Adapting classification rule induction to subgroup discovery

Nada Lavrač
Jožef Stefan Institute
Ljubljana, Slovenia
Nada.Lavrac@ijs.si

Peter Flach
University of Bristol
Bristol, UK
Peter.Flach@bristol.ac.uk

Branko Kavšek, Ljupčo Todorovski
Jožef Stefan Institute
Ljubljana, Slovenia
Branko.Kavsek@ijs.si
Ljupco.Todorovski@ijs.si

Abstract

Rule learning is typically used for solving classification and prediction tasks. However, learning of classification rules can be adapted also to subgroup discovery. This paper shows how this can be achieved by modifying the covering algorithm and the search heuristic, performing probabilistic classification of instances, and using an appropriate measure for evaluating the results of subgroup discovery. Experimental evaluation of the CN2-SD subgroup discovery algorithm on 17 UCI data sets demonstrates substantial reduction of the number of induced rules, increased rule coverage and rule significance, as well as slight improvements in terms of the area under the ROC curve.

1. Introduction

Classical rule learning algorithms were designed to construct classification and prediction rules [16, 4, 5]. 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 [1]), subgroup discovery (e.g., the MIDOS subgroup discovery systems [24]), and other approaches to non-classificatory induction.

The methodology presented in this paper can be applied to subgroup discovery. 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.

This paper investigates how to adapt classical classification rule learning approaches to subgroup discovery, by appropriately modifying the heuristics and the covering algorithm for learning sets of rules. The proposed modifications of classification rule learners can, in principle, be used on top of any rule learner using the covering approach for rule set construction. In this work we show an upgrade of the well-known CN2 rule learning algorithm [4, 3]. Alternatively, we could have upgraded RL [15], RIPPER [5], SLIPPER [6] or other more sophisticated classification rule learners. The reason for upgrading CN2 is that other more sophisticated learners include modifications that make them more effective in classification tasks, improving their classification accuracy. Improved classification accuracy is, however, not of ultimate interest for subgroup discovery, whose main goal is to find interesting population subgroups.

We have implemented the new subgroup discovery algorithm CN2-SD in Java and incorporated it in the WEKA data mining environment [23]. The proposed approach performs subgroup discovery through the following modifications of the classical rule learning algorithm CN2: (a) incorporating example weights into the covering algorithm, (b) incorporating example weights into the weighted relative accuracy search heuristic, (c) probabilistic classification based on the class distribution of covered examples by individual rules, both in the case of unordered rule sets and ordered decision lists, and (d) area under the ROC curve rule set evaluation.

This paper presents the CN2-SD subgroup discovery algorithm, together with its experimental evaluation in selected domains of the UCI Repository of Machine Learning Databases [17]. The experimental comparison with CN2 demonstrates that the subgroup discovery algorithm CN2-SD produces substantially smaller rule sets, where individual rules have higher coverage and significance. These three factors are important for subgroup discovery: smaller size enables better understanding, higher coverage means larger support, and higher significance means that rules describe discovered subgroups that are significantly different from the entire population. The appropriateness for subgroup discovery is confirmed also by slight improvements in terms of the area under the ROC curve.

This paper is organized as follows. In Section 2 the

0-7695-1754-4/02 $17.00 © 2002 IEEE

background for this work is explained: the standard CN2 rule induction algorithm, including the covering algorithm, the weighted relative accuracy heuristic, probabilistic classification and rule evaluation in the ROC space. Section 3 presents the modified CN2 algorithm, called CN2-SD, adapting CN2 for subgroup discovery. Section 4 presents the experimental evaluation on selected UCI domains. Some links to the related work are given in Section 5. Section 6 concludes by summarizing the results and presenting plans for further work.

2. Background

This section presents the backgrounds of our work: the classical CN2 rule induction algorithm, including the covering algorithm for rule set construction, the standard CN2 heuristics, the weighted relative accuracy heuristic, probabilistic classification and the basics of rule evaluation in ROC space.

2.1. The CN2 rule induction algorithm

CN2 is an algorithm for inducing propositional classification rules [4, 3]. Induced rules have the form `if` *Cond* `then` *Class*, where *Cond* is a conjunction of features (attribute values). In this paper we use the notation $Class \leftarrow Cond$.

