Rule Induction Using Multi-Objective Metaheuristics: Encouraging Rule Diversity

Alan Reynolds and Beatriz de la Iglesia

Abstract— **Previous research produced a multi-objective metaheuristic for partial classification, where rule dominance is determined through the comparison of rules based on just two objectives: rule confidence and coverage. The user is presented with a set of descriptions of the class of interest from which he may select a subset. This paper presents two enhancements to this algorithm, describing how the use of modified dominance relations may increase the diversity of rules presented to the user and how clustering techniques may be used to aid in the presentation of the potentially large sets of rules generated.**

I. Introduction

Earlier work by the authors [1], [2], [3] explored the application of multi-objective metaheuristics to the problem of partial classification [4], otherwise known as nugget discovery [5]. This problem is the search for rules that represent 'strong' or 'interesting' descriptions of a specified class, or subsets of the 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 they may gain a competitive advantage.

The 'interest' of a rule is subjective, meaning that at least some of the process of finding interesting rules must be controlled by the user. However, in order to make the user's job tractable, it is necessary to automate the removal of the majority of the uninteresting partial classification rules. This task must balance the risk of presenting too many rules to the user against the risk of removing rules that might be of interest. One approach to performing this task is to establish a partial ordering of the rules based on their confidence and coverage. In fact the best rules, according to a number of different measures of rule interest commonly used in the literature, are optimal according to this partial ordering [6]. Multi-objective algorithms applied to the task of finding these rules have found very good approximations to this optimal subset very efficiently [1], [2], [3].

The ultimate objective of multi-objective algorithms is to guide the user's decision making, through the provision of a set of solutions that have differing trade-offs between the various objectives. However, while in most multi-objective problems the user selects one solution, here it is likely that the user will wish to select a set of rules from those provided. Furthermore, the user is likely to want to select rules that differ significantly from each other. This implies an additional objective: the rules presented to the user should be diverse in structure as well as in objective values. Consideration of these aims leads to two questions:

1) If dominance is determined solely through the confidence and coverage of rules, is it possible that some rules that the user may find interesting are dominated and hence removed?
2) If the algorithm produces a non-dominated set containing a large number of rules, how should these rules be presented to the user, so as to aid his decision making?

In answer to the first question, the user may find a rule interesting because he finds it novel, either when compared with other rules selected or based on previous knowledge of the data. It will be shown that novel rules of reasonable confidence and coverage may indeed be dominated. Methods will be developed for relaxing the dominance relation in such a way as to prevent this from happening, without reducing the quality of the rules presented to the user.

The authors have already researched methods that provide an answer to the second question, by using clustering algorithms to produce more concise summaries of the rules generated [2], [7], [8]. One of these clustering algorithms will be used to improve the presentation of the rules produced by the multi-objective metaheuristic described in this paper.

The first half of this paper describes the types of rules generated and discusses the different functions and optimality criteria that may be used to evaluate rules. Section II describes the rule types and some simple objective functions, with section III describing simple dominance relations and optimality criteria that may be used when performing multi-objective optimization of rules. Section IV introduces the new, modified dominance relations, describing how these are designed with the aim of encouraging diversity and novelty in the rules produced by a multi-objective metaheuristic.

The second half of the paper describes, in section V, how a multi-objective genetic algorithm is applied to the problem, with results in section VI. The use of clustering to aid in the presentation of the rule-sets generated is discussed in section VII. Finally, section VIII presents some conclusions.

II. Simple Rules

A. Attribute Tests and Simple Rules

The algorithms described in this paper produce partial classification rules of the form

$$antecedent \rightarrow consequent,$$

where both the antecedent and the consequent are constructed from *attribute tests*.

Let Q be a finite set of attributes, which in practice corresponds with the fields in the database. Each $q \in Q$

Alan Reynolds and Beatriz de la Iglesia are with the School of Computing Sciences, University of East Anglia, Norwich, England. (E-mails: ar@cmp.uea.ac.uk and bli@cmp.uea.ac.uk)

has an associated domain, $Dom(q)$. An attribute test (AT), b, consists of an attribute, $at(b) \in Q$, and a value set, $Val(b) \subseteq Dom(at(b))$, and may be written $at(b) \in Val(b)$. A record satisfies this test if its value for attribute $at(b)$ belongs in the set $Val(b)$.

An algorithm may allow only certain types of value sets $Val(b)$. Types of categorical AT are as follows:

- **Value:** $Val(b) = \{v(b)\}$, where $v(b) \in Dom(at(b))$. This may be written $at(b) = v(b)$.
- **Inequality:** $Val(b) = \{x \in Dom(at(b)) : x \neq v(b)\}$, where $v(b) \in Dom(at(b))$. This may be written $at(b) \neq v(b)$.
- **Subset:** $Val(b)$ unrestricted, i.e. any subset of $Dom(at(b))$.

Types of numerical AT are as follows:

- **Binary partition:** $Val(b) = \{x \in Dom(at(b)) : x \leq v(b)\}$ or $Val(b) = \{x \in Dom(at(b)) : x \geq v(b)\}$, where $v(b) \in Dom(at(b))$. In this case, the AT may be written $at(b) \leq v(b)$ or $at(b) \geq v(b)$, respectively.
- **Range:** $Val(b) = \{x \in Dom(at(b)) : l(b) \leq x \leq u(b)\}$, where $l(b), u(b) \in Dom(at(b))$. Here, the AT is written $l(b) \leq at(b) \leq u(b)$.

