Profiling Novel Classification Algorithms: Artificial Immune Systems

Peter van der Putten
LIACS
Leiden University
Leiden, the Netherlands
Email: putten@liacs.nl

Lingjun Meng
Shandong University
Shandong, P.R.China
Email: mlj@sdu.edu.cn

Joost N. Kok
LIACS
Leiden University
Leiden, the Netherlands
Email: joost@liacs.nl

***Abstract*—In this paper we present an approach for benchmarking and profiling novel classification algorithms. We apply it to AIRS, an Artificial Immune System algorithm inspired by how the natural immune system recognizes and remembers intruders. We provide basic benchmarking results for AIRS, to our knowledge the first such test under standardised conditions. We also investigate how data set properties (data set size) relate to AIRS performance, and what other algorithms produce similar patterns over over- and underperformance on specific data sets. We present three methods for computing algorithm similarity that may be useful for profiling novel algorithms in general.**

I. Introduction

The introduction of new technologies generally follows a typical hype cycle, and novel data mining algorithms are no exception to this rule. Neural networks are a good example. Algorithms inspired by neural processing have been around since the forties (for instance [1]) but really gained traction in the eighties of the last century after publication of the PDP Handbooks [2]. The amount of neural network research exploded, and neural networks were pitched as a superior set of algorithms for classification, clustering and optimization, in some cases with no more justification than its biological origins.

After this period of excitement but also overinflated claims a more realistic approach was taken. Some researchers went the direction of using neural networks strictly for the purpose of neurobiological modeling, but the data mining and machine learning community started to ignore the biological roots and to evaluate and benchmark neural algorithms against other approaches using generally applicable measures such as accuracy. Whilst this may have resulted in the loss of some of the initial appeal, research interest and promise, it actually led to the incorporation of of neural networks into the standard toolkit of a much wider community.

So for the maturity and wider acceptance of a novel algorithm it is key that it is benchmarked against and compared with existing approaches, preferably by researchers who have not been involved in the development and evangelization of the particular novel algorithm. However, it should be noted that this only provides basic reassurance that the algorithm provides reasonably valid results. The No Free Lunch theorem loosely states that there is no algorithm that will consistently outperform all other algorithms on all problem domains [3]. So it is important to take the analysis further than basic accuracy benchmarking, and for instance investigate on what kind of problems and data it works well, or explore to what algorithms it is similar in its behavior. This will help the data miner to decide when best to apply these methods. We refer to this as algorithm profiling rather than basic benchmarking. Existing methods such as learning curves and bias variance analysis can be used, but there is also a lot of opportunity to develop new methods (see figure 1)

In this paper we provide an example approach for such an analysis. As the candidate novel algorithm we have chosen for AIRS, a so called Artificial Immune System or Immunocomputing algorithm, a prediction method inspired by the learning capabilities of the immune system ([4]). The analogy with neural networks is not a coincidence - we wanted to pick a field that is likely to be in a similar position as neural networks previously. Whilst our approach goes further than basic benchmarking ([5], [6]) we have chosen to keep it fairly straightforward and simple, so that the same approach can easily be used to benchmark, profile and characterize other novel algorithms for classification.

The paper is organized as follows. Section II provides an overview of natural immune systems, and section III outlines the AIRS algorithm. The basic benchmarking, data set properties and algorithm similarity experiment results are described in sections IV, V and VI respectively. We conclude this paper with section VII.

II. Natural Immune Systems

The natural immune system offers two lines of defense, the innate and adaptive immune system. The innate immune system consists of cells that can neutralize a predefined set of attackers, or antigens, without requiring previous exposure to them. The antigen can be an intruder or part of cells or molecules of the organism itself. In addition, higher animals like vertebrates possess an adaptive immune system that can learn to recognize, eliminate and remember specific new antigens.