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 classification accuracy of the rule as a heuristic function. The accuracy of the propositional classification rule $Class \leftarrow Cond$ is equal to the conditional probability of class *Class*, given that the condition *Cond* is satisfied: $Acc(Class \leftarrow Cond) = p(Class|Cond)$. Different probability estimates, like the Laplace [3] or the m-estimate [2, 7], can be used in CN2 for estimating the above probability. The standard CN2 algorithm used in this work uses the Laplace estimate.

CN2 can apply a significance test to an induced rule. A rule is considered to be significant if it expresses a regularity unlikely to have occurred by chance. To test significance, CN2 uses the likelihood ratio statistic [4] that measures the difference between the class probability distribution in the set of examples covered by the rule and the class probability distribution in the set of all training examples. Empirical evaluation in [3] shows that applying a significance test reduces the number of induced rules (at a cost of slightly reduced predictive accuracy).

Two different control procedures are used in CN2: one for inducing an ordered list of rules and the other for the unordered case. In both procedures, a default rule (providing for majority class assignment) is added as the final rule in an induced rule set.

When inducing an ordered list of rules, the search procedure looks for the best rule, according to the heuristic measure, in the current set of training examples. The rule predicts the most frequent class in the set of examples, covered by the induced rule. Before starting another search iteration, all examples covered by the induced rule are removed. The control procedure invokes a new search, until all the examples are covered.

In the unordered case, the control procedure is iterated, inducing rules for each class in turn. For each induced rule, only covered examples belonging to that class are removed, instead of removing all covered examples, like in the ordered case. The negative training examples (i.e., examples that belong to other classes) remain and positives are removed in order to prevent CN2 finding the same rule again.

2.2. The weighted relative accuracy heuristic

Weighted relative accuracy is a variant of rule accuracy that 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 $Class \leftarrow Cond$, $n(Class)$ 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. In our work, the Laplace estimate is used for probability estimation. Rule accuracy can be expressed as $Acc(Class \leftarrow Cond) = p(Class|Cond) = \frac{p(Class.Cond)}{p(Cond)}$. Weighted relative accuracy [14, 22], a reformulation of one of the heuristics used in EXPLORA [10] and MIDOS [24], is defined as follows.

$$WRAcc(Class \leftarrow Cond) = p(Cond).(p(Class|Cond) - p(Class)). \quad (1)$$

Like most of the other heuristics used in subgroup discovery systems, weighted relative accuracy consists of two components, providing a tradeoff between rule generality (or the size of a group $p(Cond)$) and distributional unusualness (or relative accuracy, i.e., the difference between rule accuracy $p(Class|Cond)$ and default accuracy $p(Class)$). This difference can also be seen as the accuracy gain relative to the fixed rule $Class \leftarrow true$. The latter rule predicts all instances to satisfy *Class*; a rule 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 *Cond* with a given rule head *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 ($p(Class|Cond) - p(Class)$).

In [10], these quantities are referred to as g (generality), p (rule accuracy) and p_0 (default accuracy), and the function $g(p - p_0)$ is investigated in the so-called 'p-g-space'. Klösgen also investigates other tradeoffs reducing the influence of generality, e.g. $\sqrt{g}(p - p_0)$ or $\sqrt{\frac{g}{1-g}}(p - p_0)$. Here we favor weighted relative accuracy because it has an intuitive interpretation in the ROC space (see Section 2.4).

2.3. Probabilistic classification

The induced rules can be ordered or unordered. Ordered rules are interpreted as a decision list [19] in a straightforward manner: when classifying a new example, the rules are sequentially tried and the first rule that covers the example is used for prediction.

In the case of unordered rule sets, the distribution of covered training examples among classes is attached to each rule. Rules of the form:

$$\texttt{if } Cond \texttt{ then } Class\ [ClassDistribution]$$

are induced, where numbers in the $ClassDistribution$ list denote, for each individual class, how many training examples of this class are covered by the rule. When classifying a new example, all rules are tried and those covering the example are collected. If a clash occurs (several rules with different class predictions cover the example), a voting mechanism is used to obtain the final prediction: the class distributions attached to the rules are summed to determine the most probable class. If no rule fires, a default rule is invoked which predicts the majority class of uncovered training instances.

