Developments on a Multi-Objective Metaheuristic (MOMH) Algorithm for Finding Interesting Sets of Classification Rules

Beatriz de la Iglesia, Alan Reynolds, and Vic J Rayward-Smith[1]

University of East Anglia, Norwich, Norfolk, NR4 7TJ, UK,
bli,ar,vjrs@cmp.uea.ac.uk

Abstract. In this paper, we experiment with a combination of innovative approaches to rule induction to encourage the production of interesting sets of classification rules. These include multi-objective metaheuristics to induce the rules; measures of rule dissimilarity to encourage the production of dissimilar rules; and rule clustering algorithms to evaluate the results obtained.
Our previous implementation of NSGA-II for rule induction produces a set of cc-optimal rules (coverage-confidence optimal rules). Among the set of rules produced there may be rules that are very similar. We explore the concept of rule similarity and experiment with a number of modifications of the crowding distance to increasing the diversity of the partial classification rules produced by the multi-objective algorithm.

1 Introduction

Data mining is concerned with the extraction of patterns from large databases. One particular task of data mining which is attracting increased research attention is the extraction of classification rules. Partial classification, also known as nugget discovery, involves the production of accurate yet simple rules (nuggets) that describe subsets of interest within a database.

Recently, we have developed a multi-objective metaheuristic algorithm for the extraction of partial classification rules [7]. The problem of nugget discovery was formulated as a multi-objective optimisation problem by using some of the frequently used measures of interest, namely confidence and coverage of a nugget, as objectives to be optimised. NSGA-II was then used to perform the search for Pareto-optimal rules according to the defined objectives.

The approach was evaluated by comparison to another algorithm, ARAC [18] which is guaranteed to find all cc-optimal rules subject to certain constraints. The constraints may affect the number of attribute tests that are allowed in the antecedent of the rule, or the maximum cardinality allowed for any attribute that participates in a test. For small datasets, where constraints do not have to be applied, ARAC can deliver all the cc-optimal (i.e. Pareto-optimal) rules efficiently, so it provides a perfect point of comparison.

Results showed the strength of the new multi-objective approach for finding a good approximation to the Pareto front in a number of datasets. For the larger

datasets, the multi-objective approach showed real advantage as it could find good sets of solutions in a fraction of the time, with predictable termination times, and without having to apply any restrictions to the number of attributes or their cardinality.

One question raised was whether the set of rules delivered may contain very similar rules or rules that appear to be different but match similar records. There may also be rules that are interesting because they cover different subsets of records, but which are dominated in terms of coverage and confidence and are, therefore, never found.

In this paper we investigate the quality of the rule sets obtained. In particular, we investigate various options for refining the quality of the rule sets obtained in order to deliver an *interesting* set of rules or nuggets. Defining interest in rule induction has been an area of research for some time. Most methods of measuring individual rule interest use a combination of confidence and support for the rule [14]. Considerations about rule novelty or surprise for individual rules are sometimes included [9, 10]. We study the novelty of the rules in relation to other rules within the set; that is, we would like to deliver a set of rules of high quality in terms of confidence and coverage, but also where rules are as diverse as possible with respect to other rules in the same set. This should increase the interest of the rule set, as opposed to the interest of the individual rules. We examine the interest of the rule sets obtained with our previous approach and attempt various modifications to improve our rule sets.

Section 2 covers the basic concepts and terminology used in the paper. Section 3 describes briefly the original multi-objective nugget discovery algorithm. Section 4 describes measures of rule dissimilarity and their applicability to the algorithm and introduces the concept of clustering rules for better interpretation of results. We describe some initial experimentation in section 5 and give our conclusions and ideas for further work in section 6.

2 Concepts and terminology

2.1 Nugget Discovery

The task of partial classification [1] is also known as nugget discovery; it seeks to find patterns that represent "strong" descriptions of a specified class, even when that class has few representative cases in the data. For example, in insurance data, groups of people that constitute an unacceptably high risk are in a minority. However, if an insurer can identify such groups, with their defining characteristics, they may gain a competitive advantage.

Let Q be a finite set of attributes where each $q \in Q$ has an associated domain, $\mathrm{Dom}(q)$. Then a record specifies values for each attribute in Q. A tabular database, D, is defined to be a finite set of such records. A classification tabular dataset is one in which a class attribute is present.

Rules that represent a partial classification are of the general form

$$\text{antecedent} \Rightarrow \text{consequent}$$

where the antecedent and consequent are predicates that are used to define subsets of records from the database D and the rule underlines an association between these subsets. In nugget discovery, the antecedent comprises a conjunction of Attribute Tests, ATs, and the consequent comprises a single AT representing the class description. The strength of the rule may be expressed by various measures, as described in section 2.2

Attributes may be described as ordinal or nominal (categorical). An ordinal attribute is defined as an attribute that has some explicit or implicit ordering, so numeric attributes are usually ordinal. Nominal attributes are those that are not ordinal, i.e. they have no implied ordering.

