CLONAL-GP Framework for Artificial Immune System Inspired Genetic Programming for Classification

Hajira Jabeen and Abdul Rauf Baig

National University of Computer and Emerging Sciences, Islamabad, Pakistan
{hajira.jabeen,rauf.baig}@nu.edu.pk

Abstract. This paper presents a novel framework for artificial immune system (AIS) inspired evolution in Genetic Programming (GP). A typical GP system uses the reproduction operators mimicking the phenomena of natural evolution to search for efficient classifiers. The proposed framework uses AIS inspired clonal selection algorithm to evolve classifiers using GP. The clonal selection principle states that, in human immune system, high affinity cells that recognize the invading antigens are selected to proliferate. Furthermore, these cells undergo hyper mutation and receptor editing for maturation. In this paper, we propose a computational implementation of the clonal selection principle. The motivation for using non-Darwinian evolution includes avoidance of bloat, training time reduction and simpler classifiers. We have performed empirical analysis of proposed framework over a benchmark dataset from UCI repository. The CLONAL-GP is contrasted with two variants of GP based classification mechanisms and results are found encouraging.

Keywords: Artificial Immune Systems, Genetic Programming, Classification.

1 Introduction

Genetic Programming was originally introduced as an extension of Genetic Algorithm to automatically evolve computer programs. It posses several outstanding features when compared to other traditional evolutionary algorithms. These include: variable sized solution representation, ability to work with little or no knowledge about the solution structure, transparency, data independence and efficient data modeling ability. These features make GP readily applicable to evolve classifiers. This property of GP has been recognized since its inception [1].Numerous researchers have developed different techniques to solve classification problems using GP. One of the profound techniques [2] [3] is evolution of arithmetic expressions as discriminating function between different classes. An arithmetic classifier expression (ACE) is created using numerical attributes of the data and some random constants. The value of expression is evaluated for every instance of the data where the output is a real value. This real output is mapped onto different classes of the data.

We have investigated the proposition to evolve the ACE using AIS principles. AIS [4]mimic the principles of human immune system, and are capable of performing many tasks in various areas. In this work, we will review the clonal selection concept, together with the affinity maturation process, and demonstrate that these biological

R. Setchi et al. (Eds.): KES 2010, Part I, LNAI 6276, pp. 61–68, 2010.

principles can lead to the development of powerful computational tools. AIS operates on the principle of human immune system [5], and, is capable of performing many tasks in various areas [6]. In real life when a human is exposed to an antigen (Ag), some subpopulation of its bone marrow derived cells (B cells) respond by producing antibodies (Ab).Each cell secretes a single type of antibody, which is relatively specific for the antigen. The antigen simulates the B-cell by binding to antibodies and to divide and mature into final (nondividing) secreting cells known as Plasma cells. These cells are most active antibody secretors. On the other hand B lymphocytes, which divide rapidly, also secrete antibodies, at a lower rate. T cells play a central role in regulating B cell response, but will not be explicitly accounted for the development of our model. Lymphocytes, in addition to proliferating and/or differentiating into plasma cells, can differentiate into long-lived B memory cells. Memory cells circulate through the blood, lymph and tissues, and when exposed to a second antigenic stimulus commence to differentiate into large lymphocytes capable of producing high affinity antibodies, pre-selected for the specific antigen that had stimulated the primary response.

The main features of the clonal selection explored in this paper are:

- Proliferation and differentiation on stimulation of cells with antigens
- Generation of new random genetic changes, subsequently expressed as diverse antibody patterns, by a form of depth Limited mutation (a process called affinity maturation)
- Elimination of newly differentiated lymphocytes carrying low affinity antigenic receptors.

2 GP for Data Classification

GP has been applied for classification in various ways. One of these is evolution of classification algorithms e.g. 'decision trees' [7] or evolution of rules for classifier evolution [8]. In [9] the grammar to GP evolution is evolved in such a way that ultimate result is a feasible classification algorithm. For algorithm evolution GP use some type of constrained syntax/grammar so that the trees transform into an algorithm and remain valid after application of evolutionary operators. Another method is evolution of classification rules [10] [11] [12] [13].In this type of methods GP trees are evolved as logical rules, these methods are applied on numerical and nominal data. A newer and somewhat GP specific method evolution of classifiers is in the form of mathematical expressions [2] [3] [14] [15] [16]. In this type of classification the classifiers are evolved in the form mathematical discriminating expressions. Expressions having attribute values as terminals and arithmetic operators as functions are evolved such that the real valued output is used to extract the classification result.