An important role is played by lymphocytes, cells that recognize and destroy antigens. There are different types of lymphocytes, cells that recognize antigens directly (B-cells) or cells that recognize antigens that are bound to so called

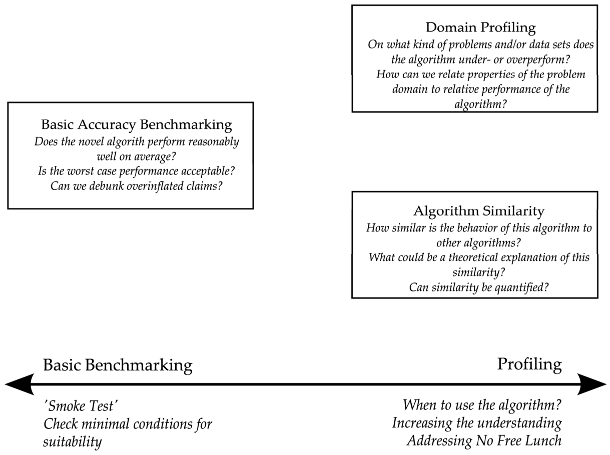

Fig. 1. Basic algorithm benchmarking versus algorithm profiling

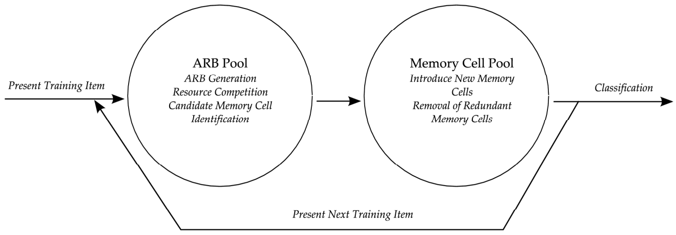

Fig. 2. An overview of the AIRS algorithm

presenter cells (T-cells). Each lymphocyte codes for a specific antigen, but there may be more possible types of antigens than there are specific lymphocytes.

This is solved by a form of natural selection. The bone marrow and thymus continuously produce lymphocytes and each of these cells can counteract a specific type of antigen. Now if for example a B-cell lymphocyte encounters an antigen it codes for, it will produce antibody molecules that neutralize the antigen and in addition a large number of cloned B-cells are produced that code for the same antigen (clonal expansion or clonal selection).

The immediate reaction of the innate and adaptive immune system cells is called the primary immune response. The immune system also keeps a record of past intrusions. A selection of the activated lymphocytes is turned into sleeper memory cells that can be activated again if a new intrusion occurs of the same antigen, resulting in a quicker response. This is called the secondary immune response [7].

III. AIRS

The Artificial Immune System algorithm (AIRS) can be applied to classification problems, which is a very common real world data mining task. Most other artificial immune system research concerns unsupervised learning and clustering. The only other attempt to use immune systems for supervised learning is the work of Carter [8]. The AIRS design refers to many immune system metaphors including resource competition, clonal selection, affinity maturation, memory cell retention, and so on.

A. History and Background

AIRS builds on the concept of resource limited clustering as introduced by Timmis and Neal and de Castro and von Zuben [9], [10].

According to the introductory paper, AIRS seems to perform well on various classification and machine learning problems [4]. Watkins claimed the performance of AIRS is comparable, and in some cases superior, to the performance of other highly-regarded supervised learning techniques for these benchmarks. Goodman, Boggess, and Watkins investigated the 'source of power for AIRS' and its performance on multiple-class problems. The authors compared the results of AIRS on several datasets including iris, ionosphere, diabetes, sonar, and Cleveland heart with the results from a large number of other classifiers from literature. Based on this comparison, the authors claim 'AIRS is competitive with the top five to eight classifiers out of 10-30 best classifiers on those problems', 'it was surprisingly successful as a general purpose classifier' and it 'performed consistently strong across large scope of classification problems' [11], [12].

B. The Algorithm

From a data mining point of view, AIRS is a cluster-based approach to classification. It first learns the structure of the input space by mapping a codebook of cluster centers to it and then performs a k-nearest neighbor search on the cluster centers for classification, just like k-means clustering for classification or Self Organizing Maps (SOMs, [13]). The attractive point of AIRS is its supervised procedure for discovering both the optimal number and position of the cluster centers.

In AIRS, there are two different populations, the Artificial Recognition Balls (ARBs - lymphocytes) and the memory cells, see figure 2. When a training antigen is presented , ARBs matching the antigen are activated and awarded more resources. ARBs with too few resources will be removed and new ARBs are created through mutation. This corresponds to the primary immune response in natural immune systems. On convergence a candidate memory cell is selected which is inserted into the memory cell pool if it contributes enough information. This corresponds to the secondary immune response. This process is repeated for all training instances - each training items can be seen as a separate 'attack'. Classification takes place by performing a nearest neighbor search on the memory cell population.

