An immune-based approach to document classification

Jamie Twycross, Steve Cayzer
Information Infrastructure Laboratory
HP Laboratories Bristol
HPL-2002-292
October 30th , 2002*

E-mail: {jamie.twycross, steve.cayzer}@hp.com

artificial immune system, document classification, machine learning, concept learning, coevolution

The human immune system as a biological complex adaptive system has provided inspiration for a range of innovative problem solving techniques in areas such as computer security, knowledge management and information retrieval. In this paper the construction and performance of a novel immune-based learning algorithm is explored whose distributed, dynamic and adaptive nature offers many potential advantages over more traditional models. Through a process of cooperative coevolution a classifier is generated which consists of a set of detectors whose local dynamics enable the system as a whole to group positive and negative examples of a concept. The immune-based learning algorithm is first validated on a standard dataset. Then, combined with an HTML feature extractor, it is tested on a web-based document classification task and found to outperform traditional classification paradigms. Further applications in document based searching, content filtering, recommendation systems and user profile generation are also directly relevant to the work presented.

* Internal Accession Date Only
Approved for External Publication

An immune-based approach to document classification

Jamie Twycross and Steve Cayzer

Biologically Inspired Complex Adaptive System (BICAS) Group,
Hewlett-Packard Laboratories, Filton Road, Stoke Gifford, Bristol U.K.
{jamie.twycross, steve.cayzer}@hp.com

Abstract. The human immune system as a biological complex adaptive system has provided inspiration for a range of innovative problem solving techniques in areas such as computer security, knowledge management and information retrieval. In this paper the construction and performance of a novel immune-based learning algorithm is explored whose distributed, dynamic and adaptive nature offers many potential advantages over more traditional models. Through a process of cooperative coevolution a classifier is generated which consists of a set of detectors whose local dynamics enable the system as a whole to group positive and negative examples of a concept. The immune-based learning algorithm is first validated on a standard dataset. Then, combined with an HTML feature extractor, it is tested on a web-based document classification task and found to outperform traditional classification paradigms. Further applications in document based searching, content filtering, recommendation systems and user profile generation are also directly relevant to the work presented.

1 Introduction

This paper explores the novel application of a biologically-inspired learning algorithm based on the human immune system to the problem of document classification. Its overall aim is to produce a novel, working system built on an immune-based learning algorithm which is able to perform better than the currently available learning algorithms. In order to give substance to the claims made, we first validate our methodology using a standard dataset. The performance of our system is then compared to that of other methods in a systematic and rigorous manner. The motivation for this project is drawn from the current need for techniques that address a range of web-based information retrieval tasks. In this, the introductory section, we give a brief overview of the concepts and themes central to the work presented here. Section 2 contains a review of related work. Details of the methodology used can be found in Section 3, while Section 4 contains the evaluation results. In Section 5, we discuss these results and offer some concluding remarks.

1.1 The Document Classification Problem

Document classification is an important technique in the field of information retrieval. Work in this field has grown steadily since the 1940's and the ad-

vent of computers, and has been driven by the need for systems which are able to quickly and accurately access the increasingly large amounts of data being produced and stored on computers. With the birth of the Internet and World Wide Web this need has become more pressing than ever, but the problem of effective retrieval still remain largely unsolved [1]. Much of the work within the field of information retrieval belongs to three main areas: content analysis, information structure, and evaluation. Content analysis is concerned with transforming documents into a form suitable for processing; information structure with improving the effectiveness and efficiency of information retrieval systems through the exploitation of relationships between documents; and evaluation with the assessment of the performance of information retrieval systems. In terms of the concept learner presented here, these areas can be equated to deciding what to feed into the learner (feature extraction, discussed below), the learning algorithm (concept learning, discussed in 1.2), and how to assess how well it works (evaluation, discussed in Sect. 3.5) respectively.

1.2 Concept Learning for Document Classification

Concept learning can be framed as the problem of acquiring the definition of a general category given a sample of positive and negative training examples of the category [9]. In this paper, we consider the general category of 'web pages relevant to my current task', which forms the target concept for which we wish to acquire a definition. Our sets of positive and negative training examples are a set of web pages that we have already rated as 'useful' or 'not useful'. At the heart of the concept learner is a learning algorithm, whose job it is to take the training examples and create a classifier which is then able to look at further examples and decide if they fit into the learned concept or not. In this paper, we concentrate on a subclass of the general classification problem in which the feature vectors are Boolean, and where each feature vector can belong to one of two classes. In this case, the problem of concept learning can be summarised as one of inferring a Boolean-valued function from a set of training examples.

