ISLE - A NOVEL IMMUNE-SYSTEM INSPIRED RULE EXTRACTION ALGORITHM

D.T. Pham, A.J. Soroka

Manufacturing Engineering Centre, Cardiff University,
PO Box925,Cardiff UK

Abstract: This paper proposes a novel rule extraction algorithm whose extraction technique is inspired by the pattern recognition abilities of the immune system. This algorithm is based upon a form of immune network that facilitates the implementation of incremental rule extraction. The use of such a network also enables the algorithm to learn from training examples that do not have classes specified. The algorithm is tested against several example data sets and shows itself to be at least comparable to existing algorithms.

Keywords: Artificial Intelligence, learning algorithms, machine learning, knowledge-based systems, rules, rule-based systems

1 INTRODUCTION

This paper presents a novel algorithm for rule extraction that is inspired by the learning properties of the immune system.

The immune mechanism is a highly complex biological system for the identification and elimination of material that is foreign to the human body. When foreign bodies (antigens) invade the body they are identified by antibodies which bind with them, marking them for elimination. The agents involved in the elimination process include Lymphocyte cells that mediate the immune responses, Phagocytes which ingest these marked antigens and agents of the complement system - enzymes which attack the antigens [Mayer, 1973].

To be able to perform such tasks it is widely regarded that the immune system must possess the ability to learn about its surroundings, and recognise antigens encountered before [Farmer et al, 1986].

The proposed algorithm, ISLE (Immune-System Inspired Learning Algorithm), is based upon concepts related to the immune system, rather than being an exact model of it. A theory of how the immune system works is the Immune Network Theory presented in [Jerne, 1973]. This theory has been widely adopted and used to inspire work in the artificial intelligence domain [Hunt and Cooke, 1995; Hunt et al, 1995; Hunt and Cooke, 1996, Knight 2002, Knight 2003].

The most significant difference between the proposed algorithm and other immune-system-based learning programs is that only the learning stage is related to the immune system, as opposed to both learning and identification as in [Hunt and Cooke, 1995; Hunt and Cooke, 1996; Hunt et al, 1996; Timmis 2000]. ISLE opens the potential for more widespread exploitation of learning systems and techniques based upon the immune system because the algorithm is not tied to a platform-dependent solution for classification.

2 IMMUNE-SYSTEM-INSPIRED RULE EXTRACTION ALGORITHM

2.1 Antibody / Antigen Representation

The attributes within an example in a data set are represented by elements within the antibody (prototype or final rule) and antigen (training data) as shown in [Fig 1].

Antibody:

Leak	#1.1>9.1<	96.5	shade	*

Content:

class	range 1.1≤#≤9.1	number	string	wildcard

Fig. 1 - Example Antibody

Most element types are self-explanatory except for perhaps range and wildcard. Range enables the algorithm to deal with continuous-valued attributes, so in [Fig. 1] the antibody can match with the antigen element if its value is between 1.1 and 9.1. The Wildcard element enables that element of the antibody to bind with an antigen element of any type and value.

2.2 Immune Network

ISLE employs a form of Immune Network to store the antibodies and antigens presented to the system. The structure of the network (consisting of multiple layers) used is shown in [Fig. 2]. The first layer (antibody layer) - similar to the Recognising Set in [Jerne, 1973] - is occupied by what are termed active antibodies. They are used for matching with antigens and generating the final rule set. However, the network differs from that in [Jerne, 1973] in that it only considers direct antigen-antibody relationships as opposed to antigen-antibody-antibody interdependencies within an immune network [Jerne, 1973; Perelson, 1989]. As using a full immune network can prove to be computationally expensive [Hunt et al, 1996].

The other layers within the network consist of antibody-antigen pairings - the pairs used to form the antibody in the previous layer. Layers furthest away from the active antibodies are the oldest (new antigen-antibody pairs are appended to the layers at the head of the list).

The utilisation of an immune network facilitates the implementation of incremental learning features as such algorithms need some method by which they can store their state between different learning sessions. By providing a data structure that can be saved and reloaded, it becomes possible to use the algorithm in an incremental manner.