For a more formal description of the AIRS algorithm information see [4].

IV. Basic Accuracy Benchmarking

The goal of the benchmark experiments is to evaluate the predictive performance of AIRS in a real world application setting. We assume that our users are non data mining experts, e.g., business users, who may lack knowledge or time to fine-tune models, so we used default parameters without manual fine-tuning to create a level playing field. For this reason we also decided to use simple accuracy rather than more advanced measures such as area under the ROC. To ensure consistency, the experiments for all classifiers were carried out under exactly the same conditions, in contrast to some earlier published work on AIRS (see section III-A).

A. Approach

We selected data sets with varying number of attributes, instances and classes, from simple toy data sets to difficult real world learning problems, from the UCI Machine Learning and KDD repositories [14]. The TIC data sets are derived from the standard TIC training set by down sampling the negative outcomes to get an even distribution of the target. In addition, TIC5050S only contains the most relevant according attributes according to a subset feature selection method [15].

For the experiments, we selected some representative, well known classifiers as challengers. These classifiers include naive Bayes, logistic regression, decision tables, decision trees (C45/J48), conjunctive rules, bagged decision trees, multi layer perceptrons (MLP), 1-nearest neighbor (IB1) and 7-nearest neighbor (IB7). This set of algorithms was chosen because it covers most of the algorithms used in business data mining and contains a variety of classifier types and representations - instance based learning, clustering, regression type learning, trees and rules, and so on. Furthermore we added classifiers that provide lower bound benchmark figures: majority class (ZeroR) simply predicts the majority class and decision stumps are decision trees with one split only. For AIRS we chose the 1 and 7 nearest neighbor versions of the algorithm. We used the Java version of AIRS by Janna Hamaker and the WEKA toolbox for the benchmark algorithms [16] [17]. All experiments are carried out using 10-fold stratified cross validation, and we compute averages and standard deviations on the accuracies [5], [6].

B. Results

The results of the experiments can be found in Table I. With respect to the worst case classifiers we highlight some interesting patterns. Almost all classifiers outperform majority vote. The comparison with decision stumps is more striking. For example, for all data sets with the exception of the waveform data set the conjunctive rules classifier does not perform better than decision stumps. Other examples are the TIC data sets: none of the classifiers other than C45 on TICTRAIN5050s perform better than decision stumps. This demonstrates the power of a very simple decision rule in a real world black box modeling environment (see also [18]).

To get a better picture on the relative performance of AIRS we compare it to the average classifier performance (excluding decision stump and majority vote). AIRS-1 performs better than average on 3 of these 9 datasets. AIRS-7 performs better than average on 6 of these 9 datasets. This conflicts with the claims made in earlier studies that were cited in section

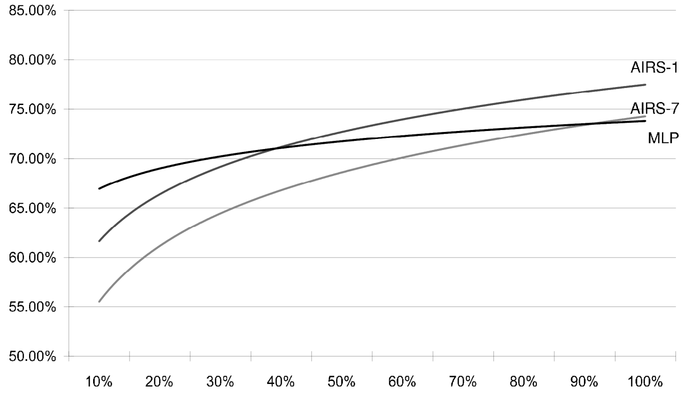

Fig. 3. Accuracy learning curves for AIRS-1 and AIRS-7 and MLP

2.2. We also made some comparisons to the IB-k algorithms, because these may be closest to a trained AIRS classifier. AIRS-1 improves on IB1 more often than the other way around; this is probably due to the fact that AIRS-1 provides some useful generalization. However IB7 performs better than AIRS-7 on all of the data sets. AIRS-7 performs better than AIRS-1on 7 out of 9 data sets. Using more clusters may give better results but not to the extent that IB7 can be beaten (basically as many cluster centers as data points).

That said, with the exception of AIRS-1 on German credit data, the AIRS algorithms produce at least around average results. This suggests that AIRS is a mature classifier that delivers reasonable results and that it can safely be used for real world classifications tasks.

