A Supervised Classifier Based on Artificial Immune System

Lingxi Peng^{1,2}, Yinqiao Peng¹, Xiaojie Liu², Caiming Liu², Jinquan Zeng²,
Feixian Sun², and Zhengtian Lu²

¹ School of Information, Guangdong Ocean University,
Zhanjiang 524025, China

manplx@163.com, liuxiaojie8@126.com

² College of Computer Science, Sichuan University,
Chengdu 610065, China

Abstract. Artificial immune recognition system (AIRS) has been convincingly proved a highly effective classifier, which has been successfully applied to pattern recognition and etc. However, there are two shortcomings that limit its further applications, one is the huge size of evolved memory cells pool, and the other is low classification accuracy. In order to overcome these limitations, a supervised artificial immune classifier, UCAIS, is presented. The implementation of UCAIS includes: the first is to create a pool of memory cells. Then, B-cell population is evolved and the memory cells pool is updated until the stopping criterion is met. Finally, classification is accomplished by majority vote of the k nearest memory cells. Compared with AIRS, UCAIS not only reduces the huge size of evolved memory cells pool, but also improves the classification accuracy on the four famous datasets, the Iris dataset, the Ionosphere dataset, the Diabetes dataset, and the Sonar dataset.

Keywords: machine learning; artificial immune system; supervised classification.

1 Introduction

In the last twenty years there has been a great deal of interest in exploiting the known properties of the immune system as metaphorical inspiration for computational problem solving. Exciting results have been obtained from the research of network intrusion detection system, pattern recognition, combination optimization, machine learning, and etc [1-4].

In machine learning field, De Castro and Von Zuben's Work examined the role of the clonal selection process within the immune system and went on to develop an unsupervised learning known as CLONALG [5]. This work was extended by employing the metaphor of the immune network theory, which led to the aiNet algorithm. Timmis et al. developed a resource limited artificial immune network [6]. All these models reported good benchmark results for cluster extraction and exploration, and indicated that immune system may be an excellent machine learning method.

Building on these previous works, in particular the ideas of CLONALG and resource limitation, Watkins presented the artificial immune recognition system (AIRS) algorithm in 2001 [7] and the revision in 2002 [8]. Watkins convincingly demonstrated not only that artificial immune systems could be used for supervised learning but also the AIRS classifier was a highly effective classifier [7-8].

Subsequently, AIRS has been successfully and widely applied to many fields. Zhang et al. found AIRS's performance was better than that of traditional K-means, ISODATA, fuzzy, and SOM on remote sensing imagery classification [9]. Xu et al. found AIRS's performance was better than that of neural network on weather forecast [10]. Polat et al. found AIRS's performance was better than that of MLP and PNN on medical diagnosis [11].

However, first of all, classification accuracy of AIRS is still lower than that of some traditional methods [7-8]. Secondly, at training stage in AIRS, the size of the evolved memory cells pool is huge, with small distances among memory cells. After the new candidate memory cell joins the memory cells pool, AIRS just calculates the affinity of match memory cell and candidate memory cell to judge whether the match memory cell should be discarded, but the affinity of match memory cell and any other memory cell is not calculated. For the Ionosphere dataset, the Diabetes dataset, and the Sonar dataset, the percentages of the size of the evolved memory cells pool for the whole training dataset are over 75% [7-8]. Finally, AIRS adopts $[0,1]^n$ feature vector to describe the features of antigens [7-8] and memory cells, and match them with Euclidean distance, which does not calculate the right weight of the feature. However, the present literatures for the right weights of features mainly include man-made factors and lack of mathematical gist.

In order to solve the above problems in AIRS, this paper proposes a supervised classifier based on artificial immune system (UCAIS). The experimental results show that UCAIS not only reduces the size of the evolved memory cells pool, but also improves the classification accuracy, compared with AIRS.

