A Learning Automata Based Artificial Immune System for Data Classification

A. H. Momeni Azandaryani M. R. Meybodi

Soft Computing Laboratory
Department of Computer Engineering and Information Technology
Amirkabir University of Technology
Tehran, Iran

a_h_momeni@aut.ac.ir *mmeybodi@aut.ac.ir*

Abstract

In this paper we propose an artificial immune system in which learning automata are used to adaptively determine the values of its parameters. Learning automata are used for altering the shape of receptor portion of antibodies to better complementarily match the confronted antigen. In order to show the effectiveness of the proposed artificial immune computer experiments have been conducted. The result of experimentations confirms the effectiveness of the proposed model.

1. Introduction

Artificial Immune Systems (AISs) represent a field of biologically inspired computing that attempts to exploit theories, principles, and concepts of modern immunology to design immune system-based applications in science and engineering [13]. One role of the immune system (IS) is to protect the host organism against attacks from antigens and eliminate those cells that have been "infected". AISs are proving to be a very general and applicable form of bio-inspired computing. A great deal of work has gone into developing algorithms that extrapolate basic immune processes such as clonal selection, negative and positive selection, danger theory, and immune networks [22]. To date, AIS have been applied to areas such as machine learning [16], optimization [15], scheduling [23], virus detection [17], data analysis [20], fault diagnosis [21], network intrusion detection [18], computer security [24], robotic systems [17], decision support systems [19], combinatorial optimization [15], anomaly detection [9], protein structure prediction and many other areas [17].

In spite of applying AIS to different areas of computer problems, many researchers are working on design of some modified structures for these systems. In the context of the framework for AIS presented in [12], in this paper we introduce an immune algorithm based on learning automata principles. We propose a new operator for mutation and simulating V-region (Variable Region) of antibodies. This operator uses an LA for learning how to alter the shape of receptor portion of antibodies to algorithm to some classification problems. The experiments elucidated that this algorithm has high classification precision and can be used as a good solution for different classification problems.

The rest of this paper is organized as follows. Section 2 describes some background the proposed algorithm, section 4 reports the results of experimentations and finally concluding remarks are presented in Section 5.

2. Background Information

In this section, we briefly introduce human immune system, artificial immune system and learning automata.

2.1. Human Immune System

The human immune system is a complex system made of cells, molecules, and organs that together constitute an identification mechanism capable of perceiving and combating dysfunction from our own cells and the action of exogenous infectious microorganisms as well. The human immune system safeguards us against infectious agents such as viruses,

bacteria, fungi, and other parasites. Any molecule that can be recognized by the adaptive immune system is known as an antigen (Ag). Lymphocytes or the white blood cells are the fundamental components of the immune system. Within the human body, Lymphocytes are found in two forms, B cells and T cells. Functionally, these two types of cells differ in their mode of antigen recognition. B-cells are capable of recognizing antigens free in solution, while T cells require antigens to be presented by other accessory cells. Each has its distinct chemical structure and produces many Y-shaped antibodies (Ab) from its surface to kill the antigens. Ab's are molecules attached primarily to the surface of B cells whose aim is to recognize and bind to Ags [1].

The clonal selection theory [2] explains how an immune response is mounted when a nonself-antigenic pattern is recognized by a B cell. When a B-cell receptor recognizes a nonself-antigen with certain affinity, it is then selected to proliferate and produce antibodies in high volumes. The antibodies are soluble forms of the B-cell receptors that are released from the B-cell surface to cope with the invading nonself-antigen. Antibodies bind themselves to antigens, thus resulting in their eventual elimination by other immune cells. In case of immune cells, proliferation is an asexual or a mitotic process; the cells divide themselves.

Once a B cell is sufficiently stimulated through close affinity to a specific antigen, it rapidly produces clones of itself. At the same time, the B-cell clones undergo a hypermutation process at particular sites in its gene, which enables the new cells to match the antigen more closely. There is a very rapid proliferation of immune cells, successive generations of which are better and better matches for the antigens of the invading pathogen [3]. The B cells that are not stimulated as they do not match any antigens in the body will eventually die. On the contrary, the activated B cells with high antigenic affinities are selected to become memory cells. When a body has successfully defended against a pathogen, memory cells remain and circulate in the blood, lymph, and tissues for very long periods of time. These memory cells recognize antigens similar to those that originally caused the immune response and created the memory cells, so that the body's response to a later invasion of the same pathogen or a very similar invader is much more rapid and powerful than to an invader never seen before in the primary response.

