Journal of Machine Learning Research 5 (2004) 153-188 Submitted 12/02; Published 2/04

Subgroup Discovery with CN2-SD

Nada Lavrač NADA.LAVRAC@IJS.SI
Jožef Stefan Institute
Jamova 39
1000 Ljubljana, Slovenia and
Nova Gorica Polytechnic
Vipavska 13
5100 Nova
Gorica, Slovenia

Branko Kavšek BRANKO.KAVSEK@IJS.SI
Jožef Stefan Institute
Jamova 39
1000 Ljubljana, Slovenia

Peter Flach PETER.FLACH@BRISTOL.AC.UK
University of Bristol
Woodland Road
Bristol BS8 1UB, United Kingdom

Ljupčo Todorovski LJUPCO.TODOROVSKI@IJS.SI
Jožef Stefan Institute
Jamova 39
1000 Ljubljana, Slovenia

Editor: Stefan Wrobel

Abstract

This paper investigates how to adapt standard classification rule learning approaches to subgroup discovery. The goal of subgroup discovery is to find rules describing subsets of the population that are sufficiently large and statistically unusual. The paper presents a subgroup discovery algorithm, *CN2-SD*, developed by modifying parts of the CN2 classification rule learner: its covering algorithm, search heuristic, probabilistic classification of instances, and evaluation measures. Experimental evaluation of *CN2-SD* on 23 UCI data sets shows 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 ROC curve, when compared with the CN2 algorithm. Application of *CN2-SD* to a large traffic accident data set confirms these findings.

Keywords: Rule Learning, Subgroup Discovery, UCI Data Sets, Traffic Accident Data Analysis

1. Introduction

Rule learning is most frequently used in the context of classification rule learning (Michalski et al., 1986, Clark and Niblett, 1989, Cohen, 1995) and association rule learning (Agrawal et al., 1996). While classification rule learning is an approach to *predictive induction* (or supervised learning), aimed at constructing a set of rules to be used for classification and/or prediction, association rule learning is a form of *descriptive induction* (non-classificatory induction or unsupervised learning), aimed at the discovery of individual rules which define interesting patterns in data.

Descriptive induction has recently gained much attention of the rule learning research community. Besides mining of association rules (e.g., the APRIORI association rule learning algorithm (Agrawal et al., 1996)), other approaches have been developed, including clausal discovery as in the CLAUDIEN system (Raedt and Dehaspe, 1997, Raedt et al., 2001), and database dependency discovery (Flach and Savnik, 1999).

1.1 Subgroup Discovery: A Task at the Intersection of Predictive and Descriptive Induction

This paper shows how classification rule learning can be adapted to *subgroup discovery*, a task at the intersection of predictive and descriptive induction, that has first been formulated by Klösgen (1996) and Wrobel (1997, 2001), and addressed by rule learning algorithms EXPLORA (Klösgen, 1996) and MIDOS (Wrobel, 1997, 2001). In the work of Klösgen (1996) and Wrobel (1997, 2001), the problem of subgroup discovery has been 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.

In subgroup discovery, rules have the form $Class \leftarrow Cond$, where the property of interest for subgroup discovery is class value *Class* that appears in the rule consequent, and the rule antecedent *Cond* is a conjunction of features (attribute-value pairs) selected from the features describing the training instances. As rules are induced from labeled training instances (labeled positive if the property of interest holds, and negative otherwise), the process of subgroup discovery is targeted at uncovering properties of a selected *target* population of individuals with the given property of interest. In this sense, subgroup discovery is a form of supervised learning. However, in many respects subgroup discovery is a form of descriptive induction as the task is to uncover individual interesting *patterns* in data. The standard assumptions made by classification rule learning algorithms (especially the ones that take the covering approach), such as 'induced rules should be as accurate as possible' or 'induced rules should be as distinct as possible, covering different parts of the population', need to be relaxed. In our approach, the first assumption, implemented in classification rule learners by heuristic which aim at optimizing predictive accuracy, is relaxed by implementing new heuristics for subgroup discovery which aim at finding 'best' subgroups in terms of rule coverage and distributional unusualness. The relaxation of the second assumption enables the discovery of overlapping subgroups, describing some population segments in a multiplicity of ways. Induced subgroup descriptions may be redundant, if viewed from a classifier perspective, but very valuable in terms of their descriptive power, uncovering genuine properties of subpopulations from different viewpoints.

Let us emphasize the difference between subgroup discovery (as a task at the intersection of predictive and descriptive induction) and classification rule learning (as a form of predictive induction). The goal of standard rule learning is to generate models, one for each class, consisting of rule sets describing class characteristics in terms of properties occurring in the descriptions of training examples. In contrast, subgroup discovery aims at discovering individual rules or 'patterns' of interest, which must be represented in explicit symbolic form and which must be relatively simple in order to be recognized as actionable by potential users. Moreover, standard classification rule learning algorithms cannot appropriately address the task of subgroup discovery as they use the covering algorithm for rule set construction which - as will be seen in this paper - hinders the applicability of classification rule induction approaches in subgroup discovery.

Subgroup discovery is usually seen as different from classification, as it addresses different goals (discovery of interesting population subgroups instead of maximizing classification accuracy of the induced rule set). This is manifested also by the fact that in subgroup discovery one can often tolerate many more false positives (negative examples incorrectly classified as positives) than in a classification task. However, both tasks, subgroup discovery and classification rule learning, can be unified under the umbrella of cost-sensitive classification. This is because when deciding which classifiers are optimal in a given context it does not matter whether we penalize false negatives as is the case in classification, or reward true positives as in subgroup discovery.

1.2 Overview of the *CN2-SD* Approach to Subgroup Discovery

This paper investigates how to adapt standard classification rule learning approaches to subgroup discovery. The proposed modifications of classification rule learners can, in principle, be used to modify any rule learner using the covering algorithm for rule set construction. In this paper, we illustrate the approach by modifying the well-known CN2 rule learning algorithm (Clark and Niblett, 1989, Clark and Boswell, 1991). Alternatively, we could have modified RL (Lee et al., 1998), RIPPER (Cohen, 1995), SLIPPER (Cohen and Singer, 1999) or other more sophisticated classification rule learners. The reason for modifying CN2 is that other more sophisticated learners include advanced techniques 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* by modifying CN2 (Clark and Niblett, 1989, Clark and Boswell, 1991). The proposed approach performs subgroup discovery through the following modifications of CN2: (a) replacing the accuracy-based search heuristic with a new weighted relative accuracy heuristic that trades off generality and accuracy of the rule, (b) incorporating example weights into the covering algorithm, (c) incorporating example weights into the weighted relative accuracy search heuristic, and (d) using 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. In addition, we have extended the ROC analysis framework to subgroup discovery and propose a set of measures appropriate for evaluating the quality of induced subgroups.

This paper presents the *CN2-SD* subgroup discovery algorithm, together with its experimental evaluation on 23 data sets of the UCI Repository of Machine Learning Databases (Murphy and Aha, 1994), as well as its application to a real world problem of traffic accident analysis. 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 have significantly different distributional characteristics compared to the entire population. The appropriateness for subgroup discovery is confirmed also by slight improvements in terms of the area under ROC curve, without decreasing predictive accuracy.

The paper is organized as follows. Section 2 introduces the background of this work which includes the description of the CN2 rule learning algorithm, the weighted relative accuracy heuristic, and probabilistic classification of new examples. Section 3 presents the subgroup discovery algorithm *CN2-SD* by describing the necessary modifications of CN2. In Section 4 we discuss subgroup discovery from the perspective of ROC analysis. Section 5 presents a range of metrics used in the experimental evaluation of *CN2-SD*. Section 6 presents the results of experiments on selected UCI data sets as well as an application of *CN2-SD* on a real-life traffic accident data set. Related work is discussed in Section 7. Section 8 concludes by summarizing the main contributions and proposing directions for further work.

2. Background

This section presents the background of our work: the classical CN2 rule induction algorithm, including the covering algorithm for rule set construction, the standard CN2 heuristic, weighted relative accuracy heuristic, and the probabilistic classification technique used in CN2.

2.1 The CN2 Rule Induction Algorithm

CN2 is an algorithm for inducing propositional classification rules (Clark and Niblett, 1989, Clark and Boswell, 1991). Induced rules have the form "`if` *Cond* `then` *Class*", where *Cond* is a conjunction of features (pairs of attributes and their values) and *Class* is the class value. In this paper we use the notation $Class \leftarrow Cond$.

CN2 consists of two main procedures: the bottom-level search procedure that performs beam search in order to find a single rule, and the top-level control procedure that repeatedly executes the bottom-level search to induce a rule set. The bottom-level performs beam search[1] using classification accuracy of the rule as a heuristic function. The accuracy of a propositional classification rule of the form $Class \leftarrow Cond$ is equal to the conditional probability of class *Class*, given that condition *Cond* is satisfied:

$$Acc(Class \leftarrow Cond) = p(Class|Cond) = \frac{p(Class.Cond)}{p(Cond)}.$$

Usually, this probability is estimated by relative frequency $\frac{n(Class.Cond)}{n(Cond)}$.[2] Different probability estimates, like the Laplace (Clark and Boswell, 1991) or the m-estimate (Cestnik, 1990, Džeroski et al., 1993), can be used in CN2 for estimating the above probability. The standard CN2 algorithm used in this work uses the Laplace estimate, which is computed as $\frac{n(Class.Cond)+1}{n(Cond)+k}$, where k is the number of classes (for a two-class problem, $k = 2$).

CN2 can also 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 (Clark and Niblett, 1989) that measures the difference between the class probability distribution in the set of training examples covered by the rule and the class probability distribution in the set of all training examples (see Equation 2 in Section 5). The empirical evaluation in the work of Clark and Boswell (1991) shows that applying the significance test reduces the number of induced rules at a cost of slightly decreased predictive accuracy.

1. CN2 constructs rules in a general-to-specific fashion, specializing only the rules in the beam (the best rules) by iteratively adding features to condition *Cond*. This procedure stops when no specialized rule can be added to the beam, because none of the specializations is more accurate than the rules in the beam.
2. Here we use the following notation: $n(Cond)$ stands for the number of instances covered by rule $Class \leftarrow Cond$, $n(Class)$ stands for the number of examples of class *Class*, and $n(Class.Cond)$ stands for the number of correctly classified examples (true positives). We use $p(\ldots)$ for the corresponding probabilities.

Two different top-level control procedures can be used in CN2. The first induces an ordered list of rules and the second an unordered set of rules. Both procedures add a default rule (providing for majority class assignment) as the final rule in the induced rule set. When inducing an ordered list of rules, the search procedure looks for the most accurate rule in the current set of training examples. The rule predicts the most frequent class in the set of covered examples. In order to prevent CN2 finding the same rule again, all the covered examples are removed before a new iteration is started at the top-level. The control procedure invokes a new search, until all the examples are covered or no significant rule can be found. 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.

2.2 The Weighted Relative Accuracy Heuristic

Weighted relative accuracy (Lavrač et al., 1999, Todorovski et al., 2000) is a variant of rule accuracy that can be applied both in the descriptive and predictive induction framework; in this paper this heuristic is applied for subgroup discovery. Weighted relative accuracy, a reformulation of one of the heuristics used in EXPLORA (Klösgen, 1996) and MIDOS (Wrobel, 1997), is defined as follows:

$$WRAcc(Class \leftarrow Cond) = p(Cond) \cdot (p(Class|Cond) - p(Class)). \tag{1}$$

Like most other heuristics used in subgroup discovery systems, weighted relative accuracy consists of two components, providing a tradeoff between rule *generality* (or the relative size of a subgroup $p(Cond)$) and distributional unusualness or *relative accuracy* (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 $Class \leftarrow true$, which predicts that all instances belong to *Class*: a rule is interesting only if it improves upon this 'default' accuracy. Another aspect of relative accuracy is that it measures the difference between true positives and the expected true positives (expected under the assumption of independence of the left and right hand-side of a rule), i.e., 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 the work of Klösgen (1996), these quantities are referred to as g (generality), p (rule accuracy or precision) and p_0 (default rule accuracy) and different tradeoffs between rule generality and precision in the so-called p-g (precision-generality) space are proposed. In addition to function $g(p-p_0)$, which is equivalent to our weighted relative accuracy heuristic, other tradeoffs that reduce the influence of generality are proposed, e.g., $\sqrt{g}(p-p_0)$ or $\sqrt{g/(1-g)}(p-p_0)$. Here, we favor the weighted relative accuracy heuristic, because it has an intuitive interpretation in ROC space, discussed in Section 4.

2.3 Probabilistic Classification

The induced rules can be ordered or unordered. Ordered rules are interpreted as a decision list (Rivest, 1987) 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.

if	legs = 2	&	feathers = yes	then	class = bird	[13,0]
if	beak = yes			then	class = bird	[20,0]
if	size = large	&	flies = no	then	class = elephant	[2,10]

Table 1: A rule set consisting of two rules for class 'bird' and one rule for class 'elephant'.

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

if *Cond* 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, the default rule is invoked to predict the majority class of training instances not covered by the other rules in the list.

Probabilistic classification is illustrated on a sample classification task, taken from Clark and Boswell (1991). Suppose we need to classify an animal which is a two-legged, feathered, large, non-flying and has a beak,[3], and the classification is based on a rule set, listed in Table 1 formed of three probabilistic rules with the [bird, elephant] class distribution assigned to each rule (for simplicity, the rule set does not include the default rule). All rules fire for the animal to be classified, resulting in a [35,10] class distribution. As a result, the animal is classified as a bird.

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 'ordered' induction algorithm, and the area under ROC curve rule set evaluation. This section describes the CN2 modifications, while ROC analysis and a novel interpretation of the weighted relative accuracy heuristic in ROC space are given in Section 4.

3.1 Weighted Covering Algorithm

If used for subgroup discovery, one of the problems of standard rule learners, such as CN2 and RIPPER, is 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 descriptions with sufficient coverage and significance. In the subsequent iterations of the covering algorithm, rules are induced from biased example subsets, i.e., subsets including only positive examples that are not covered by previously induced rules, which inappropriately biases the subgroup discovery process.

