

Using Subgroup Discovery to Analyze the UK Traffic Data

Branko Kavšek and Nada Lavrač¹

Abstract

Rule learning is typically used in solving classification and prediction tasks. However, learning of classification rules can be adapted also to subgroup discovery. Such an adaptation has already been done for the CN2 rule learning algorithm. In previous work this new algorithm, called CN2-SD, has been described in detail and applied to the well known UCI data sets. This paper summarizes the modifications needed for the adaptation of the CN2 rule learner to subgroup discovery and presents its application to a real-life data set - the UK traffic data - confirming its appropriateness for subgroup discovery in real-life applications through experimental comparison with the CN2 rule learning algorithm as well as through the evaluation of an expert. Furthermore we make the first step towards the comparison of the new CN2-SD algorithm to another state-of-the-art subgroup discovery algorithm Subgroup-Miner by applying both algorithms to a slightly different data set - the UK traffic challenge data set. The results of this application are presented in the form of ROC curves, showing CN2-SD's potential in finding descriptions (subgroups) for minority classes, while SubgroupMiner found 'better' subgroups when trying to describe the majority class given the problem at hand.

1 Introduction

Classical rule learning algorithms were designed to construct classification and prediction rules (Clark and Niblett, 1989; Michalski et al., 1986). In addition to this area of machine learning, referred to as *predictive induction*, developments in *descriptive induction* have recently gained much attention. These involve mining of association rules (e.g., the APRIORI association rule learning algorithm (Agrawal et al., 1996)), subgroup discovery (e.g., the MIDOS subgroup discovery algorithm; Wrobel, 1997), and other approaches to non-classificatory induction.

This paper summarizes the methodology, presented in Lavrač et al. (2002), which was used for upgrading the classical rule learning algorithm CN2 (Clark and Niblett, 1989; Clark and Boswell, 1991) to a subgroup discovery algorithm CN2-SD. We used the implementation of CN2-SD described in Lavrač et al. (2002), a modification of the original Boswell's implementation of the CN2 algorithm (Clark and Boswell, 1991) accommodating the changes needed to make it suitable to a subgroup discovery task. The goals of this paper are:

¹ Institute Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia; branko.kavsek@ijs.si, nada.lavrac@ijs.si

- to confirm the appropriateness of CN2-SD for subgroup discovery. While the appropriateness of CN2-SD for subgroup discovery has already been shown in Lavrač et al. (2002), where the algorithm was studied on data sets from the UCI Repository of Machine Learning (Murphy and Aha, 1994), we here apply CN2-SD to a real-life data set - the UK traffic data set - and compare it to the CN2 rule learning algorithm with the evaluation of an expert;
- to apply two different state-of-the-art subgroup discovery algorithms: CN2-SD and SubgroupMiner (Klogsen, 2002a) (an extension of the well known MIDOS algorithm; Wrobel, 1997) to a slightly different real-life problem - the UK traffic challenge problem - and discuss the results of such an application thus laying the grounds for more thorough comparisons of different existing subgroup discovery algorithms.

The paper is organized as follows. Section 2 describes the complete UK traffic data set from which samples are taken to be used in our experiments. Sections 3 and 4 describe the two subgroup discovery algorithms, CN2-SD and SubgroupMiner, respectively. In Section 3 the whole procedure of adapting the rule learner CN2 to a subgroup discovery algorithm CN2-SD is described. Section 4 briefly describes SubgroupMiner emphasizing the parts of the algorithm relevant to the application. In Section 5 the ROC (Receiver Operating Characteristic) method (Provost and Fawcett, 2001) for assessing the quality of subgroups is described. A more detailed description of issues covered in Sections 3 and 5 can be found in Lavrač et al. (2002) and in the literature referenced there. Section 6 describes the experimental evaluation together with the needed data preprocessing steps and presents the results of the experiments. Section 7 concludes by commenting these results and giving ideas for further work.

2 The UK traffic accident data

The Traffic accidents database is a large real-world database about personal injury traffic accidents in the UK for the period of 21 years from 1979 to 1999. It contains data about personal injury road accidents in the UK and their consequent casualties. The data sets are organized in three related tables: the ACCIDENT table, the VEHICLE table and the CASUALTY table. The ACCIDENT data consists of the records of all accidents happened over the given period of time (1979–1999), the VEHICLE data includes data about all the vehicles involved in those accidents, and the 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. We can also see that the three sets are related one with the other. Every separate set is described by around 20 attributes and consists of more than 5 million records.

- More specifically, the ACCIDENT table contains one record for each accident. The 30 attributes describing an accident can be divided in three groups: date

and time when the accident has occurred, description of the road where the accident has occurred, and conditions at which the accident has occurred (such as weather conditions, light and junction details). In the ACCIDENT table there are more than 5 millions of records.

