Analysis of Example Weighting in Subgroup Discovery by Comparison of Three Algorithms on a Real-life Data Set

Branko Kavšek and Nada Lavrač

Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
branko.kavsek@ijs.si, nada.lavrac@ijs.si

Abstract. This paper investigates the implications of example weighting in subgroup discovery by comparing three state-of-the-art subgroup discovery algorithms, APRIORI-SD, CN2-SD, and SubgroupMiner on a real-life data set. While both APRIORI-SD and CN2-SD use example weighting in the process of subgroup discovery, SubgroupMiner does not. Moreover, APRIORI-SD uses example weighting in the post-processing step of selecting the 'best' rules, while CN2-SD uses example weighting during rule induction. The results of the application of the three subgroup discovery algorithms on a real-life data set – the UK Traffic challenge data set are presented in the form of ROC curves showing that APRIORI-SD slightly outperforms CN2-SD; both APRIORI-SD and CN2-SD are good in finding small and highly accurate subgroups (describing minority classes), while SubgroupMiner found larger and less accurate subgroups (describing the majority class). We show by using ROC analysis that these results are not surprising and can be attributed to example weighting, that 'pushes' the search in the space of potential subgroups towards discovering small and accurate subgroups.

1 Introduction

Standard rule learning algorithms are designed to construct classification and prediction rules [13, 2, 3]. In addition to this area of machine learning, referred to as *supervised learning* or *predictive induction*, developments in *descriptive induction* have recently gained much attention, in particular *association rule learning* (e.g., the APRIORI association rule learning algorithm [1]), *subgroup discovery* (e.g., the MIDOS subgroup discovery algorithm [5]), and other approaches to non-classificatory induction.

As in the MIDOS approach, a subgroup discovery task can be defined as follows: given a population of individuals and a property of those individuals we are interested in, find population subgroups that are statistically 'most interesting', e.g., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest [5].

Some of the questions on how to adapt standard classification rule learning approaches to subgroup discovery have already been addressed in [8, 11] and well-known rule learning algorithms APRIORI [1] and CN2 [2] have been adapted to subgroup discovery. Both adapted algorithms, APRIORI-SD [8] and CN2-SD [11] use example weighting in the process of subgroup discovery.

This paper continues the research conducted in [8] where it has been experimentally shown by comparison with rule learners APRIORI-C [7], RIPPER [4], and CN2 [2] on selected UCI data sets [15] that APRIORI-SD and CN2-SD are suitable and adequate algorithms for subgroup discovery. Here we investigate the implications of example weighting in subgroup discovery by comparing three subgroup discovery algorithms, APRIORI-SD [8], CN2-SD [11], and SubgroupMiner [10] on a real-life data set – the UK Traffic challenge data set. While APRIORI-SD uses example weighting in the post-processing step of selecting the 'best' rules, CN2-SD uses example weighting during rule induction. We used APRIORI-SD's way of weighting examples as the basis for presenting example weighting in this paper. The results of the application of the three subgroup discovery algorithms on the UK Traffic challenge data set are presented in the form of ROC curves and discussed in the example weighting framework.

This paper is organized as follows. Section 2 presents the background for this work: the post-processing step of the APRIORI-SD subgroup discovery algorithm with example weighting and the weighted relative accuracy quality measure. In Section 3 ROC space is presented as a tool for evaluating the performance of subgroup discovery. Section 4 presents the experimental comparison of APRIORI-SD, CN2-SD, and SubgroupMiner on a real-life UK Traffic challenge data set together with results in the form of ROC curves. In Section 5 the results from Section 4 are discussed and example weighting investigated. Section 6 concludes by summarizing the results and presenting plans for further work.

2 Background: The APRIORI-SD Algorithm

This section presents the backgrounds for our work. The complete description of the APRIORI-SD subgroup discovery algorithm is given in [8]. Here just parts, essential for the understanding of the rest of the paper, are described.

Section 2.1 presents APRIORI-SD's post-processing procedure of selecting 'best' rules. This description is extended in Sections 2.2 and 2.3 presenting APRIORI-SD's example weighting approach and the weighted relative accuracy quality function, respectively.