The immune system as a concept learner The learning algorithm we implement and study is based on aspects of the dynamics of the human immune system (HIS), part of whose function in its role as protector of the body can be broadly seen as the classification of proteins in the body into two classes: self – belonging to the body; and non-self – not belonging to the body and potentially harmful. This classification is carried out by a set of detectors called antibodies. The question then arises how we can learn such a set of discriminating detectors (or, equivalently, classification rules). In a series of papers Potter and De Jong [13,14,16] and Potter, De Jong and Grefenstette [15] explore the use of a cooperative coevolutionary algorithm for

function optimisation and for the evolution of artificial neural networks, sequential decision rules, and learning algorithms. Their approach involves the evolution of a number of non-interbreeding subspecies, individuals of which only represent partial solutions to the problem at hand, and are combined to form a complete solution. We adopt this approach for the generation of antibodies, where each species produces one antibody. The evolved antibodies are then combined to form a serum, which performs the classification required. In order to validate this approach, we replicate and extend the results of Potter & de Jong [16] on a standard dataset, before turning our attention to web page classification. Further details of our methodology can be found in Section 3.

2 Related Work

2.1 Document Classifier Systems

Pazzani et al. [12] describe one such system, instantiated as a software agent, which learns a profile of user's interests (or, equivalently, a classifier) from a collection of user-rated web pages, and uses this profile to identify other web pages that may be relevant to the user. The agent presents users searching for information with a list of links, called an index page, some of which may be relevant to the user's current interests, some not. Several of these links are visited by the user and rated as relevant or irrelevant and the agent is then instructed to learn the concepts of relevant and irrelevant on the basis of these user-rated web pages. After learning these concepts, the agent uses them to classify the links on the index page which the user has not visited, thereby aiding the user in their search.

To construct the concept, Pazzani et al. compare several different standard learning algorithms [11,12]. They examine a naive Bayesian classifier (NBC), nearest neighbour, decision trees and neural networks, and find that the NBC generally perform best. They also investigate the role of feature selection in the predictive accuracy of the classifiers, and find that appropriate feature extraction algorithms significantly reduces classification error. They go on to implement the naive Bayesian classifier in a system, Syskill and Webert, which automatically filters search results for users.

We reimplement Pazzani's naive Bayesian classifier and in turn compare and contrast its performance with that of the immune-based concept learner.

A similar system, NewsDude [2], also developed by Pazzani and Billsus, combines a NBC with a nearest neighbour classification algorithm and is used to recommend news articles. Two user profiles are used in this system, one representing the long-term interests of a user and the other the user's short-term interests created from recently read articles. In this way the recommendation of many similar articles can be avoided. Long-term adaptive behaviour can also be changed through techniques such as reinforcement learning [23].

Many other concept learning systems exist, most employing some form of inductive learning algorithm which arrives at hypotheses by considering specific examples. Many of these algorithms are surveyed in [7,18], and we compare several with our immune-based classifier in Section 4.

2.2 Immune-based classifier systems

The learning algorithm used by the HIS is generally (but not exclusively - see, for example, [8]) thought to be based around some kind of 'negative selection', where detectors are screened against self and thus we generate a set of detectors that cover non-self space [5]. When implemented in artificial immune systems (AIS), a number of heuristic improvements can be made to the detector generation process (see for example [21]). One criticism that can be made of canonical negative selection is that it makes use of information from only one class (this, of course, is an advantage when no other feedback is available). However, when combined with clonal selection learning, the AIS can refine its classification ability. Indeed, systems based on AIS learning have been shown to be effective supervised classification algorithms [20]

In our paper, we use a coevolutionary approach for detector generation. This idea, introduced by Potter & De Jong [16], has been proven to be of use in a variety of settings. It involves the evolution of a number of non-interbreeding subspecies, individuals of which represent partial solutions to the problem at hand, and are combined to form a complete solution. Sofge et al. [17] extend this approach in an attempt to decrease the degree of epistasis, which can sometimes occur in cooperative coevolutionary approaches. Their system involves the 'blending' of the usually distinct species of individuals as evolution proceeds, which they find helps the population to escape local optima. Neri [10] also investigates the incorporation of cooperative coevolution into three learning algorithms and shows that such algorithms are able to produce efficient concept descriptions. Concept learners have also been created using a variety of other evolutionary techniques, a survey of which can be found in [19].

3 Methodology

3.1 Immune-based Classifier

The immune-based classifier is based on one described by Potter and De Jong [16], and is composed of a set of detectors, each of which is instantiated as a ternary schema of the same length as the feature vectors it will classify. Associated with each detector is a real-valued threshold, which indicates the percentage of matching bits between schema and feature vector necessary before a match is said to have occurred. The strength of the match between detector and feature vector is the percentage of matching bits in the schema

and feature vector, ignoring any positions where the schema contains a #. For example, a detector '01#1##11' will match a feature vector '11100101' in 2 out of 5 non-# bits, so the binding strength between the detector and feature vector is $2/5 = 0.4$. The calculated binding strength must be greater than the threshold of the detector to consider a match to have occurred. Detectors can be of one of two types, Type 0 or Type 1, with a Type 0 detector, as in the human immune system, classifying any feature vector it matches as nonself, while a Type 1 detector contrarily classifying matching feature vectors as self.