- The VEHICLE table contains one record for each vehicle involved in the accident from the ACCIDENT table. There can be one or many vehicles involved in a single accident. The VEHICLE table attributes are describing the type of the vehicle, maneuver and direction of the vehicle (from and to), vehicle location on the road, junction location at impact, sex and age of the driver, alcohol test results, damage on a vehicle, and the object that vehicle hit on and off carriageway. There are 24 attributes in the VEHICLE table which contains almost 9 millions of records.
- The CASUALTY table contains records about casualties for each of the vehicles in the VEHICLE table. There can be one or more casualties per vehicle. The CASUALTY table contains 16 attributes describing sex and age of casualty, type of casualty (e.g. pedestrian, cyclist, car occupant etc.), severity of casualty, if casualty type is pedestrian, what were his/her characteristics (location, movement, direction). This table contains almost 7 millions of records.

3 From CN2 to CN2-SD

This section outlines the backgrounds of the CN2-SD algorithm: classical CN2 rule induction algorithm and standard CN2 heuristics, as well as the weighted relative accuracy heuristic and other modifications needed for the adaptation of CN2 to subgroup discovery.

3.1 The CN2 Rule Induction Algorithm

CN2 is an algorithm for inducing propositional classification rules (Clark and Niblett, 1989; Clark and Boswell, 1991). CN2 consists of two main procedures: the *search procedure* that performs beam search in order to find a single rule and the *control procedure* that repeatedly executes the search.

The search procedure performs beam search using the Laplace estimate (Clark and Boswell, 1991) of the rule as a heuristic function. We replaced the Laplace measure with the weighted relative accuracy measure (Todorovski et al., 2000), defined in Equation 3.1. Additionally, CN2 can apply a significance test to the induced rule. The rule is considered to be significant, if it locates regularity unlikely to have occurred by chance. To test significance, CN2 uses the likelihood ratio statistic (Clark and Niblett, 1989).

Two different control procedures are used in CN2: one for inducing an *ordered list* of rules and the other for the *unordered* case. Both ordered and unordered control procedures induce rules in a similar fashion, running the search procedure that finds the best rule, removing the examples covered by that rule and iteratively

repeating this step until all examples have been covered. Detailed description of the difference between the two control procedures is given in Clark and Boswell (1991).

More important than how the rules are produced is how they are interpreted. In the ordered case each rule depends on the rules that precede it, while in the unordered case each rule is interpreted separately and thus each rule represents an independent ‘chunk’ of knowledge.

3.2 The Weighted Relative Accuracy Heuristic

Weighted relative accuracy (WRAcc) can be meaningfully applied both in the descriptive and predictive induction framework; in this paper we apply this heuristic for subgroup discovery.

We use the following notation. Let $n(Cond)$ stand for the number of instances covered by a rule $Cond \rightarrow Class$, $n(Class)$ stand for the number of examples of class $Class$, and $n(Class.Cond)$ stand for the number of correctly classified examples (true positives). We use $p(Class.Cond)$ etc. for the corresponding probabilities. *WRAcc* (Lavrač et al., 1999; Todorovski et al., 2000) is then defined as follows:

$$WRAcc(Cond \rightarrow Class) = p(Cond).(p(Class|Cond) - p(Class)). \quad (3.1)$$

WRAcc consists of two components: *generality* $p(Cond)$, and *relative accuracy* $p(Class|Cond) - p(Class)$. The second term, relative accuracy, is the accuracy gain relative to the fixed (default) rule $true \rightarrow Class$. However, it is easy to obtain high relative accuracy with highly specific rules, i.e., rules with low generality $p(Cond)$. To this end, generality is used as a ‘weight’, so that weighted relative accuracy trades off generality of the rule ($p(Cond)$, i.e., rule coverage) and relative accuracy $p(Class|Cond) - p(Class)$.

3.3 The CN2-SD Algorithm

The main modifications of the CN2 algorithm, making it appropriate for subgroup discovery, involve the implementation of the *weighted covering algorithm* and incorporation of example weights into the weighted relative accuracy heuristic. Both modifications are briefly described below. The complete description is given in Lavrač et al. (2002).

3.3.1 The Weighted Covering Algorithm

In the classical covering algorithm only the first few induced rules may be of interest as subgroup descriptors with sufficient coverage, since subsequently induced rules are induced from biased example subsets, i.e., subsets including only positive examples not covered by previously induced rules. This bias constrains the population for subgroup discovery in a way that is unnatural for the subgroup discovery process which is, in general, aimed at discovering interesting properties of subgroups of the entire population. In contrast, the subsequent rules induced by the weighted

covering algorithm allow for discovering interesting subgroup properties of the entire population.

