# An English Letter Recognition Algorithm Based Artificial Immune

Chunlin Liang, Lingxi Peng*, Yindie Hong, and Jing Wang

Software School, Guangdong Ocean Univ., Zhanjiang 524025, China
manplx@163.com

**Abstract.** In the study of letter recognition, the recognition accuracy is impacted by fonts and styles, which is the main bottleneck that the technology is applied. In order to enhance the accuracy, a letter recognition algorithm based artificial immune, referred to as LEBAI, is presented. Inspired by nature immune system, antibody cell (B-cell) population is evolved until the B-cell population is convergent through the learning of each training antigen and the memory cells pool is updated by the optimal B-cell. Finally, recognition is accomplished by memory cells. It is tested by the well-known letter recognition data set of UCI (University of California at Irvine). Compared with HSAC (Letter Recognition Using Holland-Style Adaptive Classifiers), LEBAI showed that recognition accuracy is increased from 82.7% to 95.58%. LEBAI achieves the same recognition accuracy for the letters of different fonts and styles, or stretched and distorted randomly.

**Keywords:** Letter recognition, Pattern recognition, Artificial immune system, Machine learning.

## 1 Introduction

Artificial immune is an intelligent information processing mechanisms simulating by biological immune system. The unique information processing mechanisms of immune is self-adaptive, self-learning, self-organized, parallel processing, and coordinately distributed, which provide a powerful paradigm to solve difficult problems. It has been widely applied to various fields. Many well-known results have been achieved in research of combinatorial optimization, machine learning, and etc [1-6].

In the study of letter recognition, the recognition accuracy is impacted by fonts and styles, which is the main bottleneck that the technology is applied. Frey & Slate [7] presented letter recognition using Holland-style adaptive classifiers as HSAC on the basis of Holland's study. HSAC is a letter recognition algorithm, which can recognize letter images that are generated by randomly distorting pixel images of the 26 uppercase letters from 20 different commercial fonts and 6 different letter styles. HSAC

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part III, LNCS 5553, pp. 371–379, 2009.

shows lower recognition accuracy. Fogarty [8] presented first nearest neighbor classification on Frey and Slate's letter recognition problem as FNNC. FNNC shows higher recognition accuracy, although the recognition accuracy is increased at the great cost of increasing sharply recognition rules.

In addition, Zhu Li et al. [9] presented a character recognition adaptive learning algorithm based on SVM and sigmoid function. The recognition accuracy of adaptive data is increased by amending self- adaptive the parameters of the sigmoid function, such that the sigmoid function to better fit the class posterior probability distribution of adaptive data output distance. Li Xu et al. [10] presented a container's character recognition algorithm based on neural networks. This algorithm combines several recognizers together based on the peculiarity of each one through the series and parallel hybrid plan, so that increases the recognition accuracy. Wu Ling-chao et al. [11] presented a character recognition based on independent component analysis. The algorithm combines the distance of Euclidean and Mahalanobis based on the principles of independent component analysis to implement character recognition. Li Zuo et al. presented a character recognition approach based on feature line necessary-sufficient condition detection [12]. The algorithm recognizes character through extracting the feature lines from bitmap of character and detecting the necessary-sufficient condition with templates.

Overall, the recognition accuracy of these algorithms is higher, which is limited to the standards character for a kind of font and style. As regards the characters of different fonts and styles, or stretched and distorted randomly, these algorithms are very inefficiently.

In order to enhance the recognition accuracy for the letter with different fonts and styles, a letter recognition algorithm based artificial immune, referred to as LEBAI, is presented. It is tested by the well-known letter recognition data set of UCI (University of California at Irvine) [13] and compared with HSAC etc. LEBAI shows that the recognition accuracy is increased. Furthermore, learning items obvious is reduced and the rate of learning convergence is improved. It also achieved the same accuracy for letters with different fonts and styles, even randomly stretched and distorted.