3.2 Naive Bayes Classifier

The naive Bayesian classifier (NBC) is a probabilistic method of classification, which calculates the probabilities of a particular feature vector belonging to each possible class and then classifies the feature vector as belonging to the class for which this probability is highest. Formally, if $\mathbf{a} = [a_1, a_2, \cdots, a_n]$ is a feature vector made up of n features, a_i, and $V = \{v_1, v_2, \cdots, v_m\}$ is a set of m classes, then the class $v_{NB} \in V$ that the NBC classifies the example $\mathbf{a}$ as belonging to is given by:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_{i=1}^{n} P(a_i|v_j), \ \forall \ v_j \in V$$

3.3 Co-Evolutionary algorithm

The scheme used to evolve an immune-based concept learner is based on a coevolutionary approach described by Potter and De Jong [16]. The cooperative coevolutionary algorithm consists of a number of non-interbreeding species of detectors, whose encoding will be described shortly, and initially starts with one randomly initialised species whose fitness is evaluated as described below. The initialisation of species is controlled by two parameters: a generality bias parameter and a type bias parameter, both in the range [0, 1]. The generality bias parameter represents the probability that any position in a newly initialised detector contains a #, as opposed to a 0 or 1. The type bias parameter is the probably that a detector will be of Type 1. At each generation, a trial population composed of the fittest detector in each species is created and the fitness of this trial population evaluated. The fitness of all individuals in a species is then evaluated, as described below. Next, child species are created by selecting two parents from the same species using fitness-proportionate selection with balanced linear scaling, which are then recombined using uniform crossover, and mutated by bit flipping to create a child detector, which forms part of the child species for the species the parents were selected from. This process continues until the child and parent species are the same size. The fitness of each individual in the new species is then evaluated. If the fitness of the trial population fails to increase

above a certain stagnation threshold over several consecutive generations, a new species is added and any species not contributing to the fitness of the trial population are removed. The parameters used in the experiments were as follows: Species size=100, crossover rate=0.6, mutation rate=2/(*genome length*), stagnation threshold=0.001, stagnation generations=2, generality and type biases both=0.5.

Encoding Detectors are encoded as proposed by Potter & De Jong [16]. This scheme employs binary genomes, each containing 4 genes. The first gene, the threshold gene, encodes the (8 bit) value for the detector's threshold. A Gray coding was used for this gene in order to reduce the probability of small changes in the genotype producing disproportionately large changes in the phenotype. The threshold value is calculated by converting the gene to base 10 and then dividing this value by 255 to get a real number in the range [0, 1]. The second and third genes, the pattern and mask genes, are combined to form the detector's schema. Each of these genes has the same number of bits as the number of bits in the feature vectors the immune-based classifier is designed to operate on. The mask gene is overlaid onto the pattern gene and any positions at which the mask gene is 1 changes the corresponding bit in the pattern gene to a #. A value of 0 in the mask gene leaves the corresponding bit of the pattern gene unchanged. In this way the schema is formed by copying the pattern gene, modified by the mask gene. The fourth gene stores the detector's type. The overall arrangement is shown in Fig. 1 for a detector recognizing 8-bit patterns.

Genome

threshold	**pattern**	**mask**	**type**
00110111	10100101	01101110	1

⇓

Detector

threshold	**schema**	**type**
0.215686	1##0###1	1

Fig. 1. Detector encoding scheme

We ran several experiments with alternative encodings without finding any significant performance improvements.

Fitness evaluation The fitness of the trial population, composed of the best individual from each species, is calculated by presenting it with each training vector in the training set in turn. The detector in the trial population which

matches the current training vector with the greatest binding strength is then found, and if this strength is greater than the detector's threshold, the detector is said to have matched the training vector, and assigns it to Class 1 (or Class 0 if the detector is Type 1), otherwise if no match occurs the training vector is assigned to Class 0. The assigned class is then compared with the actual class of the training vector, and if equal the trial population is said to have classified the training vector correctly. The number of correct classifications made by the trial population gives the predictive accuracy of the trial population on the training set.