2.4. ROC analysis for subgroup discovery

A point on the *ROC curve* (ROC: Receiver Operating Characteristic) [18] 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 rules/subgroups whose TPr/FPr tradeoff is close to the diagonal can be discarded as insignificant. Conversely, significant rules/subgroups are those sufficiently distant from the diagonal.[1] The significant rules define the points in the ROC space from which a convex hull is constructed. The area under the ROC curve (*AUC*) can be used as a quality measure for comparing the success of different learners or subgroup miners.

Weighted relative accuracy is appropriate to measure 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 $WRAcc$ because it is a familiar measure in subgroup discovery.

3. The CN2-SD subgroup discovery algorithm

The main modifications of the CN2 algorithm, making it appropriate for subgroup discovery, involve the implementation of the weighted covering algorithm, incorporation of example weights into the weighted relative accuracy heuristic, probabilistic classification also in the case of the

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

'ordered' induction algorithm, and the area under the ROC curve rule set evaluation.

3.1. Weighted covering algorithm

Subgroup discovery is, in general, aimed at discovering interesting properties of subgroups of the entire population. If used for subgroup discovery, the problem of standard rule learners like CN2 and RIPPER is in the use of the covering algorithm for rule set construction. The main deficiency of the covering algorithm is that only the first few induced rules may be of interest as subgroup descriptors with sufficient coverage and significance. Subsequently induced rules are induced from biased example subsets, i.e., subsets including only positive examples not covered by previously induced rules, which inappropriately biases the subgroup discovery process.

As a remedy to this problem we propose to use the weighted covering algorithm, in which the subsequently induced rules allow for discovering interesting subgroup properties of the entire population. The weighted covering algorithm modifies the classical covering algorithm 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 that shows how many times (with how many rules induced so far) the example has been covered so far. Weights derived from these example counts then appear in the computation of *WRAcc*. Initial weights of all positive examples e_j equal $w(e_j, 0) = 1$. We have implemented two approaches:

(a) Multiplicative weights. In the first approach, weights decrease multiplicatively. For a given parameter $\gamma < 1$, weights of covered positive examples decrease as follows: $w(e_j, i) = \gamma^i$, where $w(e_j, i)$ is the weight of example e_j being covered i times. Note that the weighted covering algorithm with $\gamma = 1$ would result in finding the same rule over and over again, whereas with $\gamma = 0$ the algorithm would perform the same as the standard CN2 algorithm.

(b) Additive weights. In the second approach, weights of covered positive examples decrease according to the formula $w(e_j, i) = \frac{1}{i+1}$.

3.2. Modified WRAcc heuristic with example weights

The modification of CN2 reported in [22] affected only the heuristic function: weighted relative accuracy was used as search heuristic, instead of the accuracy heuristic of the original CN2, while everything else remained the same. In this work, the heuristic function was further modified to enable handling 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 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 $w(e_j, 0) = 1$ means that the example hasn't been covered by any rule, meaning 'please cover this example, it hasn't been covered before', while lower weights mean 'don't try too hard on this example'. The modified *WRAcc* measure is then defined as follows.

$$WRAcc(Cl \leftarrow Cond) = \frac{n'(Cond)}{N'}\left(\frac{n'(Cl.Cond)}{n'(Cond)} - \frac{n'(Cl)}{N'}\right). \quad (2)$$

In this equation, N' is the sum of the weights of all examples, $n'(Cond)$ is the sum of the weights of all covered examples, and $n'(Cl.Cond)$ is the sum of the weights of all correctly covered examples.

To add a rule to the generated rule set, the rule with the maximum *WRAcc* measure is chosen out of those rules in the search space, which are not yet present in the rule set produced so far (all rules in the final rule set are thus distinct, without duplicates).

3.3. Probabilistic classification

Each CN2 rule returns a class distribution in terms of the numbers of examples covered, as distributed over classes. The CN2 algorithm uses class distribution in classifying unseen instances only in the case of unordered rule sets, where rules are induced separately for each class. In the case of ordered decision lists, the first rule that fires provides the classification. In our modified CN2-SD algorithm, also in the ordered case all applicable rules are taken into the account, hence the same probabilistic classification is used in both classifiers. This means that the terminology 'ordered' and 'unordered', which in CN2 distinguished between decision list and rule set induction, has a different meaning in our setting: the 'unordered' algorithm refers to learning classes one by one, while the 'ordered' algorithm refers to finding best rule conditions and assigning the majority class in the head.