Each rule created has the same consequent, consisting of just one AT. This defines the class of interest, e.g. the class of motorists who have claimed on their insurance. Rule antecedents are conjunctions of ATs, none of which may share the same attribute as the consequent.

B. Support, Confidence and Coverage

Define the support set, $S(M)$, of any conjunction, M, of ATs to be the set of records which satisify M. Further define the support, $sup(M)$, to be the cardinality of this set, i.e. $sup(M) = |S(M)|$. Given a rule, r, we designate the antecedent of the rule r^a and the consequent r^c.

The *support set* of r, $S(r)$, is defined as $S(r^a \wedge r^c)$.

The *support* for r, $sup(r)$, is defined as $|S(r)|$.

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

$$conf(r) = sup(r)/sup(r^a).$$

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

$$cov(r) = sup(r)/sup(r^c).$$

III. Simple Optimality Criteria and Dominance

There exists a number of methods that aim to eliminate uninteresting rules. Two simple approaches use the concepts of pc-dominance and cc-dominance.

- Rule q is dominated by rule r according to pc-dominance if and only if
$$\begin{array}{c} S(q) \subseteq S(r) \text{ and } conf(q) < conf(r), \text{ or} \\ S(q) \subset S(r) \text{ and } conf(q) \leq conf(r) \end{array}$$
- Rule q is dominated by rule r according to cc-dominance (or sc-dominance) if and only if
$$\begin{array}{c} cov(q) \leq cov(r) \text{ and } conf(q) < conf(r), \text{ or} \\ cov(q) < cov(r) \text{ and } conf(q) \leq conf(r) \end{array}$$

The algorithms discussed in this paper find rules that satisfy certain constraints: ATs may be constrained to be of particular types; there may be a limit on the number of ATs permitted. Within such constraints, it is possible to search for pc-optimal [6] or cc-optimal rules. A pc-optimal (or cc-optimal) rule is one that is not dominated by any other rule that satisfies the constraints.

Bayardo Jr. and Agrawal [6] showed that the best rule, according to a number of different measures of rule interest, is cc-optimal. These measures include the chi-squared value, entropy gain, conviction, lift, gini values etc. This motivated previous research into the use of cc-dominance within a multi-objective metaheuristic, producing sets of non-dominated rules. However, there is the concern that using cc-dominance may result in the removal of interesting rules. A rule may be dominated, yet have reasonable confidence and coverage. If it also differs significantly from the non-dominated rules presented to the user, either in appearance or in the records matched, then it may be of interest.

While the use of cc-dominance may result in interesting rules being dominated, using pc-dominance instead, in all-rules algorithms [9], [10] for example, results in the generation of very large sets of non-dominated rules. Different methods must be used if a better balance between rule-set size and the risk of eliminating interesting rules is to be achieved.

IV. Using Modified Dominance Relations to Encourage Novel Rules

A rule may be considered to have 'novelty' if it either provides information not provided by other rules in the population or matches records not matched by the other rules. One approach to encouraging such novelty would be to explicitly consider it as a third objective. However, there are problems with this approach:

- Since the novelty of a rule depends upon the other rules present in the population, it will change as other rules are introduced or removed. The success of novel rules may ultimately cause their own destruction, since, as the number of similar rules increases, novelty drops until such rules are no longer novel and are dominated.
- Recalculating the novelty of every rule in the population, in each generation, through comparison with each other rule, would be computationally costly.
- Some rules would survive due only to their novelty, regardless of low confidence and coverage, since a rule with maximal novelty would be non-dominated.

The approach considered here involves modifying the dominance relation without explicitly adding a third objective. When rule q would normally be dominated by rule r, the difference between the two rules is considered. If rule q provides enough novelty with respect to rule r, it is permitted to remain non-dominated. The amount of novelty required depends upon the relative quality of the two rules according to the two prime objectives: coverage and confidence.

Section IV-A describes two methods for determining the novelty of one rule with respect to another, based on the rules' support sets. Section IV-B adapts these methods to determine apparent/syntactic novelty. Section IV-C describes the degree to which one rule dominates another, according to cc-dominance. These concepts are combined in section IV-D to produce a transitive dominance relation that allows rule novelty to be valued during the search for interesting rules.

A. Rule Novelty: Support Sets

Two methods have been developed for evaluating the 'novelty' of rule q, with respect to rule r, based on the support sets of the rule. If C is the set of records in the class of interest, then:

$$nov_a(q,r) = \tfrac{|S(q)-S(r)|}{|C|}, \quad nov_r(q,r) = \tfrac{|S(q)-S(r)|}{|S(r)|}.$$

The first of these is referred to as the absolute novelty and is the probability that a member of the class of interest matches rule q but not r.

The second, relative novelty, is the absolute novelty divided by the coverage of rule r. To obtain a given amount of relative novelty, rule q must obtain the support of many new records if r has a large support set already, but only a few if r matches only a few records.

Note that both forms of novelty can be shown to satisfy the following:

$$nov(p,q) + nov(q,r) \geq nov(p,r). \qquad (1)$$

This will be used in section IV-D to prove the transitivity of a modified dominance relation.

B. Apparent Rule Novelty

The novelty measures described above assign novelty only when there are records that support rule q but not r. If q matches only a subset of those records matched by r, the novelty value is zero, regardless of how different the rules might appear.