3.4 Data sets

Two sets of test data were used in our experiments, both taken from the UCI Repository of Machine Learning Databases [4]. The first data set, the 1984 United States Congressional Voting Records , was used to validate our algorithm. This data set contains the voting records for 267 Republican and 168 Democrat members of the U.S. House of Representatives. Each record holds the vote cast by the member on 16 different issues, and the original records have been simplified to record this vote as yea (01), nay (10) or abstain (00). Each member's voting record is represented as a Boolean feature vector, with each consecutive pair of bits encoding a vote for a particular issue. Using this encoding, it is possible to generate antibodies that generalize over votes (eg '0#' matches both 'abstain' and 'yea'). Associated with each record is the class the record belonged to: 0 for Democrat, 1 for Republican.

The second data set used was the Syskill and Webert Web Page Ratings [70]. These data consist of 4 sets: Bands, BioMedical, Goats, and Sheep; each containing HTML pages related to a particular topic (the encoding of the pages is described in the next section). A user rated each page in a set as not interesting or interesting, which allowed a page to be assigned to one of two classes: Class 0 (cold) and Class 1 (hot) respectively. The task in this problem is to make predictions about whether examples from an unseen set of web pages would be interesting or not from the information contained within a training set of ranked pages. Table 3.4 provides a summary of the data sets.

Table 1. HTML Documents: Data set summary

data set	total examples	number of positives
Bands	61	15
BioMedical	131	32
Goats	70	32
Sheep	65	14

Feature Extraction In applying learning algorithms to the classification of text and HTML, the documents must be presented in the form of feature vectors. Following Pazzani et al. [12], a feature extraction algorithm was used to convert a raw HTML document into a Boolean feature vector. Each bit in the feature vector represents the absence or presence (at least once) of some associated feature, in this case a word, in the document. The task of the feature extraction algorithm is to decide from which words to compose the feature vector, and this is done using an information-based approach to extract the most informative words from a collection of documents.

Initially, the feature extraction algorithm takes the complete set of pages, S, and creates a list of all the words, W, contained in the pages. If a word, considered as a sequence of upper of lower case letters [a-zA-Z] separated by nonalphabetic characters, occurs more than once on the same page or across several pages, it is only represented once on this list. All words were converted to upper case and any words occurring on a list of frequently used words (Table 3.4) were removed. The *expected information gain*, $E(w, S)$, that the presence or absence word $w \in W$ gives towards the classification of S is:

$$E(w, S) = I(S) - [P(w = \text{pres})I(S_{w=\text{pres}}) + P(w = \text{abs})I(S_{w=\text{abs}})]$$

with $P(w = \text{pres})$ is the probability a word is present at least once on any page, $S_{w=\text{pres}}$ the set of pages containing the word w, and,

$$I(S) = \sum_{C \in \{\text{hot,cold}\}} -P(S_C) \log_2[P(S_C)]$$

where S_C is the set of pages belonging to class C, and $P(S_C)$ is the probability of a page belonging to that class.

To create n features the extraction algorithm uses the n words with the highest values of $E(w, S)$. Each HTML document is then converted to a Boolean feature vector by assigning a 1 to the appropriate feature if the document contains the word at least once, and a 0 if the document does not contain the word.

3.5 Evaluation

One of the most commonly used measures of classifier performance is that of predictive accuracy; the probability that the classifier will classify any randomly chosen example correctly. We used different accuracy measures for each data set.

For the voting data set, 10-fold crossvalidation could be used. This involves randomly dividing the complete data set into 10 equally sized disjoint sets, and then using 1 subset as a test set and the other 9 as a training set. The training set is used by the learning algorithm to create a classifier,

Table 2. Frequent words removed from the word list

AND	HREF	THE	IMG	SRC	HAVE
FOR	FONT	COM	ALIGN	ALT	COMMENTS
SIZE	INDEX	HTM	TITLE	GOPHER	WHO
ORG	NAME	THIS	WEB	YOU	PLEASE
WWW	HOME	ABOUT	INTERNET	WIDTH	ALSO
PAGE	FTP	BODY	ARE	LIST	EDU
HTML	NET	HEIGHT	LINKS	NEWS	WHAT
FROM	HEAD	STRONG	WELCOME	WITH	MORE
TOP	MAILTO	YOUR	GIFS	BOTTOM	OUR
MAIL	CGI	THAT	BIN	ALL	WILL
CENTER	WUSTL	GDB	GOV	OTHER	IBC
ANY	HAS	NOT	TOC	GNN	ADDRESS
HTTP	GIF	WIC	SERVER	HERE	CAN
	AVAILABLE			INFORMATION	

whose sample predictive accuracy is then calculated using the test set. This process is repeated for each of the 10 subsets, the mean sample predictive accuracy of these 10 trials forming an unbiased estimate of the true predictive accuracy. Averaging the 10-fold crossvalidation process over 5 trials further refined these results. A randomly constructed crossvalidation set was used in each trial and the trials were paired, meaning the same training and test sets were used to train and test the two classifiers on each iteration.