3.4. Area under the ROC curve evaluation

In subgroup discovery there are two ways in which a rule learner can give rise to a ROC curve.

(a) AUC-Method-1. The first method treats each rule as a separate subgroup which is plotted in the ROC space with its true and false positive rates. We then calculate the convex hull of this set of points, selecting the subgroups which perform optimally under a particular range of operating characteristics. The area under this ROC convex hull (*AUC*) indicates the combined quality of the optimal subgroups, in the sense that it does evaluate whether a particular subgroup

has anything to add in the context of all the other subgroups. However, the method does not take account of any overlap between subgroups, and subgroups not on the convex hull are simply ignored (the existence of many such subgroups may indicate overfitting, as we illustrate in Section 4).

Note that CN2-SD learns rules both for the positive and negative target class: rules for the positive target class represent subgroups with a higher proportion of positives than average, and rules for the negative target class represent subgroups with a lower than average proportion of positives. Consequently, this method constructs two convex hulls, one above the diagonal and one below (see Figure 1 in Section 4).

(b) AUC-Method-2. The second method employs the combined probabilistic classifications of all subgroups, as indicated below. If we always choose the most likely predicted class, this corresponds to setting a fixed threshold 0.5 on the positive probability: if the positive probability is larger than this threshold we predict positive, else negative. A ROC curve can be constructed by varying this threshold from 1 (all predictions negative, corresponding to (0,0) in the ROC space) to 0 (all predictions positive, corresponding to (1,1) in the ROC space). This results in $n + 1$ points in the ROC space, where n is the total number of classified examples. Equivalently, we can order all the examples by decreasing the predicted probability of being positive, and tracing the ROC curve by starting in (0,0), stepping up when the example is actually positive and stepping to the right when it is negative, until we reach (1,1).[2] The area under this ROC curve indicates the combined quality of all subgroups (i.e., the quality of the entire rules set). This method can be used with a test set or in cross-validation, but the resulting curve is not necessarily convex. A detailed description of this method applied to decision tree induction can be found in [11].

Which of the two methods is more appropriate for subgroup discovery is open for debate. The second method seems more appropriate if the discovered subgroups are also used for classification, while the first seems more appropriate for the selection and evaluation of a subset of potentially optimal individual subgroups. One advantage of the second method is that it is easier to apply cross-validation. A disadvantage of the first method is that it ignores redundant subgroups which may indicate overfitting. In the experimental evaluation in Section 4 we used AUC-Method-2, while the ROC convex hull obtained by AUC-Method-1 is only illustrated in one domain in Figure 1.

[2] In the case of ties, we make the appropriate number of steps up and to the right at once, drawing a diagonal line segment.

4. Experimental evaluation

For subgroup discovery, expert's evaluation of results is of ultimate interest. Nevertheless, before applying the proposed approach to a particular problem of interest, we wanted to verify our claims that the mechanisms implemented in the CN2-SD algorithm are indeed appropriate for subgroup discovery.

We experimentally evaluated our approach on 17 data sets from the UCI Repository of Machine Learning Databases [17]. In Table 1, the selected data sets are summarized in terms of the number of attributes, the number of examples, and the percentage of examples of the majority class. These data sets have been widely used in other comparative studies. Since currently our Java re-implementation of CN2 in WEKA does not support continuous attributes and can not handle missing values, all continuous attributes were discretized and data sets that contain no missing values were chosen. The discretization described in [8] was performed using the WEKA tool [23]. Moreover, in our experiments all of the data sets have two classes, either originally or by selecting one class as 'positive' and joining all the other in the 'negative' class (in Table 1, the selected positive class is indicated by {ClassName}); this was done for the purpose of enabling the area under the ROC curve evaluation.

Table 1. Data set characteristics.

