Introduction
In previous work we have used multiobjective optimization to produce the following:

**Simple rules:** e.g. if haircol = red and exp $\geq 1$ hr and month = May then burn = yes.

**Expression trees:**

An alternative approach is to generate sets of simple rules. A record is predicted to belong to the ‘class of interest’ if it matches any one of the rules in the set.
Rule sets could be generated from scratch.

However, simple rule induction algorithms may have already been applied to the data. These typically produce a large number of rules.

An algorithm could be designed to select a subset of these rules.

Minimize error rate and rule set complexity.
The quality of the resulting classifier will depend on:

- the rule selection algorithm AND
- the set of rules from which it selects.

This raises two questions.

- How should this initial rule set be generated?
- If such a set already exists, how likely is it that a rule selection algorithm can produce an effective binary classifier.
Rule Induction
Define the support set, $S(M)$, of a conjunction of simple tests, $M$, to be the set of records that satisfy $M$.

Then the support for $M$ is $sup(M) = |S(M)|$.

If $r$ is a rule, let $r^a$ be the rule antecedent and $r^c$ be the consequent.

The support set of $r$, $S(r)$, is $S(r^a \land r^c)$.

The support for $r$ is $sup(r) = |S(r)|$. 
The confidence and coverage of \( r \) are defined by

\[
\text{conf}(r) = \frac{\text{sup}(r)}{\text{sup}(r^a)}, \quad \text{cov}(r) = \frac{\text{sup}(r)}{\text{sup}(r^c)}.
\]

Rule \( q \) is dominated by rule \( r \) according to \textit{cc-dominance} if and only if

\[
\text{cov}(q) \leq \text{cov}(r) \text{ and } \text{conf}(q) < \text{conf}(r), \text{ or } \\
\text{cov}(q) < \text{cov}(r) \text{ and } \text{conf}(q) \leq \text{conf}(r).
\]
Novelty and Modified Dominance

- If \( C \) is the class of interest, then:

\[
nov_a(q, r) = \frac{|S(q) - S(r)|}{|C|}, \quad nov_r(q, r) = \frac{|S(q) - S(r)|}{|S(r)|}.
\]

- If rule \( q \) is cc-dominated by rule \( r \), then the dominance margin is defined as

\[
dm(r, q) = \min(\text{conf}(r) - \text{conf}(q), \text{cov}(r) - \text{cov}(q)).
\]

- Now let rule \( q \) be dominated by rule \( r \) if and only if

\[
r >_{cc} q \text{ and } \lambda nov(q, r) \leq dm(r, q).
\]
Clustered Rules

Introduction

Rule Induction
Support
Confidence, Coverage and cc-Dominance
Novelty and Modified Dominance

Clustered Rules
Epsilon Dominance
A Third Objective

Rule Selection

Results
Rule $q$ is $\epsilon$-dominated by rule $r$ if and only if

$$cov(q) + \epsilon \leq cov(r) \text{ and } conf(q) + \epsilon < conf(r), \text{ or}$$

$$cov(q) + \epsilon < cov(r) \text{ and } conf(q) + \epsilon \leq conf(r).$$

While this increases the number of non-dominated rules, no effort is made to ensure that the additional rules provide useful new information.
A rule selection algorithm needs access to simple, high quality rules.

The complexity, $\text{comp}(r)$, of a rule, $r$, may be added as a third objective.

If the novelty measures are used then rule $q$ is dominated by rule $r$ if and only if

$$r \succ_{\text{ccc}} q \text{ and } \lambda_{\text{nov}}(q, r) \leq dm(r, q).$$
GRASP (Greedy Randomized Adaptive Search Procedure) is an iterative optimization algorithm, where each iteration consists of two phases:

- A randomized greedy algorithm.
- Local search.

In a typical algorithm, the first phase constructs a solution one element at a time.

- The best elements form a restricted candidate list (RCL).
- The element to be added is selected at random from the RCL.
GRASP has rarely been applied as a multiobjective optimization algorithm.

One approach used is to select a random linear combination of the objectives to optimize prior to application of the two phases.

We attempt to create a MO GRASP (MOG) based solely on the use of dominance.
Function MOG\((G, N, I)\)

\[
\text{bestFront} := \emptyset
\]

\[
\text{for } (x := 1 \text{ to } G) \quad \text{newFront} := \text{OneGeneration}(N, I) \quad \text{bestFront} := \text{bestFront} \cup \text{newFront}
\]

Remove any dominated solutions from \text{bestFront}

\[
\text{endfor}
\]

\[
\text{return bestFront}
\]

Function OneGeneration\((N, I)\)

\[
\text{front} := \text{BuildFront}(N)
\]

\[
\text{front} := \text{LocalSearch}(\text{front}, I)
\]

\[
\text{return front}
\]
Function \textbf{BuildFront}(N)

\begin{align*}
\text{front} & := \emptyset \\
\text{for } (x := 1 \text{ to } N) & \\
\text{newFront} & := \text{GreedyRun()} \\
\text{front} & := \text{front} \cup \text{newFront} \\
\text{endfor} \\
\text{Remove any dominated solutions from front} \\
\text{return front}
\end{align*}
MOG: Local Search

