

Inverted Heuristics in Subgroup Discovery

Anita Valmarska
Jožef Stefan Institute
Jožef Stefan International
Postgraduate School
Jamova 39
Ljubljana, Slovenia
anita.valmarska@ijs.si

Marko Robnik-Šikonja
Faculty of Computer and
Information Science
Večna pot 113
Ljubljana, Slovenia
marko.robnik@fri.uni-lj.si

Nada Lavrač
Jožef Stefan Institute
Jožef Stefan International
Postgraduate School
Jamova 39
Ljubljana, Slovenia
nada.lavrac@ijs.si

ABSTRACT

In rule learning, rules are typically induced in two phases, rule refinement and rule selection. It was recently argued that the usage of two separate heuristics for each phase—in particular using the so-called inverted heuristic in the refinement phase—produces longer rules with comparable classification accuracy. In this paper we test the utility of inverted heuristics in the context of subgroup discovery. For this purpose we developed a DoubleBeam subgroup discovery algorithm that allows for combining various heuristics for rule refinement and selection. The algorithm was experimentally evaluated on 20 UCI datasets using 10-fold double-loop cross validation. The experimental results suggest that a variant of the DoubleBeam algorithm using a specific combination of refinement and selection heuristics generates longer rules without compromising rule quality. However, the DoubleBeam algorithm using inverted heuristics does not outperform the standard CN2-SD and SD algorithms.

1. INTRODUCTION

Rule learning is one of the earliest machine learning techniques and has been used in numerous applications [5]. It is a symbolic data analysis technique whose aim is to discover comprehensible patterns or models of data [10]. The key advantage of rule learning compared to other statistical learning techniques is its inherent simplicity and human comprehensible output models and patterns.

Symbolic data analysis techniques can be divided into two categories. Techniques for *predictive induction* produce models, typically induced from class labeled data, which are used to predict the class of previously unseen examples. The second category consists of techniques for *descriptive induction*, where the aim is to find comprehensible patterns, typically induced from unlabeled data. There are also descriptive induction techniques that discover patterns in the form of rules from labeled data, which are referred to as *supervised descriptive rule discovery* approaches [10]. Typical representatives of these techniques are contrast set mining (CSM) [2], emerging pattern mining (EPM) [4], and subgroup discovery (SD) [9, 16].

The task of subgroup discovery is to find interesting subgroups in the population i.e., subgroups that have a significantly different class distribution than the entire population. The result of subgroup discovery is a set of individual rules, where the rule consequence is a class value. The main difference between learning of classification rules and subgroup discovery is that the latter induces only individual rules of interest, revealing interesting properties of groups

of instances, and not necessarily forming a rule set covering the entire problem space, which is required for classification.

An important characteristic of subgroup discovery task is a combination of predictive and descriptive induction. It provides short and understandable descriptions of subgroups regarding the property of interest. This feature of subgroup discovery has inspired many researchers to investigate new methods that will be more effective in finding more interesting patterns in the data. Most subgroup discovery approaches build on classification algorithms, e.g., EXPLORA [9], MIDOS [16], SD [6], CN2-SD [13], and RSD [14], or on the algorithms for association rule learning, e.g., APRIORI-SD [8], SD-MAP [1], and Merge-SD [7].

In rule learning, during the process of rule construction, conditions that optimize a certain heuristic are added. Typically, the heuristics are used in two different phases of the process: (i) to evaluate *rule refinements*, i.e., to select which of the refinements of the current rule will be further explored, and (ii) for *rule selection*, i.e., to decide which of the refinements that have been explored is added to the rule set. Stecher et al. [15] proposed using separate heuristics for each of the two rule construction phases. In the rule refinement phase they proposed to use the *inverted heuristics*, i.e., the heuristics whose isometrics are rotated around the base rule. These heuristics are used to evaluate the relative gain obtained by the refinement of the current rule.

In this paper we test the utility of inverted heuristics in the context of subgroup discovery. For this purpose we developed a DoubleBeam subgroup discovery algorithm that allows for combining various heuristics for rule refinement and selection. The algorithm was experimentally evaluated on 20 UCI datasets using 10-fold double-loop cross validation.