	Data set	#Attr.	#Ex.	Maj. Class (%)
1	Anneal{3}	38	898	76.16
2	Australian	14	690	55.5
3	Balance{L}	4	625	46.08
4	Car{unacc}	6	1728	70.02
5	Credit-g	20	1000	70
6	Diabetes	8	768	65.1
7	Glass{build wind non-float}	9	214	35.51
8	Heart-stat	13	270	55.56
9	Ionosphere	34	351	64.1
10	Iris{Iris-setosa}	4	150	33.33
11	Lymph{metastases}	18	148	54.72
12	Segment{brickface}	19	2310	14.29
13	Sonar	60	208	53.36
14	Tic-tac-toe	9	958	65.34
15	Vehicle{bus}	18	846	25.77
16	Wine{2}	13	178	39.89
17	Zoo{mammal}	17	101	40.59

The comparison of CN2-SD with CN2 was performed in 17 UCI domains with AUC-Method-2 evaluation based on 10-fold stratified cross validation. Table 2 compares the CN2-SD subgroup discovery algorithm with the standard CN2 algorithm (*CN2-standard*, described in [3]) and the CN2 algorithm using *WRAcc* (*CN2-WRAcc*, described in [22]) in terms of area under the ROC curve (*AUC*). All these variants of the CN2 algorithm were first re-implemented in the WEKA data mining environment [23],

because the use of the same system makes the comparisons more impartial. Due to space restrictions we only include the results of the unordered algorithms. The results of the CN2-SD algorithm were computed using both the multiplicative weights (with $\gamma = 0.5, 0.7, 0.9$) and the additive weights. Results with $\gamma = 0.7$ are not listed, as they are always between those of $\gamma = 0.5$ and $\gamma = 0.9$, as expected. All other parameters of the CN2 algorithm were set to their default values (beam-size = 5, significance-threshold = 99%).

Table 2. Area under the ROC curve with standard deviation ($AUC \pm sd$) for different variants of the unordered algorithm using 10-fold stratified cross-validation.

#	CN2 standard $AUC \pm sd$	CN2-SD WRAcc $AUC \pm sd$	CN2-SD ($\gamma = 0.5$) $AUC \pm sd$	CN2-SD ($\gamma = 0.9$) $AUC \pm sd$	CN2-SD (add. weight.) $AUC \pm sd$
1	99.41 ± 0.01	99.72 ± 0.00	99.24 ± 0.01	98.51 ± 0.01	98.17 ± 0.01
2	35.10 ± 0.11	87.83 ± 0.05	83.15 ± 0.05	84.32 ± 0.05	84.97 ± 0.04
3	86.22 ± 0.03	89.00 ± 0.03	93.89 ± 0.02	93.56 ± 0.02	91.82 ± 0.03
4	99.93 ± 0.00	96.55 ± 0.02	94.67 ± 0.02	93.00 ± 0.02	86.78 ± 0.02
5	70.10 ± 0.09	72.11 ± 0.06	71.38 ± 0.07	72.68 ± 0.07	70.12 ± 0.06
6	69.52 ± 0.08	78.93 ± 0.05	79.89 ± 0.04	80.14 ± 0.05	79.43 ± 0.05
7	68.23 ± 0.08	73.85 ± 0.12	70.71 ± 0.16	72.91 ± 0.15	72.67 ± 0.14
8	74.75 ± 0.09	74.56 ± 0.07	82.96 ± 0.08	86.16 ± 0.11	84.76 ± 0.09
9	93.81 ± 0.03	90.21 ± 0.06	90.66 ± 0.06	91.80 ± 0.06	91.36 ± 0.05
10	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
11	94.34 ± 0.04	89.16 ± 0.08	88.15 ± 0.07	90.76 ± 0.06	88.53 ± 0.08
12	99.73 ± 0.01	99.79 ± 0.00	98.99 ± 0.01	98.19 ± 0.02	98.05 ± 0.02
13	65.32 ± 0.12	60.61 ± 0.10	69.35 ± 0.13	71.19 ± 0.16	65.10 ± 0.16
14	100.00 ± 0.00	81.00 ± 0.08	92.97 ± 0.03	91.96 ± 0.04	90.24 ± 0.04
15	97.27 ± 0.02	92.41 ± 0.03	94.38 ± 0.03	94.18 ± 0.02	93.43 ± 0.02
16	94.14 ± 0.05	96.30 ± 0.06	95.39 ± 0.05	95.53 ± 0.05	92.16 ± 0.09
17	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00
Avg	**85.17 ± 0.04**	**87.18 ± 0.05**	**88.58 ± 0.05**	**89.11 ± 0.05**	**87.51 ± 0.05**