The rest of the paper is organized as follows. In Section 2, an UCAIS classifier is presented. In Section 3, simulations and experimental results are provided. Finally, Sections 4 contains our summary and conclusions.

2 Proposed Classifier

Broadly, the training principle of UCAIS is based upon the mammalian immune system's response to an invading antigen. In a mammalian immune system, the system generates B-cells, which respond to an invader and its presenting characteristics, and through mutation, these B-cells develop greater and greater affinity for the antigen. Most B-cells have a short lifetime, but a small proportion of them become memory cells, which are longer-lived. These memory cells in nature enable a mammalian immune system to respond rapidly to a second invasion by a previously encountered threat. In UCAIS, the antigens are the training data. When a training example is presented, an initial population of B-cells is mutated; the resulting B-cells with the high

affinity to the training data continue to propagate, producing larger numbers of mutated clones, while those with less affinity produce fewer offspring and even die out.

Before the UCAIS classifier is given, let us establish the notional conventions first. Let MC represent the set of memory cell, and mc represents one memory cell where $mc \in MC$. Let AB represent the set of ARB (Artificial Recognition Ball) [9], and ab represents one B-cell where $ab \in AB$. Let AG represent the set of antigen, and ag represents one antigen where $ag \in AG$. Let $mc.c$, $ab.c$, and $ag.c$ represent the class of a given memory cell and B-cell, antigen, respectively, where $mc.c \in C = \{1, 2, \dots, nc\}$, $ab.c \in C = \{1, 2, \dots, nc\}$, and $ag.c \in C = \{1, 2, \dots, nc\}$, nc is the number of classes in the dataset. Let $ab.stim$ represent the stimulation of a given ag to AB ab . Let $TotalResource$ represent the allowed total resources of the system. Let $mc.f$, $ab.f$, and $ag.f$ represent the feature vector of a given memory cell, B-cell and antigen, respectively. Let $mc.f_i$, $ab.f_i$, and $ag.f_i$ represent the value of the i th value of $mc.f$, $ab.f$, and $ag.f$, respectively.

Definition 1. Let MC_c represent the set of memory cell of the c th class, such that $MC_c \subseteq MC = MC_1 \cup MC_2 \cup \dots \cup MC_{nc}$. If $ag.c = c$, then $mc \in MC_c$.

Definition 2. Let AB_c represent the AB set of the c th class, such that $AB_c \subseteq AB = AB_1 \cup AB_2 \cup \dots \cup AB_{nc}$. If $ab.c = c$, then $ab \in AB_c$.

Definition 3. Let AG_c represent the AG set of the c th class, such that $AG_c \subseteq AG = AG_1 \cup AG_2 \cup \dots \cup AG_{nc}$. If $ag.c = c$, then $ag \in AG_c$.

Definition 4. *MMCD (Minimum Memory Cell Distance)*: the minimum distance of the same class in the memory cell's set.

MMCD is used to control the size of memory cell pool, which is to reduce the huge size of memory cells pool in AIRS. After the mach memory cell joins into the set of memory cells pool, if the affinity of mach memory cell and any other memory cell is smaller than *MMCD*, the mach memory cell will be discarded from the memory cell's set.

During training, the first is to initialize a pool of memory cells. Then, B-cell population is evolved and the memory cells pool is updated until the stopping criterion is met, which may be thought of as generalizations of the training instances vectors. The classifier includes five following processes.

2.1 Initialization

The initialization stage is to normalize the feature vectors of the training data, and seed the memory cell if desired. The process is as follows.

- 1) Normalize the feature vectors of the training data.
- 2) Randomly select antigen from the training set of AG and join it into the set of memory cells pool.