3. The animal being classified is a weka.

As a remedy to this problem we propose the use of a weighted covering algorithm (Gamberger and Lavrač, 2002), in which the subsequently induced rules (i.e., rules induced in the later stages) also represent interesting and sufficiently large subgroups of the 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 indicating how often (with how many rules) 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 1, $w(e_j,0) = 1$. The initial example weight $w(e_j,0) = 1$ means that the example has not been covered by any rule, meaning 'please cover this example, since it has not been covered before', while lower weights, $0 < w < 1$ mean 'do not try too hard on this example'. Consequently, the examples already covered by one or more constructed rules decrease their weights while the uncovered target class examples whose weights have not been decreased will have a greater chance to be covered in the following iterations of the algorithm.

For a weighted covering algorithm to be used, we have to specify the weighting scheme, i.e., how the weight of each example decreases with the increasing number of covering rules. We have implemented two weighting schemes described below.

3.1.1 MULTIPLICATIVE WEIGHTS

In the first scheme, weights decrease multiplicatively. For a given parameter $0 < \gamma < 1$, 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 by i rules. 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 covering algorithm.

3.1.2 ADDITIVE WEIGHTS

In the second scheme, weights of covered positive examples decrease according to the formula $w(e_j,i) = \frac{1}{i+1}$. In the first iteration all target class examples contribute the same weight $w(e_j,0) = 1$, while in the following iterations the contributions of examples are inversely proportional to their coverage by previously induced rules.

3.2 Modified *WRAcc* Heuristic with Example Weights

The modification of CN2 reported in the work of Todorovski et al. (2000) affected only the heuristic function: weighted relative accuracy was used as a search heuristic, instead of the accuracy heuristic of the original CN2, while everything else remained the same. In this work, the heuristic function is further modified to handle example weights, which provide the means to consider different parts of the example 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 modified *WRAcc* measure is then defined as follows:

$$WRAcc(Class \leftarrow Cond) = \frac{n'(Cond)}{N'} \cdot \left(\frac{n'(Class.Cond)}{n'(Cond)} - \frac{n'(Class)}{N'}\right).$$

`if`	legs = 2	&	feathers = yes	`then`	class = bird	[1, 0]
`if`	beak = yes			`then`	class = bird	[1, 0]
`if`	size = large	&	flies = no	`then`	class = elephant	[0.17,0.83]

Table 2: The rule set of Table 1 as treated by *CN2-SD*.

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 $n'(Class.Cond)$ 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 number 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 account, hence 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 rule head.

Note that *CN2-SD* does not use the same probabilistic classification scheme as CN2. Unlike CN2, where the rule class distribution is computed in terms of the numbers of examples covered, *CN2-SD* treats the class distribution in terms of probabilities (computed by the relative frequency estimate). Table 2 presents the three rules of Table 1 with the class distribution expressed with probabilities. A two-legged, feathered, large, non-flying animal with a beak is again classified as a bird but now the probabilities are averaged (instead of summing the numbers of examples), resulting in the final probability distribution [0.72,0.28]. By using this voting scheme the subgroups covering a small number of examples are not so heavily penalized (as is the case in CN2) when classifying a new example.

3.4 *CN2-SD* Implementation

Two variants of *CN2-SD* have been implemented. The *CN2-SD* subgroup discovery algorithm used in the experiments in this paper is implemented in C and runs on a number of UNIX platforms. Its predecessor, used in the experiments reported by Lavrač et al. (2002), is implemented in Java and incorporated in the WEKA data mining environment (Witten and Frank, 1999). The C implementation is more efficient and less restrictive than the Java implementation, which is limited to binary class problems and to discrete attributes.

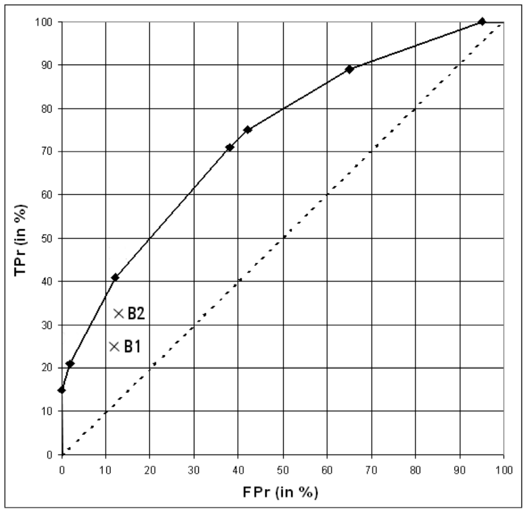

Figure 1: The ROC space with TPr on the X axis and FPr on the Y axis. The solid line connecting seven optimal subgroups (marked by ♦) is the ROC convex hull. $B1$ and $B2$ denote suboptimal subgroups (marked by x). The dotted line – the diagonal connecting points (0,0) and (100,100) – indicates positions of insignificant rules with zero relative accuracy.

4. ROC Analysis for Subgroup Discovery

In this section we describe how ROC (Receiver Operating Characteristic) analysis (Provost and Fawcett, 2001) can be used to understand subgroup discovery and to visualize and evaluate discovered subgroups.

A point in *ROC space* shows classifier performance in terms of false alarm or *false positive rate* $FPr = \frac{FP}{TN+FP}$ (plotted on the X-axis), and sensitivity or *true positive rate* $TPr = \frac{TP}{TP+FN}$ (plotted on the Y-axis). In terms of the expressions introduced in Sections 2.1 and 2.2, TP (true positives), FP (false positives), TN (true negatives) and FN (false negatives) can be expressed as: $TP = n(Class.Cond)$, $FP = n(\overline{Class}.Cond)$, $TN = n(\overline{Class}.\overline{Cond})$ and $FN = n(Class.\overline{Cond})$, where $\overline{Class}$ and $\overline{Cond}$ stand for $\neg Class$ and $\neg Cond$, respectively.

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. Significant rules define the points in ROC space from which a convex hull can be constructed. The best rules define the ROC convex hull. Figure 1 shows seven rules on the convex hull (marked by ♦), while two rules $B1$ and $B2$ below the convex hull (marked by x) are of lower quality.

4.1 The Interpretation of Weighted Relative Accuracy in ROC Space

Weighted relative accuracy is appropriate for measuring the quality of a single subgroup, because it is proportional to the distance from the diagonal in ROC space.[4] To see that this holds, 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 in terms of its TPr/FPr tradeoff in ROC space. 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. Consequently, all point on the diagonal represent insignificant subgroups.

Relative accuracy, $p(Class|Cond) - p(Class)$, 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, *WRAcc* isometrics lie on straight lines parallel to the diagonal (Flach, 2003, Fürnkranz and Flach, 2003). Consequently, a point on the line $TPr = FPr + a$, where a is the vertical distance of the line to the diagonal, has $WRAcc = a.p(Class)p(\overline{Class})$. Thus, given a fixed class distribution, *WRAcc* is proportional to the vertical distance a to the diagonal. In fact, the quantity $TPr - FPr$ would be an alternative quality measure for subgroups, with the additional advantage that it allows for comparison of subgroups from populations with different class distributions.

4.2 Methods for Constructing ROC Curves and AUC Evaluation

Subgroups obtained by CN2-SD can be evaluated in ROC space in two different ways.

4.2.1 AUC-METHOD-1

The first method treats each rule as a separate subgroup which is plotted in ROC space in terms of its true and false positive rates (TPr and FPr). We then generate 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, this method does not take account of any overlap between subgroups, and subgroups not on the convex hull are simply ignored.

Figure 2 presents two ROC curves, showing the performance of CN2 and *CN2-SD* algorithms on the Australian UCI data set.

4.2.2 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 (the probability of the target class): if the positive probabil-

4. Some of the reasoning supporting this claim is further discussed in the last two paragraphs of Section 5.1.

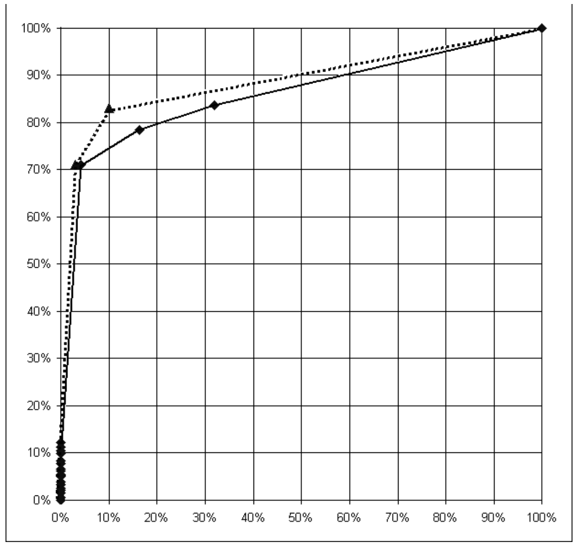

Figure 2: Example ROC curves (AUC-Method-1) on the Australian UCI data set: the solid curve for the standard CN2 classification rule learner, and the dotted curve for *CN2-SD*.

ity is larger than this threshold we predict positive, else negative. The ROC curve can be constructed by varying this threshold from 1 (all predictions negative, corresponding to (0,0) in ROC space) to 0 (all predictions positive, corresponding to (1,1) in ROC space). This results in $n+1$ points in ROC space, where n is the total number of classified examples (test instances). Equivalently, we can order all the classified examples by decreasing 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).[5] Each point on this curve corresponds to a classifier defined by a possible probability threshold, as opposed to AUC-Method-1, where a point on the ROC curve corresponds to one of the optimal subgroups. The ROC curve depicts a set of classifiers, whereas the area under this ROC curve indicates the combined quality of all subgroups (i.e., the quality of the entire rule set). This method can be used with a test set or in cross-validation, but the resulting curve is not necessarily convex.[6]

Figure 3 presents two ROC curves, showing the performance of the CN2 and *CN2-SD* algorithms on the Australian UCI data set. It is apparent from this figure that CN2 is badly overfitting on this data set, because almost all of its ROC curve is below the diagonal. This is because CN2 has learned many overly specific rules, which bias the predicted probabilities. These overly specific rules are visible in Figure 2 as points close to the origin.

5. In the case of ties, we make the appropriate number of steps up and to the right at once, drawing a diagonal line segment.
6. A description of this method applied to decision tree induction can be found in the paper by Ferri-Ramírez et al. (2002).

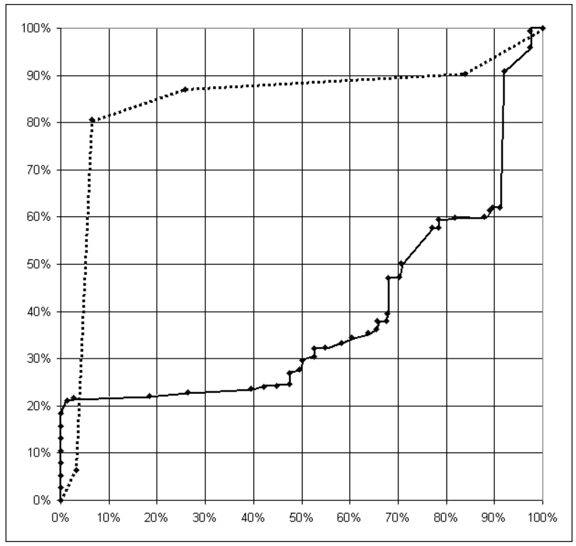

Figure 3: Example ROC curves (AUC-Method-2) on the Australian UCI data set: the solid curve for the standard CN2 classification rule learner, and the dotted curve for *CN2-SD*.

4.2.3 COMPARISON OF THE TWO AUC METHODS

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 intended to be applied also in the predictive setting, as a rule set (a model) used for classification. Its advantage is also that it is easier to apply cross-validation. In the experimental evaluation in Section 6 we use AUC-Method-2 in the comparison of the predictive performance of rule learners.

An argument in favor of using AUC-Method-1 for subgroup evaluation is based on the observation that AUC-Method-1 suggests to eliminate, from the induced set of subgroup descriptions, those rules which are not on the ROC convex hull. This seems appropriate, as the 'best' subgroups according to the *WRAcc* evaluation measure, are subgroups most distant from the ROC diagonal. However, disjoint subgroups, either on or close to the convex hull, should not be eliminated, as (due to disjoint coverage and possibly different symbolic descriptions) they may represent interesting subgroups, regardless of the fact that there is another 'better' subgroup on the ROC convex hull, with a similar TPr/FPr tradeoff.

Notice that the area under ROC curve (AUC-Method-1) cannot be used as a predictive quality measure when comparing different subgroup miners, because it does not take into account the overlapping structure of subgroups. An argument against the use of this measure is here elaborated through a simple example.[7] Consider for instance two subgroup mining results, of say 3 subgroups in each resulting rule set. The first result set consists of three disjoint subgroups of equal size that together cover all the examples of the selected *Class* value and have a 100% accuracy. Thus these three subgroups are a perfect classifier for the *Class* value. In ROC space the three subgroups collapse at the point (0,1/3). The second result set consists of three equal subgroups (having a max-

7. We are grateful to the anonymous reviewer who provided this illustrative example.

imum overlap: with different descriptions, but equal extensions), also with a 100% accuracy and covering one third of the class examples. Clearly the first result is better, but the representation of the results in ROC space (and the area under ROC curve) is the same for both cases.

5. Subgroup Evaluation Measures

In this section we distinguish between *predictive* and *descriptive* evaluation measures, which is in-line with the distinction of predictive induction and descriptive induction made in Section 1. Descriptive measures are used to evaluate the quality of individual rules (individual patterns). These quality measures are the most appropriate for subgroup discovery, as the task of subgroup discovery is to induce individual patterns of interest. Predictive measures are used in addition to descriptive measures just to show that the *CN2-SD* subgroup discovery mechanisms perform well also in the predictive induction setting, where the goal is to induce a classifier.

5.1 Descriptive Measures of Rule Interestingness

Descriptive measures of rule interestingness evaluate each individual subgroup and are thus appropriate for evaluating the success of subgroup discovery. The proposed quality measures compute the average over the induced set of subgroup descriptions, which enables the comparison between different algorithms.

Coverage. The average coverage measures the percentage of examples covered on average by one rule of the induced rule set. Coverage of a single rule R_i is defined as

$$Cov(R_i) = Cov(Class \leftarrow Cond_i) = p(Cond_i) = \frac{n(Cond_i)}{N}.$$

The average coverage of a rule set is computed as

$$COV = \frac{1}{n_R}\sum_{i=1}^{n_R} Cov(R_i),$$

where n_R is the number of induced rules.

