Proceedings of the 8th Brazilian Symposium on Neural Networks (SBRN), São Luis, MA, Los Alamitos: IEEE Computer Press, p. ?-?, 2004.

Induction of Fuzzy Classification Rules with an Artificial Immune System

Roberto T. Alves[1], Myriam R. Delgado[1], Heitor S. Lopes[1], Alex A. Freitas[2]

[1] *CPGEI, CEFET-PR, Av. Sete de Setembro, 3165, CEP: 80230-901 Curitiba, Brasil*
roberto_alves@msn.com, myriam@dainf.cefetpr.br, hslopes@cpgei.cefetpr.br
[2] *Computing Laboratory, University of Kent, Canterbury, CT2 7NF, UK*
A.A.Freitas@kent.ac.uk

Abstract

Fuzzy systems were designed to compute with uncertainties and linguistic information and allow us to develop mathematical tools for information processing. Artificial immune systems (AIS) consist of methods inspired by the biological immune system and designed for solving real-world problems. This work integrates these two kinds of systems, proposing a novel AIS for discovering fuzzy classification rules from data. The results of the proposed algorithm are compared with the results of C4.5Rules, a very popular algorithm for discovering classification rules.

1. Introduction

Data mining consists of extracting knowledge from real-world data sets, and it is the subject of extensive research [1]. This work focuses on the classification task of data mining, where the goal is to predict the class of an example (record) – out of a predefined set of classes – based on the values of attributes for that example. This work involves artificial immune systems (AIS) and fuzzy systems (FS). FS are inspired by the theory of fuzzy sets, introduced by Zadeh [2], and they have a fundamental role in a wide range of information processing areas, bridging the gap between numerical and symbolic processing [3]. AIS originated from attempts to model and apply immunological principles to the development of novel computational tools, and they have also been used in a wide range of areas, ranging from failure and anomaly detection to optimization and control to machine learning and data mining [4].

This work proposes a method based on AIS to discover fuzzy classification rules from data. The AIS obtains, via an evolutionary process extended with data mining procedures (such as rule pruning), a classifier consisting of fuzzy rules of the form: IF (fuzzy conditions) THEN (class). The interpretation of each rule is that, if an example satisfies the fuzzy conditions, then the example is assigned the class predicted by the rule. This knowledge representation has the advantage of being intuitively comprehensible to the user, and the use of fuzzy conditions improves the comprehensibility of the rule and the rule's ability to cope with uncertainties typically found in real-world data.

2. Theoretical background

2.1. Fuzzy systems

The theory of fuzzy systems uses symbols, called linguistic terms, which have well-defined semantics. After being converted into membership functions of fuzzy sets, linguistics terms allow the numerical processing of the corresponding symbols or concepts. Fuzzy systems are very effective in expressing the ambiguity and subjectivity of human reasoning.

The membership functions determine to which degree a given object belongs to a set. In a fuzzy system, this degree varies continuously in the range [0…1]. Membership functions can take different shapes, from the simplest ones (triangular functions) to more complex functions (parameterized by the user).

In a classification problem with *c* classes and *n* attributes, fuzzy rules can be written as [5]:

$$R_j: \text{ If } x_1 \text{ is } A_1^j \ldots \text{ and } x_n \text{ is } A_n^j \text{ then Class } C_j, \quad j = 1,..,N \quad (1)$$

where $\mathbf{x} = (x_1, ..., x_n)$ is an *n*-dimensional pattern vector; A_1^i is an antecedent linguistic value such as *small* or *large (i=1,…,n);* C_j is a consequent class; and *N* is the number of fuzzy IF-THEN rules. The antecedent part of each rule is specified by a combination of linguistic values, produced by the partition of the universe (i.e.,

the domain of each attribute) into a set of linguistic terms.

In this work only the rule antecedents are evolved by the AIS. (The rule consequent (predicted class) is fixed for all rules during an AIS run, as will be explained later.) The partition of the universe is fixed and specified *a priori* by the user. The number of rules and the number of conditions per rule are automatically determined by the AIS. This flexibility is essential in data mining, where the best values for these numbers are not know *a priori* for the data being mined.

