# An immunological algorithm for discovering small-disjunct rules in data mining

**Deborah R. Carvalho** [1,2]

[2] Universidade Tuiuti do Paraná (UTP)
Computer Science Dept.
Av. Comendador Franco, 1860. Curitiba-PR
80215-090 Brazil
deborah@ipnet.com.br

**Alex A. Freitas** [1]

[1] Pontifícia Universidade Católica do Paraná (PUCPR)
Postgraduate program in applied computer science
R. Imaculada Conceição, 1155. Curitiba – PR
80215-901. Brazil
alex@ppgia.pucpr.br
http://www.ppgia.pucpr.br/~alex

**Abstract**

In essence, small disjuncts are rules covering a small number of examples. Although each small disjunct covers a small number of examples, the set of all small disjuncts can collectively cover a large number of examples. Indeed, this work presents evidence that this is the case. This work also proposes a hybrid decision tree/immunological algorithm method to cope with the problem of small disjuncts. The basic idea is that examples belonging to large disjuncts are classified by rules produced by a decision-tree algorithm, while examples belonging to small disjuncts are classified by an immunological algorithm, specifically designed for this task.

## 1 INTRODUCTION

The current information age is characterized by a great expansion in the volume of data that are being generated and stored. A huge proportion of this data is recorded in the form of computer databases, in order that the computer technology may easily access it.

Some of the most popular data mining tasks are association rules, classification and clustering. In the context of the classification task of data mining, the discovered knowledge is often expressed as a set of IF-THEN rules, since this kind of knowledge representation is intuitive for the user. From a logical viewpoint, typically the discovered rules are in disjunctive normal form, where each rule represents a disjunct and each rule condition represents a conjunct. A small disjunct can be defined as a rule that covers a small number of training examples (Holte et al. 1989).

In this work we propose a hybrid decision tree/immunological algorithm method for rule discovery that copes with the problem of small disjuncts. The basic idea is that examples belonging to large disjuncts are classified by rules produced by a decision-tree algorithm, while examples belonging to small disjuncts (whose classification is considerably more difficult) are classified by rules produced by a new immunological algorithm, specifically designed for discovering small-disjunct rules.

## 2 A HYBRID DECISION -TREE / IMMUNOLOGICAL - ALGORITHM METHOD FOR RULE DISCOVERY

The basic idea of our hybrid method is to use a well-known decision-tree algorithm to classify examples belonging to large disjuncts and use a new immunological algorithm to discover rules classifying examples belonging to small disjuncts. Decision-tree algorithms have a bias towards generality that is well suited for large disjuncts, but not for small disjuncts. On the other hand, immunological algorithms are robust, adaptative algorithms that intuitively can be more easily tailored for coping with small disjuncts.

In the first phase we run C4.5, a well-known decision tree induction algorithm (Quinlan 1993). The induced, pruned tree is transformed into a set of rules. Hence, a decision tree with *d* leaves is transformed into a rule set with *d* rules (or disjuncts). Each of these rules is considered either as a small disjunct or as a “large” (non-small) disjunct, depending on whether or not its coverage (the number of examples covered by the rule) is smaller than a given threshold.

The second phase consists of using an immunological algorithm to discover rules covering the examples belonging to small disjuncts. We have developed a new immunological algorithm for this phase.

### 2.1 AN IMMUNOLOGICAL ALGORITHM FOR DISCOVERING SMALL-DISJUNCT RULES

The architecture of the natural immune system is multilayered, with defenses provided at three levels: skin and mucous membrane, innate immune system and adaptative immune response (Somayaji at al. 1997).

In our system the innate immune task is performed by a decision tree, whereas the immunological algorithm incorporates some features of adaptative immune response. The recognition and response to antigens is performed by antibodies, which in our system are represented by IF-THEN rules. The adaptative immune system possesses two types of response: primary and secondary. The primary occurs when the immune system encounters the antigen for the first time and reacts against it. In our IA this stage is simulated when a new antibody population is created and those antibodies try to cover (to ”match”) the antigens (small-disjunct examples).

### 2.2 ANTIBODY REPRESENTATION

Each antibody represents a conjunction of conditions composing a given rule antecedent. Each condition is an attribute-value pair

To represent a variable-length rule antecedent we use a fixed-length structure, for the sake of simplicity. This rule antecedent representation is described in more detail in (Carvalho & Freitas 2000). The overall structure of an antibody is illustrated in Figure 1.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $A_1$ $Op_1\{V_{1j}..\}$ | $B_1$ | … | $A_i$ $Op_i\{V_{ij}..\}$ | $B_i$ | … | $A_n$ $Op_n\{V_{nj}..\}$ | $B_n$ |

Figure 1: Structure of an antibody (rule antecedent).