With the change of AG class, the variation degree of each feature varies. Specifically, the larger variation degree of the feature, the larger right weight is given to the feature. The right weight is to reduce the size of memory cells pool and improve the classification accuracy. For the training set of AG , suppose there are m antigens, and n features constitute every feature vector. Let $ag[m]$ represent the m th antigen. Let

$\overline{AG.f_i}$ represent the average value of the i th feature. Let SD_i represent the standard deviation of the i th feature and $SD_i = \left(\frac{1}{m-1} \sum_{j=1}^m (ag[j].f_i - \overline{AG.f_i})^2 \right)^{\frac{1}{2}}$. Let v_i represent variation coefficient of the i th feature of feature vector and $v_i = SD_i / \overline{AG.f_i}$. Let ω_i represent the i th feature right weight of the feature vector, which is defined by formula (1).

$$\omega_i = v_i / \sum_{i=1}^n v_i \quad (1)$$

The calculation of affinity and stimulation are defined by formula (2) and formula (3) respectively. When the parameters are mc and ag , formula (3) and formula (3) present the affinity and stimulation of mc and ag , respectively. When the parameters are ab and ag , formula (2) and formula (3) present the affinity and stimulation of ab and ag , respectively. When the parameters are mc and mc , formula (2) present the affinity of two memory cells.

$$Affinity(x, y) = \rho \sqrt{\sum_{i=1}^n \omega_i |x.f_i - y.f_i|^p} / n \quad (2)$$

$$Stimulation(x, y) = 1 - Affinity(x, y) \quad (3)$$

2.2 Clone and Mutation of ARB

UCAIS then learn from each training antigen, and the first is to identify the memory cell which has the same class as the antigen and which is most stimulated by the antigen ag according to formula (4). If there are no memory cells having the same class as ag in the memory cells pool, add ag to the pool as mc_{match} .

$$mc_{match} = \begin{cases} ag & \text{iff } MC_{ag,c} = \emptyset \\ \arg \max_{mc \in MC_{ag,c}} stimulation(ag, mc) & \text{otherwise} \end{cases} \quad (4)$$

Once the mc_{match} with highest stimulation is identified, it generates new ARBs. Let $NumClones = clonalRate * hyperClonalRate * stimulation(mc_{match}, ag)$. The $hyperClonalRate$ and $clonalRate$ are integer values set by the user. The $clonalRate$ is used to determine how many clones are produced by ARBs and memory cells. A typical value is 10. The $hyperClonalRate$ is a multiplier, which ensures that a hypermutating memory cells produces more new cells than standard ARB. UCAIS creates $NumClones$ new clones of mc_{match} , where each feature vector of the clone can be mutated with stimulation value. Specifically, the higher the normalized stimulation value, the smaller the range of mutation allowed. The mutated ARBs of clone and mc_{match} are then joined into the set of AB .

2.3 Competition for Resources

The number of resources in the system is a fixed value, *TotalResource*. For each $ab \in AB$, if $ab.c = ag.c$, then ab is allocated a number of resources. The principle of resources allocation is that a B-cell which is highly stimulated by the given ag can own more resources, and the total resources which are allocated to B-cells can not exceed *TotalResource*. The competitive allocation of resources will result in some B-cells owning the least resources died. The goal is to control the number of B-cells. This process include following steps.

1) If ab is of the same class as the antigen ag , find the maximum stimulation and minimum stimulation among all the *ARBs* in the set of AB by formula (3). For each $ab \in AB$, normalize its stimulation by formula (5).

$$ab.stim = \frac{ab.stim - minstim}{maxstim - minstim} \quad \text{iff } ab.c = ag.c \quad (5)$$

2) Calculate ab 's resources based on stimulation level by formula (6).

$$ab.resources = ab.stim * clonalRate \quad \text{iff } ab.c = ag.c \quad (6)$$

3) Sum all resources. If the sum of resources just allocated to the *ARBs* exceeds the allowance *Totalresource*, resources are removed from the least stimulated *ARBs* first.

4) Meanwhile, all surviving *ARBs* are allowed to generate mutated clones.