Support. For subgroup discovery it is interesting to compute the overall support (the target coverage) as the percentage of target examples (positives) covered by the rules, computed as the true positive rate for the union of subgroups. Support of a rule is defined as the frequency of correctly classified covered examples:

$$Sup(R_i) = Sup(Class \leftarrow Cond_i) = p(Class.Cond_i) = \frac{n(Class.Cond_i)}{N}.$$

The overall support of a rule set is computed as

$$SUP = \frac{1}{N}\sum_{Class_j} n(Class_j \cdot \bigvee_{Class_j \leftarrow Cond_i} Cond_i),$$

where the examples covered by several rules are counted only once (hence the disjunction of rule conditions of rules with the same $Class_j$ value in the rule head).

Size. Size is a measure of complexity (the syntactical complexity of induced rules). The rule set size is computed as the number of rules in the induced rule set (including the default rule):

$$SIZE = n_R.$$

In addition to rule set size used in this paper, complexity could be measured also by the average number of rules/subgroups per class, and the average number of features per rule.

Significance. Average rule significance is computed in terms of the likelihood ratio of a rule, normalized with the likelihood ratio of the significance threshold (99%); the average is computed over all the rules. Significance (or *evidence*, in the terminology of Klösgen, 1996) indicates how significant is a finding, if measured by this statistical criterion. In the CN2 algorithm (Clark and Niblett, 1989), significance is measured in terms of the likelihood ratio statistic of a rule as follows:

$$Sig(R_i) = Sig(Class \leftarrow Cond_i) = 2 \cdot \sum_j n(Class_j.Cond_i) \cdot \log \frac{n(Class_j.Cond_i)}{n(Class_j)}. \qquad (2)$$

where for each class $Class_j$, $n(Class_j.Cond_i)$ denotes the number of instances of $Class_j$ in the set where the rule body holds true, and $n(Class_j)$ is the number of $Class_j$ instances. Note that although for each generated subgroup description one class is selected as the target class, the significance criterion measures the distributional unusualness unbiased to any particular class – as such, it measures the significance of rule condition only.

The average significance of a rule set is computed as:

$$SIG = \frac{1}{n_R} \sum_{i=1}^{n_R} Sig(R_i).$$

Unusualness. Average rule unusualness is computed as the average *WRAcc* computed over all the rules:

$$WRACC = \frac{1}{n_R} \sum_{i=1}^{n_R} WRAcc(R_i).$$

As discussed in Section 4.1, *WRAcc* is appropriate for measuring the unusualness of separate subgroups, because it is proportional to the vertical distance from the diagonal in the ROC space (see the underlying reasoning in Section 4.1).

As *WRAcc* is proportional to the distance to the diagonal in ROC space, *WRAcc* also reflects rule significance – the larger *WRAcc*, the more significant the rule, and vice versa. As both *WRAcc* and rule significance measure the distributional unusualness of a subgroup, they are the most important quality measures for subgroup discovery. However, while significance only measures distributional unusualness, *WRAcc* takes also rule coverage into account, therefore we consider *unusualness* – computed by the average *WRAcc* – to be the most appropriate measure for subgroup quality evaluation.

As pointed out in Section 4.1, the quantity $TPr - FPr$ could 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 of its '*p-g*' (precision-generality) interpretation, which is particularly suitable for subgroup discovery.

5.2 Predictive Measures of Rule Set Classification Performance

Predictive measures evaluate a rule set, interpreting a set of subgroup descriptions as a predictive model. Despite the fact that optimizing accuracy is not the intended goal of subgroup discovery algorithms, these measures can be used in order to provide a comparison of *CN2-SD* with standard classification rule learners.

Predictive accuracy. The percentage of correctly predicted instances. For a binary classification problem, rule set accuracy is computed as follows:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN}.$$

Note that ACC measures the accuracy of the whole rule set on both positive and negative examples, while rule accuracy (defined as $Acc(Class \leftarrow Cond) = p(Class|Cond)$) measures the accuracy of a single rule on positives only.

Area under ROC curve. The AUC-Method-2, described in Section 4.2, applicable to rule sets is selected as the evaluation measure. It interprets a rule set as a probabilistic model, given all the different probability thresholds as defined through the probabilistic classification of test instances.

6. Experimental Evaluation

For subgroup discovery, expert 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. For this purpose we tested it on selected UCI data sets. In this paper we use the same data sets as in the work of Todorovski et al. (2000). We have applied *CN2-SD* also to a real life problem of traffic accident analysis; these results were evaluated also by the expert.

6.1 The Experimental Setting

To test the applicability of *CN2-SD* to the subgroup discovery task, we compare its performance with the performance of the standard CN2 classification rule learning algorithm (referred to as *CN2-standard*, and described in the work of Clark and Boswell, 1991) as well as with the CN2 algorithm using *WRAcc* (*CN2-WRAcc*, described by Todorovski et al., 2000).

In this comparative study all the parameters of the CN2 algorithm are set to their default values (beam-size = 5, significance-threshold = 99%). The results of the *CN2-SD* algorithm are computed using both multiplicative weights (with $\gamma = 0.5, 0.7, 0.9$)[8] and additive weights.

We estimate the performance of the algorithms using stratified 10-fold cross-validation. The obtained estimates are presented in terms of their average values and standard deviations.

Statistical significance of the difference in performance compared to *CN2-standard* is tested using the paired t-test (exactly the same folds are used in all comparisons) with significance level of 95%: bold font and ↑ to the right of a result in all the tables means that the algorithm is significantly better than *CN2-standard* while ↓ means it is significantly worse. The same paired t-test is used to compare the different versions of our algorithm with *CN2-standard* over all the data sets.

8. Results obtained with $\gamma = 0.7$ are presented in the tables of Appendix A but not in the main part of the paper.

	Data set	#Att.	#D.att.	#C.att.	#Class	#Ex.	Maj. Class (%)
1	australian	14	8	6	2	690	56
2	breast-w	9	9	0	2	699	66
3	bridges-td	7	4	3	2	102	85
4	chess	36	36	0	2	3196	52
5	diabetes	8	0	8	2	768	65
6	echo	6	1	5	2	131	67
7	german	20	13	7	2	1000	70
8	heart	13	6	7	2	270	56
9	hepatitis	19	13	6	2	155	79
10	hypothyroid	25	18	7	2	3163	95
11	ionosphere	34	0	34	2	351	64
12	iris	4	0	4	3	150	33
13	mutagen	59	57	2	2	188	66
14	mutagen-f	57	57	0	2	188	66
15	tic-tac-toe	9	9	0	2	958	65
16	vote	16	16	0	2	435	61
17	balance	4	0	4	3	625	46
18	car	6	6	0	4	1728	70
19	glass	9	0	9	6	214	36
20	image	19	0	19	7	2310	14
21	soya	35	35	0	19	683	13
22	waveform	21	0	21	3	5000	34
23	wine	13	0	13	3	178	40

Table 3: Properties of the UCI data sets.

6.2 Experiments on UCI Data Sets

We experimentally evaluate our approach on 23 data sets from the UCI Repository of Machine Learning Databases (Murphy and Aha, 1994). Table 3 gives an overview of the selected data sets in terms of the number of attributes (total, discrete, continuous), the number of classes, the number of examples, and the percentage of examples of the majority class. These data sets have been widely used in other comparative studies (Todorovski et al., 2000). We have divided the data sets in two groups (Table 3), those with two classes (binary data sets 1–16) and those with more then two classes (multi-class data sets 17–23). This distinction is made as ROC analysis is applied only on binary data sets.[9]

6.2.1 RESULTS OF THE UNORDERED *CN2-SD*

Tables 4 and 5 present summary results of the UCI experiments, while details can be found in Tables 14–20 in Appendix A. For each performance measure, the summary table shows the average value over all the data sets, the significance of the results compared to *CN2-standard* (p-value), win/loss/draw in terms of the number of data sets in which the results are better/worse/equal compared with *CN2-standard*, as well as the number of significant wins and losses.

9. This is a simplification (as multi-class AUC could also be computed as the average of AUCs computed by comparing all pairs of classes (Hand and Till, 2001)) that still provides sufficient evidence to support the claims of this paper.

Performance Measure	Data Sets	**CN2 standard**	**CN2 WRAcc**	**CN2-SD ($\gamma = 0.5$)**	**CN2-SD ($\gamma = 0.9$)**	**CN2-SD (add.)**	**Detailed Results**
Coverage (COV)	23	0.131 ± 0.14	0.311 ± 0.17	0.403 ± 0.23	0.450 ± 0.26	**0.486** ± 0.30	Table 14
• significance – p value			0.000	0.000	0.000	**0.000**	
• win/loss/draw			22/1/0	22/1/0	23/0/0	**22/1/0**	
• sig.win/sig.loss			21/1	22/0	22/0	**21/1**	
Support (SUP)	23	0.84 ± 0.03	0.85 ± 0.03	0.90 ± 0.06	**0.92** ± 0.06	0.91 ± 0.06	Table 15
• significance – p value			0.637	0.000	**0.000**	0.001	
• win/loss/draw			13/10/0	18/5/0	**20/3/0**	16/7/0	
• sig.win/sig.loss			5/4	13/1	**18/0**	13/1	
Size (SIZE)	23	18.18 ± 21.77	**6.15** ± 4.49	6.25 ± 4.42	6.49 ± 4.57	6.35 ± 4.58	Table 16
• significance – p value			**0.006**	0.007	0.007	0.007	
• win/loss/draw			**22/1/0**	22/1/0	20/3/0	23/0/0	
• sig.win/sig.loss			**22/0**	21/0	19/2	18/0	
Significance (SIG)	23	2.11 ± 0.46	8.97 ± 4.66	15.57 ± 6.05	16.92 ± 8.90	**18.47** ± 9.00	Table 17
• significance – p value			0.000	0.000	0.000	**0.000**	
• win/loss/draw			22/1/0	23/0/0	22/1/0	**23/0/0**	
• sig.win/sig.loss			21/0	23/0	21/0	**23/0**	
Unusualness (WRACC)	23	0.017 ± 0.02	0.056 ± 0.05	0.079 ± 0.06	0.088 ± 0.06	**0.092** ± 0.07	Table 18
• significance – p value			0.001	0.000	0.000	**0.000**	
• win/loss/draw			20/1/2	22/1/0	22/1/0	**22/1/0**	
• sig.win/sig.loss			19/1	21/1	21/1	**21/1**	

Table 4: Summary of the experimental results on the UCI data sets (descriptive evaluation measures) for different variants of the unordered algorithm using 10-fold stratified cross-validation. The best results are shown in boldface.

The analysis shows that if multiplicative weights are used, most results improve with the increased value of the γ parameter. As in most cases the best *CN2-SD* variants are *CN2-SD* with $\gamma = 0.9$ and with additive weights, and as using additive weighs is the simpler method, the additive weights setting is recommended as default for experimental use.

The summary of results in terms of descriptive measures of interestingness is as follows.

- In terms of the average coverage per rule *CN2-SD* produces rules with significantly higher coverage (the higher the coverage the better the rule) than both *CN2-WRAcc* and *CN2-standard*. The coverage is increased by increasing the γ parameter and the best results are achieved by $\gamma = 0.9$ and by additive weights.

- *CN2-SD* induces rule sets with significantly larger overall support than *CN2-standard* meaning that it covers a higher percentage of target examples (positives) thus leaving a smaller number of examples unclassified.[10]

- *CN2-WRAcc* and *CN2-SD* induce rule sets that are significantly smaller than *CN2-standard* (smaller rule sets are better), while rule sets of *CN2-WRAcc* and *CN2-SD* are comparable, despite the fact that *CN2-SD* uses weights to 'recycle' examples and thus produces overlapping rules.

10. CN2 handles the unclassified examples by classifying them using the default rule – the rule predicting the majority class.

Performance Measure	Data Sets	**CN2 standard**	**CN2 WRAcc**	**CN2-SD** ($\gamma = 0.5$)	**CN2-SD** ($\gamma = 0.9$)	**CN2-SD (add.)**	**Detailed Results**
Accuracy (ACC)	23	**81.61** ± 11.66	78.12 ± 16.28	80.92 ± 16.04	81.07 ± 15.78	79.36 ± 16.24	Table 19
• significance – p value			0.150	0.771	0.818	0.344	
• win/loss/draw			10/12/1	17/6/0	19/4/0	15/8/0	
• sig.win/sig.loss			3/5	9/4	10/4	7/4	
AUC-Method-2 (AUC)	16	82.16 ± 16.81	84.37 ± 9.87	**86.75** ± 8.95	86.39 ± 10.32	86.33 ± 8.60	Table 20
• significance – p value			0.563	**0.175**	0.236	0.236	
• win/loss/draw			6/9/1	**10/6/0**	9/7/0	10/6/0	
• sig.win/sig.loss			5/5	**6/2**	7/4	6/3	

Table 5: Summary of the experimental results on the UCI data sets (predictive evaluation measures) for different variants of the unordered algorithm using 10-fold stratified cross-validation. The best results are shown in boldface.

- *CN2-SD* induces significantly better rules in terms of rule significance (rules with higher significance are better) computed by 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 in 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 terms of rule unusualness which is of ultimate interest to the subgroup discovery task, *CN2-SD* produces rules with significantly higher average weighted relative accuracy than *CN2-standard*. Like in the case of average coverage per rule the unusualness is increased by increasing the γ parameter and the best results are achieved by $\gamma = 0.9$ and by additive weights. Note that the unusualness of a rule, computed by its *WRAcc*, is a combination of rule accuracy, coverage and prior probability of the target class.

In terms of predictive measures of classification performance results can be summarized as follows.

- *CN2-SD* improves the accuracy in comparison with *CN2-WRAcc* and performs comparable to *CN2-standard* (the difference is insignificant). Notice however that while optimizing predictive accuracy is the ultimate goal of CN2, for *CN2-SD* the goal is to optimize the coverage/relative-accuracy tradeoff.

- In the computation of area under ROC curve (AUC-Method-2) due to the restriction of this method to binary class data sets, only 16 binary data sets are used in the comparisons. Notice that *CN2-SD* improves the area under ROC curve compared to *CN2-WRAcc* and compared to *CN2-standard*, but the differences are not significant. The area under ROC curve however seems not to be affected by the parameter γ or by the weighting approach of *CN2-SD*.

AUC performance is also illustrated by means of the results on the Australian UCI data set in Figures 2 and 3 of Section 4.2. The solid lines in these graphs indicate ROC curves obtained by *CN2-standard* while the dotted lines represent ROC curves for *CN2-SD* with additive weights.