The weighted covering algorithm is modified in such a way that covered positive examples are not deleted from the current training set. Instead, in each run of the covering loop, the algorithm stores with each example a count of how many times (with how many rules induced so far) the example has been covered. Weights derived from these example counts then appear in the computation of WRAcc. We have implemented two approaches: **Multiplicative weights** - in this approach, weights decrease multiplicatively. For a given parameter $0 < \gamma < 1$, weights of covered examples decrease as follows: $e(i) = \gamma^i$, where $e(i)$ is the weight of an example being covered i times. In the second approach - **Additive weights**, weights of covered examples are modified as follows: $e(i) = 1/(i + 1)$.

3.3.2 Modified WRAcc Heuristic with Example Weights

The modification of CN2 reported in Todorovski et al. (2000) affected only the heuristic function: weighted relative accuracy was used as search heuristic, instead of the Laplace heuristic of the original CN2, while everything else stayed the same. In Lavrač et al. (2002), the heuristic function was further modified to enable handling of example weights, which provide the means to consider different parts of the instance space in each iteration of the weighted covering algorithm.

In the WRAcc computation (Equation 3.1) all probabilities are computed by relative frequencies. An *example weight* measures how important it is to cover this example in the next iteration. The initial example weight $e(0) = 1$ means that the example hasn't been covered by any rule, while lower weights mean that it has already been covered by previously generated rules. The modified WRAcc measure is then defined as follows:

$$WRAcc(Cond \rightarrow Class) = \frac{n'(Cond)}{N'} \left(\frac{n'(Class.Cond)}{n'(Cond)} - \frac{n'(Class)}{N'} \right). \quad (3.2)$$

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

4 The SubgroupMiner subgroup discovery algorithm

SubgroupMiner is an advanced subgroup mining system offering the exploitation of very large databases by efficient data base integration, multirelational hypotheses, visualization based interaction options, and the discovery of causal subgroup structures. It is an extension of older subgroup discovery systems EXPLORA (Klosgen, 1996) and MIDOS (Wrobel, 1997).

Like CN2-SD, SubgroupMiner discovers subgroups in the form of decision rules using (interactive) beam search in the space of possible solutions. Differently from

CN2-SD, SubgroupMiner uses another quality function to rank the rules during the beam search. In addition, SubgroupMiner uses a special postprocessing approach to eliminate redundant subgroups.

The details of SubgroupMiner can be found in (Klosgen, 2002a). We will describe here only those parts of the SubgroupMiner system that are relevant to its application to the task at hand, namely the quality function and the redundancy elimination approach.

4.1 The subgroup quality function

The statistical significance of a subgroup is evaluated by a quality function. Quality functions must satisfy some basic monotonicity axioms, hold symmetry and equivalence properties, and can be arranged in families to adjust to user preferences (Klosgen, 2002; Klosgen, 1999). Quality functions depend on the type of the subgroup pattern. As a standard quality function, SubgroupMiner uses the classical binomial test to verify if the target share is significantly different in a subgroup:

$$Q(S) = \frac{p - p_0}{\sqrt{p_0(1 - p_0)}} \sqrt{n} \sqrt{\frac{N}{N - n}} \quad (4.1)$$

If S is a rule of the form $Cond \rightarrow Class$ then, in the notation of Equation 3.1, n can be expressed as $n(Cond)$, p is $p(Class|Cond)$, p_0 is $p(Class)$ and N is the number of all examples. Vice-versa we can re-write Equation 3.1 with the notation of Equation 4.1:

$$WRAcc(S) = \frac{n}{N}(p - p_0) \quad (4.2)$$

4.2 The redundancy elimination approach

SubgroupMiner uses the same approach as EXPLORA (Klosgen, 1996) to eliminate redundant subgroups. This approach is called *subgroup suppression*. The algorithm suppresses subgroups that are worse than, but not too different from another subgroup. A subgroup that is dissimilar to the other ones is retained, while better ones may be discarded because they are very similar to others that are a little bit better. A subgroup is evaluated as redundant relative to a subgroup with a higher significance when a constraint balancing overlap degree and significance difference is satisfied (Gebhardt, 1991):

$$S_1 \text{ suppresses } S_2 \text{ if } Q(S_2) < \text{Affinity}(S_2, S_1)Q(S_1) \quad (4.3)$$

Q is the statistical quality of a subgroup (see Equation 4.1), and the *affinity* of two subgroups is given as:

$$\text{Affinity}(S_2, S_1) = (|S_1 \cap S_2|/|S_2|)^\alpha \quad (4.4)$$

The parameter α (with default value 1) can be used to control the number of suppressions. The user can increase (or decrease) α to get fewer (or more) resulting subgroups.