This paper is organized as follows. In Section 2 we present the findings of Stecher et al. about the use of inverted heuristics in the rule learning process. Section 3 presents the DoubleBeam subgroup discovery algorithm. In Section 4 we describe the data sets used, followed by the empirical evaluation and the obtained results. Finally, in Section 5 we present our conclusions and ideas for further work.

2. INVERTED HEURISTICS

Rule learning algorithms rely on heuristic measures to determine the quality of induced rule. Stecher et al. [15] propose distinction between rule refinement and rule selection heuristics in inductive rule learning. They argue that the nature of the separate-and-conquer rule learning algorithms opens up a possibility to use two different heuristics

in two fundamental steps in the process of rule learning - rule refinement and rule selection. They show in the coverage space why it is beneficial to separate the evaluation of candidates for rule refinement and the selection of rules for the final theory. The rule refinement step in a top-down search requires *inverted heuristics*, which, in principle, results in better rules. Such heuristics evaluate rules from the point of the current base rule, instead of the empty rule. They adapt three standard heuristics with slightly different but related properties:

- **Precision:**

$$h_{prec}(p, n) = \frac{p}{p+n}; \quad (1)$$

- **Laplace:**

$$h_{lap}(p, n) = \frac{p+1}{p+n+2}; \quad (2)$$

- **m-estimate:**

$$h_{m-est}(p, n, m) = \frac{p+m \cdot \frac{P}{P+N}}{p+n+m}. \quad (3)$$

Parameters p and n denote the number of positive and negative examples in a potential subgroup, respectively, and regarding the target class of interest.

For the purpose of rule refinement an inverted heuristic is used. Isometrics of inverted heuristics do not rotate around the origin, but rotate around the base rule. Representations of the inverted heuristics in the coverage space reveal the following relationship with the basic heuristics:

$$h'(p, n) = h(N-n, P-p) \quad (4)$$

where parameters P and N denote the number of positive and negative examples in the data set with respect to the target class, and dependent on the predecessor rule. Consequently, the inverted heuristics have the following forms:

- **Inverted precision:**

$$h'_{prec}(p, n) = \frac{N-n}{(P+N)-(p+n)}; \quad (5)$$

- **Inverted Laplace:**

$$h'_{lap}(p, n) = \frac{N-n+1}{(P+N)-(p+n+2)}; \quad (6)$$

- **Inverted m-estimate:**

$$h'_{m-est}(p, n, m) = \frac{N-n+m \cdot \frac{P}{P+N}}{(P+N)-(p+n+m)}. \quad (7)$$

Overall, in [15] the combination of Laplace heuristic h_{lap} in the rule selection step and inverted Laplace heuristic h'_{lap} in rule refinement step outperformed other combinations in terms of average classification accuracy. An interesting side conclusion from [15] is that the usage of inverted heuristics in the rule refinement produces on average longer rules.

The tendency of inverted heuristics to find longer descriptions and no additional parameters make the separation of rule refinement and rule selection an appealing research approach in the domain of subgroup discovery, therefore, we

investigated the use of inverted heuristics in subgroup discovery. For that purpose we implemented a new DoubleBeam algorithm for subgroup discovery which implements the usage of separate refinement and selection heuristics with beam search.

