ARTIFICIAL IMMUNE SYSTEM CLASSIFICATION OF MULTIPLE-CLASS PROBLEMS

DONALD E. GOODMAN, JR.
Mississippi State University
Department of Psychology
Mississippi State, Mississippi

LOIS C. BOGGESS
Mississippi State University
Department of Computer Science
Mississippi State, Mississippi

ANDREW B. WATKINS
University of Kent at Canterbury
Computing Laboratory
Canterbury, Kent, United Kingdom

ABSTRACT

A new classifier, AIRS, based on the principles of resource-limited artificial immune systems, has been shown recently to consistently rank among the best five to eight known classifiers for a number of well-studied classification problems, including the Iris data, the Cleveland heart disease data, and others. However, each of these previous test problems has involved only a few classes. In this paper, we discuss the general application of AIRS to multi-class problems, and we compare it to a similar well-known classifier, Kohonen's LVQ, on both simulated and real-world data sets.

INTRODUCTION

In this paper we explore the characteristics of the Artificial Immune Recognition System (AIRS), a recently introduced classifier based on the principles of resource-limited artificial immune systems. AIRS has been tested on a number of publicly available classification problem sets. When compared against a listing of the 10-30 best classifiers on those problems (Duchs, 2000a,b), AIRS's performance was found to be competitive with the top five to eight classifiers for every task but one, in which AIRS ranked second (Watkins, 2001; Watkins and Boggess, 2002a; Duch 2000a).

Different classification tools offer different advantages: k-nearest neighbor classifiers, for example, are often among the best classifiers for certain problems, but they tend to be computationally expensive. Radial basis networks offer a certain amount of generalization, but tend to require as many centers as there are training instances for best performance. Artificial neural networks are efficient, once trained, and generalize from the training instances, but the best network architecture for a given problem is seldom known and often requires extensive experimentation

One of the advantages of AIRS is that it is not necessary to know what the appropriate settings for the classifier are in advance. The most important element of the classifier is self-determined. In our experience, almost any setting of AIRS's parameters result in a classifier that is only a few percentage points less accurate than an optimized version of the system, on each of the problems

with which we have experimented. It is possible to tweak a number of parameters to optimize the classifier empirically for any given problem. Once trained, the classifier itself is reminiscent of a k-nearest neighbor classifier. However, in practice, generalization has been incorporated into the nodes that are part of the resulting classifier, and typically AIRS uses about half as many such nodes as k-nearest neighbor uses.

The remainder of this paper is organized as follows: First, we present a brief overview of concepts from artificial immune systems in general and resource-limited artificial immune systems in particular which are relevant to the AIRS classifier. We then present a summary of results of applications of AIRS to a variety of well-known classification problems. Next we present the results of investigating the behavior of AIRS in two new directions. a) We tested the performance of AIRS in artificially constructed cases where class boundaries are complex. For these experiments, the classes were not only not linearly separable, but in many cases discontinuous. b) We also tested the performance of AIRS in real-world classification problems as a function of the number of dimensions of the feature space. For purposes of comparison, we also tested Kohonen's LVQ classifier against the same data sets, and we supply those comparisons.

As will be seen, AIRS continues to exhibit very promising behavior on the new classification tasks, including classification tasks with very high-dimensioned feature spaces.

ARTIFICIAL IMMUNE SYSTEMS

Biological immune systems are complex and not fully understood. They offer a variety of metaphors and paradigms that can be adapted to computational tasks. Researchers in the computational sciences tend to focus on only a few of the metaphors that are available from these biological systems. We will confine our discussion of biological immune systems to a very small subset of concepts.