2.2. Artificial immune systems (AIS)

AIS is a relatively new computational intelligence paradigm [6]. AIS are inspired by the biological immune system and are designed to solve real-world problems [4]. The biological immune system is a complex system that involves distinct "agents" (antibodies) interacting with each other, with the function of protecting the organism from damage caused by invading agents (antigens), such as bacteria and viruses.

Some important characteristics of the immune system from a computational viewpoint are as follows [6]:

- The immune system can recognize and classify different patterns and produce selective responses. In addition, it uses a combinatorial process to generate a diverse set of lymphocyte receptors, in order to increase the chance that at least some lymphocytes recognize a given antigen;
- The system learns, by experience, the structure of a given antigen. When B cells are activated, some of them become memory cells, with an extended lifetime. These cells help the organism to produce a faster immune response when the same antigen is encountered in the future. The system automatically determines a balance between economy and performance, maintaining approximately just the sufficient number of these cells.
- The mechanisms of immune response are self-regulated by nature. There is no central organ controlling the immune system. The regulation of the immune response can be local or systemic, depending on the kind of antigen and its location;
- The immune response and the proliferation of immune cells occur under determined affinity thresholds (strength of the binding between antibody and antigen);
- The processes of clonal expansion and somatic hypermutation produce immune cells with high affinity to the invading antigen. These are the processes on which the AIS proposed in this paper is based, and so they are explained next.

The process of clonal expansion is a form of natural selection. When a B-cell is activated, by binding – to some degree – to an antigen, it produces several clones. These clones undergo mutation, which changes their ability to recognize (bind to) the antigen. The clones that best recognize the antigen will have a higher proliferation rate, whereas clones that are bad at recognizing the antigen will die. Hence, the process is adaptive. The proliferation rate of a B-cell is proportional to the affinity between the B-cell and the antigen; whereas the mutation rate is inversely proportional to that affinity. As a result of clonal selection and somatic hypermutation, the system will produce B-cells that are highly adapted to recognize the target antigen.

3. Description of IFRAIS

The goal of the proposed AIS algorithm, called IFRAIS (Induction of Fuzzy Rules with an Artificial Immune System), is to evolve fuzzy conditions for a classification rule's antecedent, in order to maximize the classification accuracy of the rule. In essence, the algorithm maintains a population of antibodies, each of them representing a candidate classification rule antecedent (see section 3.1). Each antigen corresponds to an example (record) to be classified. The algorithm is based on the previously-explained processes of clonal selection and somatic hypermutation. It also uses a stochastic rule pruning procedure. This procedure incorporates some "knowledge" of the task being solved (classification) into the algorithm, which usually improves the performance of an evolutionary algorithm. Note that the principles of clonal selection and somatic hypermutation have been used to design other AIS algorithms [4,7] but IFRAIS is the first AIS for discovering fuzzy classification rules based on those principles.

3.1. Encoding issues and affinity function

Recall that each antibody represents a rule antecedent. Each rule antecedent is formed by a conjunction of conditions, each of them involving a continuous or categorical attribute. Continuous attributes are fuzzified, by dividing their universe (attribute domain) into three linguistic terms – *low,*

medium, high. These terms are represented by triangular functions, for the sake of simplicity. The antibody genotype can contain irrelevant conditions (indicated by a flag), which are not expressed in the decoded rule antecedent and therefore are ignored when computing the affinity between the antibody (rule antecedent) and an antigen (example to be classified). When decoding the antibody, there is a natural restriction that the number of conditions in the decoded rule has to be at least 1. The flexibility to represent rules with different number of conditions is essential in data mining, where the optimal number of conditions for each rule is not know *a priori*.

Figure 1 shows the antibody representation, where the index i denotes the i-th attribute of the data set being mined, V_i denotes the value assigned to the i-th attribute, and B_i denotes a Boolean flag indicating whether or not the corresponding condition will be expressed in the decoded rule antecedent. The figure 1 also shows that each value V_i is associated with a triangular membership function.