5 Assessing the quality of subgroups in the ROC space

A point on the *ROC curve* (ROC: Receiver Operating Characteristic; Provost and Fawcett, 2001) shows classifier performance in terms of false alarm or *false positive rate* $FPr = \frac{FP}{TN+FP}$ (plotted on the X -axis) that needs to be minimized, and sensitivity or *true positive rate* $TPr = \frac{TP}{TP+FN}$ (plotted on the Y -axis) that needs to be maximized. 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 diagonal can be discarded as insignificant. Conversely, significant subgroups are those sufficiently distant from the diagonal.² The significant rules define the points in the ROC space from which a convex hull is constructed. Best of the induced rules define the ROC convex hull.

Weighted relative accuracy is appropriate for measuring the quality of a single subgroup, because it is proportional to the distance from the diagonal in the ROC space. To see this, note first that rule accuracy $p(Class|Cond)$ is proportional to the angle between the X -axis and the line connecting the origin with the point depicting the rule's TPr/FPr tradeoff. So, for instance, the X -axis has always rule accuracy 0 (these are purely negative subgroups), the Y -axis has always rule accuracy 1 (purely positive subgroups), and the diagonal represents subgroups with rule accuracy $p(Class)$, the prior probability of the positive class.

Relative accuracy re-normalizes this such that all points on the diagonal have relative accuracy 0, all points on the Y -axis have relative accuracy $1 - p(Class) = p(\overline{Class})$ (the prior probability of the negative class), and all points on the X -axis have relative accuracy $-p(Class)$. Notice that all points on the diagonal also have $WRAcc = 0$. In terms of subgroup discovery, the diagonal represents all subgroups with the same target distribution as the whole population; only the generality of these 'average' subgroups increases when moving from left to right along the diagonal. This interpretation is slightly different in classifier learning, where the diagonal represents random classifiers that can be constructed without any training.

More generally, one can show that points with the same $WRAcc$ value lie on straight lines parallel to the diagonal. In particular, a point on the line $TPr = FPr + a$, $-1 \leq a \leq 1$ has $WRAcc = ap(Class)p(\overline{Class})$. Thus, given a fixed class distribution, $WRAcc$ is proportional to the vertical (or horizontal) distance a between the line parallel to the diagonal on which the point lies, and the diagonal. In fact, the quantity $TPr - FPr$ would be an alternative quality measure for subgroups, with the additional advantage that we can use it to compare subgroups from populations with different class distributions. However, in this paper we are only concerned with comparing subgroups from the same population, and we prefer

²Any of those subgroups may be the 'best' according to some expert-defined operating conditions.

WRAcc because of its appropriate ‘*p-g*’ interpretation for subgroup discovery.³

6 Experiments

In this section we describe the experiments that were made to meet the goals stated in the Introduction.

Section 6.1 describes the application of CN2-SD to the UK traffic data together with the data preprocessing step and the evaluation of results by the domain expert.

Section 6.2 describes the application of two different subgroup discovery algorithms: CN2-SD and SubgroupMiner to a different sample of the UK traffic data - the UK traffic challenge data and compares the two algorithms.

6.1 CN2-SD vs CN2

This section describes the application of CN2-SD to solve the problem of predicting the severity of accidents from the UK traffic data together with the preprocessing of the data that was needed to make the problem manageable and the evaluation of results by the domain expert. To show/confirm the appropriateness of CN2-SD for subgroup discovery the results are compared to those of the CN2 rule induction algorithm.

6.1.1 Preprocessing of the data

The enormous quantity of data in the TRAFFIC data set makes it practically impossible to run any data mining algorithm on the whole set.

Therefore we have decided to take samples of the data set and perform the experiments on these samples, rather than on the whole data set. We focused on the ACCIDENT set of data and decided to examine only the accidents that happened in 10 areas (called Local Authorities (LAs)) across Great Britain. We have chosen the 5 areas with the most increasing trend of accidents and 5 areas with the most decreasing trend according to the results of regression analysis of the number of accidents that happened in each LA over the years. In this way we selected 10 data sets (one for each LA) with some ten thousands of examples each. We further sampled this data taking only 10% of the examples from each of the 10 sets. The characteristics of these 10 data sets are given in Table 1. Since all 10 sets are subsets of the same data set, they all have the same number of attributes (26 including the class attribute). Therefore Table 1 only gives the number of examples in each set and the distribution of the class attribute. The sets 1 to 5 represent the 5 areas with the most decreasing trend of accidents (set 1 being the ‘best’) and sets 6 to 10 the ones with the most increasing trend (set 6 being the ‘worst’). The Code numbers 1 through 10 do not correspond to the codes 1 through 10 used for Local Authorities in the Database.

³See Equation 4.1 for the explanation of *p*; in this same notation $g = n/N$ and represents the generality of a subgroup - its size relative to the size of the whole population.