In the immune network illustrated in [Fig. 2] when an antigen is presented to the system the antibody with the best match binds. Provided that the classes of the antibody and antigen are the same and that the quality of the bind exceeds a given threshold. If no such antibody exists then the antigen is used to form an antibody. ISLE and its constituent sub-algorithms are described in the following sections.

2.3 Matching Algorithm

The matching algorithm performs the task of calculating the quality of the match between an antibody and an antigen. The algorithm uses different methods for calculating the match quality depending on the type of data being represented by the element.

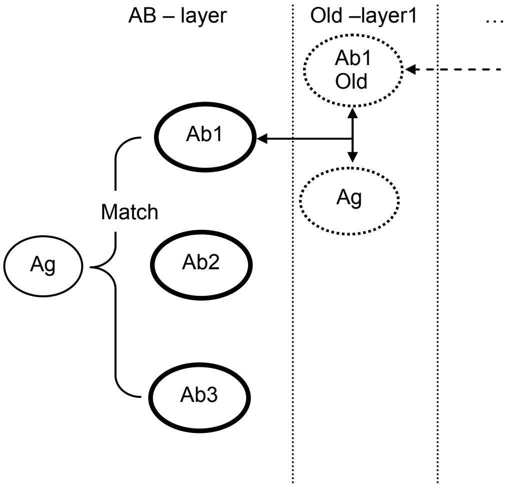

Fig. 2. Structure of Immune Network

The matching algorithm uses direct bit-to-bit matching, as opposed to the bit-shifting approaches discussed in [Farmer et al, 1986; Hunt and Cooke, 1995]. It was found that for bit-shifting approaches discrete data sets could be a problem in that the algorithm could incorrectly identify examples presented to it [Soroka, 2000].

As in some other systems, such as [Cooke and Hunt, 1995], the antibodies in ISLE may contain wildcard elements. In ISLE a match between a wildcard element and another element is considered to be of equal quality to a match between two identical elements. The method employed in [Cooke and Hunt, 1995] (wildcard match is of lower quality) could have the side effect of generating more rules. Antigens which bind 100% with an appropriate antibody containing wildcards in ISLE would bind instead with another antibody or possibly not at all.

The matching algorithm also takes into account the threshold so that, as more incorrect bindings occur, the threshold is raised and consequently the quality of match required for an antibody to bind with an antigen is increased.

2.4 Antibody Generation via Combination

This process generates a new antibody by combining an antigen and the antibody that binds with it. The process can be considered analogous to the primary response of the immune system which forms new antibodies in an attempt to deal with a previously unseen antigen. In ISLE if the nth elements of the antibody and antigen are identical then, in the new antibody, the element will remain the same. If the nth elements of the antibody and antigen do not match and if they are both strings, a wildcard operator is used to replace them. This has proved to be useful in [Cooke and Hunt, 1995] for the recognition of promoter sequences. If the two elements are different numbers then a range is generated. This removes the need for quantisation levels to be specified by the user.

This new antibody is then checked against the other antibodies within the system to ensure that there is no inconsistency, thus preventing the antibody from classifying with a score of 100% an antigen that belongs to a different class. If there is no inconsistency then the test antibody which was generated replaces the original antibody, and the latter along with the antigen is placed in the previous layer. This process will reduce, if not eliminate altogether, the possibility of misclassification.

Description of Process The generation process involves several stages: creation of a new antibody, testing of the new antibody to ensure it does not misclassify any of the antigens already presented, and placing the antibody into the network if suitable.

[Fig. 3] illustrates the process for the generation of a new antibody. If, as in the first elements of the antibody Ab and antigen Ag, the two strings do not match a wildcard is used in the resulting antibody. If, as in the second elements of the antibody and antigen the value of the antigen element is outside the limits of the range specified in the antibody element, then the range is expanded to encompass the value of the antigen element. The third elements illustrate that, when two numerical elements do not match, a range is generated which covers the two numbers. The fourth elements are two identical strings and so the element in the new antibody is the same string. The penultimate element in the antibody is a wildcard and therefore the value of the corresponding element in the new antibody remains a wildcard. The final elements are numbers of identical value and so the element in the new antibody takes the same value.

