# A genetic algorithm-based solution for the problem of small disjuncts

Deborah R. Carvalho[1, 2], Alex A. Freitas[1]
[1] Pontificia Universidade Catolica do Parana (PUCPR)
Postgraduate program in applied computer science
R. Imaculada Conceicao, 1155. Curitiba – PR 805215-901. Brazil
Tel/Fax:(55) (41) 330-1669
http://www.ppgia.pucpr.br/~alex alex@ppgia.pucpr.br
[2] Universidade Tuiuti do Parana (UTP)
Computer Science Dept.
Av. Comendador Franco, 186. Curitiba – PR 80215-090. Brazil
Tel/Fax: (55) (41) 263-3424 deborah@utp.br

**Abstract.** In essence, small disjuncts are rules covering a small number of examples. Hence, these rules are usually error-prone, which contributes to a decrease in predictive accuracy. The problem is particularly serious because, although each small disjuncts covers few examples, the set of small disjuncts can cover a large number of examples. This paper proposes a solution to the problem of discovering accurate small-disjunct rules based on genetic algorithms. The basic idea of our method is to use a hybrid decision tree / genetic algorithm approach for classification. More precisely, examples belonging to large disjuncts are classified by rules produced by a decision-tree algorithm, while examples belonging to small disjuncts are classified by a new genetic algorithm, particularly designed for discovering small-disjunct rules.

## 1 Introduction

In the context of the well-known classification task of data mining, the discovered knowledge is often expressed as a set of IF-THEN prediction rules. 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 which covers a small number of training examples [7].

In general rule induction algorithms have a bias that favors the discovery of large disjuncts, rather than small disjuncts. This preference is due to the belief that it is better to capture generalizations rather than specializations in the training set, since the latter are unlikely to be valid in the test set.

Hence, at first glance small disjuncts should not be included in the discovered rule set, since they tend to be error prone. However, a deeper study of the issue of small disjuncts reveals that in fact they can be necessary and even interesting by themselves in the context of data mining, for the following reasons:

(a) Although each disjunct covers a small number of examples, the set of all small disjuncts can cover a large number of examples. For instance [3] reports a real-world application where small disjuncts cover roughly 50% of the training examples. Therefore, if the rule induction algorithm ignores small disjuncts and discovers only large disjuncts, classification accuracy will be significantly degraded.
(b) Some small disjuncts cover examples that represent rare cases in the application domain, which constitutes an interesting concept to be discovered. Actually, bearing in mind that one of the goals of data mining is to discover previously-*unknown* rules, small-disjunct rules tend to be more interesting than large-disjunct rules, since the latter are more likely to be previously-known by the user [11].

In this paper we propose a hybrid decision tree/genetic 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 more difficult) are classified by rules produced by a new genetic algorithm.

## 2 A hybrid decision-tree/genetic-algorithm method

We propose a hybrid method for rule discovery that combines decision trees and genetic algorithms. Decision-tree algorithms have a bias towards generality that is well suited for large disjuncts, but not for small disjuncts. On the other hand, genetic algorithms are robust algorithms which tend to cope well with attribute interactions [4], [10]. Hence, they can be more easily tailored for coping with small disjuncts, which are associated with large degrees of attribute interaction [13], [9].

The proposed method discovers rules in two training phases. In the first phase we run the C4.5 decision-tree algorithm [12]. Then, the induced 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 or equal to a given threshold.

The second phase consists of using a genetic algorithm to discover rules covering examples belonging to small disjuncts. We have developed a new genetic algorithm (GA) for this phase. In our GA, each individual represents a small-disjunct rule.

Each run of our GA discovers a single rule (the best individual of the last generation) predicting a given class for examples belonging to a given small disjunct. We need to run our GA $d * c$ times, where $d$ is the number of small disjuncts and $c$ is the number of classes to be predicted. For a given small disjunct, the $i$-th run of the GA, $i = 1,...,c$, discovers a rule predicting the $i$-th class.

The genome of an individual consists of a conjunction of conditions composing the antecedent (IF part) of the rule. Each condition is an attribute-value pair - see below. The consequent (THEN part) of the rule, which specifies the predicted class, is not represented in the genome. Rather, it is fixed for a given GA run, so that all individuals have the same rule consequent during all that run.