To measure apparent (or syntactic) novelty, a model of the user's knowledge of the data may be used to produce his estimate of either absolute or relative novelty. Let Q be the event that a record matches the antecedent of rule q, let R be the event that a record matches the antecedent of rule r and let C be the event that a record is in the class of interest. The absolute novelty measure can be written

$$nov_a(q,r) = P(Q \cap \overline{R}\,|C).$$

Suppose a user with no knowledge of the relationships between the non-class fields were to estimate this probability. Such a user might start by assuming independence of the non-class fields. Let $q = \bigwedge_i q_i$, where i represents a field of the database and q_i is the conjunction obtained by collecting those ATs that concern field i. Let Q_i be the event that a record matches q_i. Then:

$$\begin{aligned} nov_a(q,r) &= P(Q\,|C) - P(Q \cap R\,|C) \\ &= P(\textstyle\bigcap_i Q_i\,|C) - P(\bigcap_i Q_i \cap \bigcap_i R_i\,|C) \\ &\approx \textstyle\prod_i P(Q_i\,|C) - \prod_i P(Q_i \cap R_i\,|C). \end{aligned}$$

If the user knows the distribution of the non-class fields over the records of interest, he knows $P(Q_i\,|C)$ and $P(Q_i \cap R_i\,|C)$ and can use the above to estimate the absolute novelty of rule Q with respect to rule R. Therefore, this estimate is used as a measure of the apparent absolute novelty:

$$app_a(q,r) = \prod_i P(Q_i\,|C) - \prod_i P(Q_i \cap R_i\,|C).$$

The same reasoning can also be used to define apparent relative novelty. Both measures of apparent novelty can be shown to satisfy inequality 1.

It is possible to create alternative measures of apparent novelty by assuming that the user has even less knowledge of the data. However, since rules are presented to the user with confidence and coverage statistics, we suppose that the user can get a pretty good feel for the distribution of the fields over the records of interest. Furthermore, a user is likely to perform some analysis of these field distributions before applying data mining tools. Therefore $app_a(q,r)$ is used to measure apparent absolute novelty in the experiments of this paper.

Note that this assumes that the user knows nothing about dependencies between non-class fields. Given the rules

- if pensioner then has savings
- if age $\geq$ 65 then has savings

this measure indicates that the second rule has novelty when compared with the first, though the user would not consider it to be novel since he knows the relationship between age and pensions. If apparent novelty is to take this into account, this knowledge would need to be incorporated into the model of the user's knowledge of the data before estimating $nov_a(q,r)$.

C. The Dominance Margin

If rule q dominates rule r, it is possible to calculate the 'dominance margin', dm. This is defined to be the smallest amount by which one of the objective values for r must be improved in order that r no longer be dominated.

$$dm(q,r) = \min(conf(q) - conf(r), cov(q) - cov(r)).$$

It is easily shown that

$$dm(p,q) + dm(q,r) \leq dm(p,r). \qquad (2)$$

D. Modifying the Dominance Relation

Having defined novelty and dominance margin, it is possible to modify the dominance relation to encourage novel rules, provided the amount of novelty, multiplied by a constant, λ, exceeds the dominance margin. So rule q dominates rule r if and only if

$$\begin{array}{l} q >_{cc} r \text{ where } >_{cc} \text{ indicates cc-dominance and} \\ \lambda nov(r,q) \leq dm(q,r). \end{array} \qquad (3)$$

Provided the form of novelty used is one of those described in sections IV-A and IV-B, the dominance relation can easily be shown to be transitive. Suppose p is dominated by q, which in turn is dominated by r. Then

$$\begin{aligned} \lambda nov(p,r) &\leq \lambda nov(p,q) + \lambda nov(q,r) \text{ (by 1)} \\ &\leq dm(q,p) + dm(r,q) \text{ (by 3)} \\ &\leq dm(r,p) \text{ (by 2)} \end{aligned}$$

i.e. p is dominated by r. This transitivity is intuitively desirable and simplifies the management of stores of non-dominated solutions.

This dominance relation allows the multi-objective metaheuristic to find novel rules that would otherwise be dominated. The parameter, λ, provides the user with control over how important such novelty is when compared with the usual objectives of confidence and coverage.

V. MULTI-OBJECTIVE METAHEURISTICS

Most multi-objective metaheuristics use the concept of domination. Given a problem with a number of possibly conflicting objectives, c_1 to c_n, to be maximized, solution s_1 is dominated by solution s_2 if and only if

$$c_i(s_1) \leq c_i(s_2) \; \forall i \in \{1, \ldots, n\},$$
$$\exists i \in \{1, \ldots, n\} : c_i(s_1) < c_i(s_2).$$

If rule coverage and confidence are the objectives to be maximized, this is the same as the definition of cc-dominance.

Many modern multi-objective metaheuristics require either no changes or only minor modifications to use the modified dominance relations, rather than the standard form of dominance above. The multi-objective genetic algorithm, NSGA II, used in this paper requires only a change in the form of crowding measure used.