When a biological immune system detects an invading pathogen, two kinds of lymphocytes are part of the response. These cells, called T cells and B cells, behave like pattern recognizers. In nature, when pathogens invade a body, special cells called antigen-presenting cells, interact with the pathogens so that their antigens – their "relevant features" – are available on the surfaces of the antigen-presenting cells. T cells and B cells that have high affinity to a given antigen can trigger an immune system response. The B cells with high affinity change state so that they begin to multiply and to mutate. Successive generations of these B cells have higher and higher affinity to the presenting antigen. Very large numbers of these B cells are produced, though the lifetime of a typical B cell is short. Some B cells, however, live indefinitely, acting like a sort of “memory”. Long after the original pathogen is destroyed, the immune system can respond more rapidly if a similar pathogen is encountered.

One role of the T cell is to protect the body against attacking its own cells. As T cells are produced and begin to mature, those which respond to "self" – to the body's own cells – are destroyed. Therefore the mature T cells that are involved in an immune response to a potential pathogen will not respond if the "pathogen" is in actuality a cell of the host body. Since B cells are unable to change state to begin multiplying and mutating unless both a B cell and a T cell have high affinity for the same antigen, B cells are thus unable to attack its host.

Even this very simplified view of part of natural immune systems offers a variety of paradigms for computational scientists. For example, the metaphor of antigen-presenting cells and T cells is used for some kinds of artificial immune systems in the field of computer security. Consider the scenario where it is very important to know whether one's data has been compromised. One kind of artificial immune system "chops up" a copy of the data into small parts, and generates a large number of random "T cells", destroying all those that have too close a match to the data fragments. The remaining cells react only to data that are different from the original source of training – which would be present only if the original data have been modified or corrupted (Forrest, et al. 1994).

Resource limited artificial immune systems (Timmis and Neal 2001), which have been the primary inspiration for AIRS, focus on B cells. In this paradigm, there is no distinction between a pathogen and its features. Indeed, since a B cell is a pattern matcher, there is no difference in the representation form of a B cell and the antigen that it matches. Each is simply a vector in the feature space. If we think of the feature vectors of a classification problem's training and test sets as antigens, then the B cells of such a system are initially random vectors in the same feature space as the training and test sets. In the work by Timmis et al., AINE, these antigens are presented to a system of B cells, and those with highest affinity begin to reproduce and to mutate. Rather than keep track of large numbers of identical B cells, as happens in such systems, AINE keeps one representative of each group of identical cells, with a resource number which stands for the proportion of the overall population of cells characterized by that representative. These representatives, called ARBs (artificial recognition balls), compete for the right to stay alive. As each new antigen is introduced to the system, new B cells are generated through cloning and mutation, and existing B cells, or rather ARBs, are eliminated if they are not stimulated enough over time and exposure to antigens. Over time only the ARBs that respond most strongly to the presented antigens survive. The resulting system is a clustering system.

AIRS – AN ARTIFICIAL IMMUNE CLASSIFICATION SYSTEM

In contrast to AINE, AIRS is a classification system. The details of the AIRS algorithm are given in (Watkins, 2001) and (Watkins and Boggess, 2002b). A supervised learning system, AIRS pays attention to the class of each antigen (feature vector in the training set) while generating ARBs that respond to the antigens. The most responsive ARBs generated are promoted to the memory cell pool (de Castro and Von Zuben 2002), which is the eventual classification tool that remains after training is complete. The mutation built into the algorithm results in ARBs that are successful in part by *not* being identical to training vectors, while being similar enough to later training vectors to be highly competitive. This is the source of AIRS's ability to generalize from the data. AIRS "grows" the memory cell pool from an initial size set by the user and seeded by training vectors. Typically, the number of cells that AIRS creates in the memory cell pool is about half the number of training cells presented to it. Once training is complete, the memory cells are used as a k-nearest neighbor classification system for test data. As can be seen below, even though AIRS typically uses half the number of classification vectors as k-Nearest Neighbor (kNN,) it often outperforms kNN on the same data.

Since AIRS uses Euclidean distance in feature space to calculate affinity, it was initially tested on classification tasks using real-valued features (the Iris data, Ionosphere data, Pima Indian Diabetes data, and Sonar data from the UCI repository (Blake and Merz 1998)) (Watkins and Boggess 2002a).

