

The Advantages of Seed Examples in First-Order Multi-class Subgroup Discovery

Tarek Abudawood and Peter A. Flach¹

1 Introduction

Subgroup discovery is halfway between predictive and descriptive rule learning: while there is a target concept, the goal of subgroup discovery is not necessarily to achieve high accuracy in predicting the target, but rather to identify subsets of the population whose class distribution is significantly different from the overall distribution. The target concept helps us to achieve a trade-off between accuracy and interestingness.

In previous work [1] we illustrated the usefulness of several multi-class subgroup evaluation measures in producing highly predictive subgroups in a propositional logic framework. In this paper we upgrade multi-class subgroup discovery to first-order logic employing a new weighted covering algorithm that takes advantage of learning from seed examples. In a rigorous experimental evaluation we show that the use of seed examples leads to considerable and statistically significant improvement of predictive power, both in terms of accuracy (3 percent points on average over the data sets) and AUC (7 percent points).

The paper is structured as follows. Section 2 describes the ingredients of our Aleph-MSD++ algorithm. An empirical evaluation of multi-class subgroup discovery for feature construction over 11 data sets is presented in Section 3. Finally, we conclude the paper in Section 4.

2 First-Order Multi-class Subgroup Discovery with Aleph-MSD++

In this section we describe our main contribution, the Aleph-MSD++ first-order multi-class subgroup discovery algorithm. There are many inductive logic programming (ILP) systems, including FOIL [7], PROGOL [5] and GOLEM [6]; they are mostly restricted to binary classification. The starting point for our study was Aleph, a Prolog-based inductive logic programming rule learner that induces first-order classification rule models [8]. Aleph is similar to PROGOL but incorporates additional search strategies and some extra options. An important feature of Aleph (and PROGOL) that we use in this paper is the use of a *seed example* to start the search for the next rule.

We present a new algorithm which extends Aleph in a number of ways, the most important of which are: the use of a novel weighted covering algorithm based on seed examples; the incorporation of multi-class subgroup discovery search heuristics; and by implementing a significance test to be applied on nominated subgroups. These extensions are discussed in the following sections.

2.1 An Improved Weighted Covering Algorithm

Most ILP induction systems implement the covering algorithm where in every iteration the training examples covered by a newly induced rule are removed from the training set before the induction process proceeds. This approach clearly biases the rules induced later in the induction process due to lack of information regarding the examples already covered. The weighted covering algorithm comes as a suitable remedy to this problem. In weighted covering, each example is assigned initial weight and gets reduced every time the example is covered by a new induced rule [4].

The weight is decreased according to one of the following two schemata: additive or multiplicative. Let $w_t(x)$ denote the weight of example x after being covered by t number of rules: in the additive weight method we have $w_t(x) = \frac{1}{1+t}$ while in the multiplicative method $w_t(x) = \gamma^t$, with γ a parameter, $0 \leq \gamma \leq 1$ (higher γ means behaviour is more different from standard covering).

While in typical systems applying the weighted covering algorithm the weights of all covered examples are decreased according to the chosen weighting schema, in Aleph-MSD++ *only the weights of examples matching the class of the selected seed example are decreased*. Restricting the weight reduction in Aleph-MSD++ is a crucial point in our paper. It ensures that we can learn diverse subgroups representing examples from all the different classes. This issue is particularly important in the case of class imbalance since the minority classes may get ignored otherwise during the discovery process.

The advantages conveyed by seed examples are not available when using CN2-based classifiers, such as CN2-SD [4] and CN2-MSD [1]. We will demonstrate these advantages by comparing the new weighted covering algorithm Aleph-MSD++ against a CN2-like version called Aleph-MSD. In the latter algorithm, the weights of *all* training examples covered by a newly induced clause are reduced – not just the ones from the same class as the seed example. The experimental results in Section 3 show that the CN2-like approach leads to subgroups that are no more predictive than the rules learned by Aleph. In contrast, the use of seed examples in Aleph-MSD++ leads to considerable and statistically significant improvement of predictive power.

2.2 Other Improvements

We employed four of the six evaluation methods for subgroup discovery proposed in [1]. Those are Gini-Split (*GS*), Chi-Squared (*Chi*²), Mutual Information (*MI*) and Weighted Multiclass Weighted Relative Accuracy (*WMWRAcc*). The reader is referred to [1] for details. Seed examples are selected randomly from examples that have never been considered as seed examples before and are not covered by any subgroup induced from a seed example of the same class.

¹ Intelligent Systems Laboratory, University of Bristol, United Kingdom, email: Dawood@cs.bris.ac.uk, Peter.Flach@bristol.ac.uk

Learning is terminated according to one of the following conditions: if all examples were covered at least once or if all the remaining uncovered training examples were considered as seed examples even if none of the induced subgroups cover them. This is an advantage over [4] and [1] in the sense that there are no minimum evaluation thresholds imposed in Aleph-MSD++. After learning we apply χ^2 significance testing at significance level $p = 0.95$, similar as in CN2 [2]. This is important especially when using a weighted covering algorithm since the number of subgroups can be large as the examples covered by previously induced subgroups do not get removed.

3 Empirical Evaluation

In this section we study the behaviour of the proposed subgroup discovery algorithm and examine its usefulness for feature generation in a classification context. We used 8 first-order and 3 propositional data sets with our implementation of Aleph-MSD++ and also its variant Aleph-MSD where the latter uses the standard weighted covering algorithm. For the purpose of evaluating the quality and utility of the induced first-order subgroup rules, we used them as boolean features and applied a decision tree learner (J48, the Weka [9] implementation of C4.5). We also used the classification rules induced by standard Aleph using the Compression search heuristic as features for J48 in order to compare the performance and behaviour of subgroup discovery methods against a typical classification method.