All above mentioned algorithms have used the principle of biological evolution to search for fitter solutions. Next section presents the proposed AIS inspired evolutionary framework. We have tested the performance of proposed framework on data classification problem.

3 Proposed Hybrid Framework

As mentioned in the previous section, the outstanding feature of GP is its ability to represent and evolve variable length solutions. This ability also introduces a drawback of increase in average tree size during evolution. We present a novel artificial immune system inspired evolution that controls bloat through depth limited receptor editing operator.

3.1 Population Representation

An ACE is represented as expression trees where the operands are (+,-,*,/) and operators are attributes of data and a few ephemeral constants. Where ephemeral constants are randomly generated constants that remain unchanged once inserted into the tree node. For example consider a four attribute data having [A1, A2, A3, A4] as attributes for a binary classification problem. Figure 1 shows an example ACE .

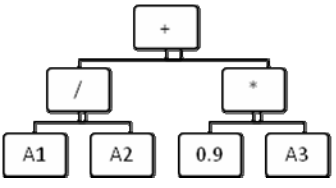


Fig. 1. Arithmetic Classifier Expression

3.2 Population Initialization

There are three well known initialization methods in GP literature. The *full* scheme creates full trees till the maximum depth allowed. It populates the tree with function nodes until maximum depth, beyond that, only terminal nodes are selected. On the other hand *grow* method randomly selects nodes from function or terminal set until maximum depth is reached and allows more diverse size and shape. The initialization method we used for ACE evolution is the well known *Ramped half and half* method [1]. The ramped half and half method utilizes advantages of both full and grow initialization schemes with equal probability. It makes use of different depths for full and grow method ranging from some minimum depth to the maximum allowed depth. This method has been widely used for initialization in many classification problems [2] [3] .

3.3 Affinity Measure

For the classification purpose the antigen population **Ag** to recognize is the set of training samples. Two possible instances are given below for the tree mentioned in Figure 1.

I1 = [1 2 3 4]Є C1
I2 = [1 2 -3 4]Є C2

For a classifier to recognize instances of particular class, we must train it to output different responses for different classes of data. We can train the expression to output a positive signal for class C1 and negative signal for C2 for a binary classification problem. In case of example expression from Figure 1, we will get the response 3.2 for I1 and -2.2 for I2. This will increase the affinity of **Ab** by 2. In this way we must calculate response for all the instances of training data to estimate the affinity measure of a particular **Ab** against antigens **Ag.** Similarly affinity measures for all the **Ab** population must be calculated. The above mentioned measure will only return the count of instances for which a classifier has output correct response. For classification we need more than correct output count.

To resolve this problem, we can see that our Ag population can be divided into two types Ag+ for which Ab should output a positive response and Ag-, for which the Ab should output a negative response. Given these definitions we can define the new affinity measure as

$$\text{Affinity of } \mathbf{Ab} = \frac{\text{number of correct positive responses}}{\text{Number of Ag +}} + \frac{\text{number of correct negative responses}}{\text{Number of Ag -}}$$

This affinity measure is can also be seen as area under the convex hull(AUCH) for one threshold(0). This would ensure that the **Ab** with better discriminative power for two different **Ag** is assigned better affinity.

3.4 Proliferation

The proliferation of affinity or increase in average fitness during the evolution process is essential for efficient search of desired solutions. In case of AIS some highest affinity individuals are selected from population and cloned in proportion to their affinity. Fitter individuals will tend to create more clones as compared to less fit members of population.

3.5 Clone Maturation (Depth Limited Mutation)

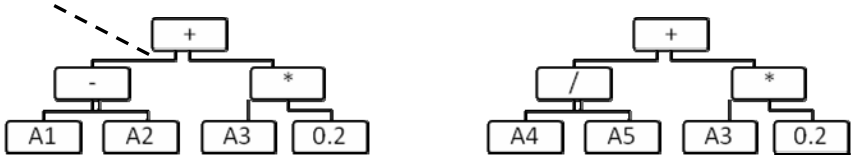
In biological immune system, the antigen activated B cell population is diversified by two mechanisms. These mechanisms are hyper mutation and receptor editing [17] [18]. Random changes are introduced via hyper mutation into the genes responsible for the Ag-Ab communication and the anticipation is that one such change can result in increase in affinity of an individual. The higher affinity variants are then collected in the memory cell repository. This hyper mutation makes sure that population is diversified and some different B cells with higher affinity may be generated. On the other hand the cells with lower affinity need elimination. [17] [19]

Some studies [18] also suggest that immune system practices molecular selection of receptors in addition to clonal selection of B cells. In such case B cell delete their low affinity receptors and developed entirely new ones. This editing operation offers the ability to escape from the local minima. In addition to hyper mutation and receptor editing, a fraction of newcomer cells from the bone marrow are added into the lymphocyte pool to ensure and maintain diversity of population. Sometimes 5% to 8% of population is replaced by newly created B Lymphocytes.