2.5 *Antibody Generation from Antigen*

Under certain circumstances antibodies cannot be generated using the process described in the previous section and have to be created directly from the antigen. As ISLE does not perform mutation, it cannot generate a new antibody from existing ones, as a real immune system would do. The need to generate new antibodies directly might arise for different reasons. The combination procedure sometimes cannot produce an antibody because the new antibody will cause misclassification. There are situations when no antibody matches the antigen sufficiently or no matching at all is achieved with the existing antibodies. The process of generating the new antibody is straightforward in that the antibody is a clone of the antigen. The new antibody is therefore able to classify that particular antigen. The antibody is then placed in the network.

Ab					
Val.A	#1.2>2.4<	3.1	X	*	1
Ag					
Val.B	2.5	4.1	X	Z	1
⇓	⇓	⇓	⇓	⇓	⇓
New Ab					
*	#1.2>2.5<	#3.1>4.1<	X	*	1

Fig. 3. - Creation of new Antibodies via Combination

However, if the antigen happens to have been for an example that has not had a class specified for it, then it is put back into the training data set, so that it can be reassessed when there are more antibodies in the system. This is only likely to happen when there are no antibodies in the system.

2.6 *Decomposition*

Antibody decomposing is a feature not present in a real immune system but it facilitates the generation of antibodies and to suppress those that misclassify.

This technique also assists the algorithm in overcoming the problems mentioned in [Cooke and Hunt, 1995] concerning wildcards. It allows the match with a wildcard element to be equivalent to a full match and yet prevents the generation of antibodies that consist entirely of wildcards.

The primary factor contributing to the requirement for decomposition with ISLE is as with [Cooke and Hunt, 1995] in that the antibody contained too many wildcards caused either by many matches with different antigens, or by one very weak match that resulted in a significant proportion of the elements becoming wildcards. For this reason, a threshold is used during matching, but it alone does not guarantee to prevent this situation from arising.

If an antibody has a 100% match with an antigen, but the antigen and antibody class are not identical, this necessitates that the antibody be decomposed. The algorithm will essentially "drill-down" through the layers of the immune network until an antibody that does not produce an inconsistent classification is retrieved. The antigens that were bound to the antibody are then appended to the list of antigens that have yet to be presented to the system. The antibody is then placed in the active layer.

The decomposition feature is also very useful in incremental learning as it allows an antibody that misclassifies an antigen to be removed and a new antibody to be generated in its place. Once this has been done, the antigens that were appended to the list are re-presented to the system for re-incorporation into the network.

Following the decomposition, the threshold is incremented using [Eqn. 1]. The equation increases the threshold quickly when the current threshold ($\text{threshold}_{\text{cu}}$) is small relative to the number of elements (numElement), thus helping to prevent antibodies that will misclassify from being generated.

$$threshold_{new}=threshold_{cur}+(0.1*numElement - threshold_{cur})$$

Eqn. 1. Threshold calculation

2.7 *Memory for Incremental Learning*

Various methods exist to perform incremental learning and therefore different approaches to the problem of memory - how to represent current knowledge so that it can be employed for future rule generation. Algorithms such as AQ15 [Michalski et al, 1986] use a "perfect memory", where all training examples together with the rules generated are saved. Other algorithms, for example, RULES-4 [Pham and Dimov, 1997], only a portion of the examples used to generate the rule set is saved.

Within the human immune system, approximately 5% of the least stimulated B-cells die off every day [Farmer et al, 1987; Hunt and Cooke, 1995]. This is the means by which the immune system controls the population of B-cells contained within the body. A system similar to this is therefore proposed for ISLE, in that 5% of the population of the immune network "dies" after rule set generation is completed (if the proportion is sufficiently large). The pairs to be removed are always the oldest. Once removed, the network can be saved to disk, to be reloaded if incremental learning is required for this rule set.

2.8 Overall Algorithm

As shown in [Fig 4] the memory of the previous network, if it exists, is re-loaded.. An antigen is then selected from the list of antigens (example is chosen from the data set). If there are no antibodies then an antibody is generated from this antigen. The algorithm runs until there are no further antigens. If antibodies are present in the system, then the antigen will be compared against all of the antibodies.