The rule antecedent contains a variable number of rule conditions. In our GA the minimum number of conditions is always 2. The maximum number of conditions, n, depends on the small disjunct, as follows.

To represent a variable-length rule antecedent (phenotype) we use a fixed-length genome. For a given GA run, the genome of an individual consists of $n$ genes, where $n = m - k$, $m$ is the total number of predictor attributes in the dataset and $k$ is the number of ancestor nodes of the decision tree leaf node identifying the small disjunct in question. Hence, the genome of a GA individual contains only the attributes that were *not* used to label any ancestor of the leaf node defining that small disjunct.

The overall structure of the genome of an individual is illustrated in Figure 1. Each gene represents a rule condition (phenotype) of the form $A_i\ Op_i\ V_{ij}$, where the subscript $i$ identifies the rule condition, $i = 1,...,n$; $A_i$ is the i-th attribute; $V_{ij}$ is the j-th value of the domain of $A_i$; and $Op_i$ is a logical/relational operator compatible with attribute $A_i$. Each gene consists of four elements, as follows:
(a) identification of a given predictor attribute, $A_i$, $i = 1,...,n$.
(b) identification of a logical/relational operator $Op_i$. For categorical (nominal) attributes, $Op_i$ is "*in*". For continuous (real-valued) attributes, $Op_i$ is either "≤" or ">".
(c) identification of a set of attribute values $\{V_{i1},...,V_{ik}\}$, if the attribute $A_i$ is categorical, or a single attribute value $V_{ij}$, if the attribute $A_i$ is continuous.
(d) a flag, called the active bit $B_i$, which takes on 1 or 0 to indicate whether or not, respectively, the i-th condition is present in the rule antecedent (phenotype).

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

**Figure 1:** Structure of the genome of an individual.

To evaluate the quality of an individual our GA uses the fitness function:

$$\text{fitness} = (TP / (TP + FN)) * (TN / (FP + TN)) , \qquad (1)$$

where TP, FN, TN and FP – standing for number of true positives, false negatives, true negatives and false positives – are well-known variables often used to evaluate the performance of classification rules – see e.g. [6].

We use tournament selection, with tournament size of 2 [8]. We also use standard one-point crossover with crossover probability of 80%, and mutation probability of 1%. Furthermore, we use elitism with an elitist factor of 1 - i.e. the best individual of each generation is passed unaltered into the next generation.

In addition to the above standard genetic operators, we have also developed a new operator especially designed for simplifying candidate rules. The basic idea of this operator, called rule-pruning operator, is to remove several conditions from a rule to make it shorter. This operator is applied to every individual of the population, right after the individual is formed.

Unlike the usually simple operators of GA, our rule-pruning operator is an elaborate procedure based on information theory [2]. Hence, it can be regarded as a way of incorporating a classification-related heuristic into a GA for rule discovery. The heuristic in question is to favor the removal of rule conditions with low information gain, while keeping the rule conditions with high information gain – see [1] for details.

Once all the $d * c$ runs of the GA are completed, examples in the test set are classified. For each test example, we push the example down the decision tree until it reaches a leaf node. If that node is a large disjunct, the example is classified by the decision tree algorithm. Otherwise we try to classify the example by using one of the $c$ rules discovered by the GA for the corresponding small disjunct. If there is no small-disjunct rule covering the test example it is classified by a default rule. We have experimented with two strategies for defining the default rule:

- a global default rule that predicts the majority class among all small-disjunct.
- a local default rule that predicts the majority class among the examples belonging to the current small disjunct.

## 3 Computational Results

We have evaluated our GA on two public domain data sets from the UCI data set repository (http://www.ics.uci.edu/~mlearn/MLRepository.html).

One of the them is the Adult data set (USA census). This dataset contains 48842 examples, 14 attributes (6 are continuous and 8 are categorical), and two classes. In our experiments we have used the predefined division of the data set into a training and a test set, with the former having 32561 examples and the latter having 16281 examples. The examples that had some missing value were removed from the data set. As a result, the number of examples was slightly reduced to 30162 and 15060 examples in the training and test set, respectively.

The other data set used in our experiments was the Wave data set. This data set contains 5000 instances, 21 attributes with values between 0 and 6 and three classes. In this data set we have run a five-fold cross-validation procedure.

In our experiments 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 $S = 15$.