2.1 Post-processing procedure

The post-processing procedure of APRIORI-SD is performed as follows:

repeat

```
- sort rules from best to worst in terms of
  the weighted relative accuracy quality measure (see Section 2.3)
- decrease the weights of covered examples (see Section 2.2)
```

until
`- all the examples have been covered`
OR
`- there are no more rules`

2.2 Example weighting for best rule selection

The weighting scheme treats examples in such a way that covered positive examples are not deleted when the currently 'best' rule is selected in the post-processing step of the algorithm (like in the standard covering approach). Instead, each time a rule is selected, the algorithm stores with each example a count of how many times (with how many rules) the example has been covered so far. Initial weights of all positive examples e_j equal 1, $w(e_j, 0) = 1$, which denotes that the example has not been covered by any rule, meaning 'among the available rules select a rule which covers this example, as this example has not been covered by other rules', while lower weights mean 'do not try too hard on this example'.

Weights of positive examples covered by the selected rule decrease according to the formula $w(e_j, i) = \frac{1}{i+1}$. In the first iteration all target class examples are assigned the same weight $w(e_j, 0) = 1$, while in the following iterations the contributions of examples are inverse proportional to their coverage by previously selected rules. In this way the examples already covered by one or more selected rules decrease their weights while uncovered target class examples whose weights have not been decreased will have a greater chance to be covered in the following iterations.

2.3 The weighted relative accuracy measure

Weighted relative accuracy is used in subgroup discovery to evaluate the quality of induced rules. We use it instead of support when selecting the 'best' rules in the post-processing step.

We use the following notation. Let $n(X)$ stand for the number of examples covered by a rule $X \rightarrow Y$, $n(Y)$ stand for the number of examples of class Y, and $n(YX)$ stand for the number of correctly classified examples (true positives). We use $p(YX)$ etc. for the corresponding probabilities. Rule accuracy, or rule confidence in the terminology of association rule learning, is defined as $Acc(X \rightarrow Y) = p(Y|X) = \frac{p(YX)}{p(X)}$. Weighted relative accuracy [17] is defined as follows.

$$WRAcc(X \rightarrow Y) = p(X).(p(Y|X) - p(Y)). \tag{1}$$

Weighted relative accuracy consists of two components: generality $p(X)$, and relative accuracy $(p(Y|X) - p(Y))$. The second term, relative accuracy, is the accuracy gain relative to the fixed rule $true \rightarrow Y$. The latter rule predicts all instances to satisfy Y; rule $X \rightarrow Y$ is only interesting if it improves upon this 'default' accuracy. Another way of viewing relative accuracy is that it measures the utility of connecting rule body X with a given rule head Y. However, it is easy to obtain high relative accuracy with highly specific rules, i.e., rules with low generality $p(X)$. To this end, generality is used as a 'weight', so that weighted relative accuracy trades off generality of the rule ($p(X)$, i.e., rule coverage) and relative accuracy ($p(Y|X) - p(Y)$). All the probabilities in Equation 1 are estimated with relative frequencies e.g., $p(X) = \frac{n(X)}{N}$, where N is the number of all instances.

Modified *WRAcc* with example weights. The rule quality measure *WRAcc* used in APRIORI-SD was further modified to enable handling example weights, which provide the means to consider different parts of the instance space with each application of a selected rule (as described in Section 2.2).

The modified *WRAcc* measure is defined as follows:

$$WRAcc(X \rightarrow Y) = \frac{n'(X)}{N'}(\frac{n'(YX)}{n'(X)} - \frac{n(Y)}{N}). \qquad (2)$$

where N' is the sum of the weights of all examples, $n'(X)$ is the sum of the weights of all covered examples, and $n'(YX)$ is the sum of the weights of all correctly covered examples.

3 Evaluation in ROC Space

A point on the *ROC curve* (ROC: Receiver Operating Characteristic) [16] shows classifier performance in terms of false alarm or *false positive rate* $FPr = \frac{FP}{TN+FP} = \frac{FP}{Neg}$ (plotted on the X-axis; Neg standing for the number of all negative examples), and sensitivity or *true positive rate* $TPr = \frac{TP}{TP+FN} = \frac{TP}{Pos}$ (plotted on the Y-axis; Pos standing for the number of all positive examples). The confusion matrix shown in Table 1 defines the notions of TP (number of true positives), FP (number of false positives), TN (number of true negatives) and FN (number of false negatives), 'actual positive' (negative) denotes the number of examples in the training set that are (actually) positive (negative), and 'predicted positive' (negative) denotes the number of examples that the rule $X \rightarrow Y$ predicted as positive (negative).