Once this has been performed, the algorithm checks if any of the best matches had a score of greater than zero. If not, then an antibody is generated from an antigen. As already mentioned this helps ensure that no antibodies consisting solely of wildcards can be generated. If the quality of the match is greater than zero, then the algorithm checks the number of antibodies that have effectively bound with the antigen to determine how many antibodies have a match score equal to the best match.

If only one antibody binds with an antigen, the algorithm checks that the classes of the antigen and antibody are not different. If identical an antibody is generated via the combination process. After an antibody has been created, the algorithm checks if there are any more antigens in the data set. If not, 5% of the older Ab-Ag pairs are removed, the trimmed network is saved and the process finishes.

If several antibodies have the same match score, they are arranged in a FIFO list. The first antibody in the list is taken and the classes of the antigen and antibody are compared. If identical or the antigen has no class the algorithm again attempts to generate a new antibody from the antigen and antibody. If unsuccessful it repeats the process until there are no antibodies in the list. When all attempts have failed, a new antibody is generated from the antigen.

If the antibody and antigen classes are different, then the antibody is decomposed as it has misclassified the antigen. The algorithm continues checking the matching antibodies against the antigen. When no valid antibodies can be found, the algorithm generates an antibody from the antigen.

Finally the rule set is extracted from the antibodies in the first or active layer.

3 RESULTS AND DISCUSSION

3.1 Complete Data

In order to test the performance of ISLE, it was compared against other existing learning algorithms. Using criteria such as the number of rules generated, accuracy of the rules, average number of conditions, and ability to classify unseen data, the comparison was carried out on three standard data sets.

Season Classification Problem The season classification problem involves a small data set that contains eleven examples. This data set has been used by several researchers including [Kottai and Bahill 1989; Pham and Aksoy, 1995; Tolun and Abu-Soud, 1998] to examine the performance of various learning algorithms. The algorithms tested were C5.0, ID3, ILA, and RULES-3 and 4. Results for ID3 and ILA taken from [Tolun and Abu-Soud, 1998].

As can be seen from the results in [Table 1] ISLE generates five rules which correctly classify the data set. This is identical to the results from the other algorithms tested with the exception of C5.0 which generates four rules plus one default rule (which can be considered as just another rule as it is the rule that fires if none of the other rules are true).

```
    START
    Load antibody (ab) if possible
A:  Get antigen (ag) from list
    Antibodies present? No → Jump to B:
    Yes → match ag with ab's
    Is bestmatch score > 0? Yes → Jump to D:
B:  No → Generate new ab using ag
C:  Anymore ag in list? Yes → Jump to A:
    No → Remove 5% of network
    Save network and rule set
    FINISH

D:  Number abs that have equal match with ag?
    >1 → Take 1st eligible ab then Jump to E:
    = 1 → Ag class same as ab class?
    No → Decompose then Jump to B:
    Yes → Generate new ab using ag & ab
    Has ab been created? No → Jump to B:
    Yes → Jump to C:

E:  Ag class same as ab?
    No → decompose then Jump to F:
    Yes → Generate new ab using ag & ab
    Has ab been created? No → Jump to F:
    Yes → Jump to C

F:  Any more abs? No → Jump to B:
    Yes → go to next ab then Jump to E:
```

Fig. 4. Overall algorithm

Table 1 Results for Season Classification Problem

Algorithm	Number of Rules	Accuracy of Rules
ISLE	5	100%
C5.0	4+1	100%
ID3	5	100%
ILA	5	100%
RULES-3	5	100%
RULES-4	5	100%

Flower Identification Problem In the flower data set [Kottai and Bahill, 1989; Aksoy, 1993], several classes are represented by only one example each, enabling assessment of the ability to deal with data sets with only a few examples of each class to learn from. The data set covers 25 classes, therefore the minimum possible number of 100% accurate rules is 25. [Table 2] shows some very interesting results. With respect to the number of rules, ISLE generates the same number as RULES-4 and one less than ID3 but four more than RULES-3. However, RULES-3 has an accuracy of 86.7% with the test data. The most striking result is in fact for C5.0, which generates one default rule and can only classify 6.9% of the examples.