### 2.3 EVALUATION FUNCTION

Assume, without loss of generality, that there are two classes. Let positive (“+”) class be the class predicted by a given rule, and negative (“-“) class be any class other than the class predicted by the rule.

To evaluate the quality of an antibody, our IA uses the following evaluation function:

$(TP / (TP + FN)) * (TN / (FP + TN))$, where

TP (true positive) = number of “+” examples that were correctly classified as “+” examples;
FP (false positive) = number of “-” examples that were wrongly classified as “+” examples;
FN (false negative) = number of “+” examples that were wrongly classified as “-” examples;
TN (true negative) = number of “-” examples that were correctly classified as “-” examples.
For a comprehensive discussion about this and related rule-quality measures in general, independent of immunological algorithms, see (Hand 1997).

### 2.4 CLONAL SELECTION

The clonal selection process is driven by antibody-antigen interactions, and it is influenced by concentrations of antigens. The evolution of the antibody population leads to maturation of the immune response, which features itself by an increase in the average match value. The match value is important to the cloning expansion process. In our algorithm, the match value is computed by a rule evaluation function (see section 2.3). At each iteration of the clonal selection algorithm, if an antibody meets the clonal expansion threshold, the system generates 50 clones of that antibody.

Hypermutation is implemented by a relatively high mutation rate (mutation rate of 10%) over an antibody. The hypermutation concept stems from the fact that the same antibody can generate 50 clones, where in each clone a gene can be mutated. At the next generation each one of these 50 clones could generate other 50 clones, and so on.

The best antibody is maintained from one iteration to the next one (elitism with factor 1).

In addition to the above immunological/evolutionary operators, we have also developed a new operator especially designed for improving the comprehensibility of rules. The basic idea of this operator, called rule-pruning operator, is to remove several conditions from a rule to make it shorter. We have devised a rule pruning procedure based on information theory. First of all, it computes the information gain of each of the $n$ rule conditions (genes) in the genome of the antibody – see e.g. (Quinlan 1993) for an explanation of how to compute the information gain. Then it removes rule conditions with a probability inversely proportional to their information gain. (The higher the information gain, the smaller the probability of removal.)

### 2.5 CLASSIFYING TEST EXAMPLES

Once training is over, examples in the test set are classified as follows. For each test example, we first check whether the example is covered by some large-disjunct rule. If so, the example is classified by the corresponding rule, which is one of the rules induced by the decision tree algorithm. Otherwise the example is classified by the following procedure.

First of all, if all training examples in the current small disjunct leaf node have the same class then assigns that class to the test example. Otherwise the system checks whether the test example is covered by some rule(s) induced by the Immunological Algorithm for that small disjunct. If so, the best rule covering the test example (according to the given rule quality measure described in section 2.3) is used to classify that example, as long as a restriction is satisfied: the rule’s quality must be greater than a predetermined threshold (see below). Finally, if there is no rule covering the test example, or if the best rule covering that example fails to have a quality greater than the threshold, then the test example is simply assigned the majority class in the training examples belonging to the current small disjunct.

## 3 COMPUTATIONAL RESULTS

In order to evaluate the proposed IA, we have used 8 public domain data sets of the well-known data repository of the UCI (University of California at Irvine): Adult, Connect, CRX, Hepatitis, Segmentation, Splice, Voting and Wave. These data sets are available at the web site http://www.ics.uci.edu/~mlearn/MLRepository.html. For the Adult data set, the examples that had some missing value were removed from the data set. For the others data sets we did not use this procedure because those data sets are relatively small, so that, intuitively, the removal of examples could degrade predictive accuracy. Instead, in the other data sets missing values were replaced by average values (continuous attibutes) or modal values (categorical attributes).

In each run of the IA, the initial antibody population size is 200, and the IA is run for 50 generations or until the best antibody has its evaluation value equal to 1.00.

Intuitively, the performance of our method will be significantly dependent on the definition of small disjunct. In our experiments we have used a commonplace definition of small disjunct, based on a fixed threshold of the number of examples covered by the disjunct. The general definition is: "A decision-tree leaf is considered a small disjunct if and only if the number of examples belonging to that leaf is smaller than or equal to a fixed size *S*." We have done experiments with four different values for the parameter *S*, namely $S = 3$, $S = 5$, $S = 10$ and $S = 15$.

For each of these four *S* values, we have done five different experiments, varying the random seed used to generate the initial population of antibodies. The results reported below, for each value of S, are an arithmetic average of the results over these five different experiments. Therefore, the total number of experiments, for each data set, is at least 20 (4 values of S * 5 different random seeds).