The immune systems possess a cross-reactive memory that is observed when an individual develops a memory to one antigen and is challenged with a related, but different one. The cross-reactive memory, clonal expansion, and programmed cell death rates allow the immune system to dynamically allocate resources as needed in a distributed environment [4].

2.2. Artificial Immune System (AIS)

In many respects, AISs are abstract computational modeling of the immune system; in fact, some AIS techniques are based on theoretical models of the Human Immune System. However, the main difference lies in the use of AISs as a problem solving technique. A theoretical model that has served as a basis for some AISs is the idiotypic network theory proposed by Jerne [5]. This theory proposed that the NIS regulates itself by forming a network of B-cells that can enhance or suppress the expression of specific antibody types. This self-regulatory mechanism maintains a stable immune memory. The formation of such a network is only possible by the presence of paratopes on the B-cells that can be recognized by other B-cells epitopes. This recognition usually extends to more than one level, resulting in the formation of complex reaction networks.

The different models/techniques use a variety of NIS aspects; however, the most relevant are the antigen-antibody (Ag-Ab) binding, idiotypic immune network theory, and the self/non-self discrimination. The modeling of the Ag-Ab binding mechanism is present in almost all the models and techniques.

Self/non-self discrimination is by T-cells mature in the thymus. There, they go through a process of selection that ensures that they are able to recognize non-self peptides presented by MHC. This process has two main phases: positive selection and negative selection [6].

During the positive selection phase, T-cells are tested for recognition of MHC molecules expressed on the cortical epithelial cells. If a T-cell fails to recognize any of the MHC molecules, it is discarded; otherwise, it is kept.

The purpose of negative selection is to test for tolerance of self cells. T-cells that recognize the combination of MHC and self peptides fail this test. This process can be seen as a filtering of a big diversity of T-cells; only those T-cells that do not recognize self peptides are kept [7].

The presence of an antigen in the system and its subsequent interaction with mature lymphocytes triggers an immune response (primary response), resulting in the proliferation of lymphocytes with a unique antigenic specificity. This process of population expansion of particular T-cells and B-cells is called clonal selection [8].

Most of these lymphocytes die when the antigen is eliminated; however, some of these lymphocytes are

kept as memory cells. The next occurrence of the same antigen can be detected quickly, activating a secondary response. This response is faster and more intense because of the availability of such memory cells [9].

To be brief, we will describe the other elements of AIS in the LAAIS section.

2.3. Learning Automata (LA)

Learning Automata are adaptive decision-making devices that operate on unknown random environments. A learning Automaton has a finite set of actions to choose from and at each stage, its choice (action) depends upon its action probability vector. For each action chosen by the automaton, the environment gives a reinforcement signal with fixed unknown probability distribution. The automaton then updates its action probability vector depending upon the reinforcement signal at that stage, and evolves to some final desired behavior. A class of learning automata is called variable structure learning automata and are represented by quadruple $\{\alpha, \beta, p, T\}$; In which, $\alpha=\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ represents the action set of the automata, $\beta=\{\beta_1, \beta_2, \ldots, \beta_r\}$ represents the input set, $p=\{p_1, p_2, \ldots, p_r\}$ represents the action probability set, and finally $p(n+1) = T(\alpha(n), \beta(n), p(n))$ represents the learning algorithm. Let α_i be the action chosen at time n, then the recurrence equation for updating p is defined as

$$\begin{aligned} p_i(n+1) &= p_i(n) + a.(1 - p_i(n)) \\ p_j(n+1) &= p_j(n) - a.p_j(n) \qquad \forall j \;\; j \neq i \end{aligned} \tag{1}$$

for favorable responses, and

$$\begin{aligned} p_i(n+1) &= (1-b).p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) \qquad \forall j \;\; j \neq i \end{aligned} \tag{2}$$

for unfavorable ones. In these equations, a and b are reward and penalty parameters respectively. If $a = b$, learning algorithm is called L_{R-P} (Linear Reward-Penalty), if $a << b$, it is called $L_{R\varepsilon P}$ (Linear Reward-Epsilon Penalty), and if $b = 0$, it is called L_{R-I} (Linear Reward-Inaction). For more information about learning automata the reader may refer to [10][11].