Table 2 - Results for Flower Identification Problem

Algorithm	Number of Rules	Accuracy of Rules
ISLE	27	100%
C5.0	0+1	6.9%
ID3	28	100%
RULES-3	23	86.7%
RULES-4	27	100%

Iris Data Set The Iris data set [Fischer, 1936] is commonly used for the benchmarking of machine learning and pattern classification algorithms [Pham and Dimov, 1997] and has been used in the evaluation of Immune System inspired algorithms [Ji, 2004]. The data set consists of 150 examples with 50 examples for each of the three Iris classes, divided into 70 examples for training and 80 examples used for testing. The 70 training examples were randomly picked from the 150 examples contained within the data set.

The results for are shown in [Table 3]. In relation to the number of rules produced, ISLE compares favourably with the other algorithms tested, generating fewer rules than even C5.0, which is considered to be amongst one of the best algorithms currently available. However, the accuracy of the four rules formed by ISLE is less than the 100% achieved by both C5.0 and RULES-3, but at least equivalent to the accuracy of RULES-4.

When the accuracy of the rules generated by ISLE in classifying unseen data is examined it can be seen that an accuracy of 95% was achieved - comparable to the other algorithms lying approximately mid-way between the worst (C5.0) and the best (RULES-3).

Table 3 - Results for Iris Data Set

Algorithm	Number of Rules	Accuracy of Rules – Training Data	Accuracy of Rules – Unseen Data
ISLE	4	97.14%	95%
C5.0	5+1	100%	91.25%
RULES-3	14	100%	97.37%
RULES-4	10	97.14%	93.2%

3.2 Incomplete Data

To examine the ability of ISLE to deal with data sets where the classes of a proportion of the training examples given are unknown, a suitable data set had to be created. To do this the Iris data set was modified. The alterations were such that a proportion of the examples within a data set, from approx. 1% to approx. 95%, were randomly selected and had their class labels removed and replaced by an identifier for a missing class. The rule set is generated, and tested against the original data set which contains no missing class labels.

Most inductive learning algorithms are not able to handle the new data sets, a C5.0 based model was generated using the part of the data set where the classes are known. This model was then presented with the entire data set (containing both examples of known and unknown classes) and it then classifies these examples according to the rules previously generated - resulting in the final rule set.

In [Fig. 5] ISLE initially is nearly identical to that of C5.0, until the point where the 60% of the classes are unknown. When the proportion reaches such a level, the accuracy of ISLE is better than that of C5.0. Indeed, when 90% of the data sets are unclassified, the accuracy is still above 80%.

4 FURTHER WORK

Antigen matching could incorporate natural language processing techniques to allow linguistically sensitive matching between antigen and antibody based upon the similarity between words. Parallel processing could be used to enable the training data to be viewed in ‘one-shot’ i.e. all the examples are processed at the same time as opposed to a process where only one example is considered at one time.

5 CONCLUSIONS

ISLE has proved itself to be comparable to existing rule generation algorithms such as C5.0, ID3 and the RULES family in many respects, particularly the size and accuracy of the rule set created. ISLE has also demonstrated its ability to create rules when data sets incorporate a proportion of unlabelled instances.

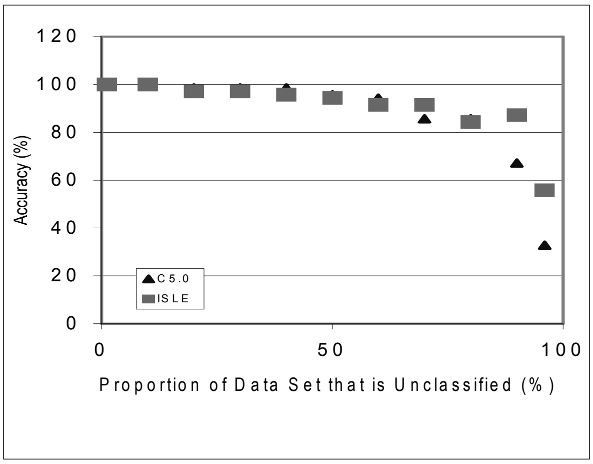

Fig. 5. Classification of unknown data

6 REFERENCES

Mayer M.M. (1973), The Complement System, *Scientific American,* 229, pp 54-66.