We now report results comparing the performance of the proposed hybrid C4.5/GA with C4.5 alone [12]. We have used C4.5's default parameters. In each GA run the population has 200 individuals, and the GA is run for 50 generations.

We have evaluated two variants of our hybrid C4.5/GA method: global and local default rule (see above). The results for the Adult and Wave data sets are shown in Tables 1 and 2. All results refer to accuracy rate on the test set. The first column of these tables indicate the size threshold $S$ used to define small disjuncts. The next three columns, labeled (a), (b), (c), report results produced by C4.5 alone. More precisely, columns (a) and (b) report the accuracy rate on the test set achieved by C4.5 separately for examples classified by large-disjunct rules and small-disjunct rules. Column (c) reports the overall accuracy rate on the test achieved by C4.5, classifying both large- and small-disjunct examples. Note that the figures in this column are of course constant across all the rows, since its results refer to the case where all test examples are classified by C4.5 rules, regardless of the definition of small disjunct.

**Table 1:** Results comparing our hybrid C4.5/GA with C4.5 in the Adult data set.

| | Accuracy rate of C4.5 only | | | Accuracy rate of C4.5/GA – global default rule | | | Accuracy rate of C4.5 / GA – local default rule | | |
|---|---|---|---|---|---|---|---|---|---|
| Dis-junct size (S) | (a) large disjuncts | (b) small disjuncts | (c) overall | (d) large disjuncts | (e) small disjuncts | (f) overall | (g) large disjuncts | (h) small disjuncts | (i) overall |
| 3 | 0.800 | 0.512 | 0.786 | 0.800 | 0.470 | 0.780 | 0.800 | 0.457 | 0.779 |
| 5 | 0.811 | 0.520 | 0.786 | 0.811 | 0.497 | 0.780 | 0.811 | 0.483 | 0.779 |
| 10 | 0.841 | 0.521 | 0.786 | 0.841 | 0.640 | 0.828 | 0.841 | 0.642 | 0.829 |
| 15 | 0.840 | 0.530 | 0.786 | 0.840 | 0.711 | 0.831 | 0.840 | 0.707 | 0.830 |

**Table 2:** Results comparing our hybrid C4.5/GA with C4.5 in the Wave data set.

| | Accuracy rate of C4.5 only | | | accuracy rate of C4.5/GA – global default rule | | | Accuracy rate of C4.5 / GA – local default rule | | |
|---|---|---|---|---|---|---|---|---|---|
| Dis-junct size (S) | (a) large disjuncts | (b) small disjuncts | (c) overall | (d) large disjuncts | (e) small disjuncts | (f) overall | (g) large disjuncts | (h) small disjuncts | (i) overall |
| 3 | 0.758 | 0.722 | 0.755 | 0.758 | 0.776 | 0.765 | 0.758 | 0.732 | 0.756 |
| 5 | 0.774 | 0.710 | 0.755 | 0.774 | 0.727 | 0.758 | 0.774 | 0.754 | 0.764 |
| 10 | 0.782 | 0.731 | 0.755 | 0.782 | 0.800 | 0.793 | 0.782 | 0.808 | 0.796 |
| 15 | 0.788 | 0.731 | 0.755 | 0.788 | 0.832 | 0.814 | 0.788 | 0.814 | 0.803 |

The next three columns, labeled (d), (e), (f), report results produced by our hybrid C4.5/GA method in the variant of global default rule. Note that the figures in column (d) are exactly the same as the figures in column (b), since our hybrid method also uses C4.5 rules for classifying examples belonging to large disjuncts. In any case, we included this redundant column in the Tables for the sakes of comprehensibility and completeness. Column (e) reports the accuracy rate on the test set for the small-disjunct rules discovered by the GA. Finally, column (f) reports the overall accuracy rate on the test achieved by our hybrid C4.5/GA method, classifying both large- and small-disjunct examples. The next three columns, labeled (g), (h), (i), refer to the results with the variant of local default rule. The meaning of these columns is analogous to the one explained for columns (d), (e), (f), respectively.