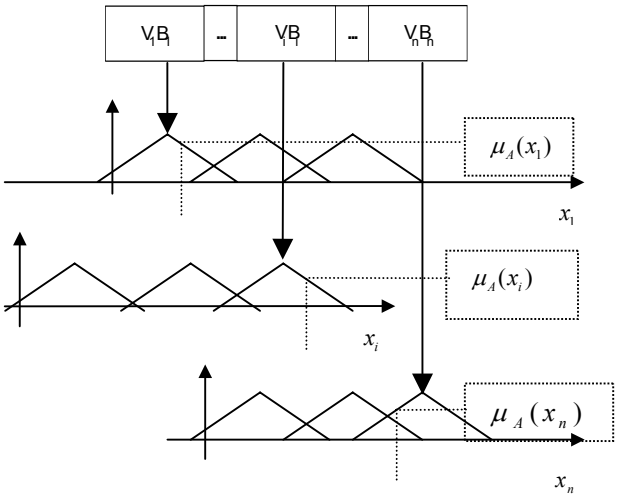

Figure 1. Relation antibody ↔ fuzzy rule.

The affinity (degree of matching) between an antigen and an antibody represents the degree to which the example satisfies the rule antecedent. This affinity is measured by a fuzzy AND of the degrees of matching ($\mu_A(x_n)$) for all the rule conditions decoded from the antibody. In this work we use the standard fuzzy AND, given by the *min* operator, so that the affinity between an antigen and an antibody is given by Equation (2):

$$\begin{aligned} Afin(j) &= f_j(\mu_{A1}(x_1),\cdots,\mu_{A_n}(x_n)) \\ &= \mu_{A1}(x_1)\wedge\cdots\wedge\mu_{A_n}(x_n) \end{aligned} \quad (2)$$

Each rule j is associated with an activation threshold L_j. An example is said to satisfy a rule antecedent (or the rule is said to cover the example) if the affinity between the example and the rule antecedent is greater than L_j. Intuitively, this threshold can have a great influence in the performance of the algorithm. Hence, this work proposes a procedure that automatically chooses the best value of this threshold – out of a range of values – for each rule. This relieves the user from the difficult task of specifying this threshold and it increases the flexibility of the system, since different rules can have different values of this threshold, depending on the conditions in the rule antecedent. More precisely, the procedure considers m uniformly distributed values of L_j in the range [0.5,…,0.7]. In this work m = 20 (a value chosen to represent a reasonably large range of values), so that the procedure considers all values of L_j in {0.50, 0.51, … 0.69, 0.70}. The procedure chooses, out of those values, the value L_j that maximizes the fitness of the antibody (computed as will be explained later).

3.2. Evolutionary Process

Each run of the algorithm discovers one fuzzy classification rule, so that the algorithm has to be run multiple times to discover multiple rules. This is obtained by using a sequential covering (SC) procedure (often used in data mining), as follows. The SC procedure starts with an empty set of discovered rules. Then it performs a loop over the classes. For each class, the algorithm will be run as many times as necessary to discover rules covering all or almost all the examples belonging to that class. More precisely, for each class the procedure initializes a variable TS with the set of all examples in the training set, and then calls the AIS algorithm (described in Figure 2) to discover a classification rule predicting the current class. The AIS returns the best evolved rule, which is added to the set of discovered rules. Next, the SC procedure removes from TS the examples that are correctly covered by the discovered rule, i.e. the examples that satisfy the rule antecedent (to a degree greater than L_j, as explained in section 3.1) and have the class predicted by the rule. Then the AIS algorithm is called again, to discover a rule from the reduced training set, and so on. This iterative process is repeated until the number of uncovered examples of the current class is smaller than a small threshold, called MaxUncovExamp (maximum number of uncovered examples) and set to 5 in our experiments. This avoids that the AIS tries to discover a rule covering a very small number of examples, in which case the rule would not be statistically reliable. This process is repeated for all the classes, producing a set of fuzzy classification rules covering almost all training examples. Examples that are not covered by

any rule are simply classified by a default rule (see section 3.3). At the end of this training phase, the fitness of all rules is recomputed by considering the entire training set, in order to have a better estimate of rule quality to be used in the classification of test examples (section 3.3).