3. The Proposed Algorithm (LAAIS)

In the section we present the proposed algorithm used for data classification. Let shape-space S be a multi-dimensional metric space where each axis stands for a physic-chemical measure characterizing a molecular shape. We assume $\Omega = \{\omega_1, \omega_2, \ldots, \omega_T\}$ be a set of classes, where each ω_t, $t=1,\ldots, T$ is described by points $x_{i,j}$, $j=1, \ldots, L_i$ in this space. Each point $x_{i,j}$ in the space S specifies a possible cell of an artificial immune system.

Like the models proposed by de castro et al. and Jerne [12][5], we make no distinction between the B cell and the Ab. The Ag-Ab affinity is measured by a distance metric (dissimilarity) between them. Oppositely, the Ab-Ab affinity is defined by a similarity metric between them. These affinities are obtained as follows:

$$Affinity(Ab, Ag) = \frac{1}{\frac{1}{p}\sum_{i=1}^{p}\sqrt{(Ab_i - Ag_i)^2}} \tag{3}$$

The following notation will be adopted in this section:

M : matrix of N memory cells;
N_c: number of clones generated from each stimulated cell;
D: dissimilarity matrix with elements $d_{i,j}$ (Ag-Ab);
S: similarity matrix with elements $s_{i,j}$ (Ab-Ab);
n: number of cells of highest affinities selected to clone and hypermutate;
ζ: percentage of the matured cells to be selected;
σ_d: natural death threshold; and
σ_s: suppression threshold.

The learning algorithm intends to build a memory pool of cells. During the course of iteration in algorithm, there may be many antibodies; however, in the final system, only their special subset constituting the memory cells will be used to classify the test antigens.

After selecting the n high affinity cells, these cells are cloned to reproduce N_c clone cells. The higher affinity leads to larger N_c like the equation [14]:

$$N_c = \sum_{j=1}^{n} Round(Kscale \times f_{ij}) \tag{4}$$

Where *Kscale* is the clonal scale and *Round()* is the operator that rounds the value in parentheses toward its closest integer value. After producing clones from the selected antibodies, these clones alter by a simple mutation operator to provide some initial diversity over the descent antibody. Then we define the hypermutation operator as follows:

$$C = C - \alpha(C - X) \tag{5}$$

Where C is a clone cell to be hypermutated, X is the confronted antigen and α is the learning rate or mutation rate. To calculate this learning rate, a learning

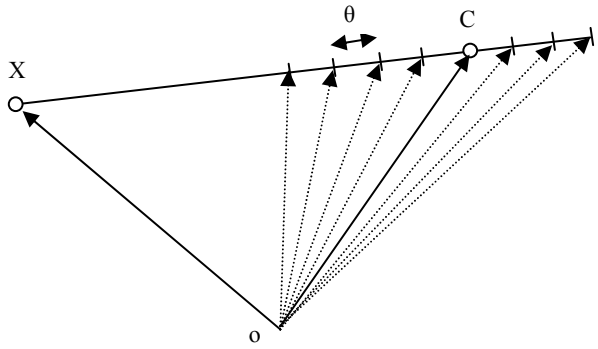

Figure 1. Possible values of the clone cell *C* after convergence of its learning automaton.

automaton (LA) is bound to each clone cell. This learning automaton has three actions:

$$\begin{cases} Increment \Rightarrow \alpha(n+1) = \alpha(n) + \theta \\ Nochange \Rightarrow \alpha(n+1) = \alpha(n) \\ Decrement \Rightarrow \alpha(n+1) = \alpha(n) - \theta \end{cases} \quad (6)$$

To have a fixed number of states for each LA, we define a constraint for changing α like: $\alpha \in [-K\theta, K'\theta]$. With this constraint, the clone cell can vary step by step in the interval $[C + K\theta(C - X), C - K\theta(C - X)]$. Figure 1 depicts the possible values of the cell C after convergence of its learning automaton.

In each LA, the reward signal is produced as follows: if in the current iteration, the cell C_i is selected to be in memory pool, the reward of the corresponding LA (LA_i) will be *0*. Otherwise it will be *1*. In this situation, we will have a competition between LAs to cause their cells appear in the memory pool. After the convergence of LAs, the candidate clone cells are inserted into the memory pool.