We evaluated the performance of Aleph-MSD++ using the four search heuristics from Section 2.2 and the following six settings of the weighted covering algorithm (the first five concern multiplicative weights: (0) $\gamma = 0.00$ (unweighted covering); (1) $\gamma = 0.25$; (2) $\gamma = 0.50$; (3) $\gamma = 0.75$; (4) $\gamma = 1.00$ (no weight reduction); (5) additive weights).

For each data set, cross-validated results were recorded for various aspects (accuracy, AUC, number of considered first-order clauses and average size of a induced first-order clauses). We observed that the use of seed example-based weighted covering outperforms standard weighted covering both with respect to accuracy (3 percent points on average over the data sets) and AUC (7 percent points). Due to space limitations we will focus in the remainder of this section on the results achieved with Aleph-MSD++.

Since averaging across data sets has limited meaning as the values may not be commensurate, for each performance aspect we record the average values as well as its average rank (1 is best, 5 is worst). We use the Friedman Test (FT) on these average ranks at $p = 0.10$ with Bonferroni-Dunn post-hoc test to check significance against Compression method as a control learner. The FT records wins and losses in the form of ranks ignoring the magnitude of these wins and losses. FT is considered to be more appropriate when comparing multiple classifiers on multiple data sets. The reader is referred to [3] for more details about the Friedman Test.

A graphical illustration of the post-hoc test results on accuracy and AUC of the Aleph-MSD++'s subgroup discovery methods and the control method is given in Figure 1. For each weight setting the corresponding critical difference diagram is shown vertically (the lower the rank the higher the performance). The figure indicates a superiority of subgroup discovery methods over Compression in almost all weighted covering algorithm settings. This demonstrates the high quality of the subgroups induced in Aleph-MSD++. GS achieves the best accuracy in all weighted covering settings except setting 4 ($\gamma = 1$) where the weights of covered training examples are not decreased. A striking result on AUC is reported for χ^2 in setting 2

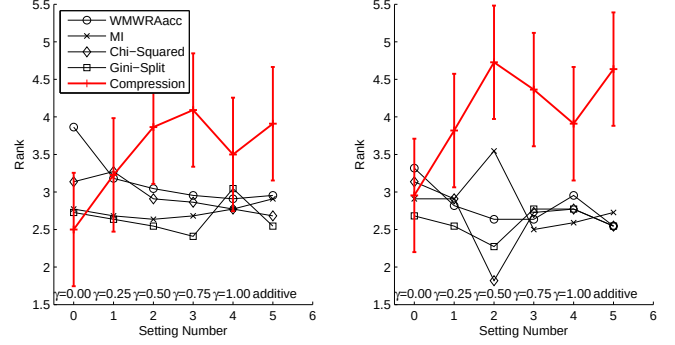


Figure 1. Average ranks achieved by Aleph-MSD++ with regard to accuracy and AUC (lower is better), using different search heuristics. The left-most setting is the default covering algorithm, while the other settings use different variants of weighted covering. Vertical lines show the critical difference with the control method (Compression): differences outside the critical difference are statistically significant.

($\gamma = 0.5$). It deserves to be mentioned that not removing the covered examples nor reducing their weights (setting 4) still achieves better accuracy and significantly better AUC than the control.

Apart from the predictive performance, Aleph-MSD++ in general considers 3 to 4 times fewer hypotheses which contributes to faster building of a small clausal theory.

4 Conclusion

In this paper we investigated multi-class subgroup discovery in inductive logic programming, where the induced subgroups are defined by conjunctions in first-order logic. We presented a novel algorithm, Aleph-MSD++, implementing a new weighted covering algorithm based on learning from seed examples. We demonstrated the usefulness and the high quality of the subgroups produced by this algorithm and compared it with the standard covering algorithm and also classification rules when used as features for a decision tree learner. We showed that our algorithm is four times faster and produce more accurate predictions in accuracy and AUC.

REFERENCES

- [1] Tarek Abudawood and Peter A. Flach, ‘Evaluation Measures for Multi-class Subgroup Discovery’, in *Proc. European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD’09)*, pp. 35–50. Springer, (2009).
- [2] Peter Clark and Robin Boswell, ‘Rule Induction with CN2: Some Recent Improvements’, in *Proc. European Conference on Machine Learning (ECML’91)*, pp. 151–163. Springer, (1991).
- [3] Janez Demšar, ‘Statistical Comparisons of Classifiers Over Multiple Data Sets’, *Journal of Machine Learning Research*, 7, 1–30, (2006).
- [4] Nada Lavrač, Branko Kavšek, Peter Flach, and Ljupco Todorovski, ‘Subgroup Discovery with CN2-SD’, *Journal of Machine Learning Research*, 5, 153–188, (2004).
- [5] Stephen Muggleton, ‘Inverse entailment and Progol’, *New Generation Computing*, 13(3), 245–286, (1995).
- [6] Stephen Muggleton and Cao Feng, ‘Efficient Induction Of Logic Programs’, in *New Generation Computing*. Academic Press, (1990).
- [7] J. Ross Quinlan and R. Mike Cameron-Jones, ‘FOIL: A Midterm Report’, in *Proc. European Conference on Machine Learning (ECML’93)*, pp. 3–20. Springer, (1993).
- [8] Ashwin Srinivasan. The Aleph manual, 1999.
- [9] Ian H. Witten and Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, San Francisco, 2nd edn., 2005.