The results comparing the performance of our hybrid C4.5/IA method against C4.5 alone are shown in the Table 1. We have used the default parameters of C4.5.

Table 1: Results (accuracy rate) comparing our hybrid C4.5/IA algorithm against C4.5

| | | Hybrid C4.5/IA | | | | C4.5(2) |
|---|---|---|---|---|---|---|
| data set | C4.5 | S = 3 | S = 5 | S = 10 | S = 15 | S = 15 |
| Adult | 0,7860 | 0,7818 | 0,7782 | **0,7927** | **0,7869** | 0,8479 |
| Wave | 0,7554 | 0,7239 | 0,6982 | 0,6135 | 0,5860 | 0,7231 |
| Connect | 0,7862 | 0,7795 | 0,7773 | 0,7681 | 0,7625 | 0,7808 |
| Splice | 0,4598 | **0,4618** | **0,4638** | **0,4680** | **0,4803** | 0,4632 |
| Hepatitis | 0,8364 | 0,8254 | 0,8058 | **0,8621** | **0,8534** | 0,8316 |
| Segmentation | 0,9767 | 0,9462 | 0,9344 | 0,9297 | 0,9326 | 0,6319 |
| Voting | 0,9463 | 0,9144 | 0,8990 | 0,8800 | 0,8869 | 0,9271 |
| CRX | 0,8453 | 0,7918 | 0,7707 | **0,8521** | **0,8683** | 0,8430 |

The results are shown in Table 1. The first column of this table indicates the data set. The second column shows the accuracy rate on the test set achieved by C4.5, classifying both large- and small-disjunct examples. The next four columns report the overall accuracy rate on the test set achieved by our hybrid C4.5/IA algorithm, i.e. using C4.5 to classify large-disjunct examples and our IA to classify small-disjuncts examples. Each of those four columns reports results for one specific value of *S* (small disjunct size).

In addition to the above comparison, we have also compared the results of our system with a "double run" of C4.5, as explained below.

The sixth column also represents the accuracy rate on the test set achieved by C4.5. The difference between the second and the sixth columns is the strategy used to build the classifier. One classifier (second column) is the result of one C4.5 run as usual. The other classifier, C4.5(2), is built by running C4.5 twice, as follows. First, we run C4.5 with default parameters, as usual. In this first run the training set is the original one, with all examples. Next the system groups all the examples belonging to small disjuncts (according to the first run of C4.5) into a single example subset. This can be thought of as a second training set. C4.5 is run again on this second training set. In order to classify a new example, the rules discovered by both runs of C4.5 are used as follows. First, the system checks whether the new example belongs to a large disjunct of the first decision tree. If so the class predicted by the corresponding leaf node is assigned to the new example. Otherwise (i.e. the example belongs to one of the small disjuncts of the first decision tree), the new examples is given to the second decision tree. Once again, the system checks whether on not the example belongs to a large disjunct of the second decision tree. If so, it is classified by the corresponding leaf node. Otherwise, it is finally classified by a default rule, which predicts the majority class in the leaf node of the first decision tree which the example belongs to. In Table 1 we report the results of this "double run" of C4.5 only for S=15, since this is the small disjunct size which leads to the largest second training set among the four values of S. Hence, the predictive accuracy of double-run C4.5 tends to be better when S=15.

In columns 3 through 6 of Table 1, the cells where the hybrid C4.5/IA achieved a higher accuracy rate than C4.5 alone or C4.5(2) are shown in bold. As shown in the table, C4.5 alone outperforms the hybrid C4.5/IA in almost all data sets (except Splice) when the definition of small disjunct is set to *S*=3 or *S*=5– i.e. any leaf node with $\leq 3$ (or 5) examples is considered a small disjunct. However, when the definition of small disjunct is set to *S*=10 or *S=15* the hybrid C4.5/IA outperforms C4.5 alone in four of the eight data sets, namely Splice, Hepatitis and Crx.

The bad results for S=3 and S=5 where somewhat expected. An explanation for these results is as follows. When a disjunct is considered as small if it covers $\leq 3$ or $\leq 5$ examples, there are very few training examples available for each IA run. With so few examples the estimate of antibody (rule) quality computed by the evaluation function is far from perfect, and the IA does not manage to do better than C4.5.

On the other hand, when a disjunct is considered as small if it covers ≤ 10 or ≤ 15 examples, the number of training examples available for the IA is considerable higher - although still relatively low. Now the estimate of rule quality computed by the evaluation function is significantly better. As a result, the IA manages, in several cases, to discover small-disjunct rules that have a better predictive accuracy than some small disjunct leaf nodes generated by C4.5.

