculation operators to extract the characteristics of training data. All extracted characteristics are kept in immune network after training. Then the class of testing data is decided by the affinity between itself and all antibodies in immune network after training. In this paper, the principle and steps of the algorithm are given in detail. The experiments using standard test data show that the algorithm has a good ability for classification. To verify the fault tolerance performance of this algorithm, some experiments are done and the experimental results are also given when some antibodies in immune network fail. The experimental results prove that GCA-IN has a good fault tolerant performance.

Keywords: Artificial Immune System; Data Mining; Immune Network; Classification; Fault tolerance

1. Introduction

The immune network theory firstly proposed by N. K. Jerne is an important theory in natural immunology [1]. So far, many researchers have successfully applied the immune network theory to design or improve clustering algorithms [2-4]. As for research on classification algorithm based on immune network theory, Farmer firstly pointed out the similarity between natural immune network and Holland's classification system [5], which proved in some sense that it was possible to design classifier based on Jerne's immune network theory. Jerome Carter designed a classification system based on B-cell, T-cell and their mutual feedback [6]; Andrew Watkins developed a classification algorithm based on Timmis's RLAIS model [7]. So far, their three papers are the only works about classification algorithms based on natural immune system.

Based on both AIS network model and the fraud detection algorithm of Jisys [3, 8], a novel classification algorithm is proposed in this paper by combining immune network theory with immune clone, immune mutation and bacterin inoculation mechanism, named after GCA-IN (a General Classification Algorithm Based on Immune Network). GCA-IN is different from previous works in some aspects. Firstly, every node of immune network in GCA-IN has a class field. Therefore, the nodes of the network can be regarded as some rules and thus the immune network is more comprehensible. Secondly, the computation method of affinity is also different from previous works. Each attribute is assigned a weight based on its information gain. And the weighted sum of match degrees of all attributes is regarded as the affinity. This method took the different importance of each attribute into account. Finally, in order to improve the training speed and classification accuracy of immune network, the algorithm introduces bacterin inoculation operator. The bacterin inoculation operator is designed in some way that an antigen is added into immune network as an inoculation when the antigen has a low affinity with all nodes in immune network.

2. GCA-IN

2.1. Immune network structure

Based on immune network theory [9], an abstract and simplified immune network structure for GCA-IN is given in fig. 1. This simplified immune network is composed of the antibodies (nodes) and their connections (edges). The affinity between two antibodies decides whether they have an edge or not. Only those two antibodies with high affinity between them have an edge. Obviously, the antigens stimulate

the antibodies in immune network to clone and mutate constantly.

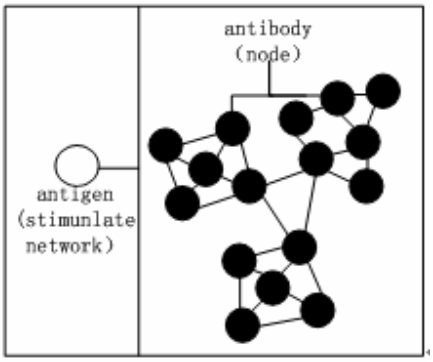

Fig.1 The structure of immune network

2.2. Steps of GCA-IN

Firstly, the structures of the antibody and antigen used in GCA-IN are given in figure 2. The i-th antibody can

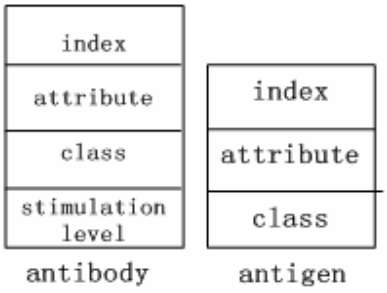

Fig.2 The structure of antibody and antigen

be denoted as $N_i = (i, A_{i1}, A_{i2}, ..., A_{in}, Class_i, Level_i)$, where i is the index of antibody, $A_{i1}, A_{i2}, ..., A_{in}$ are the attributes (data) field, $Class_i$ is class field, $Level_i$ is the stimulation level of this antibody. The i-th antigen can be denoted as $T_i = (i, A_{i1}, A_{i2}, ..., A_{in}, Class_i)$, $A_{i1}, A_{i2}, ..., A_{in}$ is the attributes (data) field and $Class_i$ is class field.

GCA-IN consists of two phases. One is immune network training phase, and the other is classification phase. The immune network training phase is the core of this algorithm.

(a) Phase I: Training the immune network

The detailed steps of the immune network training are given as follows.

The input of Phase I (Training the immune network) of GCA-IN is the training data set $T = \{T_i \mid T_i = (i, A_{i1}, A_{i2}, ..., A_{in}, Class_i)\}(1 \le i \le X)$, in which X is the number of records in training data set.