Table 1. Confusion matrix.

	predicted positive	predicted negative
actual positive	TP	FN
actual negative	FP	TN

Applying the notation used in Section 2.3, FPr and TPr can be expressed as: $FPr = \frac{n(X\overline{Y})}{Neg}$, $TPr = \frac{n(XY)}{Pos}$. In ROC space, an appropriate tradeoff, determined by the expert, can be achieved by applying different algorithms, as well as by different parameter settings of a selected data mining algorithm or by taking into the account different misclassification costs. It has been shown in [16] that by constructing ROC convex hulls we can identify potentially optimal classifiers.

ROC space is appropriate for measuring the success of subgroup discovery, since subgroups whose TPr/FPr tradeoff is close to the main diagonal (line connecting the points $(0,0)$ and $(1,1)$ in the ROC space) can be discarded as insignificant. The reason is that the rules with TPr/FPr on the main diagonal have the same distribution of covered positives and negatives as the distribution in the entire data set. On the other hand, the further away from the main diagonal we go, the more the two distributions differ. In this way, subgroups lying far away from the main diagonal in ROC space represent potentially interesting subgroups (according to the distribution of covered positive and negatives examples).

By constructing a convex hull from subgroups represented in ROC space, we take into consideration mostly those subgroups that are far away from the main diagonal and are thus both potentially interesting and each of the rules representing a subgroup that lies on the convex hull represents a potentially optimal classifier. This fact doesn't imply that all the interesting subgroups are those lying on the ROC convex hull but we can identify most of the potentially interesting subgroups by analyzing those that lie on or near the ROC convex hull. Therefore, the use of ROC convex hull construction technique for evaluation of subgroup discovery results is limited to identifying potentially interesting subgroups as well as to limit their number. It is then left to the expert to select from potentially interesting those subgroups that are really interesting.

4 Comparative Analysis on a Real-life Data Set

This section provides results of the experimental evaluation on the real-life UK Traffic challenge data set, comparing the performance of subgroup discovery algorithms APRIORI-SD, CN2-SD, and SubgroupMiner.

Section 4.1 presents the description of the UK Traffic challenge data set. In Section 4.2 the results of the comparison are presented in ROC space.

4.1 Description of the data set

In order to compare APRIORI-SD [8], CN2-SD [11], and SubgroupMiner [10], we applied these algorithms to a real-life problem data set – the UK Traffic challenge data set. This data set is a sample of a larger and more complete relational data set – the UK Traffic data set briefly described below.

The UK Traffic data set. The UK Traffic data set includes the records of all the accidents that happened on the roads of Great Britain between years 1979 and 1999. It is a relational data set consisting of 3 related sets of data: the ACCIDENT data, the VEHICLE data and the CASUALTY data. The ACCIDENT data consists of the records of all accidents that happened in the given time period; VEHICLE data includes data about all the vehicles involved in these accidents; CASUALTY data includes the data about all the casualties involved in the accidents. Consider the following example: 'Two vehicles crashed in a traffic accident and three people were seriously injured in the crash'. In terms of the TRAFFIC data set this is recorded as one record in the ACCIDENT set, two records in the VEHICLE set and three records in the CASUALTY set. Every separate set is described by around 20 attributes and consists of more than 5 million records.

The UK Traffic challenge. The task of the challenge was to produce classification models (in our case subgroup descriptions) to predict skidding and overturning for accidents from the UK Traffic data set [14]. As the class attribute *Skidding and Overturning* appears in the VEHICLE data table, the data tables ACCIDENT and VEHICLE were merged in order to make this a simple non-relational problem. Furthermore a sample of 5940 records from this merged data table was selected for learning and another sample of 1585 records was selected for testing. The class attribute *Skidding and Overturning* has six possible values. The meaning of these values and the distribution of the class values in the training and test sets are shown in Table 2.

Table 2. The meaning and the distribution of classes in the UK Traffic challenge data set.