The web page rating data sets range in size from between 61 to 131 samples, making techniques such as 10-fold crossvalidation an unreliable means of estimating the predictive accuracy due to the relatively small size of the test set produced by this method [9]. In such cases alternative methods need to be employed, one of which, and the one used here, is to create a training set by randomly selecting n samples from the original data set without replacement. The remaining unselected samples then become the test set. This method is advantageous for our purposes in that it can be used to assess the performance of classifiers over a range of training set sizes, giving a good indication of number of pages a user would have to rate in order to get reliable results from the classifier. After the training and test sets were created, the 128 most informative words were used to transform the training and test sets into Boolean feature vectors as previously explained. This process was repeated 30 times for each training set size, and the mean predictive accuracy of the resulting classifiers measured.

4 Results

All the code was written on a 1.4GHz Athlon Linux box, originally in C and then in C++, compiled using g++ v2.95.3 with level 2 optimisation, and released under the GNU General Public License. The experiments were carried

out on this Linux box and on four 1.8GHz Pentium 4's also running Linux. For the voting data set, a typical 100 generation run with 400 training examples took around 40 seconds. For the document classification problem, the immune-based classifier took around 6 seconds to train over 100 generations on a training set of size 20 with 128 features.

4.1 Classifier validation (voting data)

In order to validate our classifier, we compared the predictive accuracy of the immune-based classifier and NBC on the voting data set. The results of these experiments are shown in Fig. 2, which is a density plot of the distribution of predictive accuracies of the classifiers produced in the crossvalidation trials. Density plots can be thought of as histograms with a large number of bins, producing a smoother representation of the distribution of results over a number of trials.

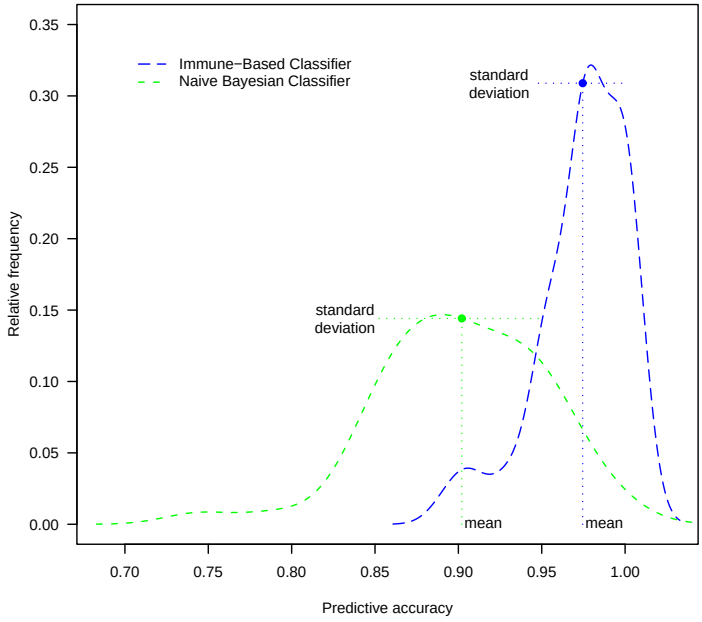

Fig. 2. 10-fold crossvalidation (voting data set). Predictive accuracy is represented along the x-axis, and the relative frequency with which a classifier with this predictive accuracy was observed during the experiments along the y-axis.

For classifiers produced by the naive Bayesian algorithm, Fig. 2 shows a right-skewed unimodal distribution with a single low, spread peak (standard deviation: 0.049) almost symmetrical about its mean predictive accuracy of 0.901. In contrast the immune system concept learner produced a less symmetric distribution with a higher mean of 0.974, rising fairly steeply and then dropping off even more steeply, giving a tighter distribution of values (standard deviation: 0.026) than the NBC. The immune-based learning algorithm thus produced classifiers which were both significantly (Wilcoxon rank sum test) more accurate and more likely to be closer to the true predictive accuracy than that of an NBC.

Table 3. Comparison of classifier performance (voting data set). The classifiers implemented in this paper are shown in **bold**.

Algorithm	predictive accuracy	standard deviation	95% confidence interval	error rate
Immune-based	0.974	0.026	0.057	0.026
Immune-based [16]	0.964		0.018	0.036
QUEST [7]	0.963			0.037
AQ15 [16]	0.956		0.023	0.044
POLYCLASS [7]	0.948			0.052
Fuzzy Classifier [4]	0.947	0.316		0.053
naive Bayesian	0.901	0.049	0.088	0.099

These results can also be compared to others reported in the literature for a number of different classifiers and summarised in Table 3. These results were calculated on the voting problem using the same 10-fold crossvalidation testing regime as the one employed here, with the exception of Lim et al. [7], who, instead of reporting predictive accuracy, reported on the rate of misclassification for the algorithms they tested. Other statistics are also given where provided by the original paper. While data to perform statistical tests were not available, the difference in the predictive accuracy suggests that the immune-based algorithm outperforms all of these algorithms.

