ROC Analysis of Example Weighting in Subgroup Discovery

Branko Kavšek and **Nada Lavrač** and **Ljupčo Todorovski**[1]

Abstract. This paper presents two new ways of example weighting for subgroup discovery. The proposed example weighting schemes are applicable to any subgroup discovery algorithm that uses the weighted covering approach to discover interesting subgroups in data. To show the implications that the new example weighting schemes have on subgroup discovery, they were implemented in the APRIORI-SD algorithm. ROC analysis was then used to study their behavior, and the behavior of APRIORI-SD's original example weighting scheme, both theoretically and practically, by application on the UK Traffic challenge data set. The findings show that the proposed example weighting schemes are a valid alternative to APRIORI-SD's original example weighting scheme when the goal is to discover fewer subgroups that are either small and highly accurate or large and less accurate.

1 INTRODUCTION

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 [15].

Many algorithms for discovering such subgroups have been developed in the last couple of decades [7, 15, 4, 6, 10]. Differently from the older subgroup discovery algorithms such as EXPLORA [7] or MIDOS [15], the new algorithms (SD [4], CN2-SD [10], APRIORI-SD [6]) all tend to use the example weighting (or weighted covering of examples) to discover interesting subgroups in the data.

In recent years ROC (Receiver Operating Characteristic) analysis [13] has gained much attention in data mining and has proved to be a useful tool for analyzing the behavior of subgroup discovery algorithms [2, 3].

This paper uses ROC analysis to investigate example weighting in subgroup discovery and presents two new ways of example weighting. The proposed example weighting schemes are applicable to any subgroup discovery algorithm that uses the weighted covering approach to discover interesting subgroups in data [6, 10, 4]. The APRIORI-SD algorithm is used as an implementation platform to show the implications that the new example weighting schemes have on subgroup discovery. ROC analysis was then used to study their behavior, and the behavior of APRIORI-SD's original example weighting scheme, both theoretically and practically, by application on the UK Traffic challenge data set.

[1] Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
email: {branko.kavsek,nada.lavrac,ljupco.todorovski}@ijs.si

This paper is organized as follows. In Section 2 the background for this work is briefly explained: the APRIORI-SD subgroup discovery algorithm with emphasis on the post-processing step of selecting the most interesting subgroups by using the weighted covering approach. Section 3 presents the theoretical analysis of example weighting using ROC isometrics [2, 3] together with the proposal of two new weighting schemes. In Section 4 the two weighting schemes are shown "in action", applied to a real-life data set. The results are depicted in ROC space and their characteristics discussed. Section 5 concludes by summarizing the findings and giving directions for further work.

2 BACKGROUND: THE APRIORI-SD ALGORITHM

This section presents the backgrounds for our work. We describe the APRIORI-SD subgroup discovery algorithm [6] by briefly describing the algorithms from which it was derived – the association rule learner APRIORI [1] and the classification rule learner APRIORI-C [5] – together with the modifications needed to transform these algorithms into APRIORI-SD.

2.1 Association rule basics

The very basics of association rules are presented here together with a brief description of the relation between association and classification rules.

APRIORI [1] is perhaps the most well known association rule learning algorithm that was extensively studied, adopted to other areas of machine learning and data mining, and successfully applied in many problem domains. This is why it will not be described in detail here; just the basics of association rules will be given.

An association rule has the following form:

$$X \rightarrow Y \tag{1}$$

where $X, Y \subset I$, X and Y are itemsets, and I is the set of all items.

The quality of each association rule is defined by its *confidence* and *support*. *Confidence* of a rule is an estimate of the conditional probability of Y given X: $p(Y|X)$. *Support* of a rule is an estimate of the probability of itemset $X \cup Y$: $p(XY)$. Confidence and support are computed as follows:

$$Confidence(X \rightarrow Y) = \frac{n(XY)}{n(X)} = \frac{p(XY)}{p(X)} = p(Y|X) \tag{2}$$

$$Support(X \rightarrow Y) = \frac{n(XY)}{N} = p(XY) \tag{3}$$

where $n(X)$ is the number of transactions that include itemset X, $n(XY)$ is the number of transactions that include both itemsets X and Y, and N is the number of all the transactions ($p(.)$ are the corresponding probabilities).

In the standard rule learning terminology, items are features (attribute values) and itemsets are conjunctions of features. In classification rules the right-hand side (Y) of a rule consists always of the class value and the left-hand side (X) is the conjunction of attribute value pairs. Confidence $p(Y|X)$ in association rule learning is called rule accuracy in classification rule learning.