3. DOUBLEBEAM ALGORITHM

We implemented a DoubleBeam algorithm for subgroup discovery. This algorithm consists of two beams, refinement and selection beam. Upon initialization, each beam is filled with the best features according to their refinement and selection quality. The algorithm then enters a loop, where it first refines the elements from the refinement beam with features from the dataset. In each step, rules from the refinement beam are refined by adding features to existing rules. Newly produced rules are added to the refinement beam if their refinement quality exceeds the refinement quality of existing rules in the refinement beam. Newly produced rules are then evaluated according to their selection quality. Selection beam is updated with newly induced rules whose selection quality is better than the selection quality of rules already in the beam. The algorithm exits the loop and stops when there are no changes in the selection beam. The DoubleBeam algorithm is outlined in Algorithm 1.

```

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:  $min\_support$ ,  $refinementBeamWidth$ ,
               $selectionBeamWidth$ ,  $refinement\_heuristics$ ,
               $selection\_heuristics$ 

CandidateList  $\leftarrow$  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  $refinementBeamWidth$  do
  |  $RefinementBeam(i) \leftarrow CandidateList(i)$ 
end
sort CandidateList according to the  $selection\_quality$ 
for  $i = 0$  to  $selectionBeamWidth$  do
  |  $SelectionBeam(i) \leftarrow CandidateList(i)$ 
end
do
  |  $RefinementCandidates \leftarrow$  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

4. EXPERIMENTAL RESULTS

The DoubleBeam algorithm was implemented in the ClowdFlows platform [12]. For the purpose of our evaluation, we used the following combinations of refinement and selection heuristics: (h'_{lap}, h_{lap}) , (h'_{prec}, h_{prec}) , (h'_{m-est}, h_{m-est}) , (h'_g, h_g) , and (h_g, h_g) , (named DB-ILL (DoubleBeam-Inverted Laplace, Laplace), DB-IPP (DoubleBeam-Inverted precision,

precision), DB-IMM (DoubleBeam-Inverted m-estimate, m-estimate), DB-IGG (DoubleBeam-Inverted generalization-quotient, generalization quotient), and DB-GG (DoubleBeam-generalization quotient, generalization quotient) respectively). The h_g heuristic is the generalization quotient proposed in [6], while h'_g is its inverted variant. The generalization quotient is a heuristic used in the SD algorithm. The SD algorithm and the algorithms CN2-SD and APRIORI-SD were already implemented in this platform.

4.1 Experimental setting

We use the same 20 UCI classification data sets as [15] to compare three state-of-the-art subgroup discovery algorithms (SD, CN2-SD, and APRIORI-SD) and the DoubleBeam algorithm with five combinations of refinement and selection heuristics (DB-ILL, DB-IPP, DB-IMM, DB-IGG, and DB-GG).

The comparison is performed in 10-fold double-loop cross validation on each dataset. For each algorithm, a grid of possible parameter values was set beforehand. The value of min_sup is set to 0.01. Each learning set (10 learning sets) was additionally split into training and test data. For each algorithm, models were built using the training data and its parameters from the grid. Parameters maximizing the value of unusualness of the produced subgroups were then chosen for building a model using the learning set. Unusualness is a measure which was presented in [13] and defined as:

$$WRAcc(Class \leftarrow Cond) = p(Cond) \cdot (p(Class|Cond) - p(Class)). \quad (8)$$

We use the subgroup discovery evaluation function implemented in Orange by Kralj et al. [11]. The function calculates the following measures: *coverage*, *support*, *size*, *complexity*, *significance*, *unusualness* i.e., WRACC, *classification accuracy*, and *AUC*.

4.2 Results