Code	Meaning of class values	Train (%)	Test (%)
0	No skidding, jack-knifing or overturning	64.26	64.67
1	Skidded	22.07	22.46
2	Skidded and overturned	7.27	6.88
3	Jack-knifed	0.20	0.06
4	Jack-knifed and overturned	0.19	0.44
5	Overturned	6.01	5.49

4.2 Experimental results

We compared the subgroup discovery algorithms APRIORI-SD [8], CN2-SD [11] and SubgroupMiner [10] by applying them to the UK Traffic challenge training data to construct subgroups and then test these subgroups on the test data. The results were plotted in ROC space. Because of the fact that only binary class problems can be plotted in ROC space, we had to transform the original problem of predicting a class with six values to six binary problems, predicting each class in turn as positive and the remaining classes as negative. All three subgroup discovery algorithms were run with default parameters (APRIORI-SD with minimal confidence 0 and minimal support 0.01; CN2-SD using the additive weighting scheme, 99% significance threshold and beam size 5; SubgroupMiner using beam size 10, max. length of rules 6 and suppression factor $\alpha = 1$).

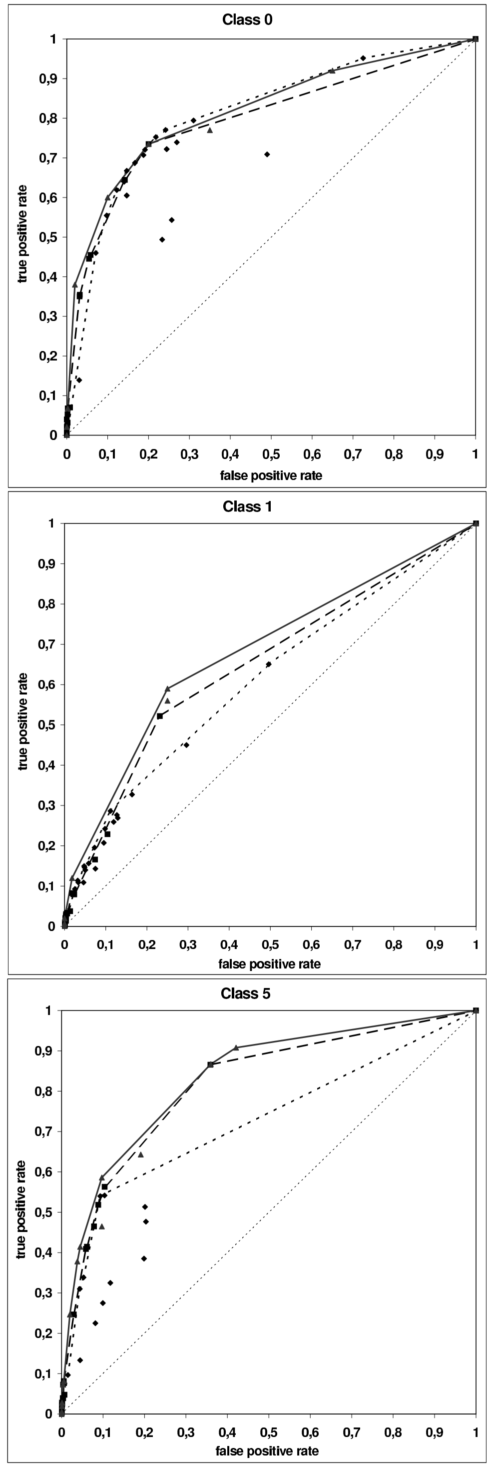

Fig. 1. The top, middle and bottom figures present the ROC curves (solid representing APRIORI-SD, dashed CN2-SD and dotted SubgroupMiner) for the problem of predicting Class 0, Class 1 and Class 5 respectively.

We discarded the problems of predicting Class 3 and Class 4 (see Table 2 for the meaning of class codes) because they contained too few test examples (see the distribution in Table 2). Furthermore, we omit the ROC plot for the problem of predicting Class 2 because it is very similar to the ROC plot for predicting Class 1 giving no additional information in comparing the subgroup discovery algorithms.

The results of the comparisons on the remaining three problems of predicting Class 0, Class 1 and Class 5 are shown in Figure 1. We can describe these problems as the problems of predicting the majority class (Class 0), the minority class (Class 5) and the class that is neither majority nor minority (Class 1).

We can see from the results shown in Figure 1 that:

1. Both APRIORI-SD and CN2-SD discovered smaller and more accurate subgroups (points near the point $(0, 0)$ in all three figures) than SubgroupMiner.
2. SubgroupMiner discovered larger but less accurate subgroups. This is especially true for the problem of predicting the majority class (Class 0 – Figure 1, top picture).
3. SubgroupMiner discovered a lot of subgroups that do not lie on the ROC convex hull (⧫ marks in Figure 1).
4. Both APRIORI-SD and CN2-SD discovered better subgroups (the distance from the diagonal $(0, 0) - (1, 1)$ is larger) when dealing with the problem of predicting a minority class (Figure 1, middle and bottom picture).
5. APRIORI-SD is 'better' than CN2-SD – its ROC curve is above the ROC curve of CN2-SD in all three cases.