The LAAIS learning algorithm works in the following manner:

Table 1. LAAIS Algorithm

for each ω_t in class list Ω **do**
  randomly construct the network cells;
  for each antigen i of the class ω_t **do**
    calculate its affinity d_{ij}, to all the network cells;
    select n highest affinity network cells;
    clone (reproduce) these n selected cells to N_c cells;
    mutate locally these clone cells;
    do
      each of the LAs selects its action;
      selected actions alter their corresponding clone cells;
      determine ***D*** for these improved cells;
      select ζ% of the highest affinity clone cells;
      deselect those clone cells that their affinity d_{ij}, is inferior to threshold σ_d ;
      calculate the network Ab-Ab affinity s_{ij} ;
      deselect $s_{ij} > \sigma_s$ (clonal suppression);
      [remained clone cells at this step are candidate to insert into memory pool]
      calculate rewards of LAs based on the candidate clone cells;
      update the action probabilities of LAs based on the received reward;
    until the convergence of LAs bound to clone cells;
    insert candidate cells to the memory pool;
    reload initial cells into network;
  end for
  determine **S** for memory cells;
  eliminate those cell whose $s_{ij} > \sigma_s$;
  Classifier ← Memory cells pool;
end for

As we see in the AIS framework description, the hypermutation should be done on each cell in proportion to its affinity. The higher affinity will result in weaker hypermutation rate and vice versa. By using learning automata, the hypermutation is a semi-stochastic process. To respect the above hidden rule, we dynamically set the value of θ for each cell based on its dissimilarity with the antigen compared.

In our proposed algorithm, we implemented all three models of learning automata. So we name these models as: $LAAIS_{R\text{-}P}$ in which the $L_{R\text{-}P}$ model is used; and $LAAIS_{R\varepsilon P}$ in which LAs are in $L_{R\varepsilon P}$ model; and finally the $LAAIS_{R\text{-}I}$ model for use of $L_{R\text{-}I}$ learning automata.

We also simply name the AIS without learning automata, AIS in which hypermutation is done by an affinity based simple random mutation.

4. Model Implementation

LAAIS was implemented in *C#* using Microsoft *.Net* framework. The classification performance was tested on three benchmark datasets Iris, Wisconsin Diagnostic Breast Cancer and Diabetes. The Iris dataset consists of 50 samples from each of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample; they are the length and the width of sepal and petal. The Wisconsin Breast Cancer dataset consists of 645 records, each introduced with 9 attributes. This dataset contain observations with missing attributes. Dealing with these missing

attributes, we filled them with average of the values of those attributes over the dataset. The Diabetes dataset is composed of 145 records with 3 attributes (Glucose, Insulin, SSPG) but hardest dataset to be classified among two others.

The size of memory pool in these models has no significant difference because of the common processes for selection and suppression. The performance of classification using the models implemented is provided in table 2.

Table 2. Comparison of performance of classification techniques

Dataset	AIS	$LAAIS_{R\text{-}I}$	$LAAIS_{R\varepsilon P}$	$LAAIS_{R\text{-}P}$
Iris	94.1%	94.8%	95%	97.3%
Wisconsin Breast Cancer	95%	91%	94%	95.4%
Diabetes	76.1%	75%	80.7%	83%

We can see from the Table 1 that $LAAIS_{R\text{-}P}$ has the maximum classification performance among other models tested. As exhibited, in the $LAAIS_{R\text{-}I}$ and $LAAIS_{R\varepsilon P}$ models the performance sometimes falls down even lower than AIS model. The cause of this behavior is that is AIS, we have implemented an affinity based random mutation that provides us some diversity about the clone cells. But the $LAAIS_{R\text{-}I}$ and $LAAIS_{R\varepsilon P}$ models, LA may quickly converge to a value that may not be the best possible value.

5. Conclusion

In this paper, a novel classification model (LAAIS) based on learning automata and artificial immune system was proposed to provide a better cell mutation and receptor variation in comparison to other AIS classifiers. This proposed model and some modifications of it were tested on three benchmark datasets Iris, Wisconsin Breast Cancer and Diabetes. One of them was found to be a very competitive classifier but the others act almost worst than the simple AIS because of their poor learning phase and quick convergence.

10. References

[1] N. K. Jerne, "The immune system", *Sci. Amer.*, vol. 229, no. 1, 1973, pp. 51–60.

[2] F. M. Burnet, "The Clonal Selection Theory of Acquired Immunity", Cambridge, MA, U.K.: *Cambridge Univ. Press*, 1959.