5) A stopping criterion is calculated by formula (7) at this point. It is met if the average stimulation level for the same class as the antigen ag is above a stimulation threshold value δ set by the user. If the stopping criterion has not been met, repeat, beginning at step 1).

$$s = \frac{\sum_{j=1}^{|AB_i|} ab_j.stim}{|AB_i|} \quad \text{iff } ab_j \in AB_i \text{ and } ab.c = ag.c \quad (7)$$

2.4 Consolidating and Controlling the Memory Cells Pool

If the new candidate for the memory cells pool, mc_{cand} , which is defined by formula (8), is a better fit for the presenting antigen than the best existing memory cells, mc_{match} , it will be added to the pool, $MC_{ag.c} = MC_{ag.c} \cup \{mc_{cand}\}$. Moreover, if the affinity between mc_{cand} and mc_{match} is less than the *MMCD*, then mc_{cand} actually replaces mc_{match} in the memory cells pool and mc_{match} is discarded. Calculate the affinity of mc_{match} and other memory cell. If the affinity is lower than *MMCD*, mc_{match} is also discarded from the memory cell's set.

$$mc_{cand} = \arg \max_{ab \in AB_{ag.c}} stimulation(ag, ab) \quad \text{iff } ab.c = ag.c \quad (8)$$

2.5 Classification Process

Classification of test data is accomplished by majority vote of the k nearest memory cells to the presented test antigen. The user can set the value of k according to the classification number of the dataset.

3 Experiments

UCAIS has been used to classify four benchmark datasets taken from the repository of the UCI [12], the Fisher iris flowers dataset, the Pima diabetes dataset, the Ionosphere dataset, and the Sonar dataset. These four datasets are famously used for testing classification algorithm. The size of the training datasets, and the user-assignable parameters include the stimulation threshold, δ , *clonalRate*, *hyperClonalRate*, *Total-resource*, and k -value are 0.8, 0.2, 1, 500, 3, respectively, which are all same with literatures [7-8]. The experiments adopt formula (3) and formula (4) to calculate affinity and stimulation respectively where $\rho=2$.

Fig. 1. Relationship of *MMCD* setting and average classification accuracy

Fig. 2. Relationship of *MMCD* setting and the size of the evolved memory cells pool

In order to test the setting of *MMCD* to the size of the evolved memory cells pool and ACA (Average Classification Accuracy), we carry out the experiments on four datasets respectively. The results are shown in Fig.1 and Fig.2. Fig.1 presents ACA increases with the increase of *MMCD*, for four datasets, when *MMCD* is 0.05, 0.03, 0.03, and 0.03, respectively, ACA gets satisfied results; meantime, the sizes of evolved memory cells pool are relatively small. However, with the continuous increase of *MMCD*, ACA decreases. Fig.2 illustrates that the size of the evolved memory cells pool decreases with the increase of *MMCD*. The results prove that the proper setting of *MMCD* can not only reduce the size of the evolved memory cells pool, but also help to improve the ACA.

UCAIS reduces the set of memory cells pool. We compared the size of evolved memory cells pool of UCAIS with AIRS1 [7], the first version of AIRS, and AIRS2 [8], the revision of AIRS1. Both AIRS1 and AIRS2 have been widely and successfully applied to many fields. The results are shown in Fig.3, which illustrates that

UCAIS is the smallest of three. Fig.4 presents the percentage reduction of the evolved set of memory cells of AIRS1, AIR2 and UCAIS, which can be seen that UCAIS is the highest on four datasets. Compared UCAIS with AIRS2, which reduces the size of the memory cells pool on formal three datasets, the percentage reduction and the improvement of UCAIS are 79.6% (+5.6%), 70% (+18%), 79.6% (+19.6%), and 38% (+31%), respectively.