For a database with n attributes, the ATs for nominal attributes can be expressed in any of the following forms:

Simple value: $AT_j = v$, where v is a value from the domain of AT_j, Dom_j, for some $1 \leq j \leq n$. A record x satisfies this test if $x[AT_j] = v$.

Subset of values: $AT_j \in \{v_1, \ldots, v_k\}$, where $\{v_1, \ldots, v_k\}$ is a subset of values in the domain of AT_j, for some $1 \leq j \leq n$. A record x satisfies this test if $x[AT_j] \in \{v_1, \ldots, v_k\}$.

Inequality test: $AT_j \neq v$, for some $1 \leq j \leq n$. A record x satisfies this test if $x[AT_j] \neq v$.

For a numeric attribute, ATs can take the following form:

Simple value: $AT_j = v$, $1 \leq j \leq n$, as for categorical attributes.

Binary partition: $AT_j \leq v$ or $AT_j \geq v$, for some $1 \leq j \leq n$, $v \in Dom_j$. A record x satisfies these tests if $x[AT_j] \leq v$ or $x[AT_j] \geq v$ respectively.

Range of values: $v_1 \leq AT_j \leq v_2$ or $AT_j \in [v_1, v_2]$, for some $1 \leq j \leq n$ and $v_1, v_2 \in Dom_j$. A record x satisfies this test if $v_1 \leq x[AT_j] \leq v_2$.

Decision tree induction [4, 15] and rule induction algorithms [5, 6] are often used to extract partial classification rules. However, decision trees often have thousands of rules, with each rule covering only a few cases; hence their use as descriptive patterns is limited. Also, both decision tree and rule induction algorithms often fail to produce patterns for minority classes. Association rule algorithms have been adapted to find patterns in classification data [2, 3], but they are predominantly developed for categorical data and often apply restrictions to the syntax of the rules to keep the search feasible. They deliver *all* rules underlying a database, which can result in output of overwhelming size.

2.2 Strength of a rule

Given a record, t, *antecedent*(t) (represented by a conjunction of ATs in nugget discovery) is true if t satisfies the predicate, *antecedent*. Similarly *consequent*(t) is true if t satisfies the predicate, *consequent*. Then the subsets defined by the antecedent or consequent are the sets of records for which the relevant predicate is true.

For a rule r, we define three sets of records.

$A(r) = \{t \in D | antecedent(t)\}$, (i.e. the set of records defined by the antecedent)
$B(r) = \{t \in D | consequent(t)\}$ (i.e. the set of records defined by the consequent)
$C(r) = \{t \in D | antecedent(t) \wedge consequent(t)\}$.

The support, $sup(M)$, for any conjunction, M, of ATs, is the number of records which satisfy M. Given a rule, r, we designate the antecedent of the rule r^a and the consequent r^c. Then, the support for the antecedent, $sup(r^a) = |A(r)| = a$ and the support for the consequent, $sup(r^c) = |B(r)| = b$ (i.e. the cardinality of the target class).

The *support* for r, $sup(r)$, is defined as $sup(r^a \wedge r^c) = |C(r)| = c$.

The *confidence* (also known as *accuracy*) of r, $conf(r)$, is defined as

$$conf(r) = \frac{sup(r)}{sup(r^a)} = \frac{c}{a}.$$

The support for a rule may be expressed as a proportion of the support for the consequent, this measure is referred to as *coverage*. In nugget discovery, it is often convenient and more intuitive to use this measure in place of rule support as we are interested in rules that represent a strong description of a predefined class.

The coverage of r, $cov(r)$, is defined as

$$cov(r) = \frac{sup(r)}{sup(r^c)} = \frac{c}{b}.$$

A strong rule may be defined as one that meets certain confidence and coverage thresholds. Those thresholds are normally set by the user (or the data owner) and are based on domain or expert knowledge about the data. Strong rules may be considered interesting if they are found to be novel and useful.

2.3 CC-optimality

The complete set of strong rules that underlie a database may be very large and many rules may be very similar. In order to address this problem and present a concise set of rules, the cc-optimal (coverage-confidence optimal) subset of rules was proposed in [3]. The cc-optimal set is a set of rules where each rule is optimal with respect to coverage and confidence.

A partial ordering, $\leq_{cc}$, is defined on rules where $r_1 <_{cc} r_2$ if and only if:

Condition A- $cov(r_1) \leq cov(r_2) \wedge conf(r_1) < conf(r_2)$,
Condition B- $cov(r_1) < cov(r_2) \wedge conf(r_1) \leq conf(r_2)$.

Also, $r_1 =_{cc} r_2$ in the partial ordering if $cov(r_1) = cov(r_2) \wedge conf(r_1) = conf(r_2)$.

It is easy to see how the concept of cc-optimality fits in with a multi-objective approach as the cc-optimal set of rules are those that lie in the Pareto optimal front when the objectives to be optimised are confidence and coverage. In other words, if $r_1 <_{cc} r_2$, rule r_2 is said to dominate r_1. r_2 is said to be Pareto optimal (or cc-optimal) if and only if there is no other rule, r_i that dominates r_2.