3.2.1 Rule evolution based on clonal selection

The evolution of fuzzy rule antecedents is performed by an AIS based on the clonal selection principle (section 2.2). The pseudocode of the algorithm (version 1.1) is shown in Figure 2.

```
Create initial population (of size N) of antibodies at random;
Prune each rule antecedent in a stochastic way;
Compute fitness of each antibody;

FOR i = 1 to Number of Generations
  NUMCLONES = 0;
  WHILE NUMCLONES =< (N - 1)
    Perform tournament selection, getting the winner to be
    cloned;
    Produce C clones of the antibody, where C is proportional to
    the antibody's fitness and 1 < C <= MAXNUMCLONES;
    NUMCLONES = NUMCLONES + C;
  END WHILE
  FOR EACH produced clone
    Mutate clone with a rate inversely proportional to its fitness;
    Prune each clone's rule antecedent in a stochastic way;
    Compute fitness of the clone;
  END FOR EACH clone;
  Elitism - maintain the best antibody in the population
  Population update - replace the other N – 1 antibodies in the
   population by the N – 1 just-produced clones;
END FOR I;
Return the rule whose antecedent consists of the antibody with
the best fitness among all antibodies produced in all generations
```

Figure 2. AIS based on clonal selection.

The AIS starts by randomly creating an initial population of antibodies, where each antibody represents the antecedent of a fuzzy classification rule. For each rule (antibody), the system prunes the rule – using the rule pruning procedure proposed by [8] – and computes the fitness of the antibody, according to equation 4, as described in section 3.2.2. Rule pruning has a twofold motivation: reducing the overfitting of the rules to the data and improving the simplicity (comprehensibility) of the rules. The basic idea of the rule pruning procedure used in this work is that, the lower the predictive power of a rule condition, the more likely it will be removed from the rule. The predictive power of a condition is estimated by its information gain, a very popular heuristic measure of predictive power in data mining [9].

After rule pruning, the outer FOR loop starts by initializing NUMCLONES to 0 and performing a tournament selection procedure, in order to select the winner antibody that will be cloned in the next step. Tournament selection is well-known and often used in evolutionary algorithms [10]. Once the winner antibody has been selected, the algorithm performs its core step, which is inspired by the clonal selection principle of the natural immune system. First, for each antibody to be cloned the algorithm produces C clones. The value of C is proportional to the fitness of the antibody. The number of clones increases linearly with the antibody fitness when $0 < Fit(Ab) < 0.5$, and any antibody with a fitness greater than or equal to 0.5 will have MAXNUMCLONES clones. We set the value of MAXNUMCLONES to just 10 to prevent the clone population from being very large, which would not only be inefficient but also possibly lead to overfitting of the rules to the data. Tournament selection and clonal expansion are performed while the total number of clones generated does not exceed $N-1$ (where N is the population size). This part of the pseudocode is different from the version 1.0 of the algorithm [11], where the number of tournaments was a user-specified parameter. The current version (1.1) avoids the need for that parameter and produced slightly better results.

Next, each of the just-produced clones undergoes a process of hypermutation. This process follows the basic idea of [4], where the mutation rate is inversely proportional to the clone's fitness (i.e., the fitness of its "parent" antibody). In other words, the lower the fitness (the worse a clone is), the higher its mutation rate. More precisely, the mutation rate for a given clone cl, denoted $mut\_rate(cl)$, is given by Equation 3:

$$mute\_rate(cl) = \alpha + ((\beta - \alpha) * (1 - fit(cl))) \qquad (3)$$

where α and β are the smallest and greatest possible mutation rates, respectively, and $fit(cl)$ is the fitness of clone cl. The fitness of a clone is a number normalized between 0 and 1, as will be explained later, so that the above formula collapses to α when the clone has the maximum fitness of 1, and it collapses to β when the clone has the minimum fitness of 0. In our experiments we have set α and β to 20% and 50%, respectively – empirically-determined values. These numbers represent the probability that each gene (rule condition) will undergo mutation. Once a clone has undergone hypermutation, its corresponding rule antecedent is pruned. Finally, the fitness of the clone is recomputed, using the current training set.

