# Sampling-Based Sequential Subgroup Mining

Martin Scholz
Artificial Intelligence Group
Department of Computer Science
University of Dortmund, Germany
scholz@ls8.cs.uni-dortmund.de

## ABSTRACT

Subgroup discovery is a learning task that aims at finding interesting rules from classified examples. The search is guided by a utility function, trading off the coverage of rules against their statistical unusualness. One shortcoming of existing approaches is that they do not incorporate prior knowledge. To this end a novel generic sampling strategy is proposed. It allows to turn pattern mining into an iterative process. In each iteration the focus of subgroup discovery lies on those patterns that are unexpected with respect to prior knowledge and previously discovered patterns. The result of this technique is a small diverse set of understandable rules that characterise a specified property of interest. As another contribution this article derives a simple connection between subgroup discovery and classifier induction. For a popular utility function this connection allows to apply any standard rule induction algorithm to the task of subgroup discovery after a step of stratified resampling. The proposed techniques are empirically compared to state of the art subgroup discovery algorithms.

## Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining;
I.2.6 [**Learning**]: Induction

## General Terms

Algorithms, Performance

## Keywords

subgroup discovery, sampling, prior knowledge

## 1. INTRODUCTION

The discipline of Knowledge Discovery in Databases is about finding useful and novel patterns, hidden in huge amounts of real-world data. Common problems are that the applied Data Mining techniques either find an unmanageable number of patterns, e.g. frequent itemsets, or that they are limited to finding "obvious" patterns only, which are already known to domain experts.

This work presents an approach towards mining interesting patterns sequentially. The most obvious patterns may either be elicited from domain experts beforehand, or they may be the result of a first application of Data Mining tools. The main question that arises in an iterative Data Mining framework is how to pre-process the data, so that subsequent steps do not report previously found patterns again, but focus on uncorrelated new patterns.

The interestingness of patterns is a crucial notion when formalising this task. A basic assumption underlying this work is that patterns are interesting to the degree to which they deviate from the user's expectation. Hence, a straightforward heuristic for rule interestingness is the degree of deviation from the user's specified domain knowledge and from previously found patterns.

For simplicity this work confines itself to probabilistic rules as the representation language. The main ideas from the literature on subgroup discovery are adopted, but extended to respecting prior knowledge. In subgroup discovery the interestingness of rules is evaluated by a utility function. This function can be regarded as a user specified parameter of the learning task. The goal of subgroup discovery is to identify subsets of the population that show an unusually high frequency of a specified property of interest. The task of characterising groups of car drivers with an unusually high risk of accidents is considered as an intuitive toy example. Assuming that the default probability, computed from all registered drivers, is about 1% per year, sufficiently large subgroups having a risk of 5% might be interesting. This illustrates a major difference to classification, where rules are only useful if they predict the most probable class.

In this work subgroups are identified with their corresponding population in the database at hand, rather than with their syntactical descriptions. This makes a difference, since many databases allow to identify the same or similar populations using correlated attributes, but these correlations are often not subject to the Data Mining step. For instance the subgroup of young drivers is almost identical to the subgroup of people that recently acquired their driving license. If the former subgroup is known to have a high risk of accidents then the same is expected for the latter. Hence, there is no need to report both rules.

The presented iterative Data Mining procedure will identify a new subgroup in each iteration, favouring subgroups that are new with respect to formalised prior knowledge and all previously found patterns.


*KDD'05,* August 21–24, 2005, Chicago, Illinois, USA.

The remainder of this paper is organised as follows: After the formal framework and basic definitions are given in section 2, existing work on subgroup discovery is described in section 3. As the main contribution a generic sampling technique to incorporate prior knowledge into subgroup discovery is presented in section 4. An algorithm operationalising this idea is presented in section 5, which is empirically evaluated in section 6. Section 7 concludes the paper.