A. NSGA II

NSGA II [11], [12], outlined in figure 1, is a multi-objective genetic algorithm that has previously been applied to rule induction [2], [3]. Note that the algorithm requires solutions to be compared on 'fitness', which is performed as follows:

```
Create an initial population, $P$, of $p$ rules;
Sort $P$ according to fitness;
$gen := 0$;
while ($gen < maxGen$)
    $numChild := 0$;
    while ($numChild < p$)
        Select 2 parents using binary tournament selection;
        Create 2 children using crossover and mutation;
        $numChild := numChild + 2$;
    endwhile
    Add the children produced to $P$;
    Sort according to fitness;
    Retain the best $p$ rules;
endwhile
```

Fig. 1. Basic pseudo-code for NSGA II.

- Rule q is fitter than rule r if q has better rank.
- If rules q and r have equal rank, q is considered fitter if it is less crowded.

The rank of each rule is calculated via a technique called non-dominated sorting. Non-dominated solutions form the first 'front' and are given a rank of one. Upon removal of these rules, new solutions become non-dominated, forming a second front of solutions of rank two. This process is continued until each solution has been assigned a rank.

The standard crowding method in NSGA II is applied to one front at a time, using only the objective values of the solutions. To obtain 'objective distances' in objective i, the front is first sorted according to this objective. The objective distance for a solution is the difference, in objective i, between its neighbours in the list. The overall crowding distance is given by the sum of the objective distances. If a solution ever appears at an end of the list it is assigned a large crowding distance to indicate no crowding.

This approach works well when using a standard definition of dominance. For example, when using cc-dominance the confidence of rules within a front decreases monotonically as the coverage increases, as shown in figure 2. The crowding distance of solution s is simply half the perimeter of the box shown, providing a good measure of how uncrowded a rule is within its front. However, when the modified dominance relation is used, confidence need not decrease monotonically as coverage increases within a front. The crowding measure no longer measures the space around solution in the objective space and so this method is no longer appropriate.

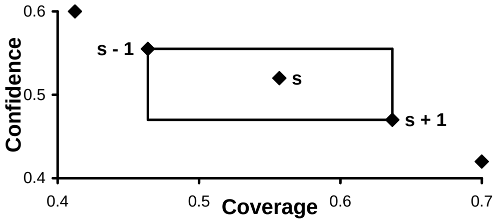

Fig. 2. Standard NSGA II crowding.

The approach used here is similar to the standard approach in two respects: it is based solely on objective values and is performed on a front-by-front basis. Rules in the front are selected in random order. Upon selection, the crowding distance of a rule is given by the distance to the closest rule *that has not yet been selected* (figure 3). If two rules are nearly identical, but are far from the other rules in the front, this method assigns a small crowding distance to one of the rules, possibly leading to its removal from the population. However, the other rule is assigned a large crowding distance and will survive. This prevents sections of the front from being entirely eliminated through crowding.

B. Rule Representation and Manipulation

Previous work [2], [3] into the application of NSGA II to rule induction used a fixed length bit-string representation for the rules. However, such a representation is inflexible: it is difficult to handle all the constraints associated with the different AT types and the AT limit. The handling of alternative AT types is necessary in order to be able to perform fair comparisons with other work. Furthermore, the representation of numeric ATs is unnatural, requiring gray-coded integers that must be translated into values in the range of the field. An alternative representation is used here,

```
Take an array of rules $r$, of size $n$;
Shuffle $r$ at random;
for $i$ from 1 to $n$
    $distance[i] := MAX$;
    for $j$ from $i+1$ to $n$
        $d := |conf(r[i]) - conf(r[j])|$
                    $+ |cov(r[i]) - cov(r[j])|$;
        $distance[i] := \min(distance[i], d)$;
    endfor
endfor
```

Fig. 3. Calculating crowding distances.

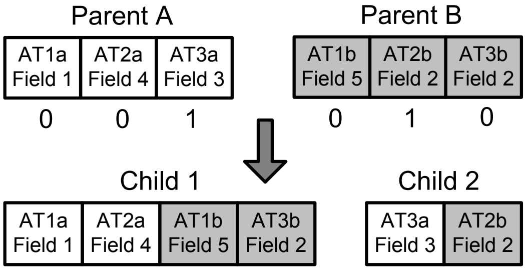

Fig. 4. Uniform crossover applied on the variable length rule representation.

providing flexibility in the choice of AT types and simple handling of constraints on the number of ATs in a rule.

In the new representation, a rule is stored as an array of ATs. Values that occur in both the numeric and categorical fields of the training set are stored in reference arrays. Indices into these arrays are then used in the representation of the ATs. Both mutation and crossover occur on an AT by AT basis. Each type of AT is represented and mutated as follows:

- **Categorical value:** Represented by the categorical field number and the category index. A mutation changes the category index to a randomly selected value.
- **Categorical inequality:** As above.
- **Categorical subset:** Represented by the categorical field number and a bit-string indicating which categories are permitted. A mutation either adds a category to the permitted list or removes one by flipping a bit.
- **Numeric binary partition:** Represented by the numeric field number, the index of the bound value and a flag indicating the type of bound. A mutation changes the index of the bound by up to 20% of the number of values that occur in the database, while ensuring that the AT does not become trivial or impossible to satisfy. The type of the bound is not changed
- **Numeric range:** Represented by the numeric field number, the index of both the lower and upper bound and two booleans indicating whether each bound is present. A mutation changes a valid bound in the same way as for a binary partition AT.

Mutations may also remove an AT entirely or add a new one.