As can be seen in Tables 1 and 2, there is little difference of performance between the two variants of our hybrid C4.5/GA, and overall both variants achieved better predictive accuracy than C4.5 alone. More precisely, comparing both columns (e) and (h) with column (b) in each of those two tables we can note two distinct patterns of results. Consider first the case where a disjunct is considered as small if it covers ≤ 3 or ≤ 5 examples, as in the first and second rows of Tables 1 and 2. In this case the accuracy rate of the small-disjunct rules produced by the GA is slightly inferior to the performance of the small-disjunct rules produced by C4.5 in the Adult data set (Table 1), while the former is somewhat superior to the latter in the Wave data set. In any case, this small difference of performance referring to small-disjunct rules has a small impact on the overall accuracy rate, as can be seen by comparing both columns (f) and (i) with column (c) in Tables 1 and 2.

A different picture emerges when a disjunct is considered as small if it covers ≤ 10 or ≤ 15 examples, as in the third and fourth rows of Tables 1 and 2. Now the performance of the small-disjunct rules produced by the GA is much better than the performance of the small-disjunct rules produced by C4.5, in both data sets. For instance, comparing both columns (e) and (h) with the column (b) in the fourth row of Table 1, the GA-discovered small disjunct rules have an accuracy rate of 71.1% and 70.7%, whereas the C4.5-discovered rules have an accuracy rate of only 53%. This improved accuracy associated with GA-discovered small disjunct rules has a considerable impact on the overall accuracy rate, as can be seen comparing both columns (f) and (i) with column (c) in the third and fourth rows of Tables 1 and 2.

A possible explanation for these results is as follows. In the first case, where a disjunct is considered as small if it covers ≤ 3 or ≤ 5 examples, there are very few training examples available for each GA run. With so few examples the estimate of rule quality computed by the fitness function is far from perfect, and the GA does not manage to do better than C4.5. On the other hand, in the second case, where a disjunct is considered as small if it covers ≤ 10 or ≤ 15 examples, the number of training examples available for the GA is significantly higher - although still relatively low. Now the estimate of rule quality computed by the fitness function is significantly better. As a result, the GA's robustness and ability to cope well with attribute interaction lead to the discovery of small-disjunct rules considerably more accurate than the corresponding rules discovered by C4.5.

Although the above results are good, they do not prove by themselves that the small-disjunct rules discovered by the GA are considerably superior to the small-disjunct rules discovered by C4.5. After all, recall that the test examples belonging to small disjuncts can be classified either by a GA-discovered rule or by the default rule. This raises the question of which of these two kinds of rule is really responsible for the good results reported above.

To answer this question we measured separately the relative frequency of use of each of the two kinds of rule, namely GA-discovered rules and default rule, in the classification of test examples belonging to small disjuncts. We found that GA-discovered rules are used much more often to classify test examples belonging to small disjuncts than the default rule. More precisely, depending on the definition of small disjunct (the value of the parameter S) used, the relative frequency of use of GA-discovered rules varies between 68.7% and 95% for the Adult data set and from 84.1% to 90.2% in the Wave data set. Hence, one can be confident that the small disjunct rules discovered by the GA are doing a good job of classifying most of the test examples belonging to small disjuncts. In any case, to get further evidence we also measured separately the predictive accuracy of GA-discovered rules and default rule in the classification of test examples belonging to small disjuncts in the case of global-default rule. Overall, as expected, the GA-discovered rules have a higher predictive accuracy than the default rule. To summarize, as expected most of the credit for the good performance in the classification of small disjuncts is to be assigned to the GA-discovered rules, rather than to the default rules. (Note that there is no need to get this kind of evidence in the case of the local-default rules, since in this case there is no other way that the GA-tree hybrid could beat the tree alone.)

Turning to computational efficiency issues, each run of the GA is relatively fast, since it uses a training set with just a few examples. However, recall that in order to discover all small disjunct rules we need to run the GA $c * d$ times, where $c$ is the number of classes and $d$ is the number of small disjuncts. The total processing time taken by the $c * d$ GA runs varies with the number of small disjuncts, which depends on both the data set and on the definition of small disjunct (the value of S). In our experiments, the processing time taken by all $c * d$ runs of the GA was about one hour for the largest data set, Adult, and the largest number of small disjuncts, associated with S = 15. The experiments were performed on a 64-Mb Pentium II. One hour seems to us a reasonable processing time and a small price to pay for the considerable increase in the predictive accuracy of the discovered rules.