2.2 The APRIORI-C algorithm

Here the adaptation of the association rule learner APRIORI to the classification rule learner APRIORI-C is briefly described.

The idea of using association rules for classification has been presented in [11]. The main advantage of APRIORI-C over its predecessors is lower memory consumption, decreased time complexity and improved understandability of results. The reader can find a detailed description of the APRIORI-C algorithm in [5]. We describe here just the parts of APRIORI-C that are essential for the understanding of the derived APRIORI-SD algorithm.

The APRIORI-C classification rule learning algorithm is derived from the association rule learning algorithm APRIORI which was adapted for classification purposes by implementing the following steps: (1) discretization of continuous attributes, (2) binarization of all (discrete) attributes, (3) running the APRIORI algorithm by taking in consideration only rules whose right-hand side consists of a single item, representing a value of the class attribute, (4) post-processing this set of rules to select the best among them and (5) using these rules to classify unclassified examples.

These steps of the APRIORI-C algorithm, as well as the approaches to feature subset selection, are described in detail in [5]. Here we describe just the fourth step, the post-processing of rules as it is crucial for the understanding of the derived APRIORI-SD algorithm.

2.2.1 Post-processing by rule subset selection

The APRIORI-C algorithm induces rules according to the parameters *minimal confidence* and *minimal support* of a rule [5]. The setting of these two parameters is often such that the algorithm induces a large number of rules, which hinders the understandability of the induced ruleset. A way to avoid this problem is to select just some best rules among all the induced rules. APRIORI-C has three ways of selecting such best rules:

Use N best rules. The algorithm first selects the best rule (the rule having the highest support), then eliminates all the covered examples, sorts the remaining rules according to support and repeats this procedure. The procedure is repeated until N rules[2] are selected or there are no more rules to select or there are no more examples to cover. The algorithm then stops and returns the classifier in the form of an IF-THEN-ELSE rule list.

Use N best rules for each class. The algorithm behaves in a similar way as in the 'use N best rules' case, selecting N best rules for each class (if so many rules exist for each class). This way the rules for the minority class will also find their way into the classifier.

Use example weighting to select the best rules. The algorithm again behaves in a similar way as 'use N best rules'. The difference is that covered examples are not eliminated, but instead their weights are decreased. They are then eliminated when their weight falls below a certain threshold (e.g., when an example has been covered more than k times[3]). The details of the weighting scheme are given in Section 2.3, describing APRIORI-SD.

2.3 APRIORI-SD

The main guidelines for adapting the classification rule learning algorithm APRIORI-C to the subgroup discovery algorithm APRIORI-SD are presented here. The details about the APRIORI-SD algorithm can be found in [6].

The main modifications of the APRIORI-C algorithm, making it appropriate for subgroup discovery, involve the implementation of:

- a new weighting scheme in post-processing,
- a different rule quality function (the weighted relative accuracy) and
- the probabilistic classification of unclassified examples.

The description of the probabilistic classification of unclassified examples is omitted here as it is not necessary for the understanding of the rest of this paper. The complete description of APRIORI-SD can be found in [6].