As it is possible to have more than one AT for a given field, a variable length representation is used. Uniform crossover is applied as shown in figure 4. Here, each AT is equally likely to be assigned to either child. The randomly generated bits shown indicate which child should take the associated AT.

After the application of crossover and mutation, rules are simplified if possible, for example by removing redundant ATs. Finally, if a simplified rule exceeds the AT limit, further ATs are removed at random.

VI. Results

A. Using cc-Dominance

To begin, experiments were performed using the standard cc-dominance. Results were compared with those obtained using an all-rules algorithm [9], [10] modified to find cc-optimal rules. We report on results obtained using two databases from the UCI machine learning repository [13]: the Adult Database, with the class of interest being those people who earn more than 50,000 dollars and the Contraceptive Method Choice Database (CMC), with the class of interest being those who do not use contraception.

In order to provide a fair comparison, both algorithms were applied with the same constraints:

- ATs on categorical fields were required to be value or inequality ATs,
- ATs on numeric fields were required to be binary partition ATs,
- The rule antecedent could have no more than six ATs.

All timings reported were obtained on an Intel Pentium 4, 3.0 GHz with 1GB of RAM.

When applied to the CMC database, the all-rules algorithm found 47 cc-optimal rules in 2 seconds. However, before the all-rules algorithm could be applied to the adult database, the ‘final weight’ field had to be removed due to the large number of values present in this field. After removal, the algorithm required over 15 hours, finding 968 cc-optimal rules. (Note that NSGA II can handle the ‘final weight’ field without difficulty. However, to maintain fairness, the ‘final weight’ field is removed in all experiments reported here.)

The results of extensive experimentation indicate that best results for NSGA II can be obtained using a population of 200 rules, with a crossover rate of 0.2 and a mutation rate of 0.2, although the algorithm is not very sensitive to changes to these parameters. (At first sight, the mutation rate seems very high. However, it must be remembered that this is the probability that an *attribute test* is mutated. If a binary representation were used instead, with a more normal looking mutation rate of 0.01, the chance of an AT being mutated would be considerably greater than 0.01 since each AT would be represented by a number of bits.) Initial rules were constructed at random using at most three ATs, though rules produced in later generations were permitted six ATs. After 150 generations, applied to the CMC data, the parent population consisted solely of cc-optimal rules, with all but one of the cc-optimal rules represented. This required a run time of 7 seconds. When applied to the Adult Database, 200 generations produced the results shown in figure 5. Results obtained using NSGA II are very close to those obtained

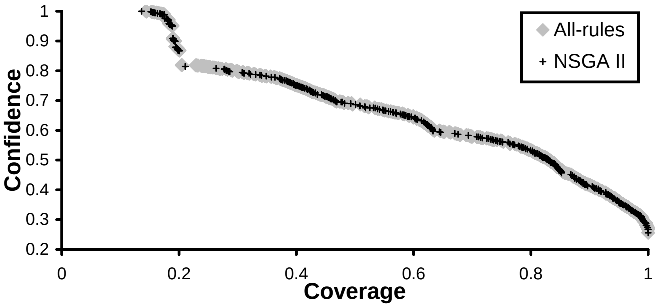

Fig. 5. NSGA II compared with the all-rules algorithm on the Adult Database.

using the all-rules algorithm, despite the fact that NSGA II required only 97 seconds.

Permitting a richer variety of rules leads to greatly increased run times for the all-rules algorithm. However, this is not a problem for the genetic algorithm. By using the subset categorical AT and placing no limit on the number of ATs in a rule, rules of even higher quality can be obtained, according to coverage and confidence.

B. Using the Modified Dominance Relations

To examine the effect of the modified dominance relations, the algorithm was applied to the Adult Database using absolute novelty (both support set based and apparent) and a range of values for λ (the novelty weight). Figures 6–8 show the confidence and coverage of the non-dominated rules produced using support set based novelty, after running the algorithm for 250 generations for three different values of λ. Rules were permitted at most six ATs, of value type for categorical fields and bound type for numeric fields. Again, results are compared with those obtained by the all-rules algorithm using cc-optimality under the same constraints.

These graphs show that as λ increases, novelty becomes a more highly prized characteristic of a rule, with more rules appearing that would previously have been dominated. A few such rules appear in figure 6. Figure 7 shows more, with rules appearing in bands. The new rules often only extend these bands, suggesting that these rules may only be variations of others in the same band. However, in figure 8, there is the suggestion of new bands forming that may contain significantly different rules.

Using apparent novelty instead, but otherwise using the same settings, produces figures 9–11. The main difference to note is that for $\lambda = 0.3$, a number of bands are produced that are not seen when using basic support set based novelty. It also appears that some of these bands consist solely of rules that would previously have been dominated.

VII. Presentation of the Rule Set

Applying NSGA II to the adult database using cc-dominance produces hundreds of non-dominated rules. The modifications to the dominance relation increase the number of non-dominated rules further. Therefore, it is necessary to consider methods for summarizing the rule sets produced in order to improve the presentation of the rule sets to the user. This section describes the use of a clustering algorithm for this purpose and as a result, further illustrates the effect of the modified dominance relations.