Finally, if necessary the processing time taken by all the $c * d$ GA runs can be considerably reduced by using parallel processing techniques [5]. Actually, our method greatly facilitates the exploitation of parallelism in the discovery of small disjunct rules, since each GA run is completely independent from the others and it needs to have access only to a small data set, which surely can be kept in the local memory of a simple processor node.

## 4 Conclusions and Future Research

The problem of how to discover good small-disjunct rules is very difficult, since these rules are error-prone due to the very nature of small disjuncts. Ideally, a data mining system should discover good small-disjunct rules without sacrificing the goodness of discovered large-disjunct rules.

Our proposed solution to this problem was a hybrid decision-tree/GA 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 genetic algorithm. In order to realize this hybrid method we have used the well-known C4.5 decision-tree algorithm and developed a new genetic algorithm tailored for the discovery of small-disjunct rules.

The proposed hybrid method was evaluated in two data sets. We found that the performance of our new GA and corresponding hybrid C4.5/GA method depends significantly on the definition of small disjunct. The results show that: (a) there is no significant difference in the accuracy rate of the rules discovered by C4.5 alone and the rules discovered by our C4.5/GA method when a disjunct is considered as small if it covers ≤ 3 or ≤ 5 examples; (b) the accuracy rate of the rules discovered by our C4.5/GA method is considerably higher than the one of the rules discovered by C4.5 alone when a disjunct is considered as small if it covers ≤ 10 or ≤ 15 examples.

A disadvantage of our hybrid C4.5/GA method is that it is much more computationally expensive than the use of C4.5 alone. More precisely, in a training set with about 30000 examples our hybrid method takes on the order of one hour, while C4.5 alone takes on the order of a few seconds. However the extra processing time is not too excessive, and it seems a small price to pay for the considerable increase in the predictive accuracy of the discovered rules.

An important direction for future research is to evaluate the performance of the proposed hybrid C4.5/GA method for different kinds of definition of small disjunct, e.g. relative size of the disjunct (rather than absolute size, as considered in this paper). Another research direction would be to compare the results of the proposed C4.5/GA method against rules discovered by the GA only, although in this case some aspects of the design of the GA would have to be modified.

## References

1. CARVALHO, D.R. and FREITAS, A.A. A hybrid decision tree/genetic algorithm for coping with the problem of small disjuncts in data mining. *To appear in Proc. 2000 Genetic and Evolutionary Computation Conf. (GECCO-2000)*. Las Vegas, NV, USA. July 2000.
2. COVER, T.M., THOMAS, J.A. (1991) *Elements of Information Theory.* John Wiley&Sons.
3. DANYLUK, A., P. and PROVOST, F.,J. (1993). Small Disjuncts in Action: Learning to Diagnose Errors in the Local Loop of the Telephone Network, *Proc. 10th International Conference Machine Learning*, 81-88.
4. FREITAS, A.A. (2000) Evolutionary Algorithms. Chapter of forthcoming *Handbook of Data Mining and Knowledge Discovery.* Oxford University Press, 2000.
5. FREITAS, A.A. and LAVINGTON, S.H. (1998) *Mining Very Large Databases with Parallel Processing*. Kluwer.
6. HAND, D.J.(1997) *Construction and Assessment of Classification Rules*. John Wiley& Sons
7. HOLTE, R.C.; ACKER, L.E. and PORTER, B.W. (1989). Concept Learning and the Problem of Small Disjuncts, *Proc. IJCAI – 89*, 813-818.
8. MICHALEWICZ, Z. (1996) *Genetic Algorithms + Data Structures = Evolution Programs.* 3rd Ed. Springer-Verlag.
9. NAZAR, K. and BRAMER, M.A. (1999) Estimating concept difficulty with cross entropy. In: Bramer, M.A. (Ed.) *Knowledge Discovery and Data Mining*, 3-31. London: IEE.
10. NODA, E.; LOPES, H.S.; FREITAS, A.A. (1999) Discovering interesting prediction rules with a genetic algorithm. *Proc. Congress on Evolutionary Comput.* (*CEC-99*), 1322-1329
11. PROVOST, F. and ARONIS, J.M. (1996). Scaling up inductive learning with massive parallelism. *Machine Learning 23(1)*, Apr. 1996, 33-46.
12. QUINLAN, J.R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann.
13. RENDELL, L. and SESHU, R. (1990) Learning hard concepts through constructive induction: framework and rationale. *Computational Intelligence 6*, 247-270.