Farmer J.D., Packard N.H., Perelson A.S., (1986), The Immune System, Adaptation and Machine Learning, *Physica*, Vol. 22, Part D, pp 187-204

Jerne N.K. (1973), The Immune System, *Scientific American*, Vol. 229, Part 1, pp 52-60

Hunt J.E. and Cooke D.E. (1995), An Adaptive, Distributed Learning System based on the Immune System, *Intelligent Systems for the 21st Century, Proceedings International Conference on Systems, Man and Cybernetics*, Vol. 3, pp 2494-2499

Hunt J.E., Cooke D.E., Holstien H. (1995), Case Memory and Retrieval Based on the Immune System, IN *Welose M. and Aomodt A., (Eds.) Lecture Notes in Artificial Intelligence 1010: Case-based Reasoning Research and Development*, pp 205-216

Hunt J.E. and Cooke D.E. (1996), Learning Using an Artificial Immune System, *Journal of Network and Computer Applications: Special Issue on Intelligent Systems: Design and Application*, Vol. 19, pp 189-212.

Knight T. and Timmis J. (2002) A Multi-Layered Immune Inspired Approach to Data Mining. IN A Lotfi, J Garibaldi, and R John, editors, *Proceedings of the 4th International Conference on Recent Advances in Soft Computing*, pages 266-271, Nottingham, UK., December 2002.

Knight T. and Timmis J. (2003) A Multi-layerd Immune Inspired Machine Learning Algorithm. In, *Applications and Science in Soft Computing*, (A. Lotfi and M. Garibaldi (Eds)) pp 195-202. Springer.

Hunt J., King C., Cooke D. (1996), Immunising Against Fraud, *Proceedings IEE Colloquium on Knowledge Discovery and Data Mining*, London: UK

Timmis J., Neal M., Hunt J., (2000) An Artificial Immune System for Data Analysis, Biosystems, Vol.55 (1/3), pp143-150

Perelson A.S. (1989). Immune Network Theory. *Immunological Reviews*. No. 110 pp 5-36

Soroka A.J. (2000), An Agent Based System for the Acquisition and Management of Fault Knowledge for Intelligent Product Manuals. PhD Thesis, Cardiff University, United Kingdom

Cooke D.E. and Hunt J.E. (1995), Recognising Promoter Sequences Using an Artificial Immune System, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology,* pp 89-97.

Michalski R.S., Mozetic I., Hong J., Lavrac N., (1986), The Multi-purpose Incremental Learning Algorithm AQ15 and its Testing, Application in Three Medical Domains, *Proceedings 5th National Conference on Artificial Intelligence,* Philadelphia, PA: USA, pp 1041-1047.

Pham D.T. and Dimov S.S., (1997), An Algorithm for Incremental Inductive Learning, *Proceedings of the Institution of Mechanical Engineers Part B: Journal of Engineering Manufacture*, Vol. 211, pp 239-249.

Kottai R.M. and Bahill A.T., (1989), Expert Systems Made with Neural Networks. *International Journal of Neural Networks,* Vol. 1, No. 4, pp 211-226

Pham DT. and Aksoy M.S. (1995), RULES: a Simple Rule Extraction Algorithm. *Expert Systems with Applications,* Vol. 8, No. 1, pp. 59-65.

Tolun M.R., and Abu-Soud S.M. (1998), ILA: an Inductive Learning Algorithm for Rule Extraction, *Expert Systems with Applications*, Vol. 14, pp 361-370.

Aksoy M.S. (1993), New Algorithms for Machine Learning, PhD. Thesis, Cardiff University.

Fisher R.A., (1936), The use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, Vol. 7, pp 179-188, 466-475

Ji Z, Dasgupta D. (2004) Real-Valued Negative Selection using Variable-Sized Detectors. Proceedings of International Conference on Genetic and Evolutionary Computation (GECCO). Seattle, USA, June 26-30, 2004.