Fig. 3. Comparison of the size of the evolved memory cells pool

Fig. 4. Comparison of the percentage reduction of the evolved memory cells pool

In order to prove that UCAIS improves classification accuracy, we compare UCAIS, AIR1, AIRS2, and Duch [13], which publishes the results of applying a large number of classifiers against many of the four benchmark classification datasets. Table.1 [7,13] presents that the comparison result of the classification accuracy. The symbol ‘-’ implies corresponding method has no classification accuracy on the data-set. Due to limited pages, lots of methods with low classification accuracy are not given. Indeed, on every benchmark dataset UCAIS outperforms a number of well-respected classifiers.

Table 1. The comparison of classification accuracy

Method	Classification accuracy			
	Iris	Ionosphere	Diabetes	Sonar
UCAIS	98.2%	96.9%	78.3%	92.3%
SSV	98.0%	-	-	-
C-MLP2LN	98.0%	-	-	-
3-NN	-	96.7%	-	-
Logdisc	-	-	77.7%	-
IncNet	-	-	77.6%	-
SVM	-	93.2	-	90.4%
AIRS2	96.0%	95.6%	74.2%	84.9%
AIRS1	96.7%	94.9%	75.8%	84.0%

4 Conclusion

A supervised classifier named UCAIS is presented in this paper. Compared with AIRS, UCAIS not only reduces the huge size of evolved memory cells pool, but also

improves the classification accuracy. The next work is to apply UCAIS to more actual problems. UCAIS is a general algorithm, and can be used in other fields. For example, if the antigen set is considered as specified patterns, or normal status of network, and etc., UCAIS can be used for pattern recognition, anomaly detection, and others.

Acknowledgments

This work was supported by 863 High Tech Project of China under Grant NO. 2006AA01Z435, the National Natural Science Foundation of China under Grant No.60373110, No.60573130, and NO.60502011.

References

1. Li, T.: Computer Immunology. Publishing House of Electronics Industry Beijing (2004)
2. Klarreich E. Inspired by Immunity. *Nature*, vol. (415) (2002) 468-470
3. Li, T., An immune based dynamic intrusion detection model. *Chinese Science Bulletin*, vol. 50(22) (2005) 2650-2657
4. Li, T.: An immunity based network security risk estimation. *Science in China Ser. F Information Sciences*, vol. 48(5) (2005) 798-816
5. De Castro, L. N., F. Von Zuben.: The clonal selection algorithm with engineering applications. in *Proc. of Genetic and Evolutionary Computation Conference*. USA: Morgan Kaufman Publishers, (2000) 36-37
6. De Castro, L. N., J. Timmis.: An Artificial Immune Network for Multimodal Optimisation. *Congress on Evolutionary Computation*. Part of the World Congress on Computational Intelligence, (2002) 699-704
7. Watkins, L. Boggess.: A Resource Limited Artificial Immune Classifier. *Proceedings of Congress on Evolutionary Computation*, Berlin Heidelberg: Springer Verlag, (2002) 926-931
8. Watkins, J. Timmis, L. Boggess.: Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm. *Genetic Programming and Evolvable Machines*, vol. 5 (3) (2004) 291-317
9. Zhong, Y.F., Zhang, L.P., Huang B., Li P.X.: An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery. *IEEE Transactions on Geosciences and Remote Sensing*, vol. 44(2) (2006) 420-431
10. Xu, C.L., Li, T., Huang X.M.: Artificial Immune Algorithm Based System for Forecasting Weather. *Journal of Sichuan University(Engineering Science Edition)*, vol. 37(5) (2005) 125-129
11. Polat, K., Sahan, S., Kodaz, H., Gunes, S.: A new classification method to diagnosis liver disorders: supervised artificial immune system (AIRS). in *Proc. of the IEEE 13th Signal Processing and Communications Applications*, New York (2005) 169-174
12. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. <http://www.ics.uci.edu/~mllearn/MLRepository.html> (1998)
13. W. Duch.: Datasets used for classification: Comparison of results. <http://www.phys.uni.torun.pl/kmk/projects/datasets.html> (2002)