6.2.2 RESULTS OF THE ORDERED *CN2-SD*

For completeness, the results for different versions of the ordered algorithm are summarized in Tables 6 and 7, without giving the results for individual data sets in Appendix A. In our view, the unordered *CN2-SD* algorithm is more appropriate for subgroup discovery than the ordered variant, as it induces a set of rules for each target class in turn.

Performance Measure	Data Sets	**CN2 standard**	**CN2 WRAcc**	**CN2-SD ($\gamma = 0.5$)**	**CN2-SD ($\gamma = 0.9$)**	**CN2-SD (add.)**
Coverage (COV)	23	0.174 ± 0.18	0.351 ± 0.18	0.439 ± 0.25	0.420 ± 0.23	**0.527** ± 0.32
• significance – *p* value			0.000	0.000	0.000	**0.000**
• win/loss/draw			21/2/0	23/0/0	23/0/0	**22/1/0**
• sig.win/sig.loss			20/1	22/0	22/0	**22/1**
Support (SUP)	23	0.85 ± 0.03	0.85 ± 0.03	0.87 ± 0.05	**0.91** ± 0.05	0.90 ± 0.06
• significance – *p* value			0.694	0.026	**0.000**	0.000
• win/loss/draw			12/11/0	14/9/0	**18/5/0**	19/4/0
• sig.win/sig.loss			4/4	11/2	**16/1**	14/0
Size (SIZE)	23	17.87 ± 28.10	**4.13** ± 2.73	4.30 ± 2.58	4.61 ± 2.64	4.27 ± 2.79
• significance – *p* value			**0.025**	0.026	0.030	0.025
• win/loss/draw			**21/1/1**	21/2/0	20/3/0	21/2/0
• sig.win/sig.loss			**21/0**	20/1	19/1	20/0
Significance (SIG)	23	1.87 ± 0.47	8.86 ± 4.81	12.70 ± 7.11	14.80 ± 8.31	**18.11** ± 9.84
• significance – *p* value			0.000	0.000	0.000	**0.000**
• win/loss/draw			22/1/0	22/1/0	23/0/0	**22/1/0**
• sig.win/sig.loss			22/0	22/0	22/0	**21/0**
Unusualness (WRACC)	23	0.024 ± 0.02	0.060 ± 0.05	0.080 ± 0.06	0.082 ± 0.06	**0.100** ± 0.07
• significance – *p* value			0.001	0.000	0.000	**0.000**
• win/loss/draw			18/5/0	21/2/0	21/2/0	**22/1/0**
• sig.win/sig.loss			17/2	20/1	20/2	**21/1**

Table 6: Summary of the experimental results on the UCI data sets (descriptive evaluation measures) for different variants of the ordered algorithm using 10-fold stratified cross-validation. The best results are shown in boldface.

6.3 Experiments in Traffic Accident Data Analysis

We have evaluated the *CN2-SD* algorithm also on a traffic accident data set. This is a large real-world database (1.5 GB) containing 21 years of police traffic accident reports (1979–1999). The analysis of this database is not straightforward because of the volume of the data, the amounts of noise and missing data, and the fact that there is no clearly defined data mining target. As described below, some preprocessing was needed before running the subgroup discovery experiments. Results of experiments were shown to the domain expert whose comments are included.

6.3.1 THE TRAFFIC ACCIDENT DATA SET

The traffic accident database contains data about traffic accidents and the vehicles and casualties involved. The data is organized in three linked tables: the ACCIDENT table, the VEHICLE table and the CASUALTY table. The ACCIDENT table consists of the records of all accidents that

Performance Measure	Data Sets	**CN2 standard**	**CN2 WRAcc**	**CN2-SD ($\gamma = 0.5$)**	**CN2-SD ($\gamma = 0.9$)**	**CN2-SD (add.)**
Accuracy (ACC)	23	**83.00** ± 10.30	78.34 ± 16.52	79.50 ± 16.68	81.10 ± 16.53	80.79 ± 16.61
• significance – p value			0.155	0.286	0.556	0.494
• win/loss/draw			8/15/0	14/9/0	15/8/0	15/8/0
• sig.win/sig.loss			3/6	15/4	8/4	7/3
AUC-Method-2 (AUC)	16	81.89 ± 10.07	82.28 ± 10.11	84.37 ± 9.19	**84.70** ± 8.53	83.79 ± 9.64
• significance – p value			0.721	0.026	**0.005**	0.049
• win/loss/draw			9/6/1	10/6/0	**12/4/0**	10/6/0
• sig.win/sig.loss			6/5	6/3	**8/4**	6/4

Table 7: Summary of the experimental results on the UCI data sets (predictive evaluation measures) for different variants of the ordered algorithm using 10-fold stratified cross-validation. The best results are shown in boldface.

happened over the given period of time (1979–1999), the VEHICLE table contains data about the vehicles involved in those accidents, and the CASUALTY table contains data about 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 table, two records in the VEHICLE table and three records in the CASUALTY table. The three tables are described in more detail below.

- The ACCIDENT table contains one record for each accident. The 30 attributes describing an accident can be divided in three groups: date and time of the accident, description of the road where the accident has occurred, and conditions under which the accident has occurred (such as weather conditions, light and junction details). In the ACCIDENT table there are more than 5 million records.
- The VEHICLE table contains one record for each vehicle involved in an accident from the ACCIDENT table. There can be one or many vehicles involved in a single accident. The VEHICLE table attributes describe 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 resulting from the accident, and the object that vehicle hit on and off carriageway. There are 24 attributes in the VEHICLE table which contains almost 9 million 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 million records.

6.3.2 DATA PREPROCESSING

The large volume of data in the traffic data set makes it practically impossible to run any data mining algorithm on the whole set. Therefore we have taken samples of the data set and performed

PFC	Number of Examples	Percentage of Sampled Accidents	Class distribution (%) fatal / serious / slight
1	2555	0.3	1.76 / 24.85 / 73.39
2	2523	1.9	2.53 / 30.87 / 66.60
3	2501	4.8	0.56 / 12.35 / 87.09
4	2499	1.9	2.16 / 27.21 / 70.63
5	2522	9.2	1.90 / 23.39 / 74.71
6	2548	2.0	1.41 / 13.69 / 84.90
7	2788	1.4	0.97 / 16.25 / 82.78

Table 8: Properties of the traffic data set.

the experiments on these samples. We focused on the ACCIDENT table and examined only the accidents that happened in 7 districts (called Police Force Codes, or PFCs) across the UK.[11] The 7 PFCs were chosen by the domain expert and represent typical PFCs from clusters of PFCs with the same accident dynamics, analyzed by Ljubič et al. (2002). In this way we obtained 7 data sets (one for each PFC) with some hundred thousands of examples each. We further sampled this data to obtain approximately 2500 examples per data set. The sample percentages are listed in Table 8 together with the other characteristics of these 7 sampled data sets.

Among the 26 attributes describing each of the 7 data sets we chose the attribute ‘accident severity’ to be the class attribute. The task that we have addressed was therefore to find subgroups of accidents of a certain severity (‘slight’, ‘serious’ or ‘fatal’) and characterize them in terms of attributes describing the accident, such as: ‘road class’, ‘speed limit’, ‘light condition’, etc.

6.3.3 RESULTS OF EXPERIMENTS

We want to investigate if by running *CN2-SD* on the data sets, described in Table 8, we are able to get some rules that are typical and different for distinct PFCs.

We used the same methodology to perform the experiments as in the case of the UCI data sets of Section 6.2. The only difference is that here we do not perform the area under ROC curve analysis, because the data sets are not two-class. The results presented in Tables 9–13 show the same advantages of *CN2-SD* over *CN2-WRAcc* and *CN2-standard* as shown by the results of experiments on the UCI data sets.[12] In particular, *CN2-SD* produces substantially smaller rule sets, where individual rules have higher coverage and significance.

It should be noticed that these data sets have a very unbalanced class distribution (most accidents are ‘slight’ and only few are ‘fatal’, see Table 8). In terms of rule set accuracy, all algorithms achieved roughly default performance which is obtained by always predicting the majority class. Since classification was not the main interest of this experiment, we omit the results.

11. For the sake of anonymity, the code numbers 1 through 7 do not correspond to the PFCs 1 through 7 used for Police Force Codes in the actual traffic accident database.
12. Like in the UCI case, only the results of the unordered versions of the algorithm are presented here, although the experiments were done with both unordered and ordered variants of the algorithms.

#	CN2 standard *COV* ± *sd*	CN2 WRAcc *COV* ± *sd*	CN2-SD ($\gamma = 0.5$) *COV* ± *sd*	CN2-SD ($\gamma = 0.7$) *COV* ± *sd*	CN2-SD ($\gamma = 0.9$) *COV* ± *sd*	CN2-SD (add.) *COV* ± *sd*
1	0.056 ± 0.01	**0.108**↑ ± 0.00	**0.111**↑ ± 0.03	**0.111**↑ ± 0.03	**0.123**↑ ± 0.03	**0.110**↑ ± 0.03
2	0.050 ± 0.10	**0.113**↑ ± 0.04	**0.127**↑ ± 0.05	**0.127**↑ ± 0.04	**0.129**↑ ± 0.05	**0.151**↑ ± 0.04
3	0.140 ± 0.03	0.118 ± 0.03	0.126 ± 0.02	0.119 ± 0.02	0.118 ± 0.01	0.154 ± 0.02
4	0.052 ± 0.01	**0.105**↑ ± 0.03	**0.105**↑ ± 0.04	**0.120**↑ ± 0.04	**0.122**↑ ± 0.04	**0.116**↑ ± 0.04
5	0.075 ± 0.08	**0.108**↑ ± 0.04	**0.115**↑ ± 0.06	**0.121**↑ ± 0.05	**0.110**↑ ± 0.05	**0.127**↑ ± 0.04
6	0.078 ± 0.06	**0.118**↑ ± 0.03	**0.134**↑ ± 0.05	**0.122**↑ ± 0.06	**0.124**↑ ± 0.06	**0.120**↑ ± 0.05
7	0.116 ± 0.08	0.110 ± 0.11	0.118 ± 0.14	0.124 ± 0.13	0.122 ± 0.13	**0.143**↑ ± 0.12
Average	0.081 ± 0.03	0.111 ± 0.01	0.120 ± 0.01	0.121 ± 0.00	0.121 ± 0.01	**0.132** ± 0.02
• significance – *p* value		0.047	0.021	0.023	0.029	**0.003**
• win/loss/draw		5/2/0	6/1/0	6/1/0	6/1/0	**7/0/0**
• sig.win/sig.loss		5/0	5/0	5/0	5/0	**6/0**

Table 9: Experimental results on the traffic accident data sets. Average coverage per rule with standard deviation (*COV* ± *sd*) for different variants of the unordered algorithm.

#	CN2 standard *SUP* ± *sd*	CN2 WRAcc *SUP* ± *sd*	CN2-SD ($\gamma = 0.5$) *SUP* ± *sd*	CN2-SD ($\gamma = 0.7$) *SUP* ± *sd*	CN2-SD ($\gamma = 0.9$) *SUP* ± *sd*	CN2-SD (add.) *SUP* ± *sd*
1	0.86 ± 0.03	0.89 ± 0.02	0.83 ± 0.06	**0.93**↑ ± 0.04	**0.96**↑ ± 0.02	**0.95**↑ ± 0.03
2	0.84 ± 0.02	0.85 ± 0.09	0.85 ± 0.02	**0.92**↑ ± 0.04	**0.93**↑ ± 0.00	0.84 ± 0.08
3	0.81 ± 0.06	0.82 ± 0.04	**0.93**↑ ± 0.02	**0.90**↑ ± 0.05	**0.97**↑ ± 0.01	0.85 ± 0.06
4	0.80 ± 0.04	**0.87**↑ ± 0.05	0.82 ± 0.05	0.83 ± 0.00	**0.91**↑ ± 0.03	0.81 ± 0.10
5	0.87 ± 0.08	0.85 ± 0.03	0.80↓ ± 0.03	0.83 ± 0.06	**0.94**↑ ± 0.02	0.83 ± 0.08
6	0.84 ± 0.06	**0.88**↑ ± 0.07	0.81 ± 0.09	**0.91**↑ ± 0.06	**0.88**↑ ± 0.07	**0.98**↑ ± 0.01
7	0.81 ± 0.08	0.83 ± 0.05	**0.90**↑ ± 0.01	0.81 ± 0.01	**0.95**↑ ± 0.02	**0.99**↑ ± 0.00
Average	0.83 ± 0.03	0.85 ± 0.02	0.85 ± 0.05	0.88 ± 0.05	**0.93** ± 0.03	0.89 ± 0.08
• significance – *p* value		0.056	0.548	0.053	**0.001**	0.092
• win/loss/draw		6/1/0	4/3/0	6/1/0	**7/0/0**	6/1/0
• sig.win/sig.loss		2/0	2/1	4/0	**7/0**	3/0

Table 10: Experimental results on the traffic accident data sets. Overall support of rule sets with standard deviation (*SUP* ± *sd*) for different variants of the unordered algorithm.

6.3.4 EVALUATION BY THE DOMAIN EXPERT

We have further examined the rules induced by the *CN2-SD* algorithm (using additive weights). We focused on rules with high coverage and rules that cover a high percentage of the predicted class as these are the rules that are likely to reflect some regularity in the data.

One of the most interesting results concerned the following. One might expect that the number of people injured would increase with the severity of the accident (up to the total number of occupants in the vehicles). Furthermore, common sense would dictate that the number of vehicles

#	CN2 standard SIZE ± sd	CN2 WRAcc SIZE ± sd	CN2-SD (γ = 0.5) SIZE ± sd	CN2-SD (γ = 0.7) SIZE ± sd	CN2-SD (γ = 0.9) SIZE ± sd	CN2-SD (add.) SIZE ± sd
1	16.7 ± 0.60	**9.3**↑ ± 0.99	**10.0**↑ ± 0.51	**10.6**↑ ± 0.46	**10.6**↑ ± 0.73	9.5 ↑ ± 0.25
2	18.7 ± 1.28	**9.2**↑ ± 0.33	**10.0**↑ ± 0.20	**10.3**↑ ± 0.21	**10.3**↑ ± 0.56	**11.1**↑ ± 0.23
3	7.0 ± 0.30	8.6 ± 0.95	9.2 ± 0.19	10.2↓ ± 0.14	9.5 ± 0.35	9.8 ↓ ± 0.19
4	18.0 ± 1.39	**9.9**↑ ± 0.59	**10.4**↑ ± 0.31	**11.2**↑ ± 0.64	**11.2**↑ ± 0.24	**10.3**↑ ± 0.56
5	12.8 ± 1.44	**9.6**↑ ± 0.19	**10.1**↑ ± 0.51	11.2 ± 0.84	11.6 ± 0.96	9.7 ↑ ± 0.21
6	12.5 ± 0.31	**8.5**↑ ± 0.35	9.3 ↑ ± 0.51	8.7 ↑ ± 0.91	9.4 ↑ ± 0.60	8.5 ↑ ± 0.39
7	8.6 ± 1.41	9.3 ± 0.41	9.9 ± 0.90	10.8↓ ± 0.73	11.1↓ ± 0.13	10.4 ± 0.59
Average	13.47 ± 4.57	**9.20** ± 0.50	9.84 ± 0.44	10.42 ± 0.86	10.53 ± 0.84	9.90 ± 0.80
• significance – p value		**0.040**	0.066	0.123	0.127	0.075
• win/loss/draw		**5/2/0**	5/2/0	5/2/0	5/2/0	5/2/0
• sig.win/sig.loss		**5/0**	5/0	4/2	4/1	5/1

Table 11: Experimental results on the traffic accident data sets. Sizes of rule sets with standard deviation (*SIZE* ± *sd*) for different variants of the unordered algorithm.