The output of Phase I (Training the immune network) of GCA-IN is an immune network which consists of a set of nodes (antibodies) and a set of edges (connections). Therefore, the immune network can be denoted as an antibody set N and a connection set C. The i-th antibody N_i in N can be expressed as $N_i = (i, A_{i1}, A_{i2}, ..., A_{in}, Class_i, Level_i)$. The connection set C is composed of several edges which connect the antibodies in immune network. Whether two antibodies have connection or not depends on their affinity. Supposing there exists an edge C_{ij} between antibody i and antibody j, C_{ij} can be recorded as $C_{ij} = (i, j, affinity_{ij})$ where $affinity_{ij}$ denotes the affinity of antibody i and antibody j, the computation method of $affinity_{ij}$ will be introduced in section 2.3.

The steps of the immune network training are given as follows.

(1) An initial network constructing set T' consisted of a selective part of the train data set, which are proportionately selected according to the class of train data set T.

Here the method in reference [8] is used to create an initial immune network. The antibody set N and the connection set C are initialized to be empty. For each data record $T_i \in T'$, T_i is added into the network in the following way. Firstly, the n nodes which have the highest affinity with T_i are selected from antibody set N. These n nodes build up a set of N'. Secondly, for each $N_j \in N'$, if the number of edges of N_j is smaller than m, an edge C_{ij} between T_i and N_j is added into the connection set C. And the weight of the edge C_{ij} is the affinity between T_i and N_j, namely $affinity(T_i, N_j)$. Otherwise, if the number of edges of N_j is larger than m, the edge with the lowest affinity is selected among all edges which are connected with N_j. If the affinity of this selected edge is lower than $affinity(T_i, N_j)$, delete this edge from C and add edge C_{ij} into C. Finally, add T_i into antibody set N. The purpose of this method is to guarantee that only the most similar nodes are connected as well as the number of each node's edges is not greater than m.

(2) For each $T_i \in T$, do:

(2.1)Find n nodes with the highest affinity between T_i and all antibodies in antibody set N. Then n nodes build up a set of N'.

(2.2) For each $N_j \in N'$, if $N_j.Class = T_i.Class$, $N_j.level = N_j.level + affinity(T_i, N_j)$ .Otherwise, $N_j.level = N_j.level - affinity(T_i, N_j)$. The computation method of $affinity(T_i, N_j)$ will be introduced in section 2.3.

(2.3) For each $N_j \in N'$, if $affinity(T_i, N_j) > \alpha$ (clone threshold)， clone N_j. The clone number is $\eta \cdot (N_j.level - \frac{\sum_{k=1}^{|N|} N_k.level}{|N|})$. Where η is the clone

rate which is a constant greater than 0. The clones are mutated with a mutation probability of β. The new antibodies after mutation are stored in a temporary antibody set K (β is a user-defined constant between 0 and 1).

(2.4)If the highest affinity between antigen T_i and all network nodes is lower than α (clone threshold), immune inoculation operator is executed. Therefore, the antigen T_i is added to the temporary antibody set K.

(3) Delete the nodes whose stimulation level are lower than the average stimulation level of all nodes in the antibody set N, and the correlative edges are synchronously deleted from the connection set C as well.

(4) If the classification accuracy trends to be stable or the maximal number of iterative steps is reached, the algorithm ends and outputs current immune network, or continues.

(5) Every antibody in temporary antibody set K is added into immune network in the same way as constructing initial network (see step 1).

(6) Renew the stimulation level of every node in antibody set N based on its correlative edges in connection set C. And assign every node's stimulation level to be the sum of stimulation degrees of all the nodes connected to it minus the sum of suppression degrees of all the nodes connected to it. The detailed computation formula is given in section 2.4.

(7) Go to step (2).

So far, the detailed process of immune network training is given. About this training process, it is necessary to note that new antibodies are generated by three ways: immune clone, mutation and inoculation. In particular, immune clone is to exactly copy the original good antibody, and to make the good antibody that matches well with the antigens proliferates. Immune mutation is to randomly change the attribute value of the original antibody to another value or to generalize the attribute value to "all". The match degree of "all" and every attribute value is equal to 1. That is, they completely match. This kind of mutation makes the antibody match a slightly different new antigen. Immune inoculation is to add the antigen whose highest affinity with all nodes in the network is lower than α (clone threshold) into the immune network as bacterin.

(b) Phase II: Classification

After training, the class of every node in immune network is known. So the classification process is much simpler than immune network train process.

In particular, for every record t in testing data set:

(1) Calculate the affinity between t and all nodes in antibody set N. Select n nodes with the highest affinity with t.

(2) Divide these n nodes into different subset based on their affinity with t. The nodes with the same or approximate affinity are in the same subset.