Function LocalSearch\((front, I)\)

for \((x := 1 \text{ to } I)\)

\(s := \text{solution selected at random from } front\)

\(s := \text{applyMove}(s)\)

\(\text{evaluate}(s)\)

if \((s \text{ is not dominated by any member of } front)\)

Add \(s\) to \(front\)

Remove any dominated solutions from \(front\)

endif

endfor

return \(front\)
Completing MOG: Cost Thresholds

Introduction
Rule Induction
Rule Selection
GRASP
Multiobjective GRASP
MOG: Overview
MOG: Construction Phase
MOG: Local Search
Completing MOG: Cost Thresholds
Completing MOG: Complexity Constraints
Implementation Details and Efficiency
Results

A Multiobjective GRASP for Rule Selection – p. 19
Completing MOG: Complexity Constraints

Introduction

Rule Induction

Rule Selection

GRASP

Multiobjective GRASP

MOG: Overview

MOG: Construction Phase

MOG: Local Search

Completing MOG: Cost Thresholds

Completing MOG: Complexity Constraints

Implementation Details and Efficiency

Results

A Multiobjective GRASP for Rule Selection – p. 20
Use a ‘match table’ and ‘match counts’.

On selection of a rule, match counts for remaining candidate rules are carefully updated.

Rules are ordered so as to minimize the number of rules that must be examined.

Given rule clusters, allow the greedy algorithm to consider only a single representative rule from each cluster.

Three methods for choosing the cluster representative have been used: random, smallest and balanced.

Local search uses three types of move: rule addition, rule removal and rule swap.
No Clusters vs. 500 Clusters
The Effect of Novelty

Introduction

Rule Induction

Rule Selection

Results

No Clusters vs. 500 Clusters

Comparison with Epsilon Dominance

Comparison with Rule Set Generation from Scratch

Example Rule Sets

Conclusions
Comparison with Epsilon Dominance

<table>
<thead>
<tr>
<th>Obj.</th>
<th>Novelty</th>
<th>( \lambda/\epsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Low</td>
</tr>
<tr>
<td>2</td>
<td>Absolute</td>
<td>3.5542</td>
</tr>
<tr>
<td></td>
<td>Relative</td>
<td>3.5510</td>
</tr>
<tr>
<td></td>
<td>Epsilon</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Absolute</td>
<td>3.4253</td>
</tr>
<tr>
<td></td>
<td>Relative</td>
<td>3.4312</td>
</tr>
<tr>
<td></td>
<td>Epsilon</td>
<td>-</td>
</tr>
</tbody>
</table>
Comparison with Rule Set Generation from Scratch

Introduction
Rule Induction
Rule Selection
Results
No Clusters vs. 500 Clusters
The Effect of Novelty
Comparison with Epsilon Dominance
Comparison with Rule Set Generation from Scratch
Example Rule Sets
Conclusions

A Multiobjective GRASP for Rule Selection – p. 26
## Example Rule Sets

<table>
<thead>
<tr>
<th>Cost</th>
<th>Rule set</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.31%</td>
<td>If cap. gain $\geq$ 5178</td>
</tr>
<tr>
<td>18.47%</td>
<td>If edu. yrs $\geq$ 13 and mar. stat. $=$ civ. spouse</td>
</tr>
<tr>
<td>16.28%</td>
<td>If edu. yrs $\geq$ 13 and mar. stat. $=$ civ. spouse or cap. gain $\geq$ 5178</td>
</tr>
<tr>
<td>16.14%</td>
<td>If hrs/week $\geq$ 31 and edu. yrs $\geq$ 13 and mar. stat. $=$ civ. spouse or cap. gain $\geq$ 5178</td>
</tr>
<tr>
<td>16.00%</td>
<td>If edu. yrs $\geq$ 13 and age $\geq$ 28 and hrs/week $\geq$ 27 and mar. stat. $=$ civ. spouse or cap. gain $\geq$ 5178</td>
</tr>
<tr>
<td>15.62%</td>
<td>If cap. gain $\geq$ 5178 $or$ cap. loss $\geq$ 1762 and cap. loss $\leq$ 1980 and relationship $=$ husband or edu. yrs $\geq$ 13 and mar. stat. $=$ civ. spouse</td>
</tr>
</tbody>
</table>
MOG has been successfully applied to another problem, using only the concept of dominance to guide the search.

If the algorithm is restricted to select from cc-optimal rules, there is insufficient variation in the rules to produce a good classifier.

Using novelty based dominance relations fixes this problem.

This justifies the use of novelty in the dominance relation, even when the production of a full classifier is not an aim.

Using novelty works better than $\epsilon$-dominance.