We also compared the sizes of the rule sets, average rule coverage, and the likelihood ratio of rules, computed from the entire data sets (not using cross-validation). Table 3 compares *CN2-SD* with *CN2-standard* and *CN2-WRAcc* in terms of the size of the rule set (S is the number of rules in a rule set, including the default rule), average rule coverage (CVG is computed as the averaged percentage of covered positive and negative examples per rule), and likelihood ratio[3] per rule.

The experimental results show that *CN2-SD* achieves improvements across the board. In terms of AUC, the smallest improvement is achieved by additive weights and slightly better improvements of 3–4% are by multiplicative weights. On the other hand, additive weights result in about 2 times less rules on average than multiplicative weights, and 6.5 times less rules than *CN2-standard*. Average rule coverage is also optimal for additive weights, improving on the average the coverage of *CN2-standard* rules with a factor of 3.5 and on *CN2-WRAcc* with a factor of 2. Note, however, that rules obtained with additive weights and multiplicative weights with high γ are highly overlapping, due to the relatively modest decrease of example weights.

In addition, there is also a substantial increase in the average likelihood ratio: while the ratios achieved by *CN2-standard* are already significant at the 99% level, this is further pushed up by *CN2-SD* with maximum values achieved by additive weights. An interesting question, to be verified with further experiments, is whether the weighted versions of the CN2 algorithm improve the significance of the induced subgroups also in the case when CN2 rules are induced without applying the significance test.

In summary, *CN2-SD* produces substantially smaller rule sets, where individual rules have higher coverage and significance.

Table 3. Average size (S), coverage (CVG) and likelihood ratio (LHR) of rules for different versions of the unordered algorithm induced from the entire data sets.

#	CN2 standard S	CVG	LHR	CN2 WRAcc S	CVG	LHR	CN2-SD ($\gamma = 0.5$) S	CVG	LHR	CN2-SD ($\gamma = 0.9$) S	CVG	LHR	CN2-SD (add. weight.) S	CVG	LHR
1	26	5.49	68.83	26	6.47	61.27	14	12.88	100.97	13	16.77	136.14	8	21.24	193.02
2	58	5.22	21.55	6	22.72	89.91	10	26.23	136.66	8	39.09	189.89	6	42.90	211.63
3	113	1.53	11.61	42	3.95	20.23	17	12.00	28.87	11	16.80	38.01	9	20.00	43.89
4	84	1.79	45.79	22	7.42	112.91	11	14.65	136.12	11	16.32	167.04	6	24.44	212.37
5	91	1.52	13.21	14	9.88	25.26	13	15.10	37.90	15	19.16	48.94	7	26.30	55.40
6	58	3.45	13.29	12	11.80	27.74	11	14.80	39.77	12	15.13	40.03	9	17.19	42.81
7	23	5.58	12.23	15	7.71	11.99	11	18.60	14.61	17	16.42	16.19	7	28.97	18.16
8	42	5.44	14.20	11	21.22	18.47	16	19.19	29.48	20	24.50	36.47	11	29.52	42.42
9	42	5.63	19.53	26	6.70	21.55	27	11.57	39.79	26	14.13	43.66	13	17.97	52.40
10	11	10.87	30.01	11	10.87	30.01	14	14.57	27.42	14	14.57	27.42	10	16.29	33.84
11	17	9.89	18.24	10	14.41	19.98	16	18.33	24.10	23	19.10	25.19	10	26.57	30.78
12	184	0.94	94.65	38	4.47	139.41	11	14.59	345.19	7	17.62	437.14	6	19.05	509.65
13	36	3.78	12.52	22	7.60	13.59	28	9.37	13.77	41	11.58	14.61	12	16.74	17.90
14	30	4.06	76.45	27	5.79	44.08	20	8.75	62.67	15	10.63	74.96	11	12.28	68.20
15	82	2.32	32.79	38	4.04	28.38	14	18.28	101.37	15	22.43	107.36	9	25.77	131.50
16	28	5.62	16.08	18	7.80	20.56	21	11.26	19.84	21	12.30	20.54	11	15.51	25.55
17	3	50.00	68.21	3	50.00	68.21	3	50.00	68.21	3	50.00	68.21	3	50.00	68.21
Avg	**54.6**	**7.24**	**33.48**	**20.0**	**11.93**	**44.33**	**15.1**	**17.07**	**72.16**	**16.9**	**19.88**	**87.75**	**8.7**	**24.16**	**103.40**