In the context of the classification problem, following terminology is adopted for abovementioned biological concepts.

Hyper mutation can be seen as point mutation where one random node from the expression is selected and replaced by its counterpart selected randomly from the primitive set. For example a function node is replaced by a new random function node and a terminal node is randomly replaced by another terminal from the terminal set.

Receptor editing means some large random change in a given B lymphocyte as opposed to small change in case of hyper mutation. In this editing operation we have proposed a new '*depthlimited' mutation.* A random node is selected from the expression and the subtree rooted at this node is replaced by a new generated tree. To ensure '*depthlimited'* mutation we have applied a restriction that the "depth" of newly created tree must be less than or equal to the replaced subtree. This phenomenon will ensure that the population of expressions does not suffer from complexity increase. This means the well known *bloat* problem is eliminated.



(a) Parent Antibody

(b) Antibody after Receptor Editing(depthlimited mutation)

Fig. 2. Receptor editing

Newcomer cells are introduced by initializing some new expression trees and making them part of new population by discarding some unfit one from the population pool.

3.6 CLONAL-GP Algorithm

The overall CLONAL-GP algorithm can be described as follows

Step 1. Begin

Step 2. Randomly initialize population of B Lymphocytes *Ab*(B-cells/Antibodies)

Step 3. While(termination condition)

- Present Antigens *Ag* to the antibody population
- Calculate affinity of all *Ab*
- Select n highest affinity antibodies *Ab'*
- Create clone population C' by Cloning Ab' proportional to their affinities (higher affinity Ab will generate more clones)
- The C' is applied population maturation process resulting in C''(high affinity clones will be mutated less).
- Determine affinity of matured clones against antigens.
- Select high affinity individuals and compare them with memory cells.

```
        ▪ If the affinity of new B Lymphocytes is greater,
          then replaced memory cells.
        ▪ Replace k lowest affinity antibodies by k newly
          generated individuals.
Step 4. End while
Step 5. Output best memory cell
Step 6. End
```

4 Results

We have experimented the proposed framework over the well know Wisconsin breast cancer dataset from the UCI repository. The experiment has been performed by applying 10 fold cross validation twice on one random sampling of data. This process is repeated five times. Therefore tenfold cross validation is repeated ten times on five different partitions of data.

Table 1. Traditional GP Parameters

S. No	Population size	600
1	Crossover Rate	0.50
2	Mutation Rate	0.25
3	Reproduction Rate	0.25
4	Selection for cross over	Tournament selection with size 7
5	Selection for mutation	Random
6	Selection for reproduction	Fitness Proportionate selection
7	Mutation type	Point Mutation
8	Initialization method	Ramped half and half method with initial depth 6
9	Initial Depth	4 Standard GP, Variable in DepthLimited GP

The parameters used in our empirical analysis for Traditional and DepthLimited GP are mentioned in Table 1. These parameters have been empirically selected in our previous work [20].

The parameters for CLONAL-GP framework proposed in this paper are mentioned in Table2. We have tried to keep the parameters consistent with our previous work.

Table 3 provides a comparison between classification accuracy achieved by three different variants of GP. The standard GP uses a typical classification method used various propositions [2] [16]. DepthLimitedGP [20] has been proposed to avoid bloat for classifier evolution. It is almost similar to standard GP with an exception of crossover operator that restricts crossover between subtrees of same depth. We can see that the proposed CLONAL-GP has performed better than both other GP variants for WBC dataset. The reason for this better performance may be the mutation operators that ensure and constantly enforce diversity in each evolutionary cycle.

Table 2. CLONAL-GP parameters for classification

S. No	Parameter	Value/Description
1	Population	600
2	Memory	10
3	Number of members selected for cloning	100
4	Number of members replaced in each population	5% (30)
5	Hypermutation	Point mutation
6	Receptor editing	Subtree mutation (depthLimited)
7	Termination Condition	100 generations or affinity =1

Table 3. Comparison of different GP variants

Dataset		Standard GP	DepthLimited GP	CLONAL-GP
WBC	Accuracy	94.5%	95.8 %	97.2%
	Tree Size	958.2	31.8	30.6
	Time	18629 sec	11762 sec	10324 sec

The point and depthlimited mutation ensure that the trees will not increase in their complexities during evolution. This automatically eliminates the bloat prevalence in population.