The WRACC values obtained in the experiments are shown in Table 1. These values are averaged over all the classes for every particular dataset. The values for the APRIORI-SD algorithm tested on the horse-colic dataset are missing as the algorithm did not converge in period over 5 days. For the datasets where the WRACC values for the APRIORI-SD algorithm are 0.000 the algorithm returned over 10, 000, 000 itemsets and did not finish properly. According to the obtained results, the CN2-SD and the SD algorithm have the best average ranks, and the Apriori-SD algorithm performs the worst.

For comparison between methods we use the methodology proposed by Demšar [3]. We operate under the null-hypothesis that all the algorithms are equivalent. Two algorithms differ significantly if the difference between their average ranks is larger than the value of the critical difference.

The results of the Nemenyi test for the average values of WRACC are shown in Figure 1. Average ranks of algorithms are written in parentheses. The critical value is 2.35. It is evident that the CN2-SD algorithm produces the most interesting subgroups, which are statistically more unusual than the ones produced by the DoubleBeam algorithm with the combinations (h'_{lap}, h_{lap}) , (h'_{prec}, h_{prec}) , (h'_{m-est}, h_{m-est}) , and the APRIORI-SD algorithm. There

are no statistically significant differences between the CN2-SD algorithm, the SD algorithm, and the DoubleBeam algorithm with the combinations (h_g, h_g) and (h'_g, h_g) .

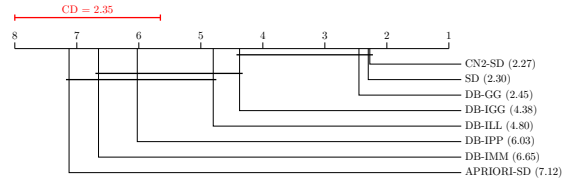


Figure 1: Nemenyi test on WRACC values with a significance level of 0.05.

The results of the Nemenyi test for the average rule size are shown in Figure 2. The DoubleBeam algorithm with the combination (h'_{prec}, h_{prec}) produces subgroups which are on average described by the longest rules. The DB-IPP algorithm generates subgroups described by rules that are statistically longer only than the ones produced by the DB-IGG algorithm. There is no statistical evidence that the DB-IPP algorithm produces longer rules than other evaluated algorithms. These results do not confirm that the DoubleBeam algorithm with inverted refinement heuristic produces statistically longer rules than other subgroup discovery algorithms.

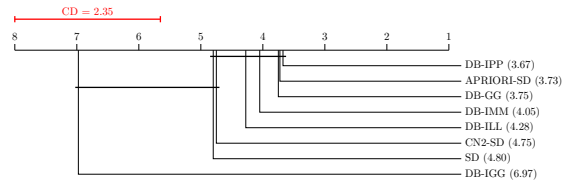


Figure 2: Nemenyi test on ranking of average rule sizes (note that larger rules produce lower rankings) with a significance level of 0.05.

5. CONCLUSIONS

The experiments indicate that subgroup describing rules created using inverted heuristics used in [15] as rule refinement heuristics in subgroup discovery are significantly less interesting than the subgroups induced by the CN2-SD algorithm, the SD algorithm, and the DB-GG algorithm. There is no significant difference of the unusualness of the subgroups induced by the CN2-SD algorithm, the SD algorithm, the DB-GG algorithm, and the DB-IGG algorithm. However, it has to be mentioned that the CN2-SD algorithm uses WRACC as its heuristics for building subgroups.

The results also suggest that when the combination (h'_{prec}, h_{prec}) of heuristics is used, the obtained rules tend to have more rule conditions than the rules built by the other state-of-the-art algorithms for subgroup discovery. However, this difference is not statistically significant. The longer rules created by the algorithms using inverted heuristics used in [15] are more specific, thus subgroups contain lower number of examples and this decreases the unusualness of the subgroups. Considering the evaluation results, we can conclude that the DoubleBeam algorithm which uses the combination (h_g, h_g) as refinement and selection heuristics can be a good choice for subgroup discovery. It induces subgroups

Table 1: Ten-fold double-loop cross validation WRACC results for subgroup discovery. Best values are written in bold.

Datasets	SD	APRIORI-SD	CN2-SD	DB-ILL	DB-IPP	DB-IMM	DB-GG	DB-IGG
breast-cancer	0.045	0.003	0.024	0.015	0.010	0.011	0.045	0.041
car	0.029	0.006	0.031	0.021	0.021	0.003	0.028	0.028
contact-lenses	0.080	0.031	0.066	0.036	0.023	0.000	0.081	0.081
futebol	0.017	0.004	0.009	0.003	0.002	0.000	0.001	0.000
glass	0.050	0.017	0.047	0.029	0.026	0.019	0.045	0.038
hepatitis	0.046	0.000	0.061	0.041	0.016	0.031	0.060	0.043