Hence, we can use a number of algorithms, including multi-objective meta-heuristics, to search the space of all possible rules and extract those that are cc-optimal, or close to cc-optimality.

The problem with cc-optimality as a criterion to choose rules is that, if two rules have the same confidence and coverage, only one of them may be kept in the cc-optimal set. However, the two rules could be very different either in attribute space or they could describe a different subset of records. In such cases it could be argued that the cc-optimal rule set may not be suitably representative of the most interesting rules underlying the database. In this paper, we investigate this claim, since that has been a doubt cast over our previous research. Intuitively, if this is the case, one would expect to find at least some rules of similar coverage and accuracy which cover different sets of records in the final Pareto front.

3 NSGA-II for nugget discovery

The algorithm NSGA-II [8] was applied to the problem of finding cc-optimal rules [7]. NSGA-II uses non-dominated sorting as a mechanism for introducing elitism in the search. It also uses a crowding operator to ensure diversity of solutions within the Pareto front.

A population of solutions is created and sorted into fronts according to non-domination with respect to the multiple objective functions. Solutions within the same front are then sorted according to crowding distance. Solutions that are non-dominated (i.e. those that belong to the first front) are given priority for reproduction. If two solutions are non-dominated, the solution that is least crowded has a higher priority for reproduction. The cycle of selection and reproduction using crossover and mutation creates a new pool which is merged with the initial pool, and the process is repeated again over a number of generations.

3.1 Implementation details

The solution to be represented is a conjunctive rule or nugget following the syntax described in section 2.1. A binary string is used for this as follows. The first part of the string is used to represent the numeric fields or attributes. Each numeric attribute is represented by a set of Gray-coded lower and upper limits, where each limit is allocated a user-defined number of bits, p ($p = 10$ is the default). There is a scaling procedure that transforms any number in the range of possible values using p bits $[0, 2^p - 1]$ to a number in the range of values that the attribute can take.

The second part of the string represents categorical attributes, with each attribute having v number of bits, where v is the number of distinct values (or

the number of labels) that the categorical attribute can take. If a bit assigned to a categorical attribute is set to 0 then the corresponding label is included as an inequality in one of the conjuncts.

In this work, we assume that the consequent of the rule is fixed and of the form of an attribute test, AT, on the class label, hence it does not need to be represented as part of the rule.

Random initialisation proved ineffective in a number of experiments. A more effective approach for this kind of problem is to use mutated forms of the default rule as initial solutions. The default is the rule in which all limits are maximally spaced and all labels are included. In other words, it predicts the class without any pre-conditions. This is the approach used in all experiments reported here, with a mutation probability of 1% which was set after parameter experimentation.

To evaluate a solution, the bit string is first decoded into a rule, and the data in the database, which has been previously loaded in memory, is scanned. For each record the values of the fields are compared against the rule, and the class is also compared. The counts of c (support for the rule) and a (support for the antecedent) are updated accordingly. The counts of b (support for the consequent) and d (cardinality of the database) are known from the data loading stage. Once all data has been examined, the measures of strength used as the objectives, in this case the coverage and confidence, are calculated for each nugget.

Parameter experimentation established the use of one-point crossover, with a crossover rate of 80%. The size of the population was set at 100 solutions.

The output of this algorithm can either be the best solutions found through the search (of which we keep a copy) or the final parents.

The rules obtained by this approach are a subset of the rules underlying the database and should be a good representation of the cc-optimal set. There is, of course, no way of guaranteeing optimality with any heuristic technique. In practical experimentation, however, when the implementation was tested on a number of standard databases against an algorithm (ARAC [18]) capable of finding all rules in the cc-optimal set, it performed very well and was shown to find a good approximation to the Pareto front of this set in each database tested (for details see [7]). The spread of solutions in the Pareto optimal front with respect to the objectives to be optimised appeared to be good, but it was difficult to know how close some of those rules may have been to one another in real terms, and which other rules (perhaps interesting rules) may have been side-stepped in the search for cc-optimal rules. In order to assess these factors, the set of rules produced by the algorithm needs to be analysed in terms of similarity of rules within the set.

4 Rule Dissimilarity

Rule dissimilarity can be measured in a number of ways. First, one could look at the specific syntactic difference between two rules, i.e. the difference in at-

tribute space. This may be considered as testing the appearance of two rules. Rules that appear to be different in attribute space may represent interesting concepts for the user. For example, they may represent different (alternative) characterisations of the same subset of records, perhaps by using attributes that are correlated.

On the other hand, we can simply examine the subset of records that is characterised by a rule. Rules that characterise different (non-overlapping) subsets of records may be considered dissimilar.

In the case of nugget discovery, we may be interested in encouraging diversity of solutions in terms of both their appearance and the population they characterise. However, the first concern must be to characterise as much of the target class as possible, so we will start by looking at dissimilarity in the sets of records that 'match' different rules. We leave rule appearance as an issue for further research.