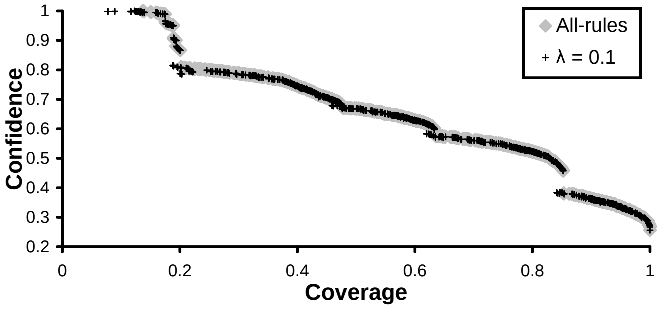

Fig. 6. Rules obtained with $\lambda = 0.1$, support set based novelty.

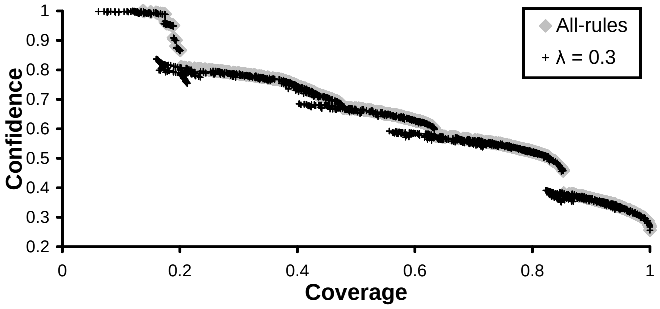

Fig. 7. Rules obtained with $\lambda = 0.3$, support set based novelty.

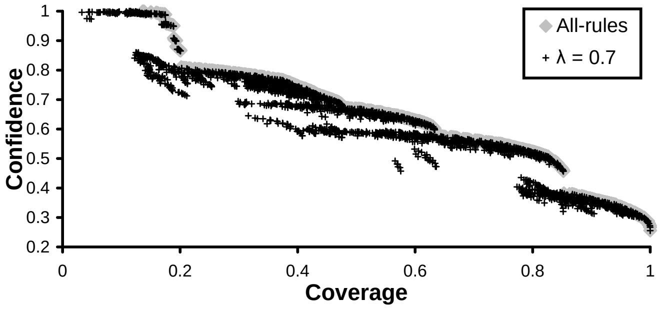

Fig. 8. Rules obtained with $\lambda = 0.7$, support set based novelty.

Previous research led to the application of a number of clustering algorithms to rules produced by both metaheuristics and by all-rule algorithms [2], [7], [8]. However, before any of these algorithms can be applied, it is necessary to be able to measure rule dissimilarity. The choice of dissimilarity measure depends on the type of novelty valued in the rules. If support set based novelty is valued in the creation of the rules, the dissimilarity between two rules should also be based on the support sets. If absolute novelty is used, a version of the simple matching coefficient [14], $d(q, r)$, is used to determine the distances, as defined by

$$d(q, r) = \frac{|S(q) - S(r)| + |S(r) - S(q)|}{|C|}.$$

If relative novelty is used, the Jaccard coefficient [15] is more

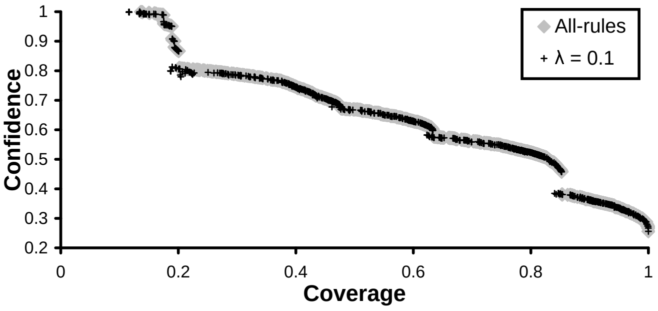

Fig. 9. Rules obtained with $\lambda = 0.1$, apparent novelty.

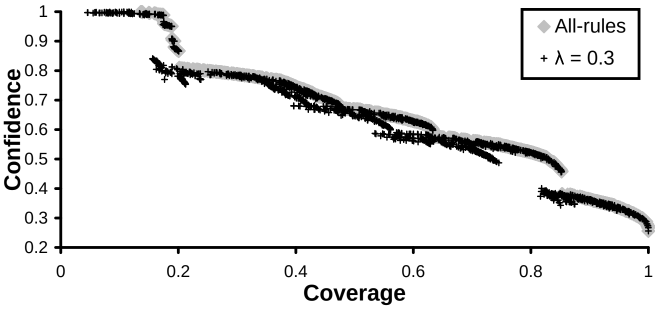

Fig. 10. Rules obtained with $\lambda = 0.3$, apparent novelty.

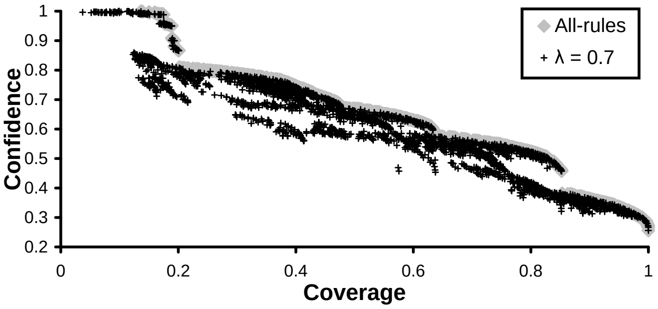

Fig. 11. Rules obtained with $\lambda = 0.7$, apparent novelty.