## 2 Proposed Algorithm

The principle of LEBAI is based on immune response of organism to antigen. First, the immune system of organism responded to the antigen, and extracted the characteristics of antigen by antibody cell (B-cell) when the organism is attacked. Afterwards, the B-cell is cloned and mutated. The B-cell competed with other in population. The optimal B-cell what the affinity is higher with the antigen is reserve and became the memory cell with the longer life cycle. Finally, the immune system can respond rapidly to the same antigen by the memory cell.

In the algorithm of LEBAI, the training data is equivalent of intrusion antigen. The clone amount and mutation probability of B-cell is adjusted dynamically according to the affinity between B-cell and antigen. On one hand, the lower affinity with the antigen, the smaller B-cell is stimulated; meantime, the fewer amount of cloning, the greater the probability of mutation. On the other hand, the B-cell is greater stimulated,

the more the amount of cloning, and the smaller the probability of mutation. After B-cell is cloned and mutated, the B-cells that can recognize the foreign antigen had a longer life cycle, which would become the memory cells.

LEBAI first learn each antigen of training data one by one and evolved a stable memory cells. Finally, the recognition of antigen is accomplished by memory cells.

Before the introduction the algorithm of LEBAI, the terminologies, symbols, as well as the formulas are defined first.

Tuple <*f*, *c*, *t*> is defined as artificial immune cell, referred to as *AIC*. *AIC* is composed of antigens set as *AG*, artificial recognition antibodies set as *AB*, and memory cells set as *MC*, such that $AG \cup ARB \cup MC = AIC$. *f* is defined as the feature vector of in <*f*, *c*, *t*>, such that $f_i$ is the same as that the *i* value of the feature vector, where $f_i$ is real number, $i=\{1,2,\cdots, L\}$, and *L* is a natural number of dimension of the feature vector. *C* is defined as a set to express all classes of antibody, and *c* is a positive integer to express the one of all classes, such that $c \in C=\{1,2,\ldots,nc\}$, where *nc* is the size of *C*. *t* is defined as a life cycle of antibody. In addition, *ag*, *ab*, and *mc* as were defined as an antigen, a *ARB* cell, and a memory cell, respectively.

The affinity of antibody with antigen is based on the similarity of each structure, which is associated with their distance. The fewer the distance, the higher the affinity they have. Let *D*(*ag*, *ab*) represent the distance of antibody with antigen, which is defined as Eq.(1).

$$D(ag, ab) = \sqrt{\sum_{i=1}^{L} | ag.f_i - ab.f_i |} \tag{1}$$

Let *affinity(ag,ab)* represent the affinity of antibody with antigen (see Eq.(2)), such that $affinity(ag,ab) \in (0,1]$.

$$affinity(ag, ab) = \frac{1}{1 + D(ag, ab)} \tag{2}$$

### 2.1 Initialization

First of all, the characteristics value of the feature vector of training antigens are standardized, which lead to the matrix. The matrix is defined as Eq. (3), where *n* is the number of training antigens.

$$AG = \begin{pmatrix} ag_1 f_1 & ag_1.f_2 & \cdots & ag_1.f_L \\ ag_2.f_1 & ag_2.f_2 & \cdots & ag_2.f_L \\ \vdots & \vdots & & \vdots \\ ag_n.f_1 & ag_n.f_2 & \cdots & ag_n.f_L \end{pmatrix} \tag{3}$$

Then, the algorithm selected randomly *m* antigens to take shape the initial set of artificial antibodies and memory cells.