2.3.1 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 measure
  (see Sections 2.3.3 and 2.3.4)
- decrease the weights of covered examples
  (see Section 2.3.2)
until
- there are no more examples to cover
  OR
- there are no more 'good' rules
```

2.3.2 Example weighting in best rule selection

In the 'use example weighting to select best rules' post-processing method of APRIORI-C described in Section 2.2.1, the examples covered by the current best rule are not eliminated; instead they are re-weighted.

The weighting scheme treats examples in such a way that covered examples are not deleted when the currently 'best' rule is selected in the post-processing step of the algorithm. 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 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'.

[2] N is a user defined parameter.

[3] The default value of $k = 5$ was used throughout this paper.

Weights of 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[4]. 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.

2.3.3 The weighted relative accuracy measure

Weighted relative accuracy (*WRAcc*), proposed in [9] as an alternative to classification 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 $Accuracy(X \rightarrow Y) = p(Y|X) = \frac{p(YX)}{p(X)}$. Weighted relative accuracy [9, 14] is defined as follows.

$$WRAcc(X \rightarrow Y) = p(X).(p(Y|X) - p(Y)) \quad (4)$$

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 4 are estimated with relative frequencies e.g., $p(X) = \frac{n(X)}{N}$, where N is the number of all instances.

2.3.4 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 2.3.2).

The modified *wWRAcc* measure is defined as follows:

$$wWRAcc(X \rightarrow Y) = \frac{n'(X)}{N'} \left(\frac{n'(YX)}{n'(X)} - \frac{n'(Y)}{N'} \right) \quad (5)$$

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.

[4] The examples are eventually deleted when the value of i exceeds a certain user-defined threshold k.

3 ROC ANALYSIS OF EXAMPLE WEIGHTING

This section shows how the effects of the *WRAcc* quality function together with example weighting in APRIORI-SD can be analyzed in ROC (Receiver Operating Characteristic) space [13] using the guidelines from [2, 3].

The section consists of three parts. In the first part ROC space and its usefulness for subgroup discovery is presented. The second part presents the analysis of the *WRAcc* quality function together with example weighting in APRIORI-SD. In the third part two new weighting schemes for APRIORI-SD are presented and analyzed in ROC space.

3.1 ROC space and subgroup discovery

ROC space [13] is a 2-dimensional space that shows a classifier's (rule's/subgroup's in our case) 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) that needs to be minimized, 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) that needs to be maximized. The confusion matrix shown in Table 1 defines the notions of TP (true positives), FP (false positives), TN (true negatives) and FN (false negatives).

Table 1. Confusion matrix.

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

Applying the notation used to define *confidence* and *support* (see Equation 2) FPr and TPr can be expressed as: $FPr = \frac{n(X\overline{Y})}{Neg}$, $TPr = \frac{n(XY)}{Pos}$. In the 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. The 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.

3.2 Analysis of *WRAcc* and example weighting

Following the guidelines from [2, 3], we used the isometrics in ROC space to represent the *WRAcc* quality function defined by Equation 4 in Section 2.3.3. A ROC isometric is a line in ROC space connecting points with equal value of the quality function. In the case of the *WRAcc* function the ROC isometrics are parallel to the main diagonal in ROC space.

In fact, the definition of *WRAcc* (Equation 4) can be rewritten in terms of TPr and FPr as $WRAcc(X \rightarrow Y) = p(Y).(1 - p(Y)).(TPr - FPr)$ [8], hence an iso–*WRAcc*–line is defined by $TPr = \frac{WRAcc(X \rightarrow Y)}{p(Y).(1-p(Y))} + FPr$.

In Figure 1 this main diagonal (line connecting the points $(0, 0)$ and $(1, 1)$ in the ROC space) is denoted with a thicker line and represents the ROC isometric for the value 0 of the *WRAcc* function.

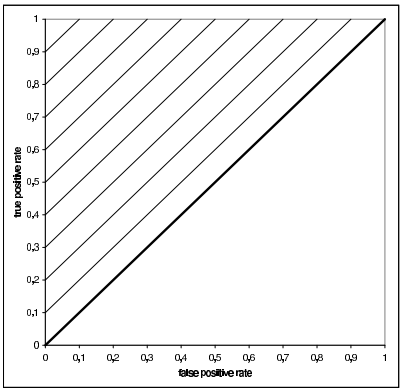

Figure 1. ROC isometrics for the *WRAcc* quality function.

Points lying on this diagonal represent subgroups with the same distribution of positive and negative examples as in the entire data set. Above the main diagonal *WRAcc* has a positive value, while below it is negative. The further away a point is from the main diagonal towards the point $(1, 1)$ in ROC space, the higher the value of the *WRAcc* function in that point (we do not take into account points with negative *WRAcc* as those points represent subgroups with the proportion of positives that is smaller than that of the entire data set).