V. Profiling: Influence of Data Set Properties

As mentioned the No Free Lunch theorem states that there is no classifier that outperforms all other classifiers across all problem domains. So it is interesting to investigate on what kind of data sets AIRS performs well, for example by relating data set properties to the performance of AIRS relative to other algorithms. We focus on a key property, the size of the data set.

A. Approach

We carried out a so called learning curve analysis on the diabetes data set. We created models using the same set of classifiers as in the previous section, using 25% hold out validation. These experiments were carried out on samples of various size, starting with 10% and with 10% increments. We then fitted logistic trend lines to the result series for each of the classifiers and eyeballed the patterns that emerged.

B. Results

Roughly three patterns of result series emerge. In figure 3 the trend lines of AIRS-1 and AIRS-7 are compared to MLP. The trend of AIRS-1 and AIRS-7 is quite similar - in this case AIRS-1 outperforms AIRS-7 regardless of data set size. The trend of the MLP curve is much flatter - i.e. it outperforms AIRS at lower data set size but AIRS starts to perform better at larger data set sizes. The opposite is true for IB1. The

	Sonar	Wisc. Breast Cancer	Wave form	Iris	Iono-sphere	Pima dia-betes	Ger-man	TIC5050 credit	TIC5050
Maj. Class	*53.4* ± 1.7	*65.5* ± 0.5	*33.8* ± 0.1	*33.3* ± 0.0	*64.1* ± 1.4	*65.1* ± 0.4	*70.0* ± 0.0	*49.7* ± 0.4	*49.7* ± 0.4
IB1	**86.6** ±7.0	95.3 ±3.4	73.6 ± 1.3	95.3 ± 5.5	86.3 ± 4.6	70.2 ± 4.7	72.0 ± 3.1	*55.9* ± 7.8	59.9 ± 5.1
IB7	80.8 ± 7.8	**96.6** ± 2.2	80.1 ± 1.1	**96.7** ± 3.5	85.2 ± 4.3	74.7 ± 5.0	74.0 ± 4.1	61.1 ± 3.2	65.4 ± 8.4
Dec. Stump	73.1 ± 8.3	*92.4* ± 4.4	*56.8* ± 1.5	*66.7* ± 0.0	*82.6* ± 4.8	71.9 ± 5.1	*70.0* ± 0.0	**68.5** ± 4.7	**68.5** ± 4.7
C45 (J48)	71.2 ± 7.1	94.6 ± 3.6	75.1 ± 1.3	**96.0** ± 5.6	**91.5** ± 3.3	73.8 ± 5.7	70.5 ± 3.6	**68.1** ± 5.5	**69.1** ± 4.4
Naive Bayes	*67.9* ± 9.3	96.0 ± 1.6	80.0 ± 2.0	**96.0** ± 4.7	*82.6* ± 5.5	**76.3** ± 5.5	**75.4** ± 4.3	62.8 ± 6.4	68.0 ± 3.3
Conj. Rules	*65.9* ± 8.7	*91.7* ± 4.5	*57.3* ± 1.3	*66.7* ± 0	*81.5* ± 5.4	*68.8* ± 8.67	*70.0* ± 0	**67.4** ± 3.7	68.3 ± 4.5
Bag-ging	77.4 ± 0.1	95.6 ± 3.1	**81.8** ± 1.4	94.0 ± 5.8	**90.9** ± 4.4	74.6 ± 3.6	**74.4** ± 4.9	59.9 ± 5.8	68.4 ± 4.1
Log. Regr.	73.1 ± 13.4	**96.6** ± 2.2	**86.6** ± 2.3	**96.0** ± 5.6	88.9 ± 4.9	**77.2** ± 4.6	**75.2** ± 3.4	62.7 ± 4.6	66.5 ± 3.4
MLP	**82.3** ± 10.7	95.3 ± 2.6	**83.6** ± 1.7	97.3 ± 3.4	**91.2** ± 2.8	**75.4** ± 4.7	71.6 ± 3.0	60.7 ± 4.3	65.4 ± 4.7
Dec. Table	74.5 ± 8.2	95.4 ± 2.7	73.8 ± 1.6	92.7 ± 5.8	89.5 ± 4.5	73.3 ± 3.6	72.2 ± 4.1	61.9 ± 4.5	**69.1** ± 5.7
AIRS1	**84.1** ± 7.4	96.1 ± 1.8	75.2 ± 1.7	**96.0** ± 5.6	86.9 ± 3.1	*67.4* ± 4.6	*68.0* ± 5.1	*56.8* ± 4.4	*55.0* ± 6.5
AIRS7	76.5 ± 8.4	**96.2** ± 1.9	79.6 ± 2 .2	95.3 ± 5.5	88.6 ± 5.0	73.6 ± 3.5	71.4 ± 3.1	57.8 ± 5.5	*59.1* ± 6.1

