INVERTED HEURISTICS IN SUBGROUP DISCOVERY

Anita Valmarska\textsuperscript{1} \quad Marko Robnik-Šikonja\textsuperscript{2} \quad Nada Lavrač\textsuperscript{1}

\textsuperscript{1}Jožef Stefan Institute, Ljubljana, Slovenia
\textsuperscript{2}Faculty of Computer and Information Science, Ljubljana, Slovenia
Outline

• Motivation
• Inverted heuristics
• DoubleBeam algorithm
• Experimental results
• Conclusions and further work
Motivation

• Two phases in rule learning:
  • Refinement phase
  • Selection phase
• Stecher et al. propose usage of inverted heuristics in refinement phase
• Subgroup discovery
Inverted heuristics

• Rule learning algorithms rely on some sort of measure to determine the quality of a rule
• Different heuristics
• Separate-and-conquer rule learning algorithms have two fundamental steps in the process learning – rule refinement and rule selection.
Inverted heuristics

• Stecher et al. propose separation of the rule refinement and rule selection heuristics in inductive rule learning

• Rule refinement step requires *inverted heuristics*, which evaluate rules from the point of view of the current base rule, and not the empty rule.
Inverted heuristics

DoubleBeam algorithm

**Input**: $E = P \cup N$ (E is the training set, $|E|$ is the training set size, $P$ are the positive (class) examples, $N$ are negative (non-target) examples), TargetClass

**Output**: subgroups

**Parameters**: $\text{min\_support}$, $\text{refinement\_Beam\_Width}$, $\text{selection\_Beam\_Width}$, $\text{refinement\_heuristics}$, $\text{selection\_heuristics}$

CandidateList ← all feature values or intervals

for each candidate in CandidateList do
    evaluate candidate with refinement\_quality
    evaluate candidate with selection\_quality
end

sort CandidateList according to the refinement\_quality

for $i = 0$ to refinement\_Beam\_Width) do
    RefinementBeam($i$) ← CandidateList($i$)
end

sort CandidateList according to the selection\_quality

for $i = 0$ to selection\_Beam\_Width) do
    SelectionBeam($i$) ← CandidateList($i$)
end

do
    RefinementCandidates ← refine RefinementBeam with CandidateList
    update RefinementBeam with RefinementCandidates using refinement\_quality
    update SelectionBeam with RefinementCandidates using selection\_quality
while while there are changes in SelectionBeam;
return SelectionBeam

Algorithm 1: DoubleBeam algorithm
Experimental results

- 20 UCI datasets
- Ten-fold double-loop cross-validation
- Measures: coverage, support, size, complexity, significance, unusualness, classification accuracy and AUC.
- Eight tested algorithms
  - SD, CN2-SD, APRIORI-SD, DB-ILL, DB-IPP, DB-IMM, DB-GG, DB-IGG
Experimental results

Figure 1: Nemenyi test on WRACC values with a significance level of 0.05.
Experimental results

Figure 2: Nemenyi test on ranking of average rule sizes (note that larger rules produce lower rankings) with a significance level of 0.05
Conclusions and further work

- Inverted heuristics in subgroup discovery produces significantly less interesting rules.
- DB-IPP tend to produce longer rules, but less unusual ones.
- DB-GG can be a good choice for subgroup discovery.
Conclusions and further work

• Why the DB-ILL algorithm, the DB-IPP algorithm and the DB-IMM algorithm produce less interesting rules?
• Research the influence of inverted heuristics in other state-of-the-art subgroup discovery algorithms.
• Test the DoubleBeam algorithm using WRACC as a refinement and selection heuristic.
Thank you!