PERFORMANCE OVER INCREASING NUMBER OF CLASSES

To explore the behavior of AIRS in response to increasing numbers of classes, we devised an artificial classification problem in two-dimensional space so that we could visualize the results. Accordingly, we created mappings from $\mathcal{R}^2$ to three classes, to five classes, to eight classes and to 12 classes.

Figures 1a. and b. represent two of these mappings. No attempt was made to optimize AIRS for the problem - that is, there are a number of parameters which can be modified by the user (number of seed cells to begin training with, maximum number of resources allowed, stimulation thresholds, mutation rate, cloning rate, the k value for kNN classification in the final classifier, and so on) - and we simply used the "as-shipped" default values of the system. In order to compare what we were seeing against a known classifier which we thought might behave similarly, we also ran Kohonen's Learning Vector Quantization (LVQ) against the same problem sets. AIRS makes its own determination of the appropriate number of memory cells in the final classifier, but LVQ expects the user to decide on the right number of output vectors. Therefore we set the number of output vectors for LVQ to be about the same number of vectors for each of the problems, about 100 to 130 output vectors. To guard against the possibility that such a large number of output vectors for LVQ might be "overkill" and that it might perform much better with a small number of output vectors, we also experimented with starting LVQ with a small number of vectors, and adding output vectors until LVQ's performance peaked and began to degrade or until it plateaued. These two different forms of LVQ are shown as "big LVQ" and "optimized LVQ" in the data below (Fig. 2). The data represent averages over three runs of 10-way cross validation. Note that "optimized LVQ" is clearly not optimal, but it does reflect what a typical researcher might use in an LVQ classifier after a reasonable attempt at determining the right number of output vectors for the researcher's classification problem.

EFFECT OF INCREASING NUMBER OF FEATURE

Finally, we investigated the response of AIRS to increasing numbers of features. Because LVQ seems to perform very well when required to use a number of output vectors comparable to what AIRS determines to be a good representation, we continued our comparisons. As before, all values are averages over runs of 10-way cross validation. With the exception of the arrhythmia problem, the AIRS classifier has been optimized. We note that the average performance of the AIRS classifier for the credit application classification problem is marginally better than the best prior classifiers known to us for this problem: a Bayesian classifier with Adaboost (Ridgeway et al. 1998) and a C4.5 decision tree developed with a special pruning algorithm (Webb 1996).

CONCLUSIONS

We have summarized the performance of a recently introduced new classifier, AIRS, which is based on concepts from resource limited artificial immune systems. We explored the behavior of this promising new classifier on an artificial but difficult problem as we increased the number of classes while holding the number of features steady. For purposes of comparison, we performed the same experiments with a well-known classifier, Kohonen's LVQ. We then traced the behavior of both AIRS and our reference classifier across a number of publicly available classification problems with increasing numbers of features. In the course of this experiment AIRS's average performance on one of the problems was the best that we are aware of for that problem.

REFERENCES

Blake, C.L. and Merz, C. J. 1998. UCI Repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html. University of California at Irvine, Department of Computer Science.

de Castro, L. N. and Von Zuben, F. J. 2002. Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation, Special issue on Artificial Immune Systems*. ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/lnunes/ieee_tec01.pdf

Duch, W. 2000a. "Datasets used for classification: Comparison of results," http://www.phys.uni.torun.pl/kmk/projects/datasets.html.

Duch, W. 2000b. "Logical rules extracted from data," http://www.phys.uni.torun.pl/kmk/projects/rules.html.

Forrest, S., Perlson,A. S., Allen, L. and Cherukuri, R. 1994. Self-Nonself Discrimination in a Computer. In Proceedings of the IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, Los Alamitos, CA, pp 202-212, 1994.