### 2.2 The Clone and Mutation of Antibody Cell

After the initialization accomplished, LEBAI then learn from each of the training antigens. First, the memory cell as $mc_{match}$ is found from memory cells based on Eq. (4), where the class of $mc_{match}$ is the same as the class of the learning antigen, and the *affinity*(*ag*, *ab*) is highest. If the $mc_{match}$ is not found, $mc_{match}$=*ag*, and *ag* will be added to the memory cells set.

$$mc_{match} = \begin{cases} ag & iff \quad MC_{ag.c} = \varphi \\ max(affinity(ag, mc)) & MC_{ag.c} \neq \phi \end{cases} \tag{4}$$

Afterwards, the $mc_{match}$ is cloned. The amount of cloning, referred to as *cCount*, is based on the affinity of $mc_{match}$ with the learning antigen. The higher the affinity, the $mc_{match}$ is stimulated greater, the more the amount. The *cCount* is decided by the two parameters, the one is the stimulated value of cloning as *cStim,* the other one is the constant of cloning as *cConst*. The *cStim* is defined by Eq. (5). The *cConst* is a input constant, where the *cConst* is used to ensure that the new B-cells were enough to be added to B-cells. The *cConst* is defined by Eq. (6).

$$cStim = \frac{1 + affinity(ag, mc_{match})}{2} \tag{5}$$

$$cCount = cStim * cConst \tag{6}$$

Finally, the feature vector of the new B-cells is mutated. The probability of mutation, referred to as *mRate*, is based on the affinity of $mc_{match}$ with the learning antigen. The higher the affinity, the $mc_{match}$ is stimulated fewer, and the lower the probability. The *mRate* is decided by the two parameters, the one is the stimulated value of mutation as *mStim,* and the other one is the constant of mutating as *mConst*. The *mStim* is defined by Eq. (7). The *mConst* is a positive constant, where the *mConst* is used to adjust the tempo that the eigenvector is mutated. The *mRate* is defined by Eq. (8).

$$mStim = \frac{1}{1 + affinity(ag, mc_{match})} \tag{7}$$

$$mRate = mStim * mConst \tag{8}$$

The mutated B-cells will be added to B-cells after the $mc_{match}$ has been cloned and mutated.

### 2.3 The Controlling of Antibody Cell Scale

After the antibody cells have been cloned and mutated, the scale of antibody cells will be expanded rapidly. When the amount of antibody cells reached a certain threshold, in order to control the scale of antibody cells and the convergence of the algorithm, some of the antibody cells would be eliminated through competition.

The life cycle of the antibody cell $t_{ab}$ and right weight $w_{ab}$ are defined by Eq. (9) and Eq. (10), respectively. The larger $w_{ab}$, the longer life cycle $t_{ab}$ has. $p$ is the amount of the antibody cells which has the same class as the learning antigen. $t_0$ is the life cycle of the antibody and the initial $t_0$ is 0, and The *tConst* is integer constant, where the *tConst* is used to expand the life cycle scale of the antibodies with same class.

$$w_{ab} = \frac{\sum_{j=1, i\neq j}^{j=p} D(ab_i, ab_j)}{p-1} \text{ iff } ab_j \in AB_i \text{ and } ab_i.c = ab_j.c \tag{9}$$

$$t_{ab} = t_0 + tConst * w_{ab} \tag{10}$$

Afterwards, some of the antibody cells would be eliminated based on the life cycle of the antibody cell. The process is as follows:

(1) Update the life cycle of the antibody cell based on Eq. (10), where the class of the antibody cell has the same as the learning antigen.
(2) Compute the average of the max life cycle, referred to as $avg_{max}$, based on Eq. (11), where the max life cycle has the highest value in the antibody cells of the same class.
(3) Compute the average of the min life cycle, referred to as $avg_{minx}$, based on Eq. (12), where the min life cycle has the lowest value in the antibody cells of the same class.
(4) If $t_{ab} > avg_{max}$, the antibody cell is eliminated by aging; If $t_{ab} < avg_{min}$, the one will be eliminated by life weak. The antibody cells between $avg_{min}$ and $avg_{max}$ will be reserved to become the candidate memory cell.

$$avg_{\max} = \frac{\sum_{c=1}^{nc} ab_c.t_{\max}}{nc} \quad iff \quad c \in C \tag{11}$$

$$avg_{\min} = \frac{\sum_{c=1}^{nc} ab_c.t_{\min}}{nc} \quad iff \quad c \in C \tag{12}$$