The point $(1, 1)$ in ROC space is sometimes referred to as "ROC heaven" because it represents the subgroup covering all the positives and none of the negatives. It is also the point in which *WRAcc* reaches its maximum value.

By using *WRAcc* as its quality function, APRIORI-SD tries to find subgroups that are as far as possible from the main diagonal in ROC space and at the same time as close as possible to the point $(1, 1)$.

What happens when we take into consideration the example weighting? Do the ROC isometrics in Figure 1 change? When we add example weights to the *WRAcc* function, we obtain the modified *wWRAcc* function defined by Equation 5 in Section 2.3.4. All three terms of *wWRAcc*: $\frac{n'(X)}{N'}$, $\frac{n'(YX)}{n'(X)}$ and $\frac{n'(Y)}{N'}$ include example weights both in the numerator and denominator of the fraction. In this way when example weights are decreased, both the values of the numerator and denominator decrease keeping the value of *wWRAcc* somehow 'balanced'.

We can observe the effect of example weighting by analyzing the *wWRAcc* function after the discovery of the first rule. Equation 6 shows the right–hand side of Equation 5: the *wWRAcc* function after the weights of examples covered by the first rule have been decreased from 1 to $1/2$.

$$\frac{n(X)/2}{N - n(X)/2}\left(\frac{n(YX)/2}{n(X)/2} - \frac{n(Y) - n(YX)/2}{N - n(X)/2}\right) \qquad (6)$$

Equation 6 shows that the number of examples covered by the rule ($n(X)$) affects the angle of the iso–*wWRAcc*–lines: the more examples covered, the lower the angle, and vice versa.

3.3 New example weighting schemes for APRIORI-SD

Can we 'push' *wWRAcc* out of 'balance' and thus change its ROC isometrics independently of rule coverage? Let's see how this can be done. The third term in the definition of *wWRAcc* (Equation 5 in Section 2.3.4): $\frac{n'(Y)}{N'}$ represents the portion of positive examples (Y) in the whole population. The problem is that when the example weights change, the value of this term changes too keeping the equation 'balanced'. Let's see what happens if we replace this term by the same term from the original definition of *WRAcc* (Equation 4 in Section 2.3.3): $\frac{n(Y)}{N}$. The new *wWRAcc'* is defined as follows:

$$wWRAcc'(X \rightarrow Y) = \frac{n'(X)}{N'}\left(\frac{n'(YX)}{n'(X)} - \frac{n(Y)}{N}\right) \qquad (7)$$

By this term replacement we forced *wWRAcc'* to reflect the improvement of a subgroup's (weighted) accuracy with respect to the accuracy of the default rule ($true \rightarrow Y$) in the original population.

The *wWRAcc'* is 'unbalanced' with respect to example weights meaning that its ROC isometrics change when the example weights change (independently of rule coverage) as shown in Figure 2.

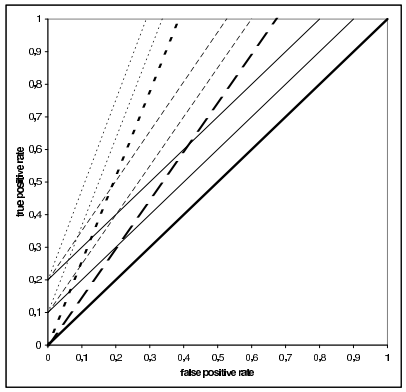

Figure 2. ROC isometrics – the effect of weighting (just positive) examples on the *wWRAcc'* quality function.

Figure 2 shows how ROC isometric lines change from solid to dashed to dotted when the weights of (positive[5]) examples decrease (thick lines in the figure denote ROC isometrics for the value 0 of the *wWRAcc'* function). In general, while still remaining parallel, the angle of ROC isometrics for *wWRAcc'* increases with the decrease of the weights of (positive) examples.

The new *wWRAcc'* quality function gives way to two new weighting schemes that can be used in conjunction with it.

3.3.1 wWRAcc' by weighting just positive examples

The weighting scheme described here is very similar to the one used by the original APRIORI-SD algorithm (see Section 2.3.2) with the difference that in this new scheme only the covered positive examples are re-weighted. The original APRIORI-SD's weighting scheme re-weights all the covered examples. The behavior of this new weighting scheme in conjunction with the *wWRAcc'* quality function is shown in Figure 2. APRIORI-SD's weighting scheme would behave very similarly to the new scheme if used in conjunction with the *wWRAcc'* function with the difference that the increase of angle of the ROC isometrics with the decrease of example weights would be less drastic then in the case of the new weighting scheme.

As seen in Figure 2, APRIORI-SD with this weighting scheme will tend to discover more accurate subgroups – ROC isometrics tend to become more and more vertical with the decrease of example weights thus 'pushing' the search to an area that contains subgroups with few negative examples.