Ridgeway G., Madigan, D., Richardson, T. and O'Kane, J. 1998. "Interpretable Boosted Naïve Bayes Classification," Proceedings, Fourth International Conference on Knowledge Discovery and Data Mining (R. Agrawal, P. Stolorz, G. Piatetsky-Shapiro, eds.), pp. 101-104.

Timmis, J. and Neal, M. 2001. A Resource Limited Artificial Immune System for Data Analysis. *Knowledge Based Systems*, 14(3-4):121-130.

Watkins, A. 2001. AIRS: A Resource Limited Artificial Immune Classifier. M.S. thesis, Department of Computer Science. Mississippi State University.

Watkins, A. and Boggess, L. 2002a. A New Classifier Based on Resource Limited Artificial Immune Systems. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC2002),* IEEE Press.

Watkins, A. and Boggess, L. 2002b. A Resource Limited Artificial Immune Classifier. In *Proceedings of the 2002 Congress on Evolutionary Computation(CEC2002),* Special Session on Artificial Immune Systems. IEEE Press.

Webb, G.I. (1996) "Further Experimental Evidence against the Utility of Occam's Razor", Journal of Artificial Intelligence Research 4 (1996) 397-417.

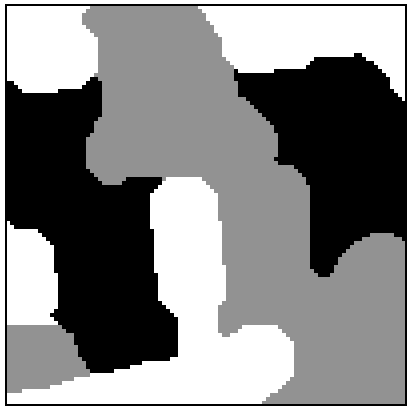

Figure 1a. Three classes in a 2D feature space.

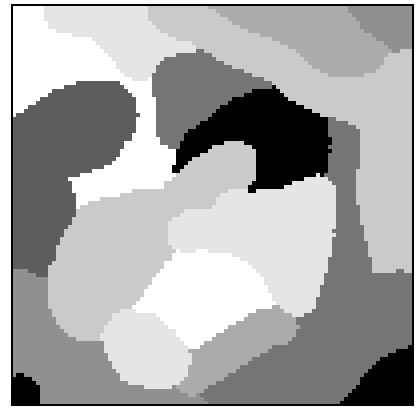

Figure 1b. Eight classes in a 2D feature space

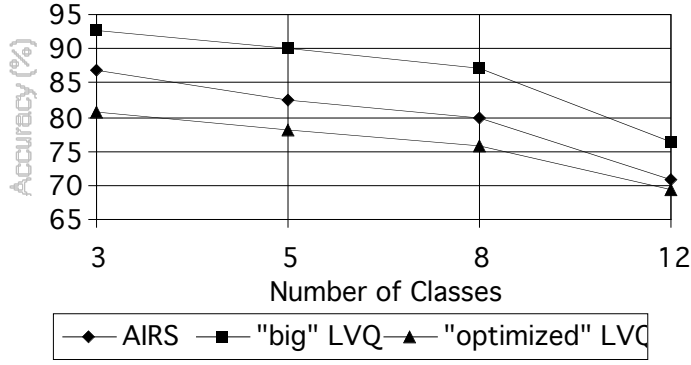

Figure 2. Plot of AIRS versus LVQ on Increasing Number of Classes

Dataset	Features	AIRS	"big" LVQ	"optimized" LVQ
balance-scale	4	96.7	92.8	94.3
pima-indian-diabetes	8	74.1	72.0	70.2
wisconsin-breast-cancer	9	97.2	96.8	96.7
credit.crx	15	85.7	68.3	66.7
ionosphere	34	94.9	85.5	86.9
arrythmia	279	59.5	68.3	69.9

Table 1: Comparison of AIRS and LVQ on Increasing Number of Features (datasets from literature).