4.2 Classification of HTML documents

The validation exercise performed above suggests that the immune-based classifier might be suitable for use on the Syskill & Webert data set. Figure 3 shows the predictive accuracy of the immune-based and NBC learning algorithms trained using a number of different training set sizes(as explained in Sect. 3.5) for each of the four data sets.

The first thing that can be noted from the graphs is the somewhat lower predictive accuracy of both classifiers on this classification problem compared

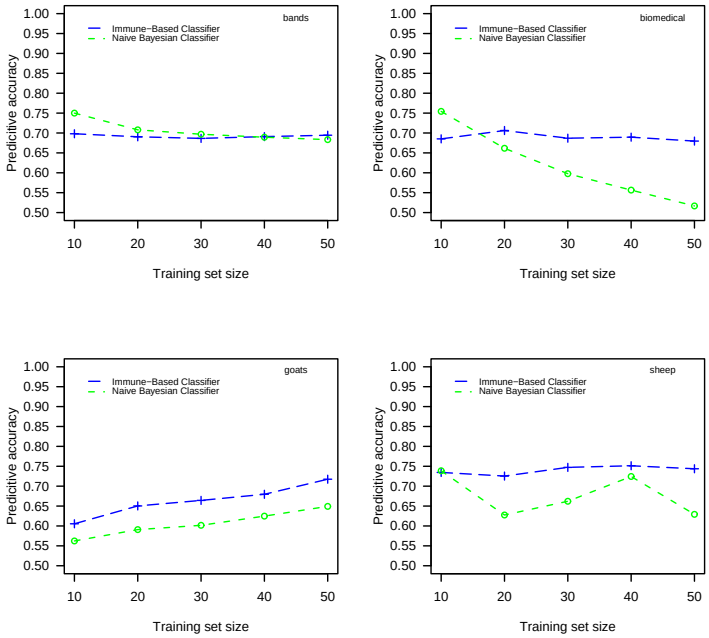

Fig. 3. Classifier performance (document classification). Predictive accuracy is plotted against training set size for each of the four datasets used.

with that of the voting problem of the previous section. This is not suprising as text classification problems are generally considered to be relatively hard classification problems [9], and in an informal survey of the literature on text classification we found the best classifiers to be achieving predictive accuracies of around 0.70 whatever the text classification problem. In this context, the performance of both classifiers is more than reasonable on the four problems. The results presented for the NBC are also similar to those of the NBC-based system of Pazzani et al. [12], offering an indication that the performance levels are due to the nature of the problem and not a result of implementation problems. This said, there are marked differences in the predictive accuracies of the NBC and immune-based classifiers on the four problems, with the immune-based algorithm at first sight appearing to generally perform better on all data sets, except the bands set, than the NBC. As before, a Wilcoxon rank sum significance test was performed on the results and this confirmed that there was a statistically significant difference between the predictive accuracies of the two algorithms, except on the bands data set. From this we can conclude that the immune-based classifier in general consistently produces better classifiers than those produced by the NBC algorithm.

Also interesting to note is the relatively constant performance of the immune-based classifiers over a range of training set sizes. Increasing the size of the training set seemed to produced little increase in classifier performance for this algorithm, while for the NBC there was a much greater fluctuation in classifier performance on an increase in training set size. This constancy of performance is particularly useful on problems such as this one because classifiers which perform well on a small training set size would be advantageous as users would be able to rate less pages but still obtain accurate predictions.

Summaries of these results, including standard deviations and confidence intervals are given below in Table 4.

Table 4. Performance of immune-based and NBC classifiers on the four data sets, for training set sizes from 10 to 50 documents. Table entries are in the form: mean predictive accuracy (standard deviation). **Bold** figures mark where the performance of the NBC (or immune-based classifier) is significantly better (Student T-test)

size	classifier	Bands	Biomedical	Goats	Sheep
10	Immune	0.698(0.056)	0.685(0.080)	**0.605(0.065)**	0.734(0.070)
	NBC	**0.750(0.000)**	**0.754(0.002)**	0.562(0.055)	0.738(0.134)
20	Immune	0.690(0.047)	**0.706(0.047)**	**0.650(0.083)**	**0.725(0.062)**
	NBC	0.707(0.067)	0.661(0.095)	0.590(0.061)	0.627(0.142)
30	Immune	0.686(0.074)	**0.686(0.056)**	**0.664(0.054)**	**0.747(0.049)**
	NBC	0.696(0.084)	0.597(0.137)	0.601(0.066)	0.662(0.097)
40	Immune	0.690(0.081)	**0.689(0.048)**	**0.679(0.087)**	0.751(0.067)
	NBC	0.689(0.056)	0.556(0.155)	0.624(0.066)	0.724(0.104)
50	Immune	0.694(0.116)	**0.679(0.070)**	**0.717(0.095)**	**0.743(0.106)**
	NBC	0.683(0.105)	0.516(0.137)	0.649(0.073)	0.629(0.153)