The next step consists of population updating. With the exception of the best antibody, which is preserved in the population by elitism, all the other N – 1

antibodies in the current population are replaced by the just-generated $N - 1$ clones, in order to keep the population size constant. The population size was set to $N = 50$, and the number of generations was set to 50. These values were empirically determined. Finally, the algorithm returns the best evolved rule, which will then be added to the set of discovered rules. The best evolved rule consists of the rule antecedent ("IF part" of the rule) represented by the antibody with the best fitness, across all antibodies produced in all generations, and of the rule consequent ("THEN part" of the rule) containing the class associated with all the antibodies in the current AIS run.

Note that the population updating method is essentially a generational one with elitism. This population updating method is often used in genetic algorithms, and it has the advantage of being a simple way to keep a constant population size. The algorithm also borrows from evolutionary algorithms the idea of tournament selection. However, it is an AIS that does not use crossover, is based on the immune-inspired principles of clonal selection and somatic hypermutation, and uses not only a fitness function but also an affinity function. In addition, it is an AIS tailored for discovering classification rules: not only its antibody encoding and fitness function are tailored for this task, but it also uses a stochastic rule pruning procedure which is very specific to this task.

3.2.2 Fitness computation

We now turn to the fitness function used by the AIS algorithm. The fitness of an antibody *Ab*, denoted by *fit(Ab),* is given by Equation 4:

$$fit(Ab) = \frac{TP}{TP + FN} \times \frac{TN}{TN + FP} \quad (4)$$

where the variables *TP*, *FN*, *TN* and *FP* have the following meaning:

- *TP* = number of true positives, i.e. number of examples satisfying the rule and having the same class as predicted by the rule;
- *FN* = number of false negatives, i.e. number of examples that do not satisfy the rule but have the class predicted by the rule;
- *TN* = number of true negatives, i.e. number of examples that do not satisfy the rule and do not have the class predicted by the rule;
- *FP* = number of false positives, i.e. number of examples that satisfy the rule but do not have the class predicted by the rule.

This fitness function was proposed by [12] and has also been used by other evolutionary algorithms for discovering classification rules [8,13]. However, in most projects using this function the discovered rules are crisp, whereas in our work the rules are fuzzy. Hence, in this work the computation of the *TP*, *FN*, *TN* and *FP* requires, for each example, measuring the degree of affinity (fuzzy matching) between the example and the rule, as described in section 3.1.

3.3. Classifying examples in the test set

The rules induced from the training set are used to classify new examples in the test set (unseen during training) as follows. For each test example, the system identifies the rule(s) activated for that example. Recall that a rule j is activated for example k if the affinity between j and k is greater than the affinity threshold for rule j.

When classifying a test example, there are three possible cases. First, if all the rules activated for that example predict the same class, then the example is simply assigned to that class. Second, if there are two or more rules predicting different classes activated for that example, the system uses a conflict resolution strategy consisting of selecting the rule with the greatest value of the product of the affinity between the rule and the example (Equation 1) by the fitness of the rule (Equation 4), i.e., it chooses the class C given by Equation 5:

$$C = C_j = \max_j (Afin_j \times Fit_j) \quad (5)$$

Third, if there is no rule activated for the example, the example is classified by the "default" rule, which predicts the most frequent class in the training set.

4. Computational results

The proposed algorithm was evaluated in 6 public domain data sets: BUPA, CRX, Wisconsin Cancer, Votes, Hepatities, Ljubljana Cancer. These data sets are available from the well-known UCI repository (*http://www.ics.uci.edu/~mlearn/MLRepository.html*). The experiments used a well-known method for estimating predictive accuracy, namely 5-fold cross-validation [9].

Table 1 shows the number of continuous and categorical attributes for each data set (recall that only continuous attributes are fuzzified). Note that the Votes data set does not have any continuous attribute to be fuzzified. This data set was included in the experiments to evaluate IFRAIS' performance in the "degenerated" case of discovering crisp rules only.

Table 2 shows the average accuracy rate in the test set (computed by cross-validation) for IFRAIS and for C4.5Rules, a very popular data mining algorithm for

discovering (crisp) classification rules [14]. The numbers after the "±" symbol are the standard deviations. For each data set, the highest accuracy rate between the two algorithms is shown in bold.