In practice we do not recommend to set the small disjunct definition to *S*=3 or *S*=5, since in this cases there would be too few examples for discovering reliable small-disjunct rules. We have done experiments with *S*=3 and *S*=5 only for the sake of completeness and for confirming the above-mentioned expectation.

The results of Table 1 show that, when *S*=10 or *S*=15, the hybrid C4.5/IA is competitive with C4.5 alone.

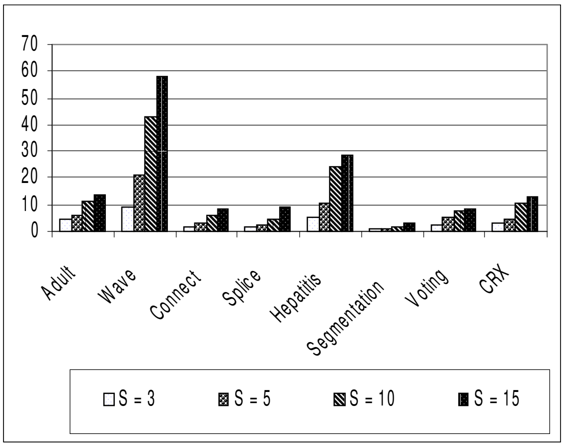


Figure 2 Frequency of the small disjuncts found in the data sets used in our experiments

Figure 2 shows the percentage of training examples which belong to small disjuncts for *S*=3, *S*=5, *S*=10 and *S*=15. As discussed in the introduction, the percentage of small-disjunct examples is representative, particularly when *S*=15. Specifically, note that in the wave data set more than 50% of the examples belong to small disjuncts.

A disadvantage of our hybrid C4.5/IA method is that it is much more computationally expensive than the use of C4.5 alone.

## 4 CONCLUSION AND FUTURE RESEARCH

In this work we have a hybrid decision-tree/immunological algorithm method, where examples belonging to large disjuncts are classified by rules produced by a decision-tree algorithm and examples belonging to small disjuncts are classified by rules produced by a immunological algorithm. In order to realize this hybrid method we have used the well-known C4.5 decision-tree algorithm and developed a new immunological algorithm tailored for the discovery of small-disjunct rules.

The computational results reported in section 4 show that, as long as we are careful to define what constitutes a small disjunct, the hybrid C4.5/IA is competitive with C4.5 alone. A reasonable definition of small disjuncts is a decision tree leaf node having ≤ 10 or ≤ 15 examples (*S*=10 and *S*=15, respectively). Smaller values of *S* would tend to imply there are two few examples for reliable generalization, and larger values of *S* would go against the notion of "small" disjuncts. Our results show that, when *S*=10 or *S*=15, the hybrid C4.5/IA method achieves better accuracy rate than C4.5 alone in 3 out of 8 data sets.

As discussed in the introduction, we showed that the percentage of small-disjunct examples is representative, reaching about 50% for one data set.

There are several possible directions for future research. An important one is to evaluate the performance of the proposed hybrid C4.5/IA method for different kinds of antibody representation, not only rules as in the current version of the system. For example, one could include a representation based on prototypes (i. e. typical instances of a class, in the nearest neighbor paradigm).

Another important research direction is to evaluate the performance of the proposed hybrid C4.5/IA method for different kinds of definition of small disjunct, e.g. relative size of the disjunct (rather than absolute size, as considered in this work).

Yet another interesting research direction would be to compare the results of the proposed C4.5/IA method against rules discovered by the IA only, although in this case the design of the IA would have to be somewhat modified.

## REFERENCES


CARVALHO, Deborah R.; FREITAS, Alex, A. (2000). A hybrid decision tree/genetic algorithm for coping with the problem of small disjunct in Data Mining. *Proc 2000 Genetic and Evolutionary Computation Conf. (GECCO-2000), 1061-1068. Las Vegas, NV, USA. July 2000.*

HAND, D. J. (1997). *Construction and Assessment of Classification Rules*, John Wiley & Sons, England.

HOLTE, Robert, C.; ACKER, Liane, E. and PORTER, Bruce, W. (1989). Concept Learning and the Problem of Small Disjuncts, *Proc. Int. Joint Conf. On Artificial Intelligence. IJCAI – 89*, 813-818.

QUINLAN, J. ROSS (1993). *C4.5 Programs for Machine Learning*, Morgan Kaufmann Publisher, USA.

SOMAYAJI, Anil; HOFMEYR, Steven; FORREST Stephanie (1997). Principles of a Computer Immune System. *New Security Paradigms Workshop, 75-82. ACM – 1998.*