[3] A. Watkins and L. Boggess, "A new classifier based on resource limited artificial immune systems", Proceeding of Congress Evolutionary Computation, Part of the World Congress on Computational Intelligence, 2002, pp. 1546–1551.

[4] Y. Zhong, L. Zhang, B. Huang, and P. Li, "An Unsupervised Artificial Immune Classifier for Multi/Hyperspectral Remote Sensing Imagery", IEEE Transaction GeoScience and Remote Sensing, vol. 44, No. 2, 2006.

[5] N. K. Jerne, "Towards a network theory of the immune system", Annals of Immunology (Inst. Pasteur), vol. 125C, 1974, pp. 373-389.

[6] A. Coutinho, "The self non-self discrimination and the nature and acquisition of the antibody repertoire", Annals of Immunology. (Inst. Pasteur), vol. 131D, 1980.

[7] J. Kappler, N. Roehm, and P. Marrack, "T Cell Tolerance by Clonal Elimination in the Thymus", Cell Magazine, no. 49, 1987, pp. 273-280.

[8] N. K. Jerne, "Clonal selection in a lymphocyte network", in Cellular selection and regulation in the immune response, Raven Press. , 1974, pp. 39-48.

[9] F. Gonzalez, "A Study of Artificial Immune Systems Applied to Anomaly Detection", A Dissertation Presented for the Doctor of Philosophy Degree, 2003.

[10] A. L. Thathachar and P. S. Sastry, "Varieties of Learning Automata: An Overview", IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, 2002, pp. 711-722.

[11] K. S. Narendra, and M. A. L. Thathachar, "Learning Automata: An Introduction", Prentice-Hall Inc, 1989.

[12] LN. de Castro and FJ. Von Juben, "An Evolutionary Immune Network for Data Clustering", In Proc. of the IEEE SBRN (Brazilian Symposium on Artificial Neural Networks) , 2000, pp. 84-89.

[13] V. Cutello, G. Nicosia, M. Pavone, and J. Timmis, "An Immune Algorithm for Protein Structure Prediction on Lattice Models", IEEE Transactions on Evolutionary Computing, Vol. 11, No. 1, 2007.

[14] VG. Arakelian, "The longway to the automatic chromatographic analysis of gases dissolved in insulating oil", IEEE Elect. Insul. Magazine, Vol. 20, No. 6, 2004, pp. 8–25.

[15] X. Wang, XZ. Gao, SJ. Ovaska, "Artificial immune optimization methods and applications: a survey", IEEE International Conference on Systems, Man and Cybernetics, 2004.

[16] JD. Farmer, NH. Packard, AS. Perelson, “The Immune System, Adaptation, And Machine Learning”, Physica 22D, 1986, pp. 187-204.

[17] LN. de Castro, FJ. Von Zuben, “Artificial Immune Systems: Part II A Survey Of Applications”, Technical Report, 2000.

[18] U. Aickelin, J. Greensmith, J. Twycross, “Immune System Approaches to Intrusion Detection - A Review”, Lecture Notes In Computer Science, 2004.

[19] D. Dasgupta, “An Artificial Immune System As A Multi-Agent Decision Support System”, IEEE International Conference on Systems, Man and Cybernetics, 1998.

[20] LN. de Castro, FJ. Von Zuben, “aiNet: An Artificial Immune Network for Data Analysis”, Chapter of Data Mining: A Heuristic Approach Book, 2001.

[21] JLM. Amaral, R. Tanscheit, M. Pacheco, “An Immune Inspired Fault Diagnosis System For Analog Circuits Using Wavelet Signatures”, Proceedings of the NASA/DoD Conference on Evolution Hardware, 2004.

[22] D. Dasgupta, Z. Ji, F. Gonzalez, “Artificial Immune System (AIS) Research In The Last Five Years”, Evolutionary Computation, 2003.

[23] CAC. Coello, DC. Rivera, NC. Cortes, “Use Of An Artificial Immune System For Job Shop Scheduling”, Lecture Notes In Computer Science, 2003.

[24] PK. Harmer, PD. Williams, GH. Gunsch, GB. Lamont, “An Artificial Immune System Architecture For Computer Security Applications”, IEEE Transactions on Evolutionary Computation, 2002.

[25] G. Nicosia, “Immune Algorithms For Optimization And Protein Structure Prediction”, Ph.D. dissertation, Mathematics Deptartment, Catania University, Italy, 2004.