As shown in Table 2, IFRAIS obtained somewhat better accuracy rates in four data sets, whereas C4.5Rules obtained considerably better accuracy rates in the other two data sets. Overall, these can be considered promising results – specially taking into account that C4.5Rules is the result of decades of research in decision tree and rule induction algorithms; whereas the whole area of AIS is still relatively new and IFRAIS can be considered (to the best of our knowledge) the first AIS algorithm to discover fuzzy classification rules.

Table 1. Data sets and number of attributes.

Data set	# Attrib.	# Cont.	# Categ.
Crx	15	6	9
Bupa	6	6	0
Hepatities	19	6	13
Votes	16	0	16
Wisconsin	9	9	0
Ljubljana	9	9	0

Table 2. Accuracy rate in the test set.

Data set	IFRAIS	C4.5 rules
Crx	86.38 ± 0.93	**90.22 ± 1.59**
Bupa	57.4 ± 1.33	**67.40 ± 1.60**
Hepatities	**81.99 ± 1.33**	76.32 ± 2.79
Votes	**95.84 ± 0.79**	94.82 ± 0.82
Wisconsin	**95.76 ± 0.90**	95.32 ± 1.09
Ljubljana	**70.42 ± 1.41**	68.80 ± 4.45

5. Conclusions and future work

This work proposed a novel AIS for fuzzy classification rule induction. The algorithm obtained accuracy rates roughly competitive with C4.5Rules, a very respectable algorithm for classification rule induction.

Future work – which has the potential to significantly improve the predictive accuracy of the discovered rules – includes: (a) developing a procedure for automatically determining the number of linguistic terms for each continuous attribute, rather than just using a fixed number as in the current version; (b) developing new heuristics based on the clonal selection principle; (c) developing a local search method to improve the performance of the algorithm in complex data sets.

6. References

[1] M.S. Chen, J.Han, and P.S. Yu, "Data mining: an overview from database perspective", *IEEE Transactions on Knowledge and Data Engineering*, 8(6), 1996, pp. 866-883.

[2] L.A. Zadeh, "Fuzzy sets", *Information and Control*, 9, 1965, pp 338-352.

[3] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*, MIT Press, Cambridge, 1998.

[4] L.N. Castro and J. Timmis, *Artificial Immune Systems: A New Computation Intelligence Approach*, Springer-Verlag, Berlin, 2002.

[5] H. Ishibuchi, and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems", *IEEE Transactions on Fuzzy Systems*, 9(4), 2001, pp. 506-515.

[6] D. Dasgupta, *Artificial Immune Systems and Their Applications,* Springer-Verlag, Berlin, 1999.

[7] L.N. De Castro and F.J. Von Zuben, "Learning and optimization using clonal selection principle", *IEEE Trans. on Evolutionary Computation,* 6(3), 2001, pp. 239-251.

[8] D.R. Carvalho and A.A. Freitas, "A hybrid decision tree/genetic algorithm for coping with the problem of small disjuncts in data mining", in: *Proceedings of Genetic and Evolutionary Computation Conference*, 2000, pp.1061-1068.

[9] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation*, Morgan Kaufmann, San Francisco, 2000.

[10] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[11] R.T. Alves, M.R. Delgado, H.S. Lopes and A.A. Freitas, "IFRAIS: An Artificial Immune System for Fuzzy-Rule Induction in Data Mining", *To appear in Proceedings of Parallel Problem Solving in Nature (PPSN)*, 2004.

[12] H.S. Lopes, M.S. Coutinho and W.C. Lima, "An evolutionary approach to simulate cognitive feedback learning in medical domain"*, in: E. Sanchez, T. Shibata, L.A. Zadeh (eds.), Genetic Algorithms and Fuzzy Logic Systems*. World Scientific, Singapore, 1997, pp. 193-207.

[13] C.C. Bojarczuk, H.S. Lopes and A.A. Freitas, "A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets", *Artificial Intelligence in Medicine*, 30(1), 2004, pp.27-48.

[14] J.R. Quinlan, C4.5: *Programs For Machine Learning*, Morgan Kaufmann, San Mateo, 1993.