5 Discussion

While the AIS classifier outperforms all the learning algorithms it has been compared against in many cases, there are marked differences in the levels of predictive accuracies it achieves on the voting and document classification problems. One factor which contributes to these differences is the variation in the size of the training sets available for each problem. In terms of available data, the voting problem represents a fairly data-rich classification problem, with 435 samples available, whereas the number of available samples for the document classification problem, between 61 and 131, makes it a relatively data-sparse problem. This leads to less samples being available for classifier training, and so it is expected that performance will generally be lower on the data-sparse problem. A second factor contributing to the difference in performance across the two problems is the feature extraction algorithm. In this paper we use a statistical feature extraction algorithm which extracts

features based on their expected information gain and, while necessary to provide a principled comparison with the NBC-based system of Pazzani et al. [12], such a method is far from ideal in the context of HTML document classification. The coarse-grained document representation produced by our feature extractor introduces a fair amount of noise into the system, making classification generally that much harder than in the voting problem. Using a more fine-grained feature extractor, such as those described by Cohen [3] or Yang et al. [22], which also exploits meta-features of HTML documents such as hyperlinks and tags, would potentially increase classifier performance on the document classification problem and close the gap in performance difference.

In practical terms, choice of a classifier not only depends upon the predictive accuracies it is able to achieve, but also on the amount of time taken in training the classifier. No matter how good the results achieved, users would often be unwilling to wait for more than a few seconds for these results, and definitely not, for example, the several days or even months taken by some of the algorithms reported by Lim et al. [7]. While our immune based classifier is obviously more computationally expensive than the NBC, for all problems examined here training times were measured in seconds. Thus as well as achieving better predictive accuracies than that of many other classifiers, our immune-based classifier is able to do so in a time which allows it to be applied to real-world problems.

In summary, we have produced a novel, working system built on an immune-based learning algorithm, which is able to perform better than the currently available learning algorithms.

5.1 Future work

Future work could include dynamic classification tasks. Take for example a system in which users have a collection of documents from which the concepts of 'related' and 'unrelated' are learnt. Over time, users may add or remove documents to and from this collection. Instead of relearning the concepts from scratch each time a document is added, as is necessary with NBC, it would be interesting to see if the AIS concept learner was able to produce accurate hypotheses starting from the previously learned concept, and so potentially offer savings in the amount of training time necessary. Gaspar and Collard [6] study the performance of an immune-based system on a time-dependent optimisation problem and find that it performs well against a standard genetic algorithm, a performance which they attribute to the central role of diversity within the adaptive dynamics of their system, giving a further indication that immune-based systems may also be advantageous in dynamic classification problems.

Another possibility is to change the form and general properties of the antibodies produced by the AIS classifier; for example the encoding and matching algorithm. The possibility of using a more fine-grained feature extraction

algorithm has already been mentioned. In terms of the mechanisms at work in the human immune system those of the artificial immune system described in this paper are, to say the least, simplistic, and present a very crude analogy to their biological counterparts. Nevertheless, even from such humble an analogy it has been shown that a powerful concept learning system can be created. Perhaps with increased fidelity to its biological counterpart, for example through processes akin to affinity maturation and clonal selection, further increases in performance can be gained.

6 Acknowledgements

Computing facilities were provided by the HP Labs Bristol BICAS Research Group. We'd like to thank members of that group for valuable comments on the manuscript.