[5] We will see later, in Section 3.3.2, that by decreasing the weights of negative examples the picture changes. However, by decreasing the weights of all covered examples we decrease the weights of mostly positives, because the algorithm is trying hard to cover as much positives and at the same time as few negatives as possible.

3.3.2 *wWRAcc' by weighting just negative examples*

The behavior of the weighting scheme described here is shown in Figure 3. The figure shows that by decreasing the weights only to the covered negative examples, the angle of the ROC isometrics decreases with the decrease of example weights behaving just the opposite as the previous weighting scheme.

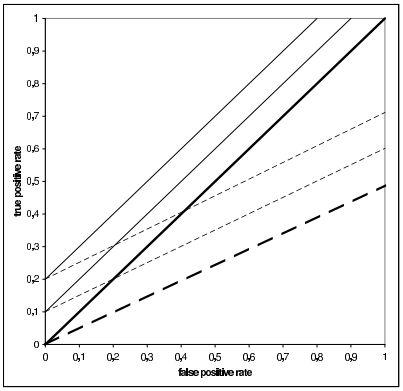

Figure 3. ROC isometrics – the effect of weighting just negative examples on the *wWRAcc'* quality function.

It can be clearly seen in Figure 3 that by decreasing the weights of covered negative examples the lower angle of the ROC isometrics allows the algorithm to find suboptimal subgroups – subgroups with positive *wWRAcc'* lying under the main diagonal in the ROC space.

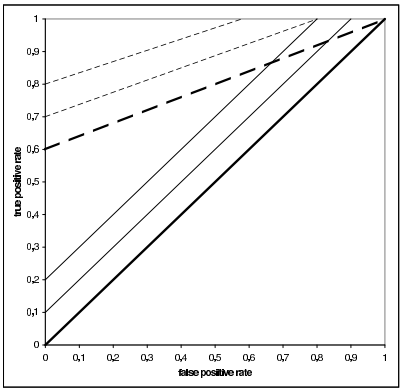

Figure 4. ROC isometrics – the corrected weighting of negatives and its effect on the *wWRAcc'* quality function.

Since we wanted subgroups with positive value of the new quality function to always lie above the main diagonal we had to 'push' the ROC isometrics (dashed lines in Figure 3) above it. We achieved this by subtracting the value of *wWRAcc'* for the default rule ($true \rightarrow Y$) from the *wWRAcc'* value of the subgroup[6]. The corrected *wWRAcc'* is expressed as follows:

$$wWRAcc'(X \rightarrow Y) - wWRAcc'(true \rightarrow Y) \qquad (8)$$

Its behavior is shown in Figure 4. As shown by this figure, APRIORI-SD with this weighting scheme will tend to discover larger subgroups – ROC isometrics tend to become more and more horizontal with the decrease of example weights thus 'pushing' the search to an area that contains subgroups which cover a large number of examples.

[6] Note that we could have achieved the same effect by changing the threshold from 0 to $wWRAcc'(true \rightarrow Y)$.

4 EXAMPLE WEIGHTING IN ACTION

After analyzing the proposed weighting schemes theoretically in ROC space (Section 3) this section presents their application in practice – on a real-life data set.

This section consists of three parts. The first part describes the real-life data set used for the analysis – the UK Traffic challenge data set. In the second part the parameters of the algorithm and performance measures used are described. The third part presents the results of the analysis and provides a brief interpretation.

4.1 Description of the data set

The real-life data set used in our experiments is the UK Traffic challenge data set [12]. This data set is a sample of a larger and more complete relational data set - the UK Traffic data set. Both data sets are briefly described below.

4.1.1 *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.

4.1.2 *The UK Traffic challenge data set*

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 [12]. 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 value	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 Application of APRIORI-SD

We applied APRIORI-SD with default parameters (*minimal confidence* $= 0$, *minimal support* $= 0.0001$, *minimal WRAcc* $= 0$, $k = 5$ and *maximal no. of terms in a subgroup* $= 10$) on the training set of 5940 examples. The algorithm was run 18 times (6 times to discover subgroups for each of the class values – one was always set as positive and the other five as negative; 3 times for each of the weighting schemes).

The performance measures used in the comparisons were: the number of discovered subgroups on the training set, the average accuracy of a subgroup on the test set (of 1585 examples) and the average size of a subgroup on the (same) test set[7].

4.3 Results

The results are shown in Table 3 and confirm the theoretical findings from Section 3. We can see from this table that when using *wWRAcc'* and weighting just positive examples, the algorithm finds subgroups that are on the average smaller and more accurate. On the other hand, by using (corrected) *wWRAcc'* and weighting just negative examples, on the average larger and less accurate subgroups are discovered by the algorithm.

Another thing that can be seen from Table 3 is that by using APRIORI-SD's original weighting scheme, more subgroups are discovered than when one of the new weighting scheme is used.

Table 3. The results of applying APRIORI-SD with different weighting schemes on the UK Traffic challenge data.

	performance measures								
	Accuracy			Size			#Subgroups		
Code	□	+	−	□	+	−	□	+	−
0	0.875	0.901	0.823	0.231	0.213	0.402	112	91	19
1	0.449	0.502	0.397	0.101	0.076	0.183	83	74	12