(3) Every subset is given a weight. Higher the affinity is, larger the weight is.

(4) For each class, sum the weight of all n nodes in that class. Thus, the final classification of t is the class with the maximal weight sum.

As shown above, GCA-IN is a supervised learning algorithm. The antigen and antibody both have class field. If the antibody wrongly recognizes the antigen (class value is different), we punish the antibody by reducing its stimulation level (see step 2.2).

2.3 Computing affinity

The affinity denotes the similarity degree of antibody and antibody or antibody and antigen. In GCA-IN, the basic computation method of affinity is to sum the weighted match degree of every attribute of the data record. The weight of attribute A_i is set as $gain^2(A_i)$, where $gain(A_i)$ denotes the information gain of attribute i. Information gain is a concept of ID3 [10]. Larger the information gain, more correlative the attribute with the class. Given two antibodies (or an antibody and an antigen) T_k and T_j whose attribute field are $(A_{k1}, A_{k2}, A_{k3}, ..., A_{ki}, A_{kn})$ and $(A_{j1}, A_{j2}, A_{j3}, ..., A_{ji}, A_{jn})$, their affinity can be calculated as:

$$affinity(T_k, T_j) = \sum gain^2(A_i) * match(A_{ki}, A_{ji})$$

Where $match(A_{ki}, A_{ji})$ denotes the match degree between the attribute values A_{ki} and A_{ji} of attribute i.

If attribute i is a category attribute:

$$match(A_{ki}, A_{ji}) = \begin{cases} 1 \text{ if } A_{ki} = A_{ji} \\ 0 \text{ if } A_{ki} \neq A_{ji} \end{cases}$$

Else $$match(A_{ki}, A_{ji}) = 1 - \frac{|A_{ki} - A_{ji}|}{\sqrt{A_{ki}^2 + A_{ji}^2}}$$

Different from the Euclidean distance method in other related works, the method above takes the different importance of every attribute into account. Especially, the problem of normalization of attribute value can be avoided by using this method when the attribute is a category attribute.

2.4 Computing the stimulation level

The stimulation level denotes the degree of the stimulation of the antibody by antigens and other antibodies. According to immune network theory, the antibody's stimulation level is calculated by integrating the following three factors [3]:

(i) The affinity between a given antibody and all the antigens.

(ii) All bordering antibodies' stimulation degree (equal affinity between the two antibodies) in immune network.

(iii) All bordering antibodies' suppression degree which are equal to ($\sum gain^2(A_i)$ -affinity. between the two antibodies). Here, $\sum gain^2(A_i)$ is the affinity of the antibodies whose attribute filed is absolutely the same.

3. Experiments

In this section, the standard data sets downloaded from the machine learning standard database (http://www.ics.uci.edu/~mlearn/MLRepository.html) are used as the experiment data sets. Based on training data set, an immune network is created using GCA-IN (phase I). Then the immune network after training is used to classify the test data.

3.1 Compared with C4.5 in classification accuracy

In this experiment, the mutation rate β is commonly 30%. The other parameter for every data set is shown in table 1, where clone threshold α is equal to the numeral in the table multiplying $\sum gain^2(A_i)$, clone rate η is equal to the numeral in the table multiplying $1/\sum gain^2(A_i)$. $\sum gain^2(A_i)$ is the affinity of the antibodies whose attribute filed is absolutely the same.

Table 1. The experiment parameters for every data set

data set / parameter	house-voting	monks-2	monks-3	iris
α	0.85	0.88	0.92	0.90
η	0.10	0.08	0.15	0.12
N	8	7	6	5
M	5	3	3	3

Table 2. Comparison of GCA-IN and C4.5 in classification accuracy

data set / algorithm	house-voting	monks-2	monks-3	iris
C4.5	0.961	0.635	0.963	0.953
GCA-IN	0.998	0.683	0.975	0.967

The classification accuracy of GCA-IN is shown in table 2 based on the parameters above. The accuracy of GCA-IN is the average result of 10 independent runs.

According to the experimental results of table 2, GCA-IN has better classification accuracy than C4.5. Compared with C4.5, the immune network structure of GCA-IN is easier to achieve the global optimization. And the immune clone, mutation, bacterin inoculation operators of GCA-IN improves the recognition ability for new pattern which leads to better classification capability.

3.2 The immune network size

In this experiment, some parameters are varied in a certain range to investigate their influence on the immune network size of GCA-IN. And most parameters have little influence on the immune network size except η. Table 3 and table 4 show the numbers of network nodes when the different η is used in monks-2 and monks-3 experiments.