Finally, we illustrate our approach in the ROC space by means of the results on the Australian data set (Figure 1). The solid lines in this graph indicate the ROC curves obtained by *CN2-SD* and *CN2-standard*, evaluated with AUC-Method-2, i.e., probabilistic classification with overlapping rules: the top line (squares) for *CN2-SD* with additive weights, and the bottom line (triangles) for *CN2-standard*. *CN2-standard* finds many more rules than *CN2-SD*, which leads to overfitting as the ROC curve is mostly below the diagonal.

For illustrative purposes we also include positive and negative convex hulls constructed from individual subgroups using AUC-Method-1 (dotted lines). The points on the X and Y-axes close to the origin are all small, purely positive and negative subgroups found by *CN2-standard*, that do not contribute to the convex hull (presumably these are the rules that lead to poor performance using probabilistic classification). Using AUC-Method-1 we can remove

[3]The likelihood ratio is used in CN2 for testing the significance of the induced rule [4]. For two-class problems this statistic is distributed approximately as χ^2 with one degree of freedom.

those overly specific subgroups, leading to reasonable positive and negative convex hulls. Notice, however, that *CN2-SD* still improves on *CN2-standard* after removing redundant subgroups.

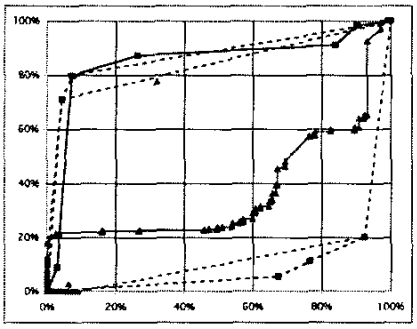

Figure 1. Example ROC curves on the Australian data set: solid curves for AUC-Method-2, and dotted positive and negative convex hulls for AUC-Method-1; squares for *CN2-SD* with additive weights, and triangles for *CN2-standard*.

5. Related work

Various rule evaluation measures and heuristics have been studied for subgroup discovery [10, 24], aimed at balancing the size of a group (referred to as factor g) with its distributional unusualness (referred to as factor p). The properties of functions that combine these two factors have been extensively studied (the so-called 'p-g-space', [10]). An alternative measure $q = \frac{TP}{FP+par}$ was proposed in [13], for expert-guided subgroup discovery in the TP/FP space, aimed at minimizing the number of false positives FP, and maximizing true positives TP, guided by generalization parameter par. Besides such 'objective' measures of interestingness, some 'subjective' measure of interestingness of a discovered pattern can be taken into the account, such as actionability ('a pattern is interesting if the user can do something with it to his or her advantage') and unexpectedness ('a pattern is interesting to the user if it is surprising to the user') [21].

Instance weights play an important role in boosting [12] and alternating decision trees [20]. Instance weights have been used also in variants of the covering algorithm implemented in rule learning approaches such as SLIPPER [6], RL [15] and DAIRY [9]. A variant of the weighted covering algorithm has been used also in the context of subgroup discovery for rule subset selection [13].

6. Conclusions

We have presented a novel approach to adapting standard classification rule learning to subgroup discovery. To this end we have appropriately adapted the covering algorithm, the search heuristics, the probabilistic classification and the performance measure. Experimental results on 17 UCI data sets demonstrate that *CN2-SD* produces substantially smaller rule sets, where individual rules have higher coverage and significance. These three factors are important for subgroup discovery: smaller size enables better understanding, higher coverage means larger support, and higher significance means that rules describe discovered subgroups that are significantly different from the entire population. We have evaluated the results of *CN2-SD* also in terms of AUC-Method-2 and shown insignificant increase in terms of the area under the ROC curve.