2	0.101	0.124	0.090	0.050	0.041	0.101	20	15	6
3	0.023	0.023	—	0.005	0.005	—	3	3	0
4	0.035	0.040	0.028	0.011	0.007	0.019	6	5	2
5	0.203	0.251	0.183	0.088	0.076	0.205	31	25	8

□ – original APRIORI-SD's weighting scheme
+ – *wWRAcc'* by weighting just positive examples
− – *wWRAcc'* by weighting just negative examples

5 CONCLUSIONS

Following the ideas presented in [2, 3] we used ROC analysis to study the behavior of APRIORI-SD's example weighting scheme. We proposed two new weighting schemes, implemented them in APRIORI-SD and studied their behavior both theoretically by means of ROC analysis and practically by application to a real-life data set.

The theoretical studies of the new weighting schemes point out that while the first scheme (*wWRAcc'* by weighting just positive examples) restricts to the search of highly accurate subgroups, the second one (*wWRAcc'* by weighting just negative examples) restricts to the search of highly general subgroups.

The results of the application to a real-life data set confirm the theoretical findings in practice. Moreover, the practical results refine the theoretical findings showing that the first weighting scheme finds small, highly accurate subgroups, while the second weighting scheme finds large, less accurate subgroups. Another finding discovered in practice is that both new weighting schemes discover a smaller number of subgroups than APRIORI-SD's original example weighting scheme. Thus, we could say that the new weighting schemes are more focused, while APRIORI-SD's original weighting scheme is more general.

We implemented the new weighting schemes in APRIORI-SD, but they can be implemented in any subgroup discovery algorithm that uses the weighted covering approach (e.g. SD [4] or CN2-SD [10]).

The implementation of the weighted schemes presented in this paper in other subgroup discovery algorithms is left for further work. Another issue for further work is the expert's analysis of the approaches presented in this paper.

[7] The accuracy and size of a single subgroup are computed as: Accuracy=$n(YX)/n(X)$, Size=$n(X)/N$, where all the terms in the equations refer to numbers of examples in the test set.

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, T. Imielinski, and R. Shrikant, 'Mining association rules between sets of items in large databases', in *Proceedings of ACM SIGMOD Conference on Management of Data*, pp. 207–216, (1993).
[2] P.A. Flach, 'The geometry of ROC space: Understanding machine learning metrics through ROC isometrics', in *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 194–201, AAAI Press, (2003).
[3] J. Fürnkranz and P.A. Flach, 'An analysis of rule evaluation metrics', in *Proceedings of the Twetieth International Conference on Machine Learning*, pp. 202–209, AAAI Press, (2003).
[4] D. Gamberger and N. Lavrač, 'Expert guided subgroup discovery: Methodology and application', *Journal of Artificial Intelligence Research*, **17**, 501–527, (2002).
[5] V. Jovanoski and N. Lavrač, 'Classification rule learning with APRIORI-C', in *Progress in Artificial Intelligence: Proceedings of the Tenth Portuguese Conference on Artificial Intelligence*, pp. 44–51, Springer, (2001).
[6] 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*, pp. 230–241, Springer, (2003).
[7] W. Klösgen, 'Explora: A multipattern and multistrategy discovery assistant', *Advances in Knowledge Discovery and Data Mining*, **MIT Press**, 249–271, (1996).
[8] N. Lavrač, B. Cestnik, D. Gamberger, and P.A. Flach, 'Decision support through subgroup discovery: Three case studies and the lessons learned', *Machine Learning*, **57**(1/2), (2004). in press.
[9] N. Lavrač, P.A. Flach, and B. Zupan, 'Rule evaluation measures: A unifying view', in *Proceedings of the Nineth International Workshop on Inductive Logic Programming*, pp. 74–185, Springer, (1999).
[10] N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski, 'Subgroup discovery with CN2-SD', *Journal of Machine Learning Research*, (5), 153–188, (2004).
[11] 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*, pp. 80–86, (1998).
[12] D. Mladenić and N. Lavrač, *Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise*, DZS, Ljubljana, 2003.
[13] F.J. Provost and T. Fawcett, 'Robust classification for imprecise environments', *Machine Learning*, **42**(3), 203–231, (2001).
[14] L. Todorovski, P.A. Flach, and N. Lavrač, 'Predictive performance of weighted relative accuracy', in *Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 255–264, Springer, (2000).
[15] S. Wrobel, 'An algorithm for multi-relational discovery of subgroups', in *Proceedings of the First European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 78–87, Springer, (1997).