Outline

1. Motivation
2. Separate-and-conquer Rule Learning
3. Search Strategies
   ▶ Hill Climbing and Beam search
   ▶ Exhaustive search
4. Rule Learning Heuristics
5. Results
   ▶ Experimental Setup
   ▶ Varying the beam size
   ▶ Individual Datasets
   ▶ Searching for single rules
6. Discussion
1. Motivation

- the phenomenon of over-searching, i.e., that more search has not to lead to better predictive accuracy, was first shown by Quinlan and Cameron-Jones (1995)
- but they only used one heuristic and no true Exhaustive Search
- we extend their work to 9 different heuristics and a true Exhaustive Search
- no experimental results about the connection between the search heuristic and the search strategy
- we want to answer the question whether Separate-and-conquer (SeCo) algorithms can improve from Exhaustive Search or bigger beams both in terms of theory size and accuracy or not
2. Separate-and-Conquer Rule Learning

In the experiments we used a simple SECO Rule Learner with the following properties:

▶ allows the usage of different heuristics and search strategies (Top-Down Beam Search)
▶ employs ordered class binarization
▶ classification is done by a decision list of rules
▶ does not perform pruning
▶ but implements Forward Pruning (important for the runtime)
  ▶ create a virtual rule that covers the same number of positive examples but no negative instances
  ▶ if the evaluation of this rule is lower than that of the best rule → stop refining this rule
3. Search Strategies

Hill-Climbing and Beam search

It is possible that a naive Beam search for $b \to \infty$ generates more rules than the Exhaustive Search.
3. Search Strategies

Exhaustive search

Note that the implemented procedure follows $OPUS^\circ$ (Webb, 1995), i.e., does not generate duplicates.
4. Rule Learning Heuristics

<table>
<thead>
<tr>
<th>heuristic</th>
<th>formula</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simple heuristics</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Precision</strong></td>
<td>$\frac{p}{p+n}$</td>
</tr>
<tr>
<td><strong>Laplace</strong></td>
<td>$\frac{p+1}{p+n+2}$</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td>$p - n$</td>
</tr>
<tr>
<td><strong>Weighted Relative Accuracy</strong></td>
<td>$\frac{p}{P} - \frac{n}{N}$</td>
</tr>
<tr>
<td><strong>Odds ratio</strong></td>
<td>$\frac{p(N-n)}{(P-p)\cdot n}$</td>
</tr>
<tr>
<td><strong>Correlation</strong></td>
<td>$\frac{p(N-n)-n(P-p)}{\sqrt{P\cdot N\cdot (p+n)\cdot (P-p+N-n)}}$</td>
</tr>
<tr>
<td><strong>Complex heuristics</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Relative Cost Measure</strong></td>
<td>$c \cdot \frac{p}{P} - (1-c) \cdot \frac{n}{N}$</td>
</tr>
<tr>
<td><strong>m-estimate</strong></td>
<td>$\frac{p+m\cdot p/(p+N)}{p+n+m}$</td>
</tr>
<tr>
<td><strong>Meta-learned</strong></td>
<td>learned $f(p,n,P,N)$</td>
</tr>
</tbody>
</table>

as suggested in (Janssen and Fürnkranz, 2008) the parameters were set to $c = 0.342$ and $m = 22.466$
5. Results
Experimental Setup

- 22 datasets from UCI Repository
- only nominal attributes in data (Exhaustive Search cannot handle numeric attributes at the moment)
- only small to medium size datasets (runtime of ES grows strongly with #attributes, #classes, #instances)
- Performance measure: macro average accuracy on many datasets estimated with 10-fold stratified CV
- expectation: runtime increases with increased beam sizes and positive effect of Exhaustive Search are
  - best visible when datasets are hard to learn
  - or when the Hill-climbing Search gets stuck in a local optimum
5. Results

Varying the beam size

Example for consistent improvement/degradation

legend: blue dotted line = # conditions, red solid line = macro-average accuracy of CV, beam size 10000 = Exhaustive Search Algorithm, # conditions = conds. of all rules summed up
5. Results

Varying the beam size

Example for strong fluctuations

Note that the final minor jump is due to different implementations of the Hill-climbing Search and the Exhaustive Search.
5. Results

Plot for individual dataset (autos-d)

legend: macro-averaged accuracy of CV
5. Results

Plot for individual dataset (breast-w-d)
5. Results

Searching for single rules

- Interestingly the performance with one rule per class plus a default class is very good (about 10% less than the complete models)

- Examples:
  - Precision: Hill-climbing Search 64.67% with 6.82 conditions, Exhaustive Search 68.55% with 9.59 conditions
  - WRA: Hill-climbing Search 68.14% with 3.23 conditions, Exhaustive Search 68.81% with 3.5 conditions

- Precision and Laplace have significantly smaller theories (about 7 times smaller) than the full size model

- All heuristics gain performance from Exhaustive Search except for the Meta-learned one
6. Discussion

- the over-searching phenomenon depends on the heuristic
  - Odds Ratio and Precision gain performance
  - more complex heuristics lose performance
- heuristics that work well in Hill-climbing Search usually do not profit from Exhaustive Search or Beam search with bigger beam sizes
- experiments show that there are different requirements for heuristics used in Hill-climbing Search and Exhaustive Search
- mandatory next step:
  - separate the search heuristic (potential of a rule of being refined into a high quality rule) and the rule evaluation function (isolated measurement of the predictive quality of a rule)
References