TABLE I

AVERAGE ACCURACY AND STANDARD DEVIATION ON ACCURACY (TEN FOLD) FOR AIRS AND A RANGE OF BENCHMARK ALGORITHMS. BEST RESULTS IN BOLDFACE, WORST RESULTS IN ITALICS.

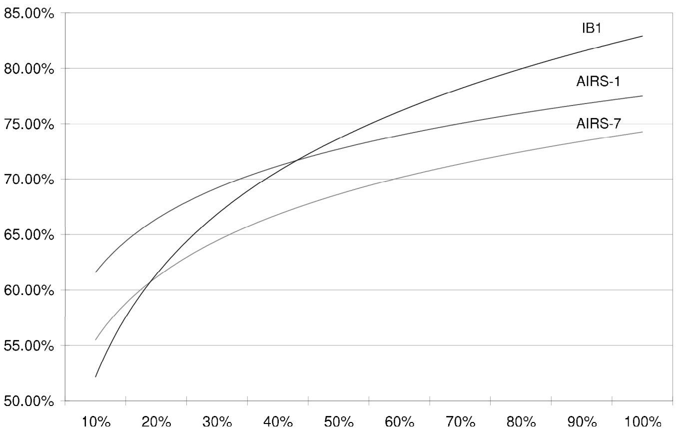

Fig. 4. Accuracy learning curves for AIRS-1 and AIRS-7 and IB1

slope of the IB1 learning curve is steeper than the AIRS learning curves, see figure 4. For the remaining classifiers, including IB7, the learning curve has a similar curve as the AIRS classifiers.

VI. PROFILING: COMPUTING ALGORITHM SIMILARITY

The experiments above provide an overview what the performance of the AIRS algorithm is and how AIRS performance may relate to data set size. Another typical question for a novel algorithm is how similar it is in its behavior compared to other algorithms. This is useful to know, as it will give a data miner an idea when to apply this technique.

Some insight can be derived by studying the theoretical properties of the algorithm. For instance, the AIRS learning process can be seen as a fairly complicated way to deliver a simple codebook of labeled cluster centers, so in theory it's behavior could be similar to nearest neighbor and or k-mean clustering style of classifiers.

The key question though is whether this behavior is validated in experiments, and whether other classifiers of very different origins may behave similarly as well, for reasons yet to be determined.

A. Approach

In our experiments we have used three different ways to measure algorithm similarity. We focussed on the accuracy of the algorithm, given that it is generally the key behavior of interest.

The basic benchmarking data provided the raw data for the analysis. To get a basic picture we simply calculated the correlation between classifiers on series of accuracies over the various data sets. A problem though with this approach is that to a large extent correlation can already be expected - on difficult problems accuracy will be low and vice versa.

So in our second method we decided to focus on performance relative to to other classifiers. For each classifier - data set combination we evaluated whether performance was better or worse than the average of all classifiers on that particular

Classifier	Correlation AIRS-1
IB1	0.99
IB7	0.98
MLP	0.97
AIRS-7	0.97
Bagging	0.94
Decisiontable	0.92
Logistic	0.89
J48	0.87
Naive Bayes	0.83
DecisionStump	0.47
Conjunctive Rules	0.46
ZeroR	-0.07

TABLE II
CORRELATION WITH AIRS-1 ACCURACY SERIES

Classifier	Correlation AIRS-7
MLP	0.99
IB7	0.99
Bagging	0.98
Logistic	0.97
AIRS-1	0.97
Decisiontable	0.96
IB1	0.95
Naive Bayes	0.93
J48	0.92
Conjunctive Rules	0.48
DecisionStump	0.45
ZeroR	-0.04

TABLE III
CORRELATION WITH AIRS-7 ACCURACY SERIES

	Agreement AIRS-1
IB1	8
AIRS-7	6
IB7	5
Conjunctive Rules	4
Bagging	4
MLP	4
J48	3
Decisiontable	3
Naive Bayes	3
Logistic	2