In further work we will evaluate the results also by using AUC-Method-1, where each subgroup establishes a separate point in the ROC space, and compare the results with the MIDOS subgroup discovery algorithm. We plan to investigate the behavior of *CN2-SD* also in multi-class problems. An interesting question, to be verified with further experiments, is whether the weighted versions of the CN2 algorithm improve the significance of the induced subgroups also in the case when CN2 rules are induced without applying the significance test. Finally, we plan to use the *CN2-SD* subgroup discovery algorithm for solving practical problems, in which expert evaluations of induced subgroup descriptions is of ultimate interest.

Acknowledgements

Thanks to Dragan Gamberger for joint work on the weighted covering algorithm, and José Hernández-Orallo and Cesar Ferri-Ramírez for joint work on AUC. The work reported in this paper was supported by the Slovenian Ministry of Education, Science and Sport, the IST-1999-11495 project Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise, and the British Council project Partnership in Science PSP-18.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo. Fast discovery of association rules. In U.M. Fayyad, G. Piatetski-Shapiro, P. Smyth and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, 307–328. AAAI Press, 1996.

[2] B. Cestnik. Estimating probabilities: A crucial task in machine learning. In L. Aiello, editor, *Proc. of the 9th European Conference on Artificial Intelligence*, 147–149. Pitman, 1990.

[3] 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.

[4] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4): 261–283, 1989.

[5] W.W. Cohen. (1995) Fast effective rule induction. In *Proc. of the 12th International Conference on Machine Learning*, 115–123. Morgan Kaufmann, 1995.

[6] 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.

[7] S. Džeroski, B. Cestnik, and I. Petrovski. (1993) Using the m-estimate in rule induction. *Journal of Computing and Information Technology*, 1(1):37 – 46, 1993.

[8] U.M. Fayyad and K.B. Irani, K.B. Multi-interval discretisation of continuous-valued attributes for classification learning. In R. Bajcsy, editor, *Proc. of the 13th International Joint Conference on Artificial Intelligence*, 1022–1027. Morgan Kaufmann, 1993.

[9] D. Hsu, O. Etzioni and S. Soderland. A redundant covering algorithm applied to text classification. In *Proc. of the AAAI Workshop on Learning from Text Categorization*. American Association for Artificial Intelligence, 1998.

[10] 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.

[11] C. Ferri-Ramírez, P.A. Flach, and J. Hernandez-Orallo. Learning decision trees using the area under the ROC curve. In *Proc. of the 19th International Conference on Machine Learning*, 139–146. Morgan Kaufmann, 2002.

[12] Y. Freund and R.E. Shapire. Experiments with a new boosting algorithm. In *Proc. of the 13th International Conference on Machine Learning*, 148–156. Morgan Kaufmann, 1996.

[13] D. Gamberger and N. Lavrač. Descriptive induction through subgroup discovery: A case study in a medical domain. In *Proc. of the 19th International Conference on Machine Learning*, 163–170. Morgan Kaufmann, 2002.

[14] N. Lavrač, P. Flach, and B. Zupan. Rule evaluation measures: A unifying view. In *Proc. of the 9th International Workshop on Inductive Logic Programming*, 74–185. Springer, 1999.

[15] Y. Lee, B.G. Buchanan, and J.M. Aronis. Knowledge-based learning in exploratory science: Learning rules to predict rodent carcinogenicity. *Machine Learning*, 30: 217–240, 1998.

[16] 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*, 1041–1045. Morgan Kaufmann, 1986.

[17] 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.

[18] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3): 203–231, 2001.

[19] R.L. Rivest. Learning decision lists. *Machine Learning*, 2(3): 229–246, 1987.

[20] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Proc. of the 11th Conference on Computational Learning Theory*, 80–91. ACM Press, 1998.

[21] A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proc. of the 1st International Conference on Knowledge Discovery and Data Mining*, 275–281, 1995.

[22] L. Todorovski, P. Flach, and N. Lavrač. Predictive performance of weighted relative accuracy. In D.A. Zighed, J. Komorowski, and J. Zytkow, editors, *Proc. of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, 255–264. Springer, 2000.

[23] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[24] S. Wrobel. 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, 1997.