#	CN2 standard SIG ± sd	CN2 WRAcc SIG ± sd	CN2-SD (γ = 0.5) SIG ± sd	CN2-SD (γ = 0.7) SIG ± sd	CN2-SD (γ = 0.9) SIG ± sd	CN2-SD (add.) SIG ± sd
1	1.9 ± 0.82	**7.0**↑ ± 0.31	**8.7**↑ ± 0.41	**9.7**↑ ± 0.59	**9.4**↑ ± 0.30	**9.6**↑ ± 0.45
2	1.9 ± 0.34	**6.2**↑ ± 0.25	**9.9**↑ ± 0.26	**9.8**↑ ± 0.20	**9.5**↑ ± 0.81	**9.8**↑ ± 0.36
3	1.3 ± 0.27	**6.6**↑ ± 0.61	**8.4**↑ ± 0.52	**9.2**↑ ± 0.54	**11.5**↑ ± 0.75	**9.3**↑ ± 0.17
4	1.6 ± 0.10	**7.6**↑ ± 0.14	**8.5**↑ ± 0.79	**11.0**↑ ± 0.84	**9.4**↑ ± 0.80	**11.1**↑ ± 0.24
5	1.6 ± 0.75	**6.0**↑ ± 0.23	**10.6**↑ ± 0.70	**9.6**↑ ± 0.76	**12.5**↑ ± 0.43	**9.1**↑ ± 0.74
6	1.5 ± 0.87	**8.5**↑ ± 0.41	**8.3**↑ ± 0.54	**9.8**↑ ± 0.24	**9.9**↑ ± 0.51	**12.5**↑ ± 0.35
7	1.7 ± 0.49	**6.8**↑ ± 0.75	**8.7**↑ ± 0.20	**9.9**↑ ± 0.63	**9.2**↑ ± 0.73	**9.7**↑ ± 0.40
Average	1.64 ± 0.20	6.95 ± 0.86	9.01 ± 0.89	9.85 ± 0.56	**10.20** ± 1.28	10.16 ± 1.21
• significance – p value		0.000	0.000	0.000	**0.000**	0.000
• win/loss/draw		7/0/0	7/0/0	7/0/0	**7/0/0**	7/0/0
• sig.win/sig.loss		7/0	7/0	7/0	**7/0**	7/0

Table 12: Experimental results on the traffic accident data sets. Average significance per rule with standard deviation (*SIG* ± *sd*) for different variants of the unordered algorithm.

involved would also increase with accident severity. Contrary to these expectations we found rules of the following two kinds:

- Rules that cover more than the average proportion of 'fatal' or 'serious' accidents when just one vehicle is involved in the accident. Examples of such rules are:
  `IF nv < 1.500 THEN sev = "1" [15 280 1024]`[13]
  `IF nv < 1.500 THEN sev = "2" [22 252 890]`

13. The 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.

#	CN2 standard *WRACC* ± *sd*	CN2 WRAcc *WRACC* ± *sd*	CN2-SD ($\gamma = 0.5$) *WRACC* ± *sd*	CN2-SD ($\gamma = 0.7$) *WRACC* ± *sd*	CN2-SD ($\gamma = 0.9$) *WRACC* ± *sd*	CN2-SD (add.) *WRACC* ± *sd*
1	0.013 ± 0.02	**0.025**↑ ± 0.05	**0.025**↑ ± 0.10	**0.026**↑ ± 0.02	**0.028**↑ ± 0.03	**0.025**↑ ± 0.09
2	0.009 ± 0.07	**0.018**↑ ± 0.05	**0.021**↑ ± 0.00	**0.021**↑ ± 0.04	**0.021**↑ ± 0.02	**0.025**↑ ± 0.04
3	0.052 ± 0.01	0.043 ± 0.00	0.046 ± 0.07	0.043 ± 0.03	0.043 ± 0.05	0.056 ± 0.02
4	0.010 ± 0.09	**0.021**↑ ± 0.06	**0.021**↑ ± 0.05	**0.024**↑ ± 0.09	**0.024**↑ ± 0.00	**0.023**↑ ± 0.07
5	0.019 ± 0.04	**0.026**↑ ± 0.06	**0.027**↑ ± 0.07	**0.029**↑ ± 0.08	**0.027**↑ ± 0.01	**0.030**↑ ± 0.07
6	0.027 ± 0.03	**0.041**↑ ± 0.06	**0.047**↑ ± 0.05	**0.042**↑ ± 0.05	**0.043**↑ ± 0.07	**0.042**↑ ± 0.07
7	0.038 ± 0.03	0.035 ± 0.01	0.038 ± 0.04	0.040 ± 0.00	0.039 ± 0.08	0.046 ± 0.04
Average	0.024 ± 0.02	0.030 ± 0.01	0.032 ± 0.01	0.032 ± 0.01	0.032 ± 0.01	**0.035** ± 0.01
• significance – *p* value		0.096	0.042	0.041	0.048	**0.000**
• win/loss/draw		5/2/0	5/2/0	6/1/0	6/1/0	**7/0/0**
• sig.win/sig.loss		5/0	5/0	5/0	5/0	**5/0**

Table 13: Experimental results on the traffic accident data sets. Unusualness of rule sets with standard deviation (*WRACC* ± *sd*) for different variants of the unordered algorithm.

- Rules that cover more than the average proportion of 'slight' accidents 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]`

Having shown the induced results to the domain expert, he pointed out the following aspects of data collection for the data in the ACCIDENT table.[14]

- The severity code in the ACCIDENT table relates to the most severe injury among those reported for that accident. Therefore a multiple vehicle accident with 1 fatal and 20 slight injuries would be classified as fatal as one fatality occurred, while each individual casualty injury severity is coded in the CASUALTY table.

- Some (slight) injuries may be unreported at the accident scene: if the policeman compiled/revised 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, these changes are not reflected in the ACCIDENT table.

The findings revealed by the rules were surprising to the domain expert and need further investigation. The analysis shows that examining the ACCIDENT table is not sufficient and that further examination of the VEHICLE and CASUALTY tables is needed in further work.

7. Related Work

Other systems have addressed the task of subgroup discovery, the best known being EXPLORA (Klösgen, 1996) and MIDOS (Wrobel, 1997, 2001). EXPLORA treats the learning task as a single relation problem, i.e., all the data are assumed to be available in one table (relation), whereas

14. We have also shown the *CN2-standard* and *CN2-WRAcc* results to the expert but he did not consider any of the rules to be interesting.

MIDOS extends this task to multi-relational databases. Other approaches deal with multi-relational databases using propositionalisation and aggregate functions can be found in the work of Knobbe et al. (2001, 2002).

Another approach to finding symbolic descriptions of groups of instances is symbolic clustering, which has been popular for many years (Michalski, 1980, Gowda and Diday, 1992). Moreover, learning of concept hierarchies also aims at discovering groups of instances, which can be induced in a supervised or unsupervised manner: decision tree induction algorithms perform supervised symbolic learning of concept hierarchies (Langley, 1996, Raedt and Blockeel, 1997), whereas hierarchical clustering algorithms (Sokal and Sneath, 1963, Gordon, 1982) are unsupervised and do not result in symbolic descriptions. Note that in decision tree learning, the rules which can be formed from paths leading from the root node to class labels in the leaves represent *discriminant descriptions*, formed from properties that best discriminate between the classes. As rules formed from decision tree paths form discriminant descriptions, they are inappropriate for solving subgroup discovery tasks which aim at describing subgroups by their characteristic properties.

Instance weights play an important role in boosting (Freund and Shapire, 1996) and alternating decision trees (Schapire and Singer, 1998). Instance weights have been used also in variants of the covering algorithm implemented in rule learning approaches such as SLIPPER (Cohen and Singer, 1999), RL (Lee et al., 1998) and DAIRY (Hsu et al., 1998). A variant of the weighted covering algorithm has been used in the subgroup discovery algorithm SD for rule subset selection (Gamberger and Lavrač, 2002).

A variety of rule evaluation measures and heuristics have been studied for subgroup discovery (Klösgen, 1996, Wrobel, 1997, 2001), 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' Klösgen, 1996). An alternative measure $q = \frac{TP}{FP+par}$ was proposed in the SD algorithm for expert-guided subgroup discovery (Gamberger and Lavrač, 2002), aimed at minimizing the number of false positives FP, and maximizing true positives TP, balanced by generalization parameter par. Besides such 'objective' measures of interestingness, some 'subjective' measure of interestingness of a discovered pattern can be taken into 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') (Silberschatz and Tuzhilin, 1995).

Note that some approaches to association rule induction can also be used for subgroup discovery. For instance, the APRIORI-C algorithm (Jovanoski and Lavrač, 2001), which applies association rule induction to classification rule induction, outputs classification rules with guaranteed support and confidence with respect to a target class. If a rule satisfies also a user-defined significance threshold, an induced APRIORI-C rule can be viewed as an independent 'chunk' of knowledge about the target class (selected property of interest for subgroup discovery), which can be viewed as a subgroup description with guaranteed significance, support and confidence. This observation led to the development of a novel subgroup discovery algorithm APRIORI-SD (Kavšek et al., 2003).

It should be noticed that in the terminology 'patient vs. greedy' of Friedman and Fisher (1999), *WRAcc* is a 'patient' rule quality measure, favoring more general subgroups than those found by using 'greedy' quality measures. As shown by our experiments in Todorovski et al. (2000), *WRAcc* heuristic improves rule coverage compared to the standard CN2 heuristic. This observation is confirmed also in the experimental evaluation in Section 6 of this paper. Further evidence confirming this claim is provided by Kavšek et al. (2003), providing experimental comparison of results of *CN2-*

SD and our novel subgroup discovery algorithm APRIORI-SD with rule learners CN2, RIPPER and APRIORI-C.

8. Conclusions and Further Work

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 heuristic, the probabilistic classification and the area under the ROC curve (AUC) performance measure. We have also proposed a set of metrics appropriate for evaluating the quality of induced subgroup descriptions.

The experimental results on 23 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 and shown a small (insignificant) increase in terms of the area under ROC curve.

We have applied *CN2-SD* also to a real-life problem of traffic accident analysis. The experimental results confirm the findings in the UCI data sets. The most interesting findings are due to interpretation by the domain expert. What was confirmed in this case study was that the result of a data mining process depends not only on the appropriateness of the selected method and the data that is at hand but also on how the data has been collected. In the traffic accident experiments examining the ACCIDENT table was not sufficient, and further examination of the VEHICLE and CASUALTY tables is needed. This will be performed using the RSD relational subgroup discovery algorithm (Lavrač et al., 2003), a recent upgrade of the *CN2-SD* algorithm which enables relational subgroup discovery.

In further work we plan to compare the results with the MIDOS subgroup discovery algorithm. We plan to investigate the behavior of *CN2-SD* in terms of AUC in multi-class problems (Hand and Till, 2001). An interesting question, to be verified in 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.

An important aspect of subgroup discovery performance, which is neglected in our study, is the degree of overlap of the induced subgroups. The challenge of our further research is to propose extensions of the weighted relative accuracy heuristic and ROC space evaluation metrics that will take into account the overlap of subgroups.

We are now moving the focus of our research in subgroup discovery from heuristic search toward exhaustive search of the space of patterns. An attempt of this kind is described by Kavšek et al. (2003) where the well known APRIORI association rule learner was adapted to the task of subgroup discovery.

Acknowledgments

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. Thanks to Peter Ljubič and Damjan Demšar for the help in upgrading the C code of the original CN2 algorithm. We are grateful to

John Bullas for the evaluation of the results of traffic accident data analysis. Thanks are also due to the anonymous reviewers for their insightful comments. 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.

Appendix A. Tables with Detailed Results for Different Variants of the Unordered Algorithm in UCI Data Sets

The tables in this appendix show detailed results of the performance of different variants of the unordered algorithm. The comparisons are made on 23 UCI data sets listed in Table 3. The results shown in Tables 14–18 of Appendix A are summarized in the paper in Table 4, and the results of Tables 19–20 in Table 5.