TABLE IV
FREQUENCY OF AGREEMENT (UNDER OR OVER PERFOMANCE COMPARED TO AVERAGE) BETWEEN AIRS-1 AND OTHER CLASSIFIERS

	Agreement AIRS-7
Bagging	7
MLP	7
IB7	6
AIRS-1	6
IB1	5
Logistic	5
J48	4
Decisiontable	4
Naive Bayes	4
Conjunctive Rules	0

TABLE V
FREQUENCY OF AGREEMENT (UNDER OR OVER PERFOMANCE COMPARED TO AVERAGE) BETWEEN AIRS-7 AND OTHER CLASSIFIERS

data set. We then counted how often classifiers agreed in terms of over or underperformance with the AIRS algorithms.

A drawback of this particular approach is that we lose how much better or worse a classifier was than average, in relative terms. So in our third approach we calculated the number of standard deviations a classifier under or over performed. To calculate similarity we then computed the correlations between these series of standard deviations.

B. Results

The correlation between the AIRS-1 accuracy series and the other algorithms can be seen in table II. As mentioned in section III-B the AIRS algorithm can be seen as a codebook learning procedure that automatically determines the optimal number of codes. Classification is done by simple nearest neighbor search on the codebook. As expected the nearest neighbor classifiers indeed have a high correlation, along with the AIRS-7 algorithm, and the IB1 algorithm indeed behaves more similar than the IB7 algorithm. An somewhat unexpected result is the high score for MLP. The AIRS-7 results (table III) show a consistent yet slightly more mixed picture with the IB1 and AIRS-1 algorithms scoring lower and MLP ranking as the first algorithm. This could have been due to the lower variance of AIRS-7 (bagging scores high, AIRS-1 scores higher than IB1).

The results of our second method can be found in tables IV and V. Note that in this method we only consider whether a method scores better or worse then average, and we count how often classifiers agree. For AIRS-1 we do see the expected behavior with respect to the AIRS-7 and the nearest neighbor algorithms, MLP now scores lower. For AIRS-7 we see a similar pattern, however nearest neighbor and AIRS are even less similar than bagging and MLP.

The results for the third method can be found in tables VI and VII. For this method the most consistent pattern as per the expectations emerge with high scores for AIRS, nearest neighbor and MLP. This method also seems to gives the widest range in similarity scores which makes it easier to discriminate across classifiers (obviously assuming the score itself is valid).

In table VIII we provide an overview of the results for the various methods. Overall it can be concluded that classifiers with theoretical similarities (AIRS-n, IB-n) indeed also behave similar, with an interesting similarity to the MLP algorithm as well.

VII. CONCLUSION

In this paper we have presented an approach to benchmarking and profiling a novel algorithm, in this case the AIRS Artificial Immune System algorithm. We are interested in immuno-computing because it is one of the newest directions in biologically inspired machine learning and focused on AIRS because it can be used for classification, which is one of the most common data mining tasks.

Classifier	Correlation with AIRS-1
IB1	0.77
AIRS-7	0.66
MLP	0.43
IB7	0.16
Bagging	-0.34
J48	-0.35
Decisiontable	-0.48
Logistic	-0.50
ZeroR	-0.52
DecisionStump	-0.54
Conjunctive Rules	-0.63
Naive Bayes	-0.66

TABLE VI
CORRELATION FOR RELATIVE UNDER OR OVER PERFORMANCE BETWEEN AIRS-1 AND OTHER CLASSIFIERS

Classifier	Correlation with AIRS-7
AIRS-1	0.66
MLP	0.52
IB1	0.48
IB7	0.23
Logistic	0.22
Bagging	0.07
Naive Bayes	-0.28
J48	-0.49
Decisiontable	-0.55
ZeroR	-0.56
DecisionStump	-0.85
Conjunctive Rules	-0.93

TABLE VII
CORRELATION FOR RELATIVE UNDER OR OVER PERFORMANCE BETWEEN AIRS-7 AND OTHER CLASSIFIERS

This was the first basic benchmark of AIRS that compared AIRS across a wide variety of data sets and algorithms, using a completely standardized experimental set up rather than referring to benchmark results from literature. In contrast to earlier claims, we find no evidence that AIRS consistently outperforms other algorithms. However, AIRS provides stable, near average results so it can safely be added to the data miners toolbox [5], [6].