Table 1: Characteristics of data sets.

LA	NO. of exs.	Class Dist (% - fatal/serious/slight)
1	6039	0.64/15.35/84.01
2	3627	1.15/16.80/82.04
3	2916	0.95/17.37/81.67
4	3182	1.10/19.60/79.29
5	2684	0.88/16.87/82.25
6	5487	1.35/12.74/85.90
7	1477	1.64/14.81/83.54
8	6381	1.66/17.31/81.02
9	1645	2.05/18.18/79.77
10	4375	1.82/17.11/81.05

6.1.2 Results of experiments

We further wanted to investigate if by running CN2-SD on the data sets described in Table 1, we are able to get some rules that are typical for the 5 areas with the most increasing trend of accidents as well as rules typical for the 5 areas with the most decreasing trend. In Table 2 CN2-SD (we used the additive weights approach) and standard CN2 are compared on the 10 LA sets in terms of: ‘number of induced rules’ (R), ‘relative average coverage’ (CVG) and ‘accuracy of rules’ (Acc).

We have used 10-fold cross-validation to compute the accuracies of induced rule sets, whereas the number of rules and the relative average coverage were computed on rules induced from all available data. The relative average coverage measures the percentage of examples covered on average by one rule from the induced rule set. It is computed as:

$$CVG = \sum_{i=1}^{n_R} \frac{covered(i)}{n_R n_{EXS}} \quad (6.1)$$

where n_R is the number of induced rules, n_{EXS} is the number of examples in the data set and $covered(i)$ is the number of examples covered by the i -th rule in the rule set.

As can be seen from the results in Table 2, CN2-SD induced rulesets with fewer rules that have a higher ‘relative average coverage’ (CVG) than CN2. The accuracy of these rulesets is, however, slightly lower than of those induced by CN2. These results confirm the findings from Lavrač et al. (2002) thus confirming the appropriateness of CN2-SD for subgroup discovery.

Unfortunately, none of the induced rules (either by CN2 or by CN2-SD) was typical either for the 5 areas with the most increasing trend of accidents or for the 5 areas with the most decreasing trend. There were, however, 3 rules that appeared in almost every ruleset induced by CN2-SD on the 10 data sets (sometimes all 3 of them or just 1 or 2) that caught our attention and the attention of the domain expert⁴. We describe these ‘interesting’ rules by providing their interpretation by a

⁴None of these 3 rules appeared in rulesets induced by CN2.

Table 2: Results of experiments.

LA	Standard CN2			CN2-SD (additive)		
	R	CVG	Acc	R	CVG	Acc
1	22	6.25	83.95	10	13.08	83.94
2	19	4.78	81.98	9	13.20	82.20
3	31	1.44	79.05	11	10.66	79.15
4	25	6.54	81.46	9	14.49	81.66
5	20	5.51	81.46	14	7.49	81.23
6	29	5.26	85.88	13	10.45	85.86
7	17	5.29	81.02	9	12.96	81.02
8	34	4.37	79.73	14	8.80	79.76
9	13	6.82	83.40	14	8.40	82.74
10	28	5.05	80.97	11	11.64	80.83
Avg	23.8	5.13	81.89	11.4	11.12	81.84

domain expert.

6.1.3 Comments of the domain expert

We examined further the rules induced by the CN2-SD algorithm (additive weights). We focused on rules with high coverage and rules that cover a high percentage of the predicted class as those are the rules that are likely to reflect some regularity in the data. We have found a very surprising fact. One might expect the more severe the accident the greater number of people hurt up to the total number of occupants in the vehicles. Also common sense would tell the more the vehicles involved in the accident the more severe the accident. Contrary to our expectations we found two types of rules:

- rules that classify an accident as ‘fatal’ or ‘serious’ when just one vehicle is involved in the accident; examples of such rules are:
IF `nv < 1.500` THEN `sev = "1"` [15 280 1024]⁵
IF `nv < 1.500` THEN `sev = "2"` [22 252 890]
- rules that classify an accident as ‘slight’ when two or more vehicles are involved and there are few casualties; an example of such a rule is:
IF `nv > 1.500` AND `nc < 2.500` THEN `sev = "3"` [8 140 1190]

We have shown these results to the domain expert who found them surprising. He tried to explain the facts described by them as a consequence of the way the data for the ACCIDENT data set was collected, as follows:

- The severity code in the ACCIDENT data set relates to the most severe injury among those reported for that accident. Therefore a multiple vehicle

⁵The rules in the example are given in the CN2-SD output format where `nv` stands for ‘number of vehicles’, `nc` is the ‘number of casualties’ and “1”, “2”, and “3” denote the class values ‘fatal’, ‘serious’ and ‘slight’ respectively.

accident with 1 fatal and 20 slight injuries would be classified as fatal as one fatality occurred. Each individual CASUALTY injury severity is coded in the CASUALTY data set.