4.1 Dissimilarity measure

When trying to define the set of records that match a particular rule, it is possible to use $A(r)$, i.e. the set of records that match the antecedent of the rule r. Another possibility is to use $C(r)$, i.e. the set of records that match both the antecedent and consequent of the rule r. In all experimentation conducted we use the set $C(r)$ to calculate rule dissimilarity. In terms of their use for calculating distances, they are interchangeable by replacing C by A in the equation below.

If r_1 and r_2 are two arbitrary rules, we can define a dissimilarity measure as

$$\begin{aligned} d(r_1, r_2) &= |C(r_1) \cup C(r_2)| - |C(r_1) \cap C(r_2)| \\ &= |C(r_1) - C(r_2)| + |C(r_2) - C(r_1)| \end{aligned}$$

This initial measure provides a count of records matching one and only one of the two rules. Dividing this measure by the number of records in the database, $|D|$ gives the simple matching coefficient [12].

Alternatively, we can use the Jaccard coefficient [11] on the sets of support for the rules and define

$$n(r_1, r_2) = d(r_1, r_2)/|C(r_1) \cup C(r_2)|.$$

Intuitively, two rules that are mutually supported by a thousand records and differ over only six are more similar than two rules that are mutually supported by no records and differ over five. Hence the Jaccard coefficient may be a better measure of rule dissimilarity for our purposes.

4.2 Clustering of rules

In order to understand the results of applying distance metrics to the rules obtained by the NSGA-II algorithm, we proceed to apply a clustering algorithm to cluster similar rules together. This should help in the presentation of results.

Our recent research on suitable approaches to clustering rules presented a number of possible clustering algorithms [16, 17] for rule clustering. Here, we use two of the algorithms: Partitioning Around Medoids (PAM) and AGlomerative NESting (AGNES). Both algorithms work on a pre-prepared dissimilarity matrix which contains the distance between each pair of rules calculated using the Jaccard coefficient. The resulting clusters are based on these distances between rules, hence rules that appear in the same cluster should apply to the same or overlapping subsets of records.

5 Experimentation

For our experimentation, we are using the *Adult* dataset from the UCI repository [13]. Initial experiments used a set of rules produced by the NSGA-II algorithm for nugget discovery. The algorithm was applied to produce rules to describe the class "Income $>$ 50 k". The AGNES clustering algorithm was applied to cluster the best rules found through the search. The clustering used the Jaccard coefficients on the sets of support for the rules. The hierarchy of rules produced was then cut at a point that lead to 8 clusters. The PAM clustering algorithm was also used to cluster the best rules obtained. The results are presented in figures 1 and 2 respectively. Both graphs show that rules that are close in terms of the values of their objective functions (coverage and confidence) are also close in terms of the set of records that support them. The clustering tends to be neatly distributed on the Pareto front, with little overlap of clusters within the front. Hence in the majority of cases, selecting sets of rules with similar coverage and confidence tends to deliver similar rules that describe a similar subsets of records. As we examine rules of different coverage/confidence we are likely to be finding rules that describe different subsets of records. A similar exercise was performed to cluster the rules in the final parent population, and this exhibited exactly the same characteristics. Other sets of rules produced from different databases provided similar clustering behavior.

5.1 Encouraging diversity in terms of support sets

It may be possible to encourage diversity of the rule set by using some measure of dissimilarity of rules as a third criterion to be optimised. However, since dissimilarity can only be measured in the context of other rules in the set, this will result in a less efficient evaluation procedure. Also, the application of dissimilarity to the non-dominated sorting may result in a new partial ordering of rules which does not reflect the requirements of nugget discovery. We consider diversity of rules in terms of support sets as a "second priority" objective, to be achieved once we can guarantee a pool of strong rules.

For our purpose, we decided to experiment with the crowding measure of NSGA-II [8]. The crowding measure in normal operation ensures that the population within the Pareto optimal front is as diverse as possible, so it acts as a secondary criterion for ordering rules. We first analysed the effect of not using a