5 Interpretation of the Results and Analysis of Example Weighting

Here we try to explain the results of the experiments by explaining each of the five findings of the previous subsection starting with the last one.

The fifth finding – APRIORI-SD being better than CN2-SD in all cases – can be explained by the fact that CN2-SD is bound to miss some 'good' subgroups by using heuristic search, while APRIORI-SD using exhaustive search takes into consideration all 'potentially good' subgroups. Note the difference between using example weighting in the induction and post-processing phase of subgroup discovery.

The fourth finding – both APRIORI-SD and CN2-SD discovered better subgroups when dealing with the problem of predicting a minority class – can be attributed to the *WRAcc* heuristic used in APRIORI-SD (in rule post-processing) and CN2-SD (used in heuristic beam search of rules). This result experimentally confirms the appropriateness of the *WRAcc* heuristic for subgroup discovery, which aims at finding subgroups maximizing the distance from the ROC diagonal [11].

The third finding – SubgroupMiner discovered a lot of subgroups that do not lie on the ROC convex hull – can also be attributed to the fact that the algorithms use different heuristics when searching the space of possible solutions (subgroups).

To explain the first two findings we use ROC isometrics described in [6]. With the help of ROC isometrics we can investigate the behavior of quality functions used by APRIORI-SD, CN2-SD and SubgroupMiner. APRIORI-SD and CN2-SD use the same quality function to find subgroups – *WRAcc* with example weights described in Section 2.3. The behavior of this quality function is depicted in Figure 2 (left-hand side) in the form of ROC isometrics (lines in the figure), each line representing some value of the quality function (see [6] for detailed description of ROC isometrics).

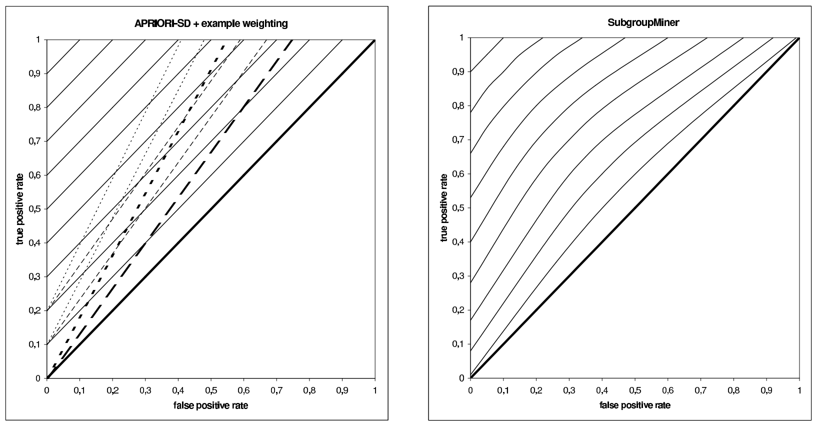

Fig. 2. The left-hand side figure shows ROC isometrics for the *WRAcc* quality function with example weights used in APRIORI-SD and CN2-SD. The right-hand side shows ROC isometrics for the SubgroupMiner's quality function.

Figure 2 (left-hand side) illustrates also the effect of weighting. Solid lines show the behavior of *WRAcc* without weights[1] (only iso-lines for positive *WRAcc* are shown). Dashed lines show the modified *WRAcc* with example weights for the case where all positive examples have a weight of 1/2. Dotted lines represent the same quality function in the case of all positive examples having a weight of 1/3. Only three iso-lines are shown in the cases of modified *WRAcc* with example weights 1/2 and 1/3 for the sake of clarity of the figure. Thick lines in Figure 2 denote that the value of the respective quality function equals 0. The behavior of SubgroupMiner's quality function is depicted in Figure 2 (right-hand side) [2].

[1] The weights of all examples equal 1.

[2] SubgroupMiner uses the following quality function to rank the subgroups during search: $Q(X \rightarrow Y) = \frac{p(Y|X)-p(Y)}{\sqrt{p(Y).(1-p(Y))}}\sqrt{n(X)}\sqrt{\frac{N}{N-n(X)}}$ A more detailed analysis of the use of quality function in subgroup discovery can be found in [9].