appropriate, defined by

$$j(q,r) = \frac{|S(q) - S(r)| + |S(r) - S(q)|}{|S(q) \cup S(r)|}.$$

When apparent novelty is used in the creation of the rule set, the dissimilarity metric should measure the apparent difference between rules. For example, to obtain the apparent dissimilarity based on the simple matching coefficient, first define events Q, R, C, Q_i and R_i as before in section IV-B. Then $d(q,r)$ can be written

$$d(q,r) = P(Q \cap \overline{R} \cup \overline{Q} \cap R \,|\, C).$$

If the user estimates this probability, assuming independence of the non-class fields, then:

$$\begin{aligned} d(q,r) &= P(Q \,|\, C) + P(R \,|\, C) - 2P(Q \cap R \,|\, C) \\ &= P(\textstyle\bigcap_i Q_i \,|\, C) + P(\bigcap_i R_i \,|\, C) - 2P(\bigcap_i (Q_i \cap R_i) \,|\, C) \\ &\approx \textstyle\prod_i P(Q_i \,|\, C) + \prod_i P(R_i \,|\, C) - \prod_i P(Q_i \cap R_i \,|\, C). \end{aligned}$$

This approximation gives a dissimilarity metric suitable for clustering rules generated using apparent, absolute novelty.

Having selected appropriate dissimilarity metrics, the hierarchical clustering algorithm known as AGNES (AGglomerative NESting) [14] was applied to the sets of non-dominated rules produced by NSGA II. Inter-cluster distances were calculated using the weighted average linkage [16], for reasons described by Reynolds et al. [7]. Figure 12 shows the results of clustering the rule set obtained using apparent novelty, with $\lambda = 0.3$. Here AGNES has been halted at the point where there are eight clusters of rules. Note that rules that are in the same band tend to be clustered together. This supports the contention in section VI-B that new rules that merely extend the bands are similar to rules already present, and as such add little information of interest. However, note that some clusters consist entirely of rules that would previously have been dominated. These clusters will be shown to contain useful information that would have been lost without the modifications made to the dominance relation.

Table I presents a summary of each of the clusters. This gives the cluster size, the central rule, or *medoid*, and the ATs present, explicitly or implicitly, in all the rules of the cluster. Clusters 2, 4 and 6 are those that would have previously been dominated. These are marked by the presence of the ATs `Sex = Male` and `Relationship = Husband`. These ATs are absent from all the rules in the other clusters, with the exception of a minority of rules from cluster 8 — the cluster of rules of highest confidence and lowest coverage. Furthermore, these ATs are entirely absent from the non-dominated rules obtained using cc-dominance. Since `Sex`

TABLE I
CLUSTER DETAILS.

Cluster	Size	Medoid antecedent	Commonality
1	314	Age $\geq$ 29, EduYrs $\geq$ 8, HoursPerWeek $\geq$ 10	Age $\geq$ 19, EduYrs $\geq$ 2
2	23	Age $\geq$ 25, Sex = Male	Age $\geq$ 23, Sex = Male
3	338	Age $\geq$ 31, EduYrs $\geq$ 9, HoursPerWeek $\geq$ 23, MarStat = Civ-spouse	Age $\geq$ 20, MarStat = Civ-spouse
4	205	Age $\geq$ 30, EduYrs $\geq$ 8, HoursPerWeek $\geq$ 32, Relat = Husband	Age $\geq$ 22, EduYrs $\geq$ 6, Relat = Husband
5	216	Age $\geq$ 31, Age $\leq$ 85, EduYrs $\geq$ 10, HoursPerWeek $\geq$ 23, MarStat = Civ-spouse	Age $\geq$ 21, EduYrs $\geq$ 10, MarStat = Civ-spouse
6	272	Age $\geq$ 29, Age $\leq$ 85, EduYrs $\geq$ 11, HoursPerWeek $\geq$ 32, Relat = Husband	EduYrs $\geq$ 10, Relat = Husband
7	437	Age $\geq$ 31, Age $\leq$ 83, EduYrs $\geq$ 13, HoursPerWeek $\geq$ 32, MarStat = Civ-spouse	EduYrs $\geq$ 11
8	200	Age $\geq$ 22, EduYrs $\geq$ 4, CapGain $\geq$ 7298	CapGain $\geq$ 3103