- Some injuries may be unreported at the accident scene, if the policeman compiles/revises the report after the event, new casualty/injury details can be reported (injuries that came to light after the event or reported for reasons relating to injury/insurance claims). However this is a very surprising fact that needs to be further investigated. We agreed with the expert that examining the ACCIDENT data set was not enough. Further examination of the VEHICLE and CASUALTY set is needed.

We applied another subgroup discovery algorithm – SubgroupMiner, described in Section 4 – to the problem at hand. The results of the SubgroupMiner algorithm, not reported here, were similar to those obtained by CN2-SD with the difference that SubgroupMiner did not discover the third ‘interesting’ rule describing ‘slight’ accidents.

The reason that we do not report the results of SubgroupMiner at this point is due to their similarity with results of CN2-SD for the problem at hand. However, we compare SubgroupMiner and CN2-SD on a slightly different problem – the UK traffic challenge problem, showing the differences between the two.

6.2 CN2-SD and SubgroupMiner

This section describes the application of two different state-of-the-art subgroup discovery algorithms: CN2-SD and SubgroupMiner to the problem of the UK traffic challenge - a sample problem from the UK traffic data. The results of the application are shown in the form of ROC graphs (see Section 5 for the description of ROC space).

6.2.1 The UK traffic challenge

The task of the challenge was to produce classification models (in our case subgroup descriptions) to predict the class *Skidding and Overturning* for accidents from the UK traffic data set (see Section 2). Because 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 is described in Table 3. Table 4 presents the distribution of the class attribute in the training and test sets.

6.2.2 Results

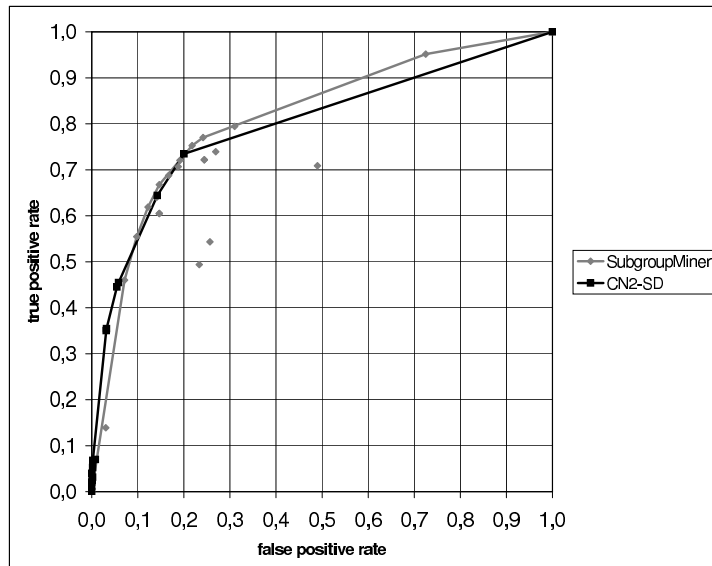
We applied the subgroup discovery algorithms CN2-SD and SubgroupMiner to the UK traffic challenge training data to construct subgroups and then test these

Table 3: Meaning of classes.

code	meaning
0	No skidding, jack-knifing or overturning
1	Skidded
2	Skidded and overturned
3	Jack-knifed
4	Jack-knifed and overturned
5	Overturned

Table 4: Distribution of classes (%).

set/class	0	1	2	3	4	5
training	64.26	22.07	7.27	0.20	0.19	6.01
test	64.67	22.46	6.88	0.06	0.44	5.49

**Figure 1:** ROC curve for the problem of predicting Class 0.

subgroups on the test data (see Table 4). The results were then plotted in the ROC space. Because of the fact that only binary class problems can be plotted in the 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. Both subgroup discovery algorithms were run with default parameters (CN2-SD using additive weighting scheme, 99% significance threshold and beam size 5, SubgroupMiner using beam size 10, max. length of rules 6 and suppression factor $\alpha = 1$).

We discarded the problems of predicting Class 3 and Class 4 (see Table 3 for the meaning of class codes) because they contained too few test examples (see the distribution in Table 4). 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