Table 4. Comparison with other classification methods

Classifiers	CLONAL-GP	SVM	ANN	C4.5
Accuracy	97.2%	96.7%	95.6%	96%

Table 4 compares the performance of proposed algorithm with other classification algorithms and CLONAL-GP has achieved better performance over WBC data.

5 Conclusion

This paper presents a novel framework for using clonal selection in GP for classification purpose. The framework offers several advantages including, elimination of bloat, simpler classifiers. All these issues tend to overburden traditional GP which is a powerful tool for classifier evolution. The proposed framework is a part of ongoing research and various problems must be addressed in the future. This includes determination of optimal parameters. Moreover, we have investigated our proposition over binary classification problem. The work can be extended to incorporate multiclass classification problems.

References

[1] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
[2] Kishore, J.K., et al.: Application of Genetic Programming for Multicategory Pattern Classification. IEEE Transactions on Evolutionary Computation (2000)
[3] Muni, D.P., Pal, N.R., Das, J.: A Novel Approach To Design Classifiers Using GP. IEEE Transactions of Evolutionary Computation (2004)
[4] Ishida, Y.: The Immune System as a Self Identification Process: A Survey and a Proposal. In: Proceedings of the IMBS 1996 (1996)
[5] Burnet, F.M.: Clonal Selection and After. Theorotical Immunology, 63–85 (1978)
[6] Castro, L.N., Zuben, F.J.V.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transaction on Evolutionary Computation (2001)
[7] Koza, J.R.: Concept formation and decision tree induction using the genetic programming paradigm. In: Schwefel, H.-P., Männer, R. (eds.) PPSN 1990. LNCS, vol. 496. Springer, Heidelberg (1991)
[8] South, M.C.: The Application of Genetic Algorithms to Rule Finding in Data Analysis (1994)
[9] Pappa, G.A., Freitas, A.A.: Evolving Rule Induction Algorithms with Multiobjective Grammer based Genetic Programming. Knowledge and Information Systems (2008)
[10] Engelbrecht, A.P., Schoeman, L., Rouwhorst, S.: A Building Block Approach to Genetic Programming for Rule Discovery. In: Abbass, H.A., Sarkar, R., Newton, C. (eds.) Data Mining: A Heuristic Approach, pp. 175–189. Idea Group Publishing, USA
[11] Mendes, R.R.F., et al.: Discovering Fuzzy Classification Rules with Genetic Programming and Co-Evolution. In: Genetic and Evolutionary Computation Conference (2001) (Late Breaking Papers)
[12] Bojarczuk, C.C., Lopes, H.S., Freitas, A.A.: Discovering Comprehensible Classification Rules using Genetic Programming: A Case Study in a Medical Domain. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 953–958. Morgan Kaufmann, San Francisco (1999)
[13] Falco, I.D., Cioppa, A.D., Tarantino, E.: Discovering Interesting Classification Rules With GP. In: Applied Soft Computing, pp. 257–269 (2002)
[14] Zhang, M., Wong, P.: Genetic Programming for Medical Classification: A Program Simplification Approach. In: Genetic Programming and Evolvable Machines, pp. 229–255 (2008)
[15] Zhang, M., Ciesielski, V.: Genetic Programming For Multiple Class object Detection. In: Proceedings of the 12th Australian Joint Conference on Artificial Intelligence, Australia, pp. 180–192 (1999)
[16] Bojarczuk, C.C., Lopes, H.S., Freitas, A.A.: Genetic programming for knowledge discovery in chest-pain diagnosis. IEEE Engineering in Medicine and Biology Magazine, 38–44 (2000)
[17] Berek, C., Ziegner, M.: The Maturation of Immune Response. Imm Today, 400–402 (1993)
[18] Tonegawa, S.: Somatic Generation of Antibody Diversity. Nature, 575–581 (1983)
[19] Nussenzweug, M.C.: Immune Recepto Editing; Revise and Select. Cell, 875–878 (1998)
[20] Jabeen, H., Baig, A.R.: DepthLimited Crossover in Genetic Programming for Classifier Evolution. In: Computers in Human Behaviour. Elsevier, Ulsan (2009) (accepted)
[21] Loveard, T., Ciesielski, V.: Representing Classification Problems in Genetic Programming. In: IEEE Congress on Evolutionary Computation, pp. 1070–1077 (2001)
[22] Smart, W., Zhang, M.: Using Genetic Programming For Multiclass Classification By Simultaneously Solving Component Binary Classification Problems. LNCS. Springer, Heidelberg (2005)