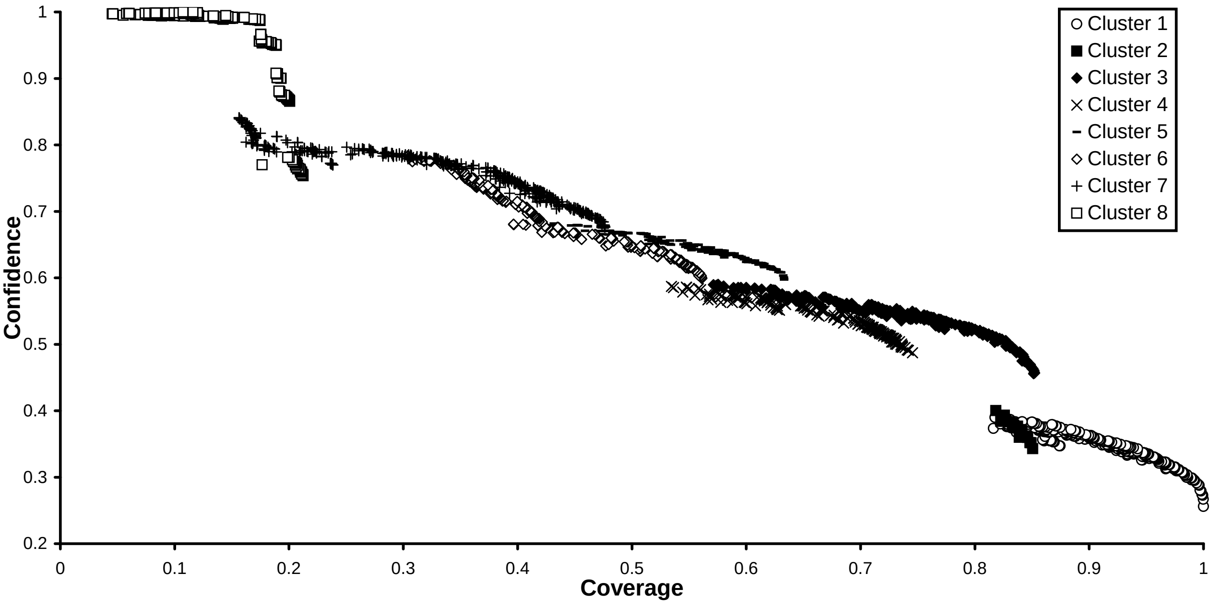

Fig. 12. Clustering applied to results using apparent novelty, $\lambda = 0.3$.

= `Male` is true for 85.2% of the class of interest but only 61.7% of the other records, the rules in these clusters are of interest, despite originally being dominated.

Note that AGNES provides a hierarchy of clusters, with each of the clusters shown divided into sub-clusters. This feature allows the user to explore clusters of interest more thoroughly by examining sub-clusters if desired.

VIII. Conclusions

It had been hypothesized that the use of cc-dominance might result in interesting descriptions of the class of interest being dominated. This paper described modified dominance relations designed to remedy this situation. It has been shown that, while some of the rules that were newly permitted to remain non-dominated were merely minor modifications of old rules, others did indeed provide additional information of interest to the user.

The use of modified dominance relations allows the user to be presented with a more diverse set of rules, without allowing the size of the rule set to grow excessively. Hierarchical clustering may then be used to aid the presentation of the rule set produced.

References

[1] B. de la Iglesia, M. S. Philpott, A. J. Bagnall, and V. J. Rayward-Smith, “Data Mining Rules Using Multi-Objective Evolutionary Algorithms,” in *Proceedings of 2003 IEEE Congress on Evolutionary Computation*, 2003, pp. 1552–1559.

[2] B. de la Iglesia, A. Reynolds, and V. J. Rayward-Smith, “Developments on a Multi-Objective Metaheuristic (MOMH) Algorithm for Finding Interesting Sets of Classification Rules,” in *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005*, ser. Lecture Notes in Computer Science, no. 3410, March 2005, pp. 826–840.

[3] B. de la Iglesia, G. Richards, M. S. Philpott, and V. J. Rayward-Smith, “The application and effectiveness of a multi-objective metaheuristic algorithm for partial classification,” *European Journal of Operational Research*, vol. 169, no. 3, pp. 898–917, 2006.

[4] K. Ali, S. Manganaris, and R. Srikant, “Partial Classification using Association Rules,” in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*. The AAAI Press, 1997, pp. 115–118.

[5] P. Riddle, R. Segal, and O. Etzioni, “Representation Design and Brute-force Induction in a Boeing Manufacturing Domain,” *Applied Artificial Intelligence*, vol. 8, pp. 125–147, 1994.

[6] R. J. Bayardo Jr. and R. Agrawal, “Mining the Most Interesting Rules,” in *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD ’99)*. ACM Press, 1999, pp. 145–153.

[7] A. P. Reynolds, G. Richards, B. de la Iglesia, and V. J. Rayward-Smith, “Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithms,” *Journal of Mathematical Modelling and Algorithms (to appear)*, 2006.

[8] 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)*, ser. Lecture Notes in Computer Science, no. 3177. Springer-Verlag, 2004, pp. 173–178.

[9] G. Richards and V. J. Rayward-Smith, “The Discovery of Association Rules from Tabular Databases Comprising Nominal and Ordinal Attributes,” *Intelligent Data Analysis*, vol. 9, no. 3, 2004.

[10] G. Richards and V. J. Rayward-Smith, “Discovery of association rules in tabular data,” in *Proceedings of IEEE First International Conference on Data Mining, San Jose, California, USA*. IEEE Computer Society, 2001, pp. 465–473.

[11] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons Ltd, 2001.

[12] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, “A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II,” in *Proceedings of the Parallel Problem Solving from Nature VI Conference*, ser. Lecture Notes in Computer Science, no. 1917. Springer, 2000, pp. 849–858.

[13] C. L. Blake and C. J. Merz, “UCI Repository of machine learning databases,” 1998, http://www.ics.uci.edu/~mlearn/MLRepository.html.

[14] L. Kaufman and P. J. Rousseuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, ser. Wiley Series in probability and mathematical statistics. John Wiley and Sons Inc., 1990.

[15] 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*, vol. 37, pp. 547–579, 1901.

[16] R. R. Sokal and P. H. A. Sneath, *Principles of Numerical Taxonomy*. San Francisco: Freeman, 1963.