#	**CN2 standard** *COV ± sd*	**CN2 WRAcc** *COV ± sd*	**CN2-SD** ($\gamma = 0.5$) *COV ± sd*	**CN2-SD** ($\gamma = 0.7$) *COV ± sd*	**CN2-SD** ($\gamma = 0.9$) *COV ± sd*	**CN2-SD (add.)** *COV ± sd*
1	0.071 ± 0.01	**0.416**↑ ± 0.00	**0.473**↑ ± 0.03	**0.492**↑ ± 0.03	**0.480**↑ ± 0.03	**0.424**↑ ± 0.03
2	0.079 ± 0.10	**0.150**↑ ± 0.04	**0.208**↑ ± 0.05	**0.174**↑ ± 0.04	**0.218**↑ ± 0.05	**0.260**↑ ± 0.04
3	0.625 ± 0.03	0.322↓ ± 0.03	0.612 ± 0.02	0.617 ± 0.02	0.721 ± 0.01	0.330↓ ± 0.02
4	0.048 ± 0.01	**0.496**↑ ± 0.03	**0.504**↑ ± 0.04	**0.513**↑ ± 0.04	**0.504**↑ ± 0.04	**0.507**↑ ± 0.04
5	0.057 ± 0.08	**0.275**↑ ± 0.04	**0.296**↑ ± 0.06	**0.344**↑ ± 0.05	**0.299**↑ ± 0.05	**0.381**↑ ± 0.04
6	0.312 ± 0.06	**0.576**↑ ± 0.03	**0.936**↑ ± 0.05	**1.039**↑ ± 0.06	**1.006**↑ ± 0.06	**1.295**↑ ± 0.05
7	0.053 ± 0.08	**0.092**↑ ± 0.11	**0.141**↑ ± 0.14	**0.153**↑ ± 0.13	**0.138**↑ ± 0.13	**0.151**↑ ± 0.12
8	0.107 ± 0.09	**0.240**↑ ± 0.07	**0.419**↑ ± 0.09	**0.376**↑ ± 0.12	**0.366**↑ ± 0.11	**0.435**↑ ± 0.09
9	0.207 ± 0.04	**0.430**↑ ± 0.06	**0.637**↑ ± 0.04	**0.829**↑ ± 0.04	**0.826**↑ ± 0.04	**0.686**↑ ± 0.03
10	0.093 ± 0.00	**0.495**↑ ± 0.00	**0.509**↑ ± 0.00	**0.509**↑ ± 0.00	**0.516**↑ ± 0.00	**0.513**↑ ± 0.00
11	0.099 ± 0.05	**0.168**↑ ± 0.08	**0.229**↑ ± 0.05	**0.234**↑ ± 0.04	**0.246**↑ ± 0.04	**0.354**↑ ± 0.06
12	0.378 ± 0.01	0.386 ± 0.01	**0.619**↑ ± 0.00	0.444 ± 0.00	**0.768**↑ ± 0.00	**0.668**↑ ± 0.01
13	0.160 ± 0.11	**0.408**↑ ± 0.09	**0.639**↑ ± 0.15	**0.467**↑ ± 0.16	**0.424**↑ ± 0.18	**0.621**↑ ± 0.17
14	0.142 ± 0.01	**0.356**↑ ± 0.07	**0.461**↑ ± 0.02	**0.668**↑ ± 0.03	**0.569**↑ ± 0.03	**0.720**↑ ± 0.03
15	0.030 ± 0.01	**0.113**↑ ± 0.07	**0.129**↑ ± 0.02	**0.146**↑ ± 0.03	**0.182**↑ ± 0.03	**0.117**↑ ± 0.03
16	0.129 ± 0.01	**0.650**↑ ± 0.07	**0.703**↑ ± 0.02	**0.711**↑ ± 0.03	**0.674**↑ ± 0.03	**0.831**↑ ± 0.03
17	0.021 ± 0.00	**0.216**↑ ± 0.00	**0.225**↑ ± 0.00	**0.270**↑ ± 0.00	**0.307**↑ ± 0.00	**0.324**↑ ± 0.00
18	0.022 ± 0.05	**0.146**↑ ± 0.08	**0.155**↑ ± 0.05	**0.157**↑ ± 0.04	**0.166**↑ ± 0.04	**0.200**↑ ± 0.06
19	0.066 ± 0.01	**0.331**↑ ± 0.01	**0.357**↑ ± 0.00	**0.628**↑ ± 0.00	**0.616**↑ ± 0.00	**0.759**↑ ± 0.01
20	0.039 ± 0.11	**0.139**↑ ± 0.09	**0.151**↑ ± 0.15	**0.159**↑ ± 0.16	**0.149**↑ ± 0.18	**0.169**↑ ± 0.17
21	0.040 ± 0.01	**0.076**↑ ± 0.07	**0.115**↑ ± 0.02	**0.177**↑ ± 0.03	**0.172**↑ ± 0.03	**0.216**↑ ± 0.03
22	0.004 ± 0.01	**0.185**↑ ± 0.07	**0.194**↑ ± 0.02	**0.185**↑ ± 0.03	**0.188**↑ ± 0.03	**0.191**↑ ± 0.03
23	0.231 ± 0.01	**0.477**↑ ± 0.07	**0.552**↑ ± 0.02	**0.715**↑ ± 0.03	**0.818**↑ ± 0.03	**1.022**↑ ± 0.03
Average	0.131 ± 0.14	0.311 ± 0.17	0.403 ± 0.23	0.435 ± 0.25	0.450 ± 0.26	**0.486** ± 0.30
• significance – p value		0.000	0.000	0.000	0.000	**0.000**
• win/loss/draw		22/1/0	22/1/0	22/1/0	23/0/0	**22/1/0**
• sig.win/sig.loss		21/1	22/0	21/0	22/0	**22/1**

Table 14: Relative average coverage per rule with standard deviation (*COV* ± *sd*) for different variants of the unordered algorithm using 10-fold stratified cross-validation.

#	CN2 standard $SUP \pm sd$	CN2 WRAcc $SUP \pm sd$	CN2-SD ($\gamma = 0.5$) $SUP \pm sd$	CN2-SD ($\gamma = 0.7$) $SUP \pm sd$	CN2-SD ($\gamma = 0.9$) $SUP \pm sd$	CN2-SD (add.) $SUP \pm sd$
1	0.81 ± 0.09	**0.89**↑ ± 0.02	**0.87**↑ ± 0.00	**0.97**↑ ± 0.01	**0.84**↑ ± 0.00	**0.89**↑ ± 0.04
2	0.88 ± 0.01	0.90 ± 0.02	0.89 ± 0.09	0.84 ± 0.04	**0.93**↑ ± 0.02	0.86 ± 0.05
3	0.87 ± 0.05	0.87 ± 0.09	0.84 ± 0.05	**0.93**↑ ± 0.02	0.84 ± 0.07	**0.95**↑ ± 0.01
4	0.87 ± 0.06	0.81↓ ± 0.09	0.90 ± 0.02	0.81↓ ± 0.04	**0.97**↑ ± 0.00	**0.93**↑ ± 0.02
5	0.80 ± 0.01	0.82 ± 0.03	**0.92**↑ ± 0.06	0.85 ± 0.01	**0.95**↑ ± 0.01	**0.87**↑ ± 0.05
6	0.90 ± 0.03	0.81↓ ± 0.01	**0.95**↑ ± 0.01	0.85 ± 0.03	**0.98**↑ ± 0.00	0.82↓ ± 0.02
7	0.89 ± 0.03	0.88 ± 0.03	0.90 ± 0.02	0.81↓ ± 0.07	**0.97**↑ ± 0.01	**0.96**↑ ± 0.01
8	0.84 ± 0.03	0.87 ± 0.04	**0.94**↑ ± 0.01	0.83 ± 0.03	0.89 ± 0.09	**0.98**↑ ± 0.00
9	0.87 ± 0.10	0.81↓ ± 0.02	0.85 ± 0.10	**0.94**↑ ± 0.00	0.90 ± 0.02	**0.99**↑ ± 0.00
10	0.84 ± 0.01	0.83 ± 0.08	0.82 ± 0.07	**1.00**↑ ± 0.00	**0.90**↑ ± 0.02	**0.95**↑ ± 0.02
11	0.83 ± 0.03	0.85 ± 0.07	**0.96**↑ ± 0.01	**0.95**↑ ± 0.01	**0.89**↑ ± 0.09	**0.98**↑ ± 0.01
12	0.82 ± 0.04	**0.89**↑ ± 0.00	0.83 ± 0.10	**0.91**↑ ± 0.01	**0.88**↑ ± 0.03	**0.95**↑ ± 0.01
13	0.87 ± 0.10	0.90 ± 0.06	0.81↓ ± 0.02	0.80↓ ± 0.09	0.85 ± 0.04	0.85 ± 0.03
14	0.84 ± 0.05	0.85 ± 0.07	0.83 ± 0.06	**0.89**↑ ± 0.06	**0.93**↑ ± 0.02	0.86 ± 0.05
15	0.83 ± 0.04	0.80 ± 0.07	**0.96**↑ ± 0.01	0.86 ± 0.09	0.80 ± 0.08	0.81 ± 0.00
16	0.85 ± 0.07	0.82 ± 0.02	**1.00**↑ ± 0.00	0.84 ± 0.06	**0.96**↑ ± 0.01	0.85 ± 0.10
17	0.86 ± 0.08	**0.90**↑ ± 0.03	0.86 ± 0.07	0.82 ± 0.06	**1.00**↑ ± 0.00	0.85 ± 0.06
18	0.81 ± 0.06	**0.85**↑ ± 0.07	**0.96**↑ ± 0.01	**0.89**↑ ± 0.05	**0.95**↑ ± 0.01	**0.97**↑ ± 0.00
19	0.83 ± 0.01	0.85 ± 0.05	**0.92**↑ ± 0.04	**0.95**↑ ± 0.01	**0.90**↑ ± 0.02	0.84 ± 0.05
20	0.90 ± 0.06	0.82↓ ± 0.07	**0.99**↑ ± 0.00	0.90 ± 0.03	**0.99**↑ ± 0.00	0.90 ± 0.04
21	0.81 ± 0.05	0.80 ± 0.04	**0.87**↑ ± 0.08	**0.90**↑ ± 0.04	**0.93**↑ ± 0.02	0.82 ± 0.06
22	0.81 ± 0.02	**0.89**↑ ± 0.06	**0.94**↑ ± 0.02	**0.96**↑ ± 0.01	**1.00**↑ ± 0.00	**0.96**↑ ± 0.01
23	0.82 ± 0.05	0.82 ± 0.04	**0.94**↑ ± 0.03	**0.87**↑ ± 0.07	**0.99**↑ ± 0.00	**0.99**↑ ± 0.00
Average	0.84 ± 0.03	0.85 ± 0.03	0.90 ± 0.06	0.89 ± 0.06	**0.92** ± 0.06	0.91 ± 0.06
• significance – *p* value		0.637	0.000	0.017	**0.000**	0.001
• win/loss/draw		13/10/0	18/5/0	14/9/0	**20/3/0**	16/7/0
• sig.win/sig.loss		5/4	13/1	11/3	**18/0**	13/1

Table 15: Overall rule set support with standard deviation ($SUP \pm sd$) for different variants of the unordered algorithm using 10-fold stratified cross-validation.

#	CN2 standard $SIZE \pm sd$	CN2 WRAcc $SIZE \pm sd$	CN2-SD ($\gamma = 0.5$) $SIZE \pm sd$	CN2-SD ($\gamma = 0.7$) $SIZE \pm sd$	CN2-SD ($\gamma = 0.9$) $SIZE \pm sd$	CN2-SD (add.) $SIZE \pm sd$
1	12.4 ± 1.95	**2.0**↑ ± 0.75	**2.7**↑ ± 0.02	**2.6**↑ ± 0.87	**2.2**↑ ± 0.85	**3.5**↑ ± 0.79
2	12.6 ± 1.04	**8.8**↑ ± 0.95	**7.9**↑ ± 0.50	**8.5**↑ ± 1.75	**9.0**↑ ± 0.24	**9.2**↑ ± 1.24
3	1.8 ± 0.10	2.0 ± 0.41	2.0 ± 0.70	2.7 ↓ ± 0.44	1.9 ± 0.27	1.8 ± 0.29
4	14.6 ± 1.81	**7.9**↑ ± 1.78	**8.1**↑ ± 1.02	**7.9**↑ ± 0.97	**8.5**↑ ± 0.47	**8.5**↑ ± 0.41
5	12.8 ± 1.56	**5.2**↑ ± 0.79	**6.0**↑ ± 0.68	**5.6**↑ ± 1.35	**5.4**↑ ± 0.30	**4.6**↑ ± 0.86
6	3.7 ± 1.37	**2.5**↑ ± 0.79	3.1 ± 0.72	3.8 ± 1.61	4.7 ↓ ± 1.22	3.4 ± 0.02
7	15.1 ± 1.89	**7.8**↑ ± 1.49	**8.4**↑ ± 1.32	**8.7**↑ ± 0.46	**9.1**↑ ± 1.26	**8.8**↑ ± 1.13
8	6.4 ± 1.53	**3.0**↑ ± 1.20	**2.9**↑ ± 0.98	**2.7**↑ ± 0.67	**2.7**↑ ± 0.90	**1.8**↑ ± 0.38
9	3.0 ± 0.29	**2.1**↑ ± 0.50	**1.7**↑ ± 0.93	2.7 ± 0.53	3.6 ↓ ± 1.83	2.7 ± 0.00
10	10.1 ± 1.02	**3.9**↑ ± 0.31	**3.9**↑ ± 0.85	**3.4**↑ ± 1.10	**3.3**↑ ± 1.90	**2.5**↑ ± 0.54
11	7.6 ± 1.01	**3.0**↑ ± 1.78	**3.9**↑ ± 1.84	**4.0**↑ ± 0.18	**3.6**↑ ± 0.87	**4.2**↑ ± 0.41
12	3.8 ± 1.24	**3.0**↑ ± 1.24	**3.2**↑ ± 0.42	**3.4**↑ ± 0.39	**2.9**↑ ± 0.05	3.6 ± 0.69
13	4.7 ± 1.30	**3.1**↑ ± 1.15	**3.4**↑ ± 0.54	**3.9**↑ ± 0.98	4.6 ± 1.19	4.5 ± 0.71
14	5.2 ± 0.90	**2.7**↑ ± 0.91	**2.1**↑ ± 0.95	**1.9**↑ ± 0.10	**1.7**↑ ± 1.73	**2.1**↑ ± 0.78
15	21.2 ± 3.48	**10.5**↑ ± 1.85	**11.2**↑ ± 1.12	**10.3**↑ ± 1.99	**9.6**↑ ± 1.32	**10.2**↑ ± 1.30
16	7.1 ± 1.59	**2.0**↑ ± 0.81	**2.4**↑ ± 0.56	**2.4**↑ ± 0.75	**2.9**↑ ± 0.56	**1.8**↑ ± 0.45
17	28.7 ± 3.89	**9.9**↑ ± 1.22	**9.4**↑ ± 1.61	**8.9**↑ ± 1.80	**9.5**↑ ± 1.03	**8.3**↑ ± 1.17
18	83.8 ± 5.37	**10.9**↑ ± 2.37	**11.3**↑ ± 2.78	**11.8**↑ ± 1.45	**11.7**↑ ± 1.67	**12.8**↑ ± 1.74
19	12.9 ± 1.68	**7.7**↑ ± 1.00	**8.6**↑ ± 1.21	**9.1**↑ ± 1.85	**8.4**↑ ± 1.09	**10.1**↑ ± 1.83
20	32.8 ± 2.64	**8.7**↑ ± 1.82	**8.9**↑ ± 1.48	**9.8**↑ ± 1.01	**10.5**↑ ± 1.37	**9.2**↑ ± 1.49
21	35.1 ± 3.54	**19.6**↑ ± 1.80	**19.3**↑ ± 2.91	**19.7**↑ ± 2.99	**19.8**↑ ± 2.58	**19.2**↑ ± 2.90
22	77.3 ± 4.07	**12.2**↑ ± 1.79	**11.4**↑ ± 2.87	**12.4**↑ ± 2.29	**12.4**↑ ± 2.09	**11.7**↑ ± 2.81
23	5.5 ± 1.26	**3.0**↑ ± 0.36	**2.1**↑ ± 0.70	**2.1**↑ ± 0.57	**1.2**↑ ± 0.73	**1.4**↑ ± 0.90
Average	18.18 ± 21.77	**6.15** ± 4.49	6.25 ± 4.42	6.45 ± 4.48	6.49 ± 4.57	6.35 ± 4.58
• significance – p value		**0.006**	0.007	0.007	0.007	0.007
• win/loss/draw		**22/1/0**	22/1/0	21/2/0	20/3/0	23/0/0
• sig.win/sig.loss		**22/0**	21/0	20/1	19/2	18/0

Table 16: Average rule set sizes with standard deviation (*SIZE* ± *sd*) for different variants of the unordered algorithm using 10-fold stratified cross-validation.