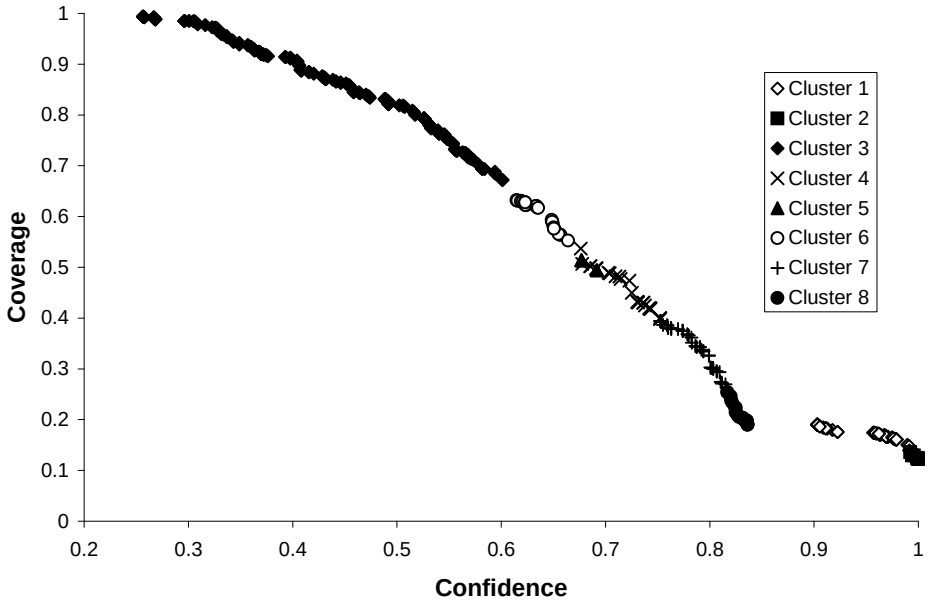

Fig. 1. Clustering of rules for Adult dataset using AGNES - 8 clusters

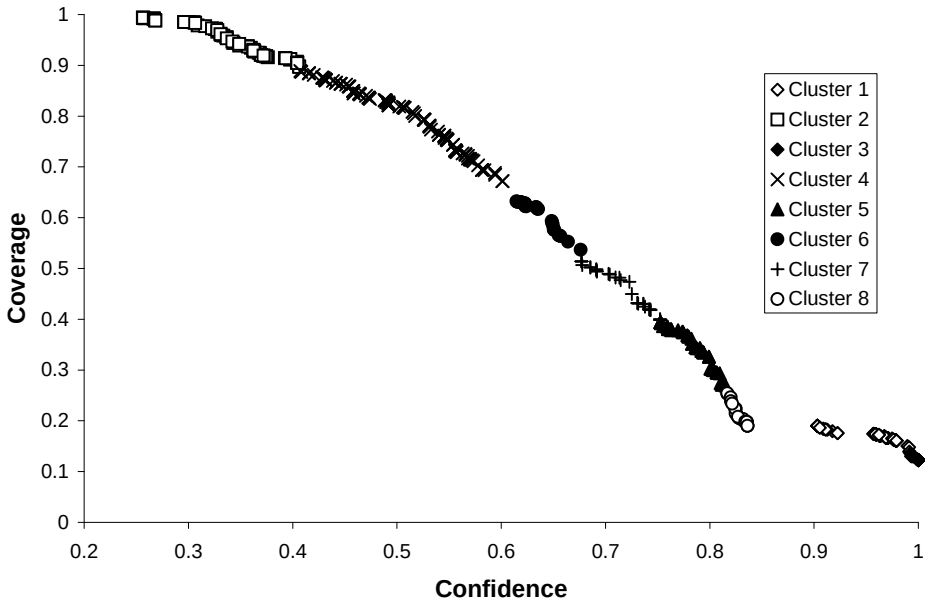

Fig. 2. Clustering of rules for Adult dataset using PAM - 8 clusters

crowding measure for this application of NSGA-II for rule induction, so within the algorithm all rules where considered to be equally crowded at all times. The Pareto front obtained by running our algorithm on the Adult data with equal crowding is shown in figure 3. In this section, we show the solutions in the final population as well as the best solutions found during the search. To aid the analysis of results, the approximation to the Pareto front obtained by the ARAC algorithm is also plotted as a line. ARAC had to impose some restrictions on the search due to the size of this database, hence it can only give us an approximation to the true Pareto front. However, the 976 rules found by this algorithm represent the best basis for comparison of results.

A similar graph showing the results of using the crowding measure as described in the original NSGA-II algorithm is also presented in figure 4.

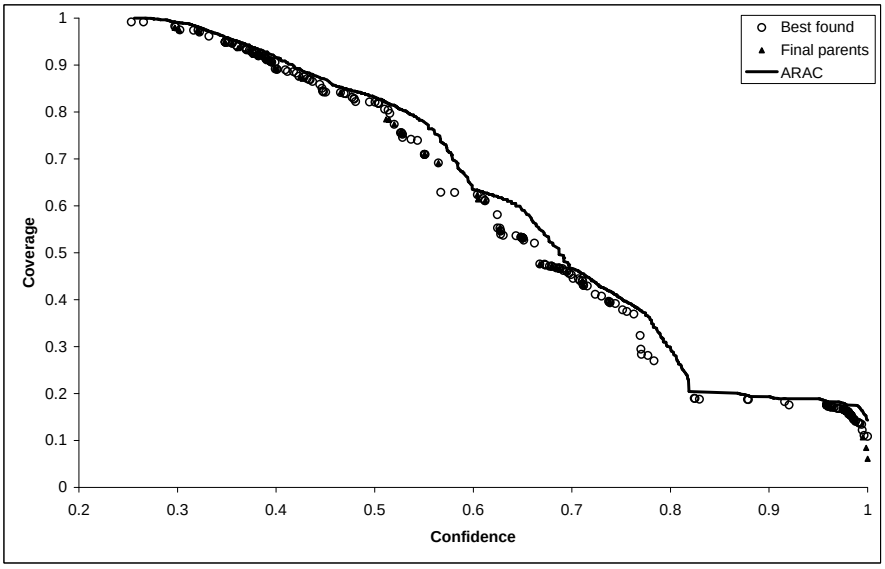

Fig. 3. Rules obtained with NSGA-II for the Adult database with equal crowding

It can be seen that using the standard crowding measure proposed by the NSGA-II algorithm produces a much better spread of solutions in the Pareto optimal front, both in terms of the best solutions found as well as in terms of the final parents.