horse-colic	0.131		0.071	0.045	0.022	0.054	0.131	0.084
hypothyroid	0.025	0.000	0.031	0.019	0.009	0.007	0.032	0.024
idh	0.088	0.053	0.094	0.078	0.068	0.000	0.081	0.088
ionosphere	0.115	0.000	0.111	0.084	0.049	0.062	0.120	0.101
iris	0.162	0.119	0.197	0.148	0.098	0.045	0.165	0.169
labor	0.074	0.018	0.106	0.082	0.021	0.029	0.090	0.063
lymphography	0.058	0.015	0.046	0.045	0.042	0.038	0.056	0.036
monk3	0.068	0.014	0.065	0.041	0.024	0.007	0.068	0.058
mushroom	0.128	0.000	0.163	0.147	0.025	0.023	0.146	0.122
primary-tumor	0.018	0.004	0.009	0.007	0.009	0.004	0.011	0.008
soybean	0.033	0.000	0.037	0.023	0.023	0.009		0.028
tic-tac-toe	0.053	0.007	0.021	0.017	0.011	0.007	0.053	0.053
vote	0.184	0.052	0.201	0.157	0.076	0.120	0.188	0.164
zoo	0.083	0.000	0.096	0.041	0.020	0.025	0.086	0.058

that are comparable to the subgroups induced by the CN2-SD algorithm and the SD algorithm in terms of their unusualness. The subgroups induced by the DB-GG algorithm are on average described by longer rules (see Figure 2).

No additional parameters required with inverted heuristics and the obtained results regarding the average rule length make the proposed approach an interesting research direction. In future we want to focus on the reasons why the rules induced by the DB-ILL algorithm, the DB-IPP algorithm and the DB-IMM algorithm are less interesting than the ones produced by the standard subgroup discovery algorithms and implement an approach which will solve this issue. We also want to research the influence of inverted heuristics in other state-of-the-art subgroup discovery algorithms.

6. ACKNOWLEDGMENTS

We would like to acknowledge the support of the Slovenian Research Agency (through a young researcher grant to AV and the programme grant Knowledge Technologies) and the European Commission (through the project HBP - The Human Brain Project - grant FP7-ICT-604102).

7. REFERENCES

- [1] M. Atzmüller and F. Puppe. SD-map - A fast algorithm for exhaustive subgroup discovery. In *Proceedings of Knowledge Discovery in Databases, PKDD 2006*, pages 6–17, 2006.
- [2] S. D. Bay and M. J. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001.
- [3] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- [4] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999*, pages 43–52, 1999.
- [5] J. Fürnkranz, D. Gamberger, and N. Lavrač. *Foundations of Rule Learning*. Springer, 2012.
- [6] D. Gamberger and N. Lavrač. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*, 17:501–527, 2002.
- [7] H. Grosskreutz and S. Rüping. On subgroup discovery in numerical domains. *Data Mining and Knowledge Discovery*, 19(2):210–226, 2009.
- [8] B. Kavšek and N. Lavrač. APRIORI-SD: adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, 20(7):543–583, 2006.
- [9] W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. 1996.
- [10] P. Kralj Novak, N. Lavrač, and G. I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 10:377–403, 2009.
- [11] P. Kralj Novak, N. Lavrač, B. Zupan, and D. Gamberger. Experimental comparison of three subgroup discovery algorithms: Analysing brain ischemia data. In *Proceedings of the 8th International Multiconference Information Society*, pages 220–223, 2005.
- [12] J. Kranjc, V. Podpečan, and N. Lavrač. ClowdFlows: A cloud based scientific workflow platform. In *Proceedings of Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012*, pages 816–819, 2012.
- [13] N. Lavrač, B. Kavšek, P. A. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.
- [14] N. Lavrač, F. Železný, and P. A. Flach. RSD: relational subgroup discovery through first-order feature construction. In *Proceedings of the 12th International Conference on Inductive Logic Programming*, pages 149–165, 2002.
- [15] J. Stecher, F. Janssen, and J. Fürnkranz. Separating rule refinement and rule selection heuristics in inductive rule learning. In *Proceedings of Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014*, pages 114–129, 2014.
- [16] S. Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery, PKDD 1997*, pages 78–87, 1997.