#	**CN2 standard** *SIG* ± *sd*	**CN2 WRAcc** *SIG* ± *sd*	**CN2-SD** ($\gamma = 0.5$) *SIG* ± *sd*	**CN2-SD** ($\gamma = 0.7$) *SIG* ± *sd*	**CN2-SD** ($\gamma = 0.9$) *SIG* ± *sd*	**CN2-SD (add.)** *SIG* ± *sd*
1	2.0 ± 0.05	**7.8**↑ ± 1.49	**14.6**↑ ± 1.05	**24.0**↑ ± 1.01	**15.6**↑ ± 1.54	**4.6**↑ ± 0.52
2	2.7 ± 0.10	**13.3**↑ ± 1.69	**27.1**↑ ± 3.37	2.1 ± 0.02	**20.5**↑ ± 2.45	**26.6**↑ ± 3.43
3	2.1 ± 0.01	**7.8**↑ ± 0.64	**13.3**↑ ± 1.39	2.5 ± 0.01	**21.2**↑ ± 2.55	**22.9**↑ ± 2.43
4	2.4 ± 0.06	**9.1**↑ ± 0.58	**14.1**↑ ± 1.72	**16.9**↑ ± 1.28	**22.5**↑ ± 2.49	**30.2**↑ ± 3.98
5	2.0 ± 0.01	**15.8**↑ ± 1.07	**14.9**↑ ± 1.95	**11.0**↑ ± 1.43	**15.2**↑ ± 1.85	2.1 ± 0.01
6	1.9 ± 0.03	**10.0**↑ ± 1.63	**11.0**↑ ± 1.12	**30.5**↑ ± 2.12	**30.1**↑ ± 2.27	**23.1**↑ ± 2.97
7	2.0 ± 0.02	2.7 ± 0.83	**19.8**↑ ± 1.21	**17.7**↑ ± 1.63	**11.1**↑ ± 1.03	**16.3**↑ ± 1.49
8	1.9 ± 0.09	**4.6**↑ ± 0.59	**23.2**↑ ± 1.82	**5.3**↑ ± 0.36	**4.0**↑ ± 0.03	**30.6**↑ ± 2.96
9	2.7 ± 0.03	**9.7**↑ ± 0.86	**12.3**↑ ± 1.00	**9.3**↑ ± 0.65	**8.5**↑ ± 0.89	**25.0**↑ ± 2.60
10	1.4 ± 0.04	**3.6**↑ ± 0.74	**5.8**↑ ± 0.48	**28.3**↑ ± 2.27	**24.9**↑ ± 2.27	**13.5**↑ ± 1.84
11	2.0 ± 0.04	1.8 ± 0.07	**16.7**↑ ± 1.42	**23.9**↑ ± 2.41	**30.9**↑ ± 2.18	**14.9**↑ ± 1.52
12	1.9 ± 0.03	**7.1**↑ ± 0.07	**17.0**↑ ± 1.61	1.3 ± 0.09	**17.6**↑ ± 1.45	**4.0**↑ ± 0.00
13	2.1 ± 0.00	**15.1**↑ ± 1.80	**19.4**↑ ± 1.77	**21.9**↑ ± 2.38	**21.4**↑ ± 2.39	**9.7**↑ ± 0.61
14	2.5 ± 0.08	**14.9**↑ ± 1.93	**18.0**↑ ± 1.57	**13.9**↑ ± 1.28	3.0 ± 0.09	**18.1**↑ ± 1.73
15	2.5 ± 0.05	**4.2**↑ ± 0.42	**17.5**↑ ± 1.79	**5.7**↑ ± 0.46	**21.9**↑ ± 2.83	**26.5**↑ ± 2.22
16	2.6 ± 0.04	**11.7**↑ ± 1.90	**9.6**↑ ± 0.56	**22.7**↑ ± 2.59	2.3 ± 0.08	**6.0**↑ ± 0.00
17	2.7 ± 0.03	**4.8**↑ ± 0.53	**11.7**↑ ± 1.67	**21.8**↑ ± 2.55	**15.0**↑ ± 1.82	**24.3**↑ ± 2.26
18	1.5 ± 0.00	**14.1**↑ ± 1.11	**6.0**↑ ± 0.93	**26.8**↑ ± 2.53	**12.6**↑ ± 1.35	**19.3**↑ ± 1.09
19	1.0 ± 0.07	**2.4**↑ ± 0.01	**22.0**↑ ± 1.20	**17.0**↑ ± 1.78	**16.4**↑ ± 1.74	**9.1**↑ ± 0.02
20	1.5 ± 0.00	**16.0**↑ ± 2.52	**24.3**↑ ± 1.52	**11.4**↑ ± 1.25	**29.9**↑ ± 3.25	**21.7**↑ ± 2.88
21	2.4 ± 0.02	**6.8**↑ ± 0.88	**15.6**↑ ± 1.98	**12.9**↑ ± 1.47	**8.2**↑ ± 0.06	**30.6**↑ ± 2.39
22	2.6 ± 0.04	**9.7**↑ ± 1.56	**3.4**↑ ± 0.09	**14.2**↑ ± 1.20	**7.1**↑ ± 0.47	**20.2**↑ ± 2.71
23	2.0 ± 0.07	**13.5**↑ ± 1.57	**20.7**↑ ± 1.93	**2.7**↑ ± 0.02	**29.4**↑ ± 3.51	**25.7**↑ ± 2.48
Average	2.11 ± 0.46	8.97 ± 4.66	15.57 ± 6.05	14.95 ± 9.02	16.92 ± 8.90	**18.47** ± 9.00
• significance – *p* value		0.000	0.000	0.000	0.000	**0.000**
• win/loss/draw		22/1/0	23/0/0	21/2/0	22/1/0	**23/0/0**
• sig.win/sig.loss		21/0	23/0	20/0	21/0	**22/0**

Table 17: Average rule significance with standard deviation (*SIG* ± *sd*) for different variants of the unordered algorithm using 10-fold stratified cross-validation.

#	CN2 standard $WRACC \pm sd$	CN2 WRAcc $WRACC \pm sd$	CN2-SD ($\gamma = 0.5$) $WRACC \pm sd$	CN2-SD ($\gamma = 0.7$) $WRACC \pm sd$	CN2-SD ($\gamma = 0.9$) $WRACC \pm sd$	CN2-SD (add.) $WRACC \pm sd$
1	0.022 ± 0.09	**0.148**↑ ± 0.03	**0.186**↑ ± 0.09	**0.185**↑ ± 0.04	**0.181**↑ ± 0.07	**0.162**↑ ± 0.01
2	0.034 ± 0.04	**0.063**↑ ± 0.04	**0.095**↑ ± 0.02	**0.079**↑ ± 0.01	**0.093**↑ ± 0.07	**0.111**↑ ± 0.04
3	-0.016 ± 0.08	-0.012 ± 0.01	-0.005 ± 0.03	-0.006 ± 0.09	-0.001 ± 0.02	-0.012 ± 0.01
4	0.020 ± 0.04	**0.210**↑ ± 0.02	**0.228**↑ ± 0.02	**0.233**↑ ± 0.04	**0.224**↑ ± 0.03	**0.224**↑ ± 0.10
5	0.013 ± 0.06	**0.065**↑ ± 0.06	**0.085**↑ ± 0.07	**0.099**↑ ± 0.04	**0.086**↑ ± 0.07	**0.092**↑ ± 0.03
6	0.058 ± 0.07	**0.099**↑ ± 0.10	**0.174**↑ ± 0.05	**0.208**↑ ± 0.00	**0.213**↑ ± 0.01	**0.243**↑ ± 0.10
7	0.012 ± 0.02	**0.020**↑ ± 0.01	**0.034**↑ ± 0.00	**0.040**↑ ± 0.05	**0.034**↑ ± 0.08	**0.034**↑ ± 0.08
8	0.026 ± 0.04	**0.065**↑ ± 0.04	**0.124**↑ ± 0.02	**0.104**↑ ± 0.06	**0.104**↑ ± 0.09	**0.122**↑ ± 0.03
9	0.004 ± 0.07	**0.018**↑ ± 0.04	**0.057**↑ ± 0.10	**0.073**↑ ± 0.09	**0.066**↑ ± 0.04	**0.049**↑ ± 0.02
10	0.013 ± 0.04	**0.067**↑ ± 0.02	**0.076**↑ ± 0.01	**0.073**↑ ± 0.09	**0.076**↑ ± 0.04	**0.072**↑ ± 0.07
11	0.041 ± 0.02	**0.065**↑ ± 0.03	**0.099**↑ ± 0.04	**0.095**↑ ± 0.05	**0.104**↑ ± 0.10	**0.145**↑ ± 0.00
12	0.024 ± 0.04	0.024 ± 0.05	**0.062**↑ ± 0.02	**0.042**↑ ± 0.02	**0.052**↑ ± 0.03	**0.045**↑ ± 0.06
13	0.024 ± 0.03	**0.056**↑ ± 0.03	**0.114**↑ ± 0.10	**0.085**↑ ± 0.04	**0.065**↑ ± 0.07	**0.092**↑ ± 0.03
14	0.009 ± 0.10	**0.038**↑ ± 0.10	**0.053**↑ ± 0.03	**0.082**↑ ± 0.10	**0.082**↑ ± 0.02	**0.085**↑ ± 0.08
15	0.015 ± 0.07	**0.030**↑ ± 0.07	**0.036**↑ ± 0.09	**0.041**↑ ± 0.03	**0.055**↑ ± 0.08	**0.032**↑ ± 0.06
16	0.017 ± 0.00	**0.095**↑ ± 0.10	**0.117**↑ ± 0.04	**0.129**↑ ± 0.04	**0.127**↑ ± 0.06	**0.138**↑ ± 0.02
17	0.005 ± 0.03	**0.048**↑ ± 0.07	**0.051**↑ ± 0.02	**0.073**↑ ± 0.08	**0.083**↑ ± 0.02	**0.073**↑ ± 0.09
18	0.009 ± 0.06	**0.030**↑ ± 0.00	**0.037**↑ ± 0.01	**0.032**↑ ± 0.00	**0.034**↑ ± 0.07	**0.045**↑ ± 0.03
19	0.007 ± 0.07	**0.060**↑ ± 0.00	**0.081**↑ ± 0.08	**0.133**↑ ± 0.05	**0.132**↑ ± 0.03	**0.147**↑ ± 0.04
20	0.004 ± 0.01	-0.045↓ ± 0.10	-0.042↓ ± 0.04	-0.048↓ ± 0.02	-0.042↓ ± 0.03	-0.051↓ ± 0.06
21	0.015 ± 0.08	0.015 ± 0.03	**0.024**↑ ± 0.04	**0.039**↑ ± 0.08	**0.042**↑ ± 0.06	**0.045**↑ ± 0.05
22	0.001 ± 0.03	**0.045**↑ ± 0.06	**0.054**↑ ± 0.05	**0.054**↑ ± 0.09	**0.054**↑ ± 0.05	**0.049**↑ ± 0.05
23	0.033 ± 0.01	**0.076**↑ ± 0.05	**0.089**↑ ± 0.03	**0.144**↑ ± 0.05	**0.149**↑ ± 0.06	**0.167**↑ ± 0.01
Average	0.017 ± 0.02	0.056 ± 0.05	0.079 ± 0.06	0.086 ± 0.07	0.088 ± 0.06	**0.092** ± 0.07
• significance – p value		0.001	0.000	0.000	0.000	**0.000**
• win/loss/draw		20/1/2	22/1/0	22/1/0	22/1/0	**22/1/0**
• sig.win/sig.loss		19/1	21/1	21/1	21/1	**21/1**

Table 18: Average rule unusualness with standard deviation (*WRACC* ± *sd*) for different variants of the unordered algorithm using 10-fold stratified cross-validation.