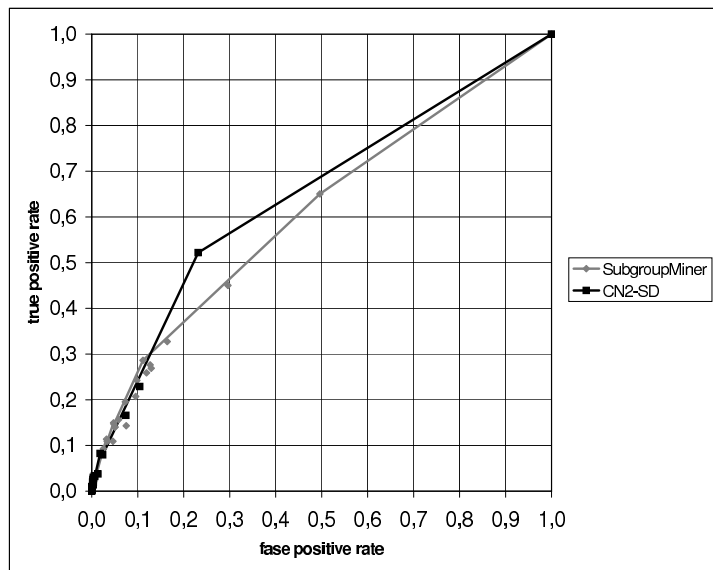


Figure 2: ROC curve for the problem of predicting Class 1.

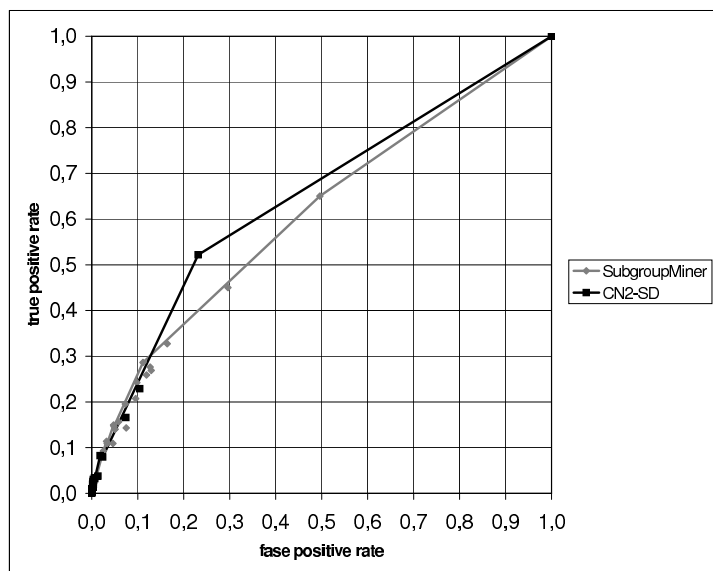


Figure 3: ROC curve for the problem of predicting Class 5.

1 giving no additional information for discussing the similarities/differences of the two subgroup discovery algorithms.

The results of the application on the remaining three problems of predicting Class 0, Class 1 and Class 5 are shown (plotted in the ROC space) in Figures 1, 2 and 3 respectively. We can describe these three 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 in Figures 1, 2 and 3 that CN2-SD produces smaller more accurate subgroups (points on the Y axis near the point $(0,0)$ in all three figures). On the other hand, SubgroupMiner produces larger subgroups that are at

the same time less accurate (points nearer to the point $(1, 1)$). This is especially true for the problem of predicting the majority class (Class 0 - Figure 1). There are two other facts that can be seen from the ROC plots. The first is that SubgroupMiner produces a lot of subgroups that do not lie on the ROC convex hull and are thus sub-optimal. The second thing is that CN2-SD produces ‘better’ subgroups (the distance from the diagonal $(0, 0) - (1, 1)$ is larger) when dealing with the problem of predicting a minority class (we can see this if we look at Figures 2 and 3).

Although ROC analysis provided us with some useful insights in the differences between subgroup discovery algorithms SubgroupMiner and CN2-SD through their application on the UK traffic accidents data, it is the final word of the domain expert that will ultimately tell us the true qualitative difference between these two algorithms.

7 Conclusions

The comparative results in Section 6.1 (Table 2) show that CN2-SD induced on average smaller rule sets that included rules that had on average a higher coverage than those induced by the standard CN2 algorithm. The latter fact makes CN2-SD more suitable for the subgroup discovery task as each rule with high coverage represents a potentially interesting subgroup in the data. On the other hand, the average accuracy of the CN2-SD rule sets was more or less the same as the accuracy of standard CN2 rules, which is very good given that the CN2-SD algorithm does not optimize rule accuracy. The above findings are not new and reflect/confirm the findings in Lavrač et al. (2002).

It is worth noticing that both CN2-SD and standard CN2 performed ‘worse than default’ in terms of accuracy of induced rules, meaning that if we predicted the majority class (Table 1) we would have got better classification accuracy than by applying the induced rules. This fact is not surprising due to a very unbalanced class distribution and the way how the experiments were performed. Since classification was not the task addressed, we were not really interested in the accuracy of the rule sets but more in detecting interesting subgroups that the rules represented.

The most interesting finding was the rule interpretation by the domain expert. What we found in our case study was that the result of a data mining process depends not only on the accuracy of the chosen method and the data that is at hand but also on how the data was collected.