### 2.4 Updating of Memory Cells

The end of the algorithm is to select the memory cells from the antibody cells. If an antibody cell can recognize the learning antigen, it would become the candidate memory cell and be added to the memory cells set to replace the one of the memory cells, on condition that the candidate memory is better than the one. The process leads to a convergence memory cells set and stable immune system, which will have a rapid second response to intrusion antigen. The process is as follows:

$$mc_{cand} = max(affinity(ag, ab)) \quad iff \quad ag.c = ab.c \tag{13}$$

First, the algorithm selects the candidate memory cell based on Eq. (13) from the antibody cells.

$$mc_{replaced} = max(affinity(ag, mc)) \ \ iff \ \ ag.c = mc.c \tag{14}$$

Afterwards, the algorithm selects the replaced memory cell based on Eq. (14) from the memory cells.

Finally, if the affinity of the learning antigen with the $mc_{\text{cand}}$ is higher than with $mc_{\text{replaced}}$, $mc_{\text{replaced}}$ will be replaced by $mc_{\text{cand}}$.

### 2.5 Recognition

After the end of the learning, the class of testing antigen will be decided by the memory cell that has the highest affinity with the testing antigen. The process is as follows:

1. Compute the affinity of each one of memory cells with the testing antigen based on Eq. (2).
2. Select the memory cell which has the highest affinity with the testing antigen. The class of the testing antigen has the same class with the memory cell.

## 3 Experiments

### 3.1 Dataset

The experiment used the well-known letter recognition dataset of UCI (University of California at Irvine) to test the recognition performance of LEBAI [13]. The data set comprised 20,000 learning items. The data of each learning item came from the different image of the letters.

### 3.2 Experiment Parameters

The experiment selects randomly the 4000 learning item as the test set from the data set, the other 16000 learning item as a training set. Table 1. show the parameters.

**Table 1.** Parameters

| Parameter | Value |
|---|---|
| cConst | 1000 |
| mConst | 0.5 |
| tConst | 3000 |
| nc | 26 |

### 3.3 Experiment Results

Different sizes of memory cells were adopted to test the recognition accuracy, which is shown in Fig. 1. LEBAI shows that the greater the size of memory cells, the higher the recognition accuracy. The ultimate recognition accuracy achieved maximum 95.58 %.

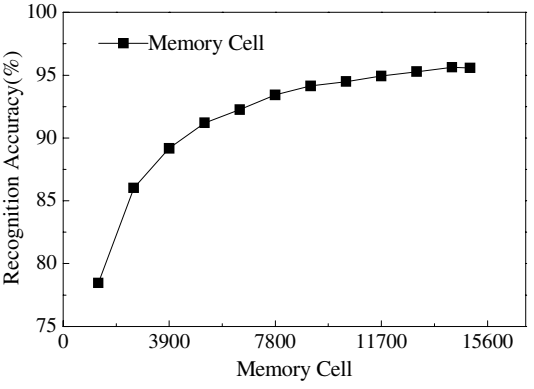


**Fig. 1.** The size of memory cells to recognition accuracies

Different sizes of training antigens were adopted to test the recognition accuracy, which is shown in Fig.2 where sizes of memory cell are 14850, 10000, and 5200, respectively. The experiments show that with the increase of training antigen, the recognition accuracy is increased. However, when the size of memory cell is over 10000, the experiment results is stable to 95.87% because the convergence of the memory cell.