A number of approaches were tried to adapt the crowding measure to encourage diversity within the rule set in terms of support sets. We only discuss the most successful approach here.

The Jaccard dissimilarity measure was calculated for each pair of rules using the support set, C, for calculations. The crowding measure was then modified to be a count of the number of rules within a certain threshold distance, T, from

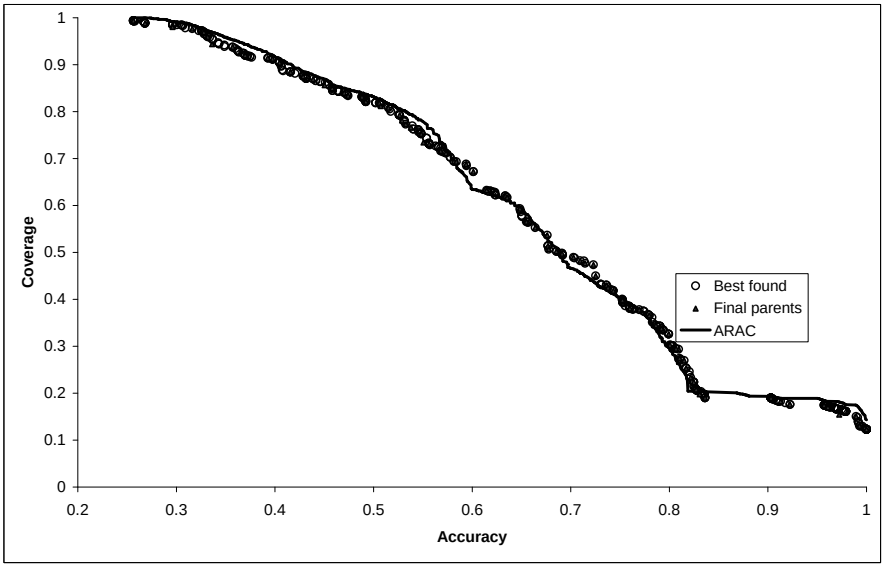

Fig. 4. Rules obtained with NSGA-II for the Adult database with standard crowding

the rule being examined, according to the Jaccard dissimilarity measure. Three values were tried for this distance threshold: 0.1, 0.2, 0.5. In the standard crowding measure, the crowding distance is calculated by using solutions within the same front. In our proposed approach it is possible to use the whole population to calculate the new crowding distance, as well as the solutions within the same front only. We experimented with both options.

The results are shown in figure 5. The left hand column of graphs shows the results using crowding based on the distance of the solutions in the front only, whereas the right column represents the results using the whole population. The first row of graphs represents the threshold value $T = 0.5$; for the second row $T = 0.2$; for the final row $T = 0.1$.

When crowding is calculated using threshold distances of 0.2 and 0.1 the spread of best found solutions seems to cover most of the Pareto front. However, the final parent solutions show less coverage of the front. A threshold of 0.5 produces poor coverage of the Pareto front. Some of these observations are expected: since we are no longer encouraging diversity as per the objective functions, some of that diversity will be lost in the population.

For each of the sets of final parents created with different crowding mechanisms (since there are always the same number of parents in each set), we calculate the sum of distances between rules. This is reported in table 1. The sum of distances increases in all cases with the new crowding mechanism with respect to standard crowding. The sum of distances also increases for equal crowding