For further information on BICAS see http://www.hpl.hp.com/research/bicas

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press, New York, 1999.
2. D. Billsus and M. Pazzani. A hybrid user model for news story classification. In J. Kay, editor, *Proc. of the Seventh Int. Conf. on User Modeling*, pages 99–108. Springer-Verlag, Banff, Canada, 1999.
3. W. W. Cohen. Automatically extracting features for concept learning from the web. In P. Langley, editor, *Proc. of the Seventeenth Int. Conf. on Machine Learning (ICML-2000)*, pages 159–166. Morgan-Kaufmann, San Francisco, CA, 2000.
4. D. Dasgupta and F. A. González. Evolving complex fuzzy classifier rules using a linear genetic representation. In L. Spector, D. Whitley, D. Goldberg, E. Cantu-Paz, I. Parmee, and H. Beyer, editors, *Proc. of the Int. Conf. on Genetic and Evolutionary Computation (GECCO-2001)*, pages 299–305. Morgan-Kaufmann, San Francisco, CA, 2001.
5. S. Forrest, A. Perelson, L.Allen, and R. Cherukuri. Self nonself discrimination in a computer. In *IEEE Symposium on Research in Security and Privacy*, pages 202–212, Oakland, CA, May 16-18 1994.
6. A. Gaspar and P. Collard. From GAs to artificial immune systems: improving adaptation in time dependent optimization. In P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, editors, *Proc. of the Congress on Evolutionary Computation*, volume 3, pages 1859–1866. IEEE Press, Mayflower Hotel, Washington D.C., 1999.
7. T. Lim, W. Loh, and Y. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40(3):203–228, 2000.
8. P. Matzinger. The danger model: A renewed sense of self. *Science*, 296:301–305, 2002.
9. T. M. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

10. F. Neri. A study on the effect of cooperative evolution on concept learning. In E. J. W. Boers, S. Cagnoni, J. Gottlieb, E. Hart, P. L. Lanzi, G. Raidl, R. E. Smith, and H. Tijink, editors, *Applications of Evolutionary Computing, EvoWorkshop 2001*, pages 314–320. Springer-Verlag, Berlin, 2001.
11. M. Pazzani and D. Billsus. Learning and revising user profiles: the identification of interesting web sites. *Machine Learning*, 27:313–331, 1997.
12. M. Pazzani, J. Muramatsu, and D. Billsus. Syskill and Webert: identifying interesting websites. In W. J. Clancey and D. Weld, editors, *Proc. of the Thirteenth Amer. Nat. Conf. on Artificial Intelligence (AAAI-96)*, volume 1, pages 54–61. AAAI Press, Portland, OR, 1996.
13. M. A. Potter and K. A. De Jong. A cooperative coevolutionary approach to function optimization. In Y. Davidor, H. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature – PPSN-94*, pages 249–257. Springer-Verlag, Berlin, 1994.
14. M. A. Potter and K. A. De Jong. Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.
15. M. A. Potter, K. A. De Jong, and J. J. Grefenstette. A coevolutionary approach to learning sequential decision rules. In L. Eshelman, editor, *Proc. of the Sixth Int. Conf. on Genetic Algorithms*, pages 366–372. Morgan-Kaufmann, San Francisco, CA, 1995.
16. M. A. Potter and K. A. De Jong. The coevolution of antibodies for concept learning. In A. E. Eiben, T. Bäck, M. Schoenauer, and H. Schwefel, editors, *Proc. of the Fifth Int. Conf. on Parallel Problem Solving from Nature (PPSN-98)*, pages 530–539. Springer-Verlag, Amsterdam, 1998.
17. D. Sofge, K. A. De Jong, and A. Schultz. A blended population approach to cooperative coevolution for decomposition of complex problems. In P. Fogel, editor, *Congress on Evolutionary Computation 2002 (CEC-2002)*. IEEE Press, Honolulu, Hawaii, 2002 (*forthcoming*).
18. S. Thrun, J. Bala, E. Bloedorn, I. Bratko, B. Cestnik, J. Cheng, K. A. De Jong, S. Dzeroski, S. E. Fahlman, D. Fisher, R. Hamann, K. Kaufman, S. Keller, I. Kononenko, J. Kreuziger, R. S. Michalski, T. Mitchell, P. Pachowicz, Y. Reich, H. Vafaie, W. Van de Welde, W. Wenzel, J. Wnek, and J. Zhang. The MONK's problems: a performance comparison of different learning algorithms. Technical Report CMU-CS-91-197, School of Computer Science, Carnegie Mellon University, Pittsburg, PA, 1991.
19. J. Twycross. An immune system approach to document classification. Master's thesis, COGS, University of Sussex, U.K., 2002. Submitted.
20. A Watkins. AIRS: A resource limited artificial immune classifier. Master's thesis, Department of Computer Science, Mississippi State University, 2001.
21. S. T. Wierzchon. Generating optimal repertoire of antibody strings in an artificial immune system. In S. T. Wierzchon M. Klopotek, M. Michalewicz, editor, *Intelligent Information Systems*, Advances in Soft Computing, pages 119–133. Physica-Verlag/Springer Verlag, Heidelberg/New York, 2000.
22. Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2):219–241, 2002.
23. B. Zhang and Y. Seo. Personalized web-document filtering using reinforcement learning. *Applied Artificial Intelligence*, 15(7):665–685, 2001.