Table 3. The size of immune network (monks-2)

η	0.06	0.07	0.08	0.09	0.l0
the network Size	215	244	283	295	317

Table 4. The size of immune network (monks-3)

η	0.12	0.13	0.14	0.15	0.16
the network Size	182	205	243	251	279

According to table 3 and table 4, the clone rate η has great influence to network size. With the increasing of η, the number of network nodes after training will increase rapidly.

3.3 Fault tolerance analyses

Table 5. Fault-tolerance capability of immune network (monks-2)

η	0.06	0.07	0.08	0.09	0.l0
0%	0.6806	0.6806	0.6991	0.6921	0.6898
1%	0.6806	0.6806	0.6991	0.6921	0.6898
2%	0.6806	0.6806	0.6991	0.6921	0.6898
5%	0.6806	0.6806	0.6991	0.6921	0.6898
10%	0.6782	0.6782	0.6944	0.6875	0.6898
15%	0.6759	0.6736	0.6944	0.6875	0.6852

Table 6. Fault-tolerance capability of immune network (monks-3)

η	0.12	0.13	0.14	0.15	0.16
0%	0.9676	0.9769	0.9769	0.9722	0.9792
1%	0.9676	0.9769	0.9769	0.9722	0.9792
2%	0.9676	0.9769	0.9769	0.9722	0.9792
5%	0.9653	0.9769	0.9745	0.9722	0.9769
10%	0.9607	0.9722	0.9745	0.9699	0.9769
15%	0.9584	0.9722	0.9745	0.9722	0.9792

The Fault tolerance characteristic of immune network is mainly dependent on the redundancy degree of the network, namely the number of immune network nodes. Because Parameter η has great influence to the size of immune network, the fault tolerance of the network has something to do with the parameter η.

Table 5 and table 6 show the fault-tolerant capability of the network nodes when the different η is used in monks-2 and monks-3 experiments. The line corresponding to (x%) denotes the classification accuracy of immune network after x% node of the network failed. And the corresponding size of immune network can be respectively found in table 3 and table 4.

As above, the immune network have a good fault-tolerant capability. Generally, more nodes the immune network has, greater the network's fault-tolerance capability is. How to handle the tradeoff between the network size and fault-tolerant capability is our future work.

4. Conclusions

This paper proposes a classification algorithm based on immune network theory, i.e. GCA-IN. To effectively construct the immune network, GCA-IN introduces immune clone, immune mutation and bacterin inoculation operators to extract the characteristic of training data. In this paper, GCA-IN also adopts a novel computation method of affinity. That is to say, each attribute is assigned a weight based on its information gain according to the different importance of each attribute. After immune network training phase, a simple and effective classification phase is given in this paper. The experiment results show that GCA-IN has a good ability for classification. Also, the fault-tolerant performance of GCA-IN is analyzed by experiments when some nodes which are randomly selected in immune network fail. The experimental results prove that GCA-IN can generate an immune network with good fault-tolerant performance.

Acknowledgements: This work is supported by NSFC Foundation (NO. 60404004), Chinese Post-doctor Foundation (NO. 2003034433) and Nature Science Major Foundation from Anhui Education Bureau (NO. 2004kj360zd).

6. References

[1] N K Jerne, "Towards a network theory of the immune system". *Annals of Immunology*, 125C: 272-289, 1972.

[2] J E Hunt, D E Cooke, "Learning using an artificial immune system". *Journal of Network and Computer Applications*, 19 (2):189-212, 1996.

[3] J Timmis, M Neal, J Hunt, "Artificial immune system for data analysis". *Bio Systems*, 55(1-3):143-150, 2000.

[4] J Timmis, M Neal, "A resource limited immune system for data analysis". *Knowledge Based Systems*, 14(4-4):121-130, 2001.

[5] J D Farmer, N H Packarcl, "The immune systems, adaptation, and machine learning". *Physical*, 22D: 187-204, 1986.

[6] J Carter, "The immune system as a model for pattern recognition and classification". *Journal of American Medical Informatics Association*, 7(1):28-41, 2000.

[7] Andrew Watkins, Lois Boggess, "A New Classifier Based on Resource Limited Artificial Immune Systems". *Proceedings of Congress on Evolutionary Computation, Part of the 2002 IEEE World Congress on Computational Intelligence*, Honolulu, USA, May 12-17, pp.1546-1551, 2002.

[8] J Hunt, J Timmis, D Cookie, "Jisys: the development of an artificial immune system for real world applications". *Artificial Immune System and Their Application*, D. Dasgupta (Eds), Springer-Verlag,pp.157-185, 1998.

[9] L N de Castro, J Timmis, "Artificial immune systems: a new computational intelligence approach". Springer-Verlag, London, September, 2002.

[10] J R Quinlan, "Induction of decision trees". *Machine Learning*, 1(1):81-106, 1986.