#	**CN2 standard**	**CN2 WRAcc**	**CN2-SD** (γ = 0.5)	**CN2-SD** (γ = 0.7)	**CN2-SD** (γ = 0.9)	**CN2-SD (add.)**
	ACC ± *sd*	*ACC* ± *sd*	*ACC* ± *sd*	*ACC* ± *sd*	*ACC* ± *sd*	*ACC* ± *sd*
1	81.62 ± 3.55	**85.53**↑ ± 0.14	**89.27**↑ ± 8.04	**87.61**↑ ± 8.71	**87.81**↑ ± 6.54	**88.35**↑ ± 8.60
2	92.28 ± 1.07	92.13 ± 5.95	95.80 ± 1.21	95.56 ± 3.15	92.53 ± 1.52	92.60 ± 2.28
3	82.45 ± 3.89	81.36 ± 1.30	84.13 ± 8.88	84.07 ± 6.11	84.81 ± 1.06	81.46 ± 2.24
4	94.18 ± 3.71	94.34 ± 2.25	97.19 ± 0.35	97.37 ± 0.42	96.54 ± 1.77	96.08 ± 1.22
5	72.77 ± 9.33	73.81 ± 0.91	**78.66**↑ ± 8.65	**78.80**↑ ± 0.04	**78.81**↑ ± 5.55	74.12 ± 9.97
6	68.71 ± 1.79	67.12 ± 6.55	68.62 ± 0.96	70.08 ± 6.28	**71.20**↑ ± 9.94	68.75 ± 5.32
7	72.40 ± 7.60	71.40 ± 7.57	73.73 ± 0.72	**75.82**↑ ± 8.07	74.67 ± 5.85	72.40 ± 7.36
8	74.10 ± 4.15	77.06 ± 7.06	**79.64**↑ ± 6.98	77.53 ± 4.77	**78.48**↑ ± 3.16	**78.03**↑ ± 2.70
9	80.74 ± 7.59	83.26 ± 0.83	**87.87**↑ ± 2.29	**87.75**↑ ± 0.36	**86.97**↑ ± 6.88	**86.14**↑ ± 1.99
10	98.58 ± 0.60	98.54 ± 0.11	99.86 ± 0.03	99.37 ± 0.06	99.77 ± 0.02	99.10 ± 0.40
11	91.44 ± 6.62	88.87↓ ± 7.26	93.25 ± 2.89	90.53 ± 1.44	92.41 ± 4.96	91.10 ± 3.76
12	91.33 ± 2.04	91.33 ± 7.02	**95.08**↑ ± 2.08	94.40 ± 0.94	91.77 ± 6.33	91.75 ± 2.28
13	80.87 ± 1.32	79.74 ± 1.74	83.81 ± 6.59	**84.23**↑ ± 7.59	81.41 ± 0.76	80.86 ± 7.26
14	72.28 ± 2.81	76.60 ± 3.10	**77.59**↑ ± 2.84	**78.35**↑ ± 5.11	**80.40**↑ ± 3.31	**77.74**↑ ± 1.69
15	98.01 ± 0.60	76.40↓ ± 3.75	77.59↓ ± 1.81	77.94↓ ± 0.63	80.26↓ ± 7.99	77.38↓ ± 4.97
16	94.24 ± 0.39	95.63 ± 1.83	**97.67**↑ ± 1.62	**99.09**↑ ± 0.14	**99.85**↑ ± 0.04	**97.62**↑ ± 1.05
17	74.71 ± 8.62	72.49 ± 0.48	72.55 ± 9.85	**77.08**↑ ± 8.89	76.90 ± 0.86	72.51 ± 5.30
18	89.82 ± 5.33	70.33↓ ± 7.94	74.21↓ ± 5.66	70.37↓ ± 7.81	70.56↓ ± 7.49	72.48↓ ± 1.62
19	60.60 ± 1.83	**68.13**↑ ± 3.76	**72.70**↑ ± 8.05	**71.12**↑ ± 5.40	**71.46**↑ ± 7.62	**69.32**↑ ± 0.08
20	58.88 ± 5.70	17.84↓ ± 2.33	22.47↓ ± 1.06	19.84↓ ± 1.48	21.98↓ ± 1.86	19.49↓ ± 1.18
21	88.73 ± 3.01	69.68↓ ± 4.14	70.71↓ ± 9.94	72.29↓ ± 8.70	74.23↓ ± 1.22	71.04↓ ± 7.45
22	69.18 ± 8.92	**74.26**↑ ± 1.32	**77.71**↑ ± 9.31	**79.11**↑ ± 1.26	**78.56**↑ ± 9.60	**75.70**↑ ± 7.67
23	89.16 ± 1.33	90.90 ± 1.18	91.08 ± 5.23	**95.12**↑ ± 1.01	**93.26**↑ ± 0.67	91.32 ± 2.97
Average	**81.61** ± 11.66	78.12 ± 16.28	80.92 ± 16.04	81.02 ± 16.44	81.07 ± 15.78	79.36 ± 16.24
• significance – *p* value		0.150	0.771	0.812	0.818	0.344
• win/loss/draw		10/12/1	17/6/0	18/5/0	19/4/0	15/8/0
• sig.win/sig.loss		3/5	9/4	11/4	10/4	7/4

Table 19: Average rule set accuracy with standard deviation (*ACC* ± *sd*) for different variants of the unordered algorithm using 10-fold stratified cross-validation.

#	CN2 standard *AUC* ± *sd*	CN2 WRAcc *AUC* ± *sd*	CN2-SD ($\gamma = 0.5$) *AUC* ± *sd*	CN2-SD ($\gamma = 0.7$) *AUC* ± *sd*	CN2-SD ($\gamma = 0.9$) *AUC* ± *sd*	CN2-SD (add.) *AUC* ± *sd*
1	33.39 ± 5.61	**86.12**↑ ± 0.05	**83.31**↑ ± 2.01	**84.27**↑ ± 9.44	**84.47**↑ ± 6.05	**85.12**↑ ± 5.16
2	90.74 ± 3.57	89.52 ± 7.26	**94.37**↑ ± 2.29	**96.28**↑ ± 1.47	**97.33**↑ ± 0.98	**94.52**↑ ± 1.67
3	84.51 ± 0.15	80.11↓ ± 9.84	82.58 ± 5.60	80.98↓ ± 8.12	78.38↓ ± 7.44	83.03 ± 1.88
4	96.22 ± 2.55	93.59 ± 2.26	97.19 ± 0.76	92.37↓ ± 2.33	96.54 ± 1.90	92.87↓ ± 2.66
5	71.33 ± 7.86	**80.75**↑ ± 0.51	**80.52**↑ ± 1.82	**80.56**↑ ± 8.17	**80.76**↑ ± 5.02	**80.06**↑ ± 3.49
6	70.53 ± 5.99	64.42↓ ± 3.29	68.09 ± 7.34	68.63 ± 2.44	64.02↓ ± 8.71	70.61 ± 2.46
7	71.99 ± 5.76	74.00 ± 7.19	73.99 ± 7.63	73.92 ± 6.01	**75.29**↑ ± 7.70	72.73 ± 3.84
8	74.17 ± 5.35	73.98 ± 0.90	**83.82**↑ ± 9.76	**84.69**↑ ± 0.63	**87.02**↑ ± 9.80	**85.62**↑ ± 1.84
9	78.81 ± 4.64	**85.65**↑ ± 0.33	**84.82**↑ ± 2.78	**82.80**↑ ± 5.19	78.66 ± 6.12	**81.29**↑ ± 0.23
10	96.22 ± 2.31	**98.59**↑ ± 0.10	97.13 ± 0.78	96.54 ± 0.13	**99.65**↑ ± 0.04	97.42 ± 0.24
11	94.46 ± 1.52	90.86↓ ± 0.32	93.17 ± 2.68	93.99 ± 2.83	94.30 ± 2.10	93.87 ± 1.07
12	99.17 ± 0.23	99.17 ± 0.16	99.96 ± 0.01	99.38 ± 0.15	99.92 ± 0.03	99.46 ± 0.06
13	83.20 ± 8.68	78.38↓ ± 2.33	82.11 ± 1.04	84.74 ± 4.51	80.12↓ ± 4.12	83.06 ± 6.97
14	75.06 ± 6.13	**79.41**↑ ± 5.12	**81.62**↑ ± 7.61	**79.97**↑ ± 1.29	**80.12**↑ ± 5.34	**78.51**↑ ± 1.15
15	97.90 ± 0.36	78.90↓ ± 6.95	91.88↓ ± 2.73	91.28↓ ± 2.63	90.87↓ ± 2.01	89.15↓ ± 4.32
16	96.88 ± 1.67	96.41 ± 1.63	93.44↓ ± 2.97	95.35 ± 0.18	94.82 ± 1.06	93.95↓ ± 2.06
Average	82.16 ± 16.81	84.37 ± 9.87	**86.75** ± 8.95	86.61 ± 8.81	86.39 ± 10.32	86.33 ± 8.60
• significance – p value		0.563	**0.175**	0.198	0.236	0.236
• win/loss/draw		6/9/1	**10/6/0**	10/6/0	9/7/0	10/6/0
• sig.win/sig.loss		5/5	**6/2**	6/3	7/4	6/3

Table 20: Area under the ROC curve (AUC-Method-2) with standard deviation (*AUC* ± *sd*) for different variants of the unordered algorithm using 10-fold stratified cross-validation.

References

Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. *Advances in Knowledge Discovery and Data Mining*, AAAI Press:307–328, 1996.

Bojan Cestnik. Estimating probabilities: A crucial task in machine learning. In *Proceedings of the Ninth European Conference on Artificial Intelligence*, pages 147–149, Pitman, 1990.

Peter Clark and Robin Boswell. Rule induction with cn2: Some recent improvements. In *Proceedings of the Fifth European Working Session on Learning*, pages 151–163, Springer, 1991.

Peter Clark and Tim Niblett. The cn2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.

William W. Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 115–123, Morgan Kaufmann, 1995.

William W. Cohen and Yoram Singer. A simple, fast, and effective rule learner. In *Proceedings of AAAI/IAAI*, pages 335–342, AAAI Press, 1999.

Sašo Džeroski, Bojan Cestnik, and Igor Petrovski. Using the m-estimate in rule induction. *Journal of Computing and Information Technology*, 1(1):37–46, 1993.

Cesar Ferri-Ramírez, Peter A. Flach, and Jose Hernandez-Orallo. Learning decision trees using the area under the roc curve. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 139–146, Morgan Kaufmann, 2002.

Peter A. Flach. The geometry of roc space: Understanding machine learning metrics through roc isometrics. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 194–201, AAAI Press, 2003.

Peter A. Flach and Iztok Savnik. Database dependency discovery: A machine learning approach. *AI Communications*, 12(3):139–160, 1999.

Yoav Freund and Robert E. Shapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, Morgan Kaufmann, 1996.

Jerome H. Friedman and Nicholas I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9:123–143, 1999.

Johannes Fürnkranz and Peter A. Flach. An analysis of rule evaluation metrics. In *Proceedings of the Twetieth International Conference on Machine Learning*, pages 202–209, AAAI Press, 2003.

Dragan Gamberger and Nada Lavrač. Expert guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*, 17:501–527, 2002.

A.D. Gordon. *Classification*. Chapman and Hall, London, 1982.

K. Chidananda Gowda and Edwin Diday. Symbolic clustering using a new dissimilarity measure. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):567–578, 1992.

David J. Hand and Robert J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45:171–186, 2001.

David Hsu, Oren Etzioni, and Stephen Soderland. A redundant covering algorithm applied to text classification. In *Proceedings of the AAAI Workshop on Learning from Text Categorization*, AAAI Press, 1998.

Viktor Jovanoski and Nada Lavrač. Classification rule learning with apriori-c. In *Progress in Artificial Intelligence: Proceedings of the Tenth Portuguese Conference on Artificial Intelligence*, pages 44–51, Springer, 2001.

Branko Kavšek, Nada Lavrač, and Viktor Jovanoski. Apriori-sd: Adapting association rule learning to subgroup discovery. In *Proceedings of the Fifth International Symposium on Intelligent Data Analysis*, pages 230–241, Springer, 2003.

Willi Klösgen. Explora: A multipattern and multistrategy discovery assistant. *Advances in Knowledge Discovery and Data Mining*, MIT Press:249–271, 1996.

Arno J. Knobbe, Marc de Haas, and Arno Siebes. Propositionalisation and aggregates. In *Proceedings of the Fifth European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 277–288, Springer, 2001.

Arno J. Knobbe, Arno Siebes, and Bart Marseille. Involving aggregate functions in multi-relational search. In *Proceedings of the Sixth European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 287–298, Springer, 2002.

Pat Langley. *Elements of Machine Learning*. Morgan Kaufmann, 1996.

Nada Lavrač, Peter A. Flach, Branko Kavšek, and Ljupčo Todorovski. Adapting classification rule induction to subgroup discovery. In *Proceedings of the Second IEEE International Conference on Data Mining*, pages 266–273, IEEE Computer Society, 2002.

Nada Lavrač, Peter A. Flach, and Blaž Zupan. Rule evaluation measures: A unifying view. In *Proceedings of the Nineth International Workshop on Inductive Logic Programming*, pages 74–185, Springer, 1999.

Nada Lavrač, Filip Železný, and Peter A. Flach. Rsd: Relational subgroup discovery through first-order feature construction. In *Proceedings of the Twelfth International Conference on Inductive Logic Programming*, pages 149–165, Springer, 2003.

Yongwon Lee, Bruce G. Buchanan, and John M. Aronis. Knowledge-based learning in exploratory science: Learning rules to predict rodent carcinogenicity. *Machine Learning*, 30:217–240, 1998.

Peter Ljubič, Ljupčo Todorovski, Nada Lavrač, and John C. Bullas. Time-series analysis of uk traffic accident data. In *Proceedings of the Fifth International Multi-conference Information Society*, pages 131–134, 2002.

Ryszard S. Michalski. Pattern recognition as rule-guided inductive inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4):349–361, 1980.

Ryszard S. Michalski, Igor Mozetič, Jiarong Hong, and N. Lavrač. The multi-purpose incremental learning system aq15 and its testing application on three medical domains. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 1041–1045, Morgan Kaufmann, 1986.

Patrick M. Murphy and David W. Aha. Uci repository of machine learning databases. Available electronically at http://www.ics.uci.edu/ mlearn/MLRepository.html, 1994.

Foster J. Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.

Luc De Raedt and Hendrik Blockeel. Using logical decision trees for clustering. In *Proceedings of the Seventh International Workshop on Inductive Logic Programming*, pages 133–140, Springer, 1997.

Luc De Raedt, Hendrik Blockeel, Luc Dehaspe, and Wim Van Laer. Three companions for data mining in first order logic. *Relational Data Mining*, Springer:106–139, 2001.

Luc De Raedt and Luc Dehaspe. Clausal discovery. *Machine Learning*, 26:99–146, 1997.

Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the Eleventh Conference on Computational Learning Theory*, pages 80–91, ACM Press, 1998.

Avi Silberschatz and Alexander Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *Proceedigns of the First International Conference on Knowledge Discovery and Data Mining*, pages 275–281, 1995.

R.R. Sokal and Peter H.A. Sneath. *Principles of Numerical Taxonomy*. Freeman, San Francisco, 1963.

Ljupčo Todorovski, Peter A. Flach, and Nada Lavrač. Predictive performance of weighted relative accuracy. In *Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery*, pages 255–264, Springer, 2000.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *Proceedings of the First European Conference on Principles of Data Mining and Knowledge Discovery*, pages 78–87, Springer, 1997.

Stefan Wrobel. Inductive logic programming for knowledge discovery in databases. *Relational Data Mining*, Springer:74–101, 2001.