When comparing the two subgroup discovery algorithms CN2-SD and SubgroupMiner applied to real-life challenge data we can observe that CN2-SD did better in predicting minority classes, while SubgroupMiner found larger and more accurate subgroups when dealing with classes containing the majority of the examples.

On the other hand, SubgroupMiner produced large subgroups and was thus more oriented towards the scope of subgroup discovery by finding large and significant ‘chunks’ of knowledge given the problem at hand.

Another issue that is worth mentioning here is that CN2-SD is able to deal only with simple single relation data while SubgroupMiner is a multirelational tool.

In conclusion, we can claim that neither of the two algorithms was better than

the other one on the given problem. They provided us with different insights of the data. It is then the task of an expert to have a final word on which of the two algorithms produced better subgroups in relation to his/her needs. In further work we plan to show the discovered subgroups to the domain expert who will be able to tell us the true qualitative difference between the two algorithms and their usefulness.

Furthermore the application presented in this paper can be seen as the first step towards a more thorough comparison of the two subgroup discovery algorithms where statistical tests are needed to support hypotheses on significant differences between the two algorithms. In further work we intend to perform a more thorough study of these differences involving also the study on how different parameter settings can affect the results of subgroup discovery.

Acknowledgements

Thanks to John C. Bullas (the domain expert) for the interpretation of the results. Thanks to Willi Klösger for providing the results of the SubgroupMiner algorithm and for many valuable suggestions in conducting the experiments. The work reported in this paper was supported by the Slovenian Ministry of Education, Science and Sport.

References

- [1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A.I. (1996): Fast discovery of association rules. In U.M. Fayyad, G. Piatetski-Shapiro, P. Smyth, and R. Uthurusamy (Eds.): *Advances in Knowledge Discovery and Data Mining*, 307-328. AAAI Press.
- [2] Clark, P. and Boswell, R. (1991): Rule induction with CN2: Some recent improvements. In Y. Kodratoff (Eds.): *Proc. of the 5th European Working Session on Learning*, 151-163. Springer.
- [3] Clark, P. and Niblett, T. (1989): The CN2 induction algorithm. *Machine Learning*, **3**, 261-283.
- [4] Gebhardt, F. (1991): Choosing among Competing Generalizations. *Knowledge Acquisition*, **3**, 361-380.
- [5] Klösger, W. (1996): Explora: A multipattern and multistrategy discovery assistant. In U.M. Fayyad, G. Piatetski-Shapiro, P. Smyth, and R. Uthurusamy (Eds.): *Advances in Knowledge Discovery and Data Mining*, 249-271. MIT Press.
- [6] Klösger, W. (1999): Applications and research problems of subgroup mining. Foundations of intelligent systems. In Z. Ras and A. Skowron (Eds.): *Lecture Notes in Artificial Intelligence*. Berlin, Heidelberg, New York: Springer, 115.

-
- [7] Klösgen, W. (2002): Subgroup discovery. Chapter 16.3. In W. Klösgen and J. Zytkow (Eds.): *Handbook of Data Mining and Knowledge Discovery*. New York: Oxford University Press.
- [8] Klösgen, W. and May, M. (2002a): Census data mining – an application. In *Proceedings of the 6th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Helsinki, Finland.
- [9] Lavrač, N., Flach, P., and Zupan, B. (1999): Rule evaluation measures: A unifying view. In *Proc. of the 9th International Workshop on Inductive Logic Programming*, 74-185. Springer.
- [10] Lavrač, N., Flach, P., Kavšek, B., and Todorovski, L. (2002): Adapting classification rule induction to subgroup discovery. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'02)*, Maebashi City, Japan, 266-273.
- [11] Michalski, R.S., Mozetič, I., Hong, J., and Lavrač, N. (1986): The multi-purpose incremental learning system AQ15 and its testing application on three medical domains. In *Proc. 5th National Conference on Artificial Intelligence*, 1041-1045. Morgan Kaufmann.
- [12] Murphy, P.M. and Aha, D.W.(1994): *UCI Repository of Machine Learning Databases*. <http://www.ics.uci.edu/~mlearn/MLRepository.html>. Irvine, University of California, Department of Information and Computer Science.
- [13] Provost, F. and Fawcett, T. (2001): Robust classification for imprecise environments. *Machine Learning*, **42**, 203-231.
- [14] Todorovski, L., Flach, P., and Lavrač, N. (2000). Predictive performance of weighted relative accuracy. In D.A. Zighed, J. Komorowski, and J. Zytkow (Eds.): *Proc. of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, 255-264. Springer.
- [15] Wrobel, S. (1997). An algorithm for multi-relational discovery of subgroups. In *Proc. of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery*, 78-87. Springer.