We can now proceed explaining the first two findings from the results by looking at Figure 2. Figure 2 (left-hand side) shows that the *WRAcc* quality function with example weights used by APRIORI-SD and CN2-SD 'tries harder' to discover more accurate subgroups – lowering the weights on positive examples makes the lines in the figure more vertical. Since there are no large subgroups that are at the same time highly accurate, the effect of weighting in our case results in finding small, highly accurate subgroups (explanation of the first finding – subgroups near the point $(0,0)$ in Figure 1). The second finding can be explained by looking at Figure 2 (right-hand side). We can see that SubgroupMiner's quality function tends to discover small and accurate subgroups and at the same time large and inaccurate ones (note the bending of iso-lines towards the points $(0,0)$ and $(1,1)$ in ROC space). The latter fact (finding large and inaccurate subgroups that both CN2-SD and APRIORI-SD 'disregard' because of the use of example weighting – see the left-hand side of Figure 1) explains the second finding from the results. Why did SubgroupMiner not discover small and accurate subgroups (in such a number as APRIORI-SD and CN2-SD did) can again be attributed to the heuristics used by the algorithms in searching the space of potentially 'good' subgroups but mostly to the suppression of overlapping subgroups that SubgroupMiner uses.

Let us illustrate the effect of weighting on an example.

Illustration of weighting on the example of predicting Class 0. We illustrate the effect of weighting by explaining step-by-step the discovery of subgroups by APRIORI-SD on the example of predicting Class 0 in the UK Traffic challenge data set (see Figure 1 (top picture) in Section 4.1). We explain the discovery of the first three subgroups. The procedure, illustrated in Figure 3 (which equals Figure 1 (top picture) with ROC curves for CN2-SD and SubgroupMiner removed), goes as follows:

- All the examples have a weight of 1. APRIORI-SD selects the 'best' subgroup – the subgroup with the maximal value of *WRAcc*. In Figure 3 the subgroup is depicted as a small triangle in the point $(0.65, 0.92)$. The *WRAcc* value of the subgroup is 0.062 (the absolute maximum for *WRAcc* being 0.23). The solid line going through the subgroup is the iso-line for $WRAcc = 0.062$. The thick solid line represents the iso-line for $WRAcc = 0$. The weights of positive examples covered by the subgroup are lowered to the value of $1/2$.
- The calculation of *WRAcc* is recomputed taking into account the new weights. Again the 'best' subgroup is selected. In the figure this newly selected subgroup is depicted as a small triangle in the point $(0.35, 0.77)$. The modified *WRAcc* value of the subgroup is 0.02 (the absolute maximum for the modified *WRAcc* being 0.13). The meaning of the lines is the same as before, only this time the lines are dashed (large dash). The weights of all positive examples covered by the newly selected subgroup are reduced (from 1 to $1/2$ and from $1/2$ to $1/3$).
- Again the calculation of *WRAcc* is recomputed taking into account the new weights and again the 'best' subgroup is selected. In the figure the new sub-

group is depicted as a small triangle in the point (0.20, 0.73). The modified *WRAcc* value of the new subgroup is 0.007 (the absolute maximum for the modified *WRAcc* being 0.07). Dashed lines (small dash) are used to depict the iso-lines. The weights of positive examples covered by the new subgroup are again reduced (from 1 to 1/2, from 1/2 to 1/3 and from 1/3 to 1/4) and the algorithm is run iteratively until all the subgroups are discovered.

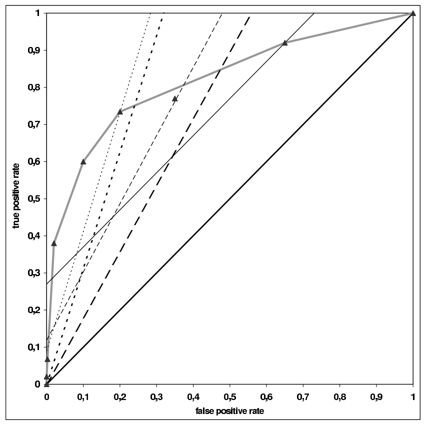

Fig. 3. The effect of example weighting used in APRIORI-SD illustrated on the problem of discovering subgroups for predicting Class 0.