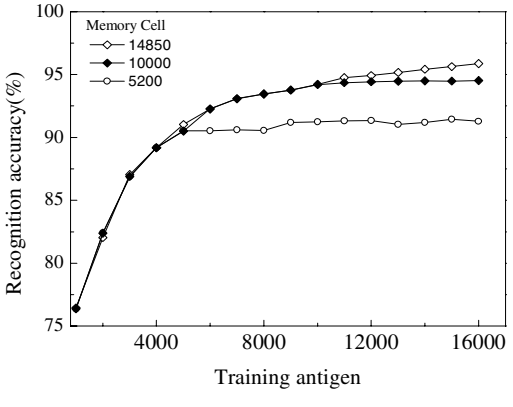


**Fig. 2.** Training antigens to recognition accuracies

**Table 2.** The comparison of recognition accuracy

| Algorithm | Accuracy (%) |
|---|---|
| HSAC [7] | 82.70 |
| Genetic programming [14] | 92.00 |
| Neural networks [15] | 94.13 |
| FNNC [8] | 95.67 |
| LEBAI | 95.87 |

In order to prove that LEBAI improves the recognition accuracy, the comparison of recognition accuracy is showed in Table.2, compared with FNNC, HSAC and etc recognition algorithms. LEBAI shows that the recognition accuracy achieves the highest recognition accuracy.

## 4 Conclusion

The reasons that the existed letter recognition algorithms have low recognition accuracy are analyzed. Afterwards, a novel letter recognition algorithm based artificial immune is presented. Compared with some well-known letter recognition algorithms, LEBAI shows higher recognition accuracy.

## References

1. Albert, R., Jeong, H., Barabasi, A.: Attack and Error Tolerance of Complex Networks. Nature 406, 378–382 (2002)
2. Li, T.: Dynamic Detection for Computer Virus based on Immune System, Science In China. Series F: Information Science 51(10), 1475–1486 (2008)
3. Omkar, S., Khandelwal, R., Yathindra, S., et al.: Artificial Immune System For Multi-Objective Design Optimization of Composite Structures. Engineering Applications of Artificial Intelligence 21(8), 1416–1429 (2008)
4. Vijayalakshmi, K., Radhakrishnan, S.: Artificial Immune based Hybrid GA for QoS based Multicast Routing in Large Scale Networks (AISMR). Computer Communications 31(17), 3984–3994 (2008)
5. Wang, L., Singh, C.: Population-based Intelligent Search in Reliability Evaluation of Generation Systems with Wind Power Penetration. IEEE Transactions on Power Systems 23(3), 1336–1345 (2008)
6. Ye, F., Xu, S., Xiong, Y.: Two-step Image Registration by Artificial Immune System and Chamfer Matching. Chinese Optics Letters 6(9), 651–653 (2008)
7. Frey, P.W., Slate, D.J.: Letter Recognition Using Holland-style Adaptive Classifiers. Machine Learning 6(2), 161–182 (1991)
8. Fogarty: First Nearest Neighbor Classification on Frey and Slate's Letter Recognition Problem. Machine Learning 9(4), 387–388 (1992)
9. Zhu, L., Sun, G.: Character Recognition Adaptive Learning Algorithm based on SVM and Sigmoid Function. Application of Electronic Technique 32(4), 16–17 (2006)
10. Li, X., Yang, J.: Container's Character Recognition Algorithm based on Neural Networks. Computer and Communications 19(z1), 89–91 (2001)

11. Wu, L., Mo, Y.: Character Recognition based on Independent Component Analysis. Journal of Shanghai University 9(3), 193–196 (2003)
12. Li, Z., Wang, S., Cai, S.: Character Recognition Approach based on Feature Line necessary-sufficient condition detection. Journal of Software 13(1), 85–91 (2002)
13. Frey, P.W., Slate, D.J.: UCI Repository of Machine Learning Databases, Letter Recognition Datasets (1991), http://archive.ics.uci.edu/ml/datasets/Letter+Recognition
14. Ahluwalia, M., Bull, L.: Coevolving Functions in Genetic Programming. Systems Architecture 47 (2001)
15. Daqi, G., Chao, X., et al.: Combinative Neural-network-based Classifiers for Optical Handwritten Character and Letter Recognition. International Joint Conference on Neural Networks, 3, 2232–2237 (2003)