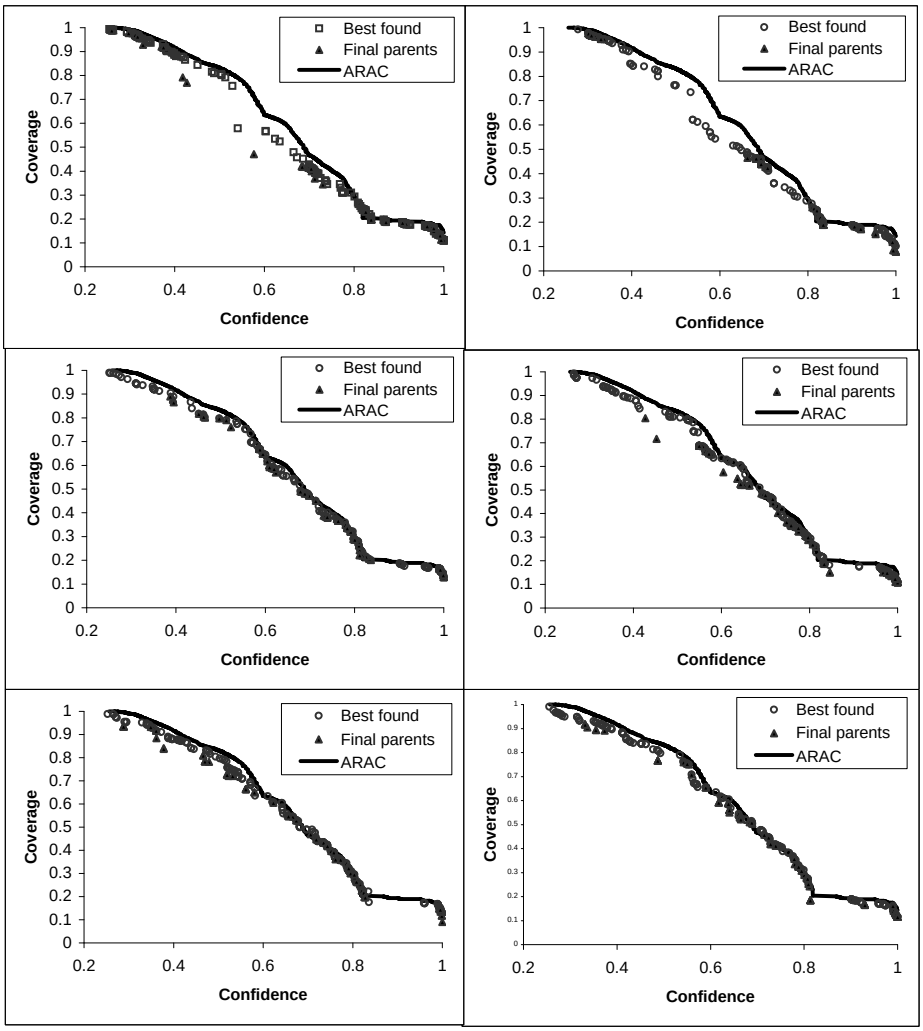

Fig. 5. Rules obtained with NSGA-II for the Adult database with new crowding mechanisms. From top to bottom graphs represent thresholds of 0.5, 0.2 and 0.1 respectively.

Table 1. Sum of distances between rules using different crowding mechanisms

Approach	Sum	Approach	Sum
Standard	4,570.51	Equal	5,057.4
Crowding based on distances by front		Crowding based on distances by population	
0.1	5,081	0.1	5,510.52
0.2	5,007.25	0.2	5,301.09
0.5	5,776.27	0.5	5,784.05

but the coverage of the front is poorer, with some areas not represented in the final parent population.

To further assess the increase in diversity of solutions in terms of support sets, we use PAM to cluster some of the rules produced and observe the degree of overlap. For this purpose, we choose the rules produced using the new crowding measure with the whole population and a threshold of 0.2. We use the support set, C, for dissimilarity calculations and clustering. We feed the clustering algorithm the set of best rules found. As before, we aim to produce 8 clusters for comparative purposes. The results of this process are shown in figure 6. There is some overlap now in clusters 5 and 2, so in the high confidence / low coverage area of the Pareto front we have managed to increase the diversity of solutions according to support sets. It seems reasonable that this is the area in which we have created diversity with our approach, as high coverage rules would apply to a high percentage of records within the population, and therefore finding alternative high coverage rules that apply to different sets of records is unlikely. Within the low coverage rules, there is obviously more scope for creating diversity as we have managed to encourage this with our new crowding measures. Hence we may now be able to present a set of rules which includes more diverse (in terms of support sets) rules of high accuracy.

6 Conclusions and further Research

In this paper we have combined a number of innovative approaches to rule induction, exploiting the power of multi-objective metaheuristics to obtain interesting rule sets. In particular, we have experimented with the crowding mechanism in NSGA-II to improve the quality of rule sets obtained. We have also assessed the quality of rule sets obtained by using innovative approaches to cluster rules according to dissimilarity measures. This combined approach, when fully tested, may become a very powerful tool for rule induction.

We have shown that the rule sets obtained by NSGA-II in the standard implementation do not contain many cases of rules that are close in the objective space but far apart in terms of their support sets.