In addition we have presented a methodology to further profile a novel algorithm. We have performed some selected learning curve experiments that showed a more or less standard curve for AIRS, steeper than MLP but flatter than nearest neighbor. We have explored a variety of methods for computing algorithm similarity that confirmed it behaved indeed similar to nearest neighbor based methods, as expected, but it also has shown to be similar to MLP in its behavior.

	Similar to AIRS-1	Similar to AIRS-7
Method 1	IB1, IB7, MLP, AIRS-7	MLP, IB7, Bagging, Logistic
Method 2	IB1, AIRS-7, IB7	MLP, Bagging, IB7, AIRS-1
Method 3	IB1, AIRS-7, MLP, IB7	AIRS-1, MLP, IB1,IB7
Summary	IB1, AIRS-7, IB7	MLP, IB7, AIRS-1

TABLE VIII
ALGORITHM SIMILARITY FOR THE COMPUTATION METHODS

Whilst even proper basic benchmarking is lacking for novel algorithms, as was also the case with AIRS, we propose that more attention is being paid to the area of algorithm profiling, both from the perspective of actually applying it to novel methods as well as an independent area for research, as there are not a lot of generally accepted methods for algorithm profiling yet.

ACKNOWLEDGMENT

We would like thank Janna Hamaker for providing the AIRS source code and Andrew Watkins for his assistance with understanding the details of the AIRS algorithm.

REFERENCES

[1] W. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943.
[2] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: explorations in the microstructure of cognition*. Cambridge, Mass.: MIT Press, 1986.
[3] D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," The Santa Fe Institute, Tech. Rep. SFI-TR-95-02-010, 1995.
[4] A. Watkins, J. Timmis, and L. Boggess, "Artificial immune recognition system (AIRS): An immune-inspired supervised learning algorithm," *Genetic Programming and Evolvable Machines*, vol. 5, no. 3, pp. 291–317, 2004.
[5] P. van der Putten and L. Meng, "Benchmarking artificial immune systems," in *BNAIC*, 2005, pp. 393–394.
[6] L. Meng, P. van der Putten, and H. Wang, "A comprehensive benchmark of the artificial immune recognition system (airs)," in *ADMA*, 2005, pp. 575–582.
[7] L. D. Castro and J. Timmis, *Artificial Immune Systems: a New Computational Intelligence Approach*. Springer Verlag, 2002.
[8] J. Carter, "The immune systems as a model for pattern recognition and classification," *Journal of the American Medical Informatics Association*, vol. 7, no. 1, pp. 28–41, 2000.
[9] L. D. Castro and F. von Zuben, "The clonal selection algorithm with engineering applications," in *Proceedings of Genetic and Evolutionary Computation*, D. Whitley, D. Goldberg, E. CantuPaz, L. Spector, I. Parmee, and H. Beyer, Eds. Morgan Kaufman, 2000, pp. 36–37.
[10] J. Timmis and M. Neal, "A resource limited artificial immune system. knowledge based systems," *Knowledge Based Systems*, vol. 14, no. 3/4, pp. 121–130, 2001.
[11] D. Goodman, L. Boggess, and A. Watkins, "Artificial immune system classification of multiple-class problems," in *Artificial Neural Networks in Engineering (ANNIE)*, 2002.
[12] ——, "An investigation into the source of power for AIRS, an artificial immune classification system," in *IJCNN-03*, 2003.
[13] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, pp. 59–69, 1982.
[14] C. Blake and C. Merz, "UCI repository of machine learning databases," 1998. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html
[15] P. van der Putten and M. van Someren, "Coil challenge 2000: The insurance company case," Leiden Institute of Advanced Computer Science, Universiteit van Leiden, Tech. Rep. 2000-09, 2000, http://www.liacs.nl/∼putten/library/cc2000.
[16] J. Hamaker and A. Watkins, *Artificial Immune Recognition System (AIRS), Java source code*, 2003.
[17] I. Witten and E. Frank, *Data Mining: Practical Machine LearningTools and Techniques with Java Implementations*. San Francisco: Morgan Kaufmann Publishers, 2000.
[18] R. Holte, "Very simple classification rules perform well on most commonly used datasets," *Machine Learning*, vol. 11, pp. 63–91, 1993.