6 Conclusions

The comparison of subgroup discovery algorithms APRIORI-SD, CN2-SD, and SubgroupMiner on a real-life UK Traffic challenge data set shows that APRIORI-SD performs very similarly to (but slightly better than) CN2-SD; both are suitable for finding small highly accurate subgroups describing minority classes, while SubgroupMiner finds larger and less accurate subgroups when dealing with classes containing the majority of the examples.

We have shown by using ROC analysis that the above findings are caused by example weighting, that ‘pushes’ the search in the space of potential subgroups towards discovering small and accurate subgroups. Moreover, using example weighting in the post-processing stage in subgroup discovery results in slightly better results than using it during induction.

The downside of the example weighting used by APRIORI-SD and CN2-SD is that it restricts the search space of potential subgroups. It is very unlikely for an algorithm that uses example weighting to find large and less accurate subgroups that might be sometimes preferred to small and accurate ones. Can the example weighting be modified to find these subgroups?

Moreover, example weighting is just one of the aspects of subgroup discovery. Another important aspect of subgroup discovery, which is neglected in our study, is the degree of overlap between the discovered subgroups. The challenge for our further research is to propose extensions of the weighted relative accuracy heuristic and ROC space evaluation metrics that will take into account the overlap between subgroups.

Another issue, left for further work, is the expert evaluation of the results. Does weighting result in the discovery of subgroups that are really interesting for an expert?

Acknowledgements

The work reported in this paper was supported by the Slovenian Ministry of Education, Science and Sport. We acknowledge also the support of the cInQ (Consortium on discovering knowledge with Inductive Queries) project, funded by the European Commission under contract IST-2000-26469.

References

1. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proceedings of the 20th International Conference on Very Large Databases*, Santiago, Chile, 1994. pp. 207–216.
2. P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In Y. Kodratoff, editor, *Proc. of the 5th European Working Session on Learning*, 151–163. Springer, 1991.
3. W.W. Cohen. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, 1995. pp. 115–123.
4. W.W. Cohen and Y. Singer. A simple, fast, and effective rule learner. In *Proc. of AAAI/IAAI*, 335–342. American Association for Artificial Intelligence, 1999.
5. S. Wrobel. Inductive Logic Programming for Knowledge Discovery in Databases. Chapter 4 in: S. Džeroski, N. Lavrač. (Eds). *Relational Data Mining*, Springer, 2001. pp. 74–101.
6. P. Flach. The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. In *Proceedings of the Twentieth International Conference on Machine Learning*, 194–201. AAAI Press, 2003.
7. V. Jovanoski, N. Lavrač. Classification Rule Learning with APRIORI-C. In P. Brazdil, A. Jorge (Eds.): *10th Portuguese Conference on Artificial Intelligence*, EPIA 2001, Porto, Portugal, 2001. pp. 44–51.
8. B. Kavšek, N. Lavrač, and V. Jovanoski. APRIORI-SD: Adapting association rule learning to subgroup discovery. In *Proceedings of the Fifth International Symposium on Intelligent Data Analysis*, IDA 2003, Berlin, Germany, 2003. pp. 230–241.
9. W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In U.M. Fayyad, G. Piatetski-Shapiro, P. Smyth and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, 249–271. MIT Press, 1996.
10. W. Klösgen and M. May. Census data mining – an application. In *Proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Helsinki, Finland, 2002.

11. N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5: 153–188, 2004.
12. B. Liu, W. Hsu and Y. Ma. Integrating Classification and Association Rule Mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining KDD'98*, New York, USA, 1998. pp. 80–86.
13. R.S. Michalski, I. Mozetič, J. Hong, and N. Lavrač. The multi-purpose incremental learning system AQ15 and its testing application on three medical domains. In *Proc. 5th National Conference on Artificial Intelligence*, Morgan Kaufmann, 1986. pp. 1041–1045.
14. D. Mladenić and N. Lavrač (eds). Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise: Results of the Sol-Eu-Net project: January 2000–March 2003, Ljubljana: DZS, 2003.
15. P.M. Murphy and D.W. Aha. *UCI repository of machine learning databases* [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1994.
16. F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3): 203–231, 2001.
17. L. Todorovski, P. Flach, and N. Lavrač. Predictive performance of weighted relative accuracy. In D.A. Zighed, J. Komorowski, and J. Zytkow, editors, *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, Springer, 2000. pp. 255–264.