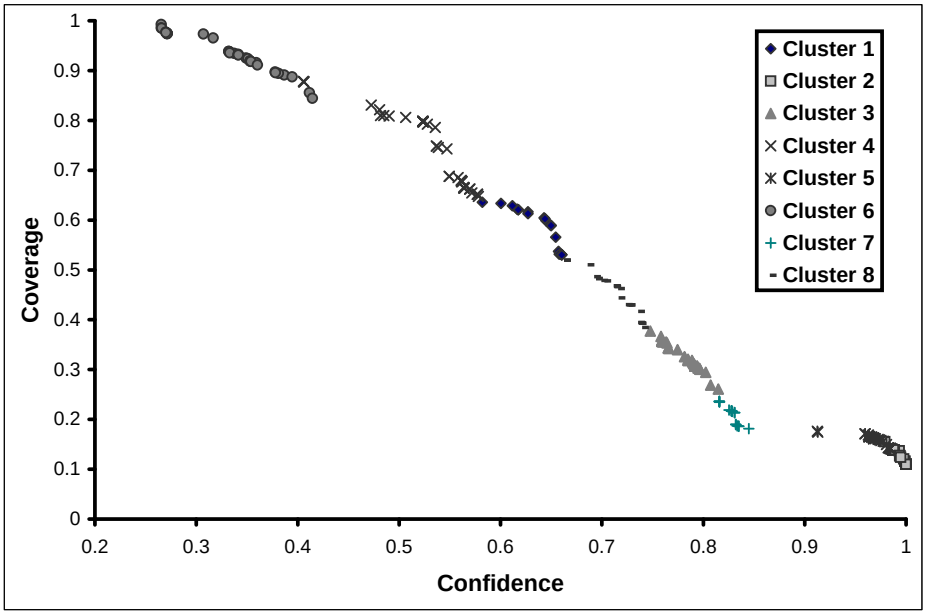

Fig. 6. Clustering of rules for Adult Database produced with dissimilarity crowding (PAM) - 8 clusters

We have created a modification of NSGA-II by introducing the concept of rule dissimilarity in the crowding measure. This has allowed us to increase the diversity of rules in some areas of the Pareto front in terms of support sets.

The work presented here is only in its initial stages, and there is scope for extending it and improving in a number of ways. More experimentation is required to draw conclusive results. As further work, other measures of rule dissimilarity may be used to encourage diversity of the rule set. This may include considering the appearance of rules. Modifications of the NSGA-II algorithm to include our criteria may not be limited to the crowding measure, but may be more drastic using different selection criteria altogether. Experimentation with other multi-objective metaheuristics may also be beneficial.

References

1. S. Ali, K. Manganaris and R. Srikant. Partial classification using association rules. In D. Heckerman, H. Mannila, D. Pregibon, and R. Uthurusamy, editors, *Proceedings of the Third Int. Conf. on Knowledge Discovery and Data Mining*, pages 115–118. AAAI Press, 1997.
2. R. Bayardo and R. Agrawal. Constraint based rule mining in large, dense databases. *Data Mining and Knowledge Discovery Journal*, 4:217–240, 2000.

3. R. Bayardo and Agrawal R. Mining the most interesting rules. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining, (KDD 99)*, pages 145–152. AAAI Press, 1999.
4. L. Breiman, J. H. Friedman, R. A. Olshen, and C.J. Stone. *Classification and regression trees*. Wadsworth, Pacific Grove, CA, 1984.
5. P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–284, 1989.
6. W. Cohen. Fast effective rule induction. In *Proceedings of Twelfth International Conference on Machine Learning (ICML-95)*, pages 115–123. Morgan Kaufman, 1995.
7. Beatriz de la Iglesia, Graeme Richards, Mark S. Philpott, and Vic J. Rayward Smith. The application and effectiveness of a multi-objective metaheuristic algorithm for partial classification. *European Journal of Operational Research*, 2004, to appear.
8. Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, J. J. Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.
9. A. A. Freitas. On objective measures of rule surprisingness. In *Principles of Data Mining and Knowledge Discovery (Proc. 2nd European Symp., PKDD'98. Nantes, France),Lecturer Notes on Artificial Intelligence, 1510, 1-9*. 1998.
10. A. A. Freitas. On rule interestingness measures. *Knowledge-Based Systems Journal*, 12(5-6):209–315, 1999.
11. P. Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise de la Sciences Naturelles*, 37:547–579, 1901.
12. Leonard Kaufman and Peter J. Rousseuw. *Finding Groups in Data: An introduction to Cluster Analisys*. Wiley Series in probability and mathematical statistics. John Wiley and Sons Inc., 1990.
13. C. J Merz and P. M. Murphy. UCI repository of machine learning databases. Univ. California, Irvine, 1998.
14. G. Piatetsky-Shapiro. *Discovery, Analysis, and Presentation of Strong Rules*, chapter 13, pages 229–248. AAAI/MIT Press, 1991.
15. J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.
16. A. P. Reynolds, G. Richards, B. de la Iglesia, and V. J. Rayward-Smith. Nugget clustering: A comparison of partitioning and hierarchical clustering algorithms. *TBA*, In preparation 2004.
17. A. P. Reynolds, G. Richards, and V. J. Rayward-Smith. The Application of K-medoids and PAM to the Clustering of Rules. In *Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'04). Lecture Notes in Computer Science No. 3177*, pages 173–178. Springer-Verlag, 2004.
18. G. Richards and V.J. Rayward-Smith. The discovery of association rules from tabular databases comprising nominal and ordinal attributes. *Intelligent Data Analysis*, 9(3), 2004.