## 2. BACKGROUND

Two different learning tasks are subject to this paper, subgroup discovery and classifier induction. Both tasks are supervised, so the learning step is based on a classified sample. Examples are defined as classified elements of an instance space $\mathbf{X}$, assumed to be sampled i.i.d with respect to a distribution $D : \mathbf{X} \to \mathbb{R}^+$. To simplify formal aspects $\mathbf{X}$ is assumed to be finite, although all results are easily generalised to continuous domains. The probability to observe an instance $x \in \mathbf{X}$ under $D$ is denoted as $Pr_{x\sim D}(x)$. The probability to observe an instance from a subset $W \subseteq \mathbf{X}$ is denoted as $Pr_D[W]$. If the underlying distribution is clear from the context the subscripts are omitted. Each example is assigned a label from $\mathbf{Y}$, the set of all possible labels by the target function $C : \mathbf{X} \to \mathbf{Y}$. This work considers only supervised learning, unless noted otherwise with a boolean target attribute $\mathbf{Y} = \{0, 1\}$. $C$ is assumed to be fixed but unknown to the learner, whose task is to find a good approximation. For subgroup discovery, the main learning task in this work, Horn logic rules are the main representation language for patterns.

*Definition 1.* A Horn logic rule consists of a body A, which is a conjunction of atoms over attributes describing $\mathbf{X}$, and a head $B$, predicting a value for the target attribute. It is notated as $A \to B$. If the body evaluates to true the rule is said to be *applicable*, if the head also evaluates to true it is called *correct*.

The focus of this work lies on propositional logic, so the bodies of rules are conjunctions of attribute-value pairs. Rules can be identified with the subset they apply for, defined by an indicator function $h : \mathbf{X} \to \{0, 1\}$, and the label they predict. To ease notation the following abbreviations are used whenever $\mathbf{Y} = \{0, 1\}$:

$$h := \{x \in \mathbf{X} \mid h(x) = 1\} \quad , \quad \overline{h} := \mathbf{X} \setminus h$$
$$Y_+ := \{x \in \mathbf{X} \mid C(x) = 1\} \quad , \quad Y_- := \mathbf{X} \setminus Y_+$$

Using this notation, the Horn logic rules predicting a boolean target are of the form $h \to Y_+$ and $h \to Y_-$. The former rule states that the target class is positive ($Y_+$) if the rule is applicable ($h$), the latter that it is negative in this case.

As illustrated by the example of road accidents, rules are not expected to match the data exactly. It is sufficient if they point to interesting regularities in the data, for example that the subgroup of young drivers faces an unusually high risk of accidents. Hence, the intended semantics is that the conditional probability $Pr(Y_+|h)$ (or $Pr(Y_-|h)$) is higher than the class prior $P(Y_+)$ (or $P(Y_-)$). Probabilistic rules are often annotated by their corresponding conditional probabilities:

$$h \to Y_+ \ \ [5\%] \qquad :\Leftrightarrow \qquad Pr_{x\sim D}[C(x) = 1 \mid h(x) = 1] = 5\%$$

For now any form of prior knowledge is assumed to be represented by rules of this form. Subsection 4.3 extends the presented techniques to more general forms of prior knowledge.

Performance metrics are functions that heuristically assign a utility score to each rule under consideration. For the notion of rule interestingness different formalisations have been proposed in the literature (e.g. [24]). In this work interestingness is considered equal to unexpectedness. The following paragraphs discuss a few of the most important metrics for rule selection.

The goal when training classifiers is to select a predictive model that separates positive and negative examples accurately.

*Definition 2.* The *accuracy* of rule $A \to B$ is defined as

$$\mathbf{Acc}(A \to B) \quad := \quad Pr[A \cap B] + Pr\left[\overline{A} \cap \overline{B}\right]$$

*Definition 3.* The *precision* of a rule is defined as its probability of being correct, given that it is applicable:

$$\mathbf{Prec}(A \to B) := Pr[B \mid A]$$

Subgroup discovery has a different focus. Rules are interesting only, if for the covered subset the target attribute's distribution deviates from the default distribution.

*Definition 4.* The **Bias** of a rule is the difference between conditional and default probability (prior) of the target:

$$\mathbf{Bias}(A \to B) := Pr[B \mid A] - Pr[B]$$

The following metric has been used to measure interest in the domain of frequent itemset mining [3]. In the supervised context it measures the change in the target attribute's frequency for the subset covered by a rule.

*Definition 5.* For any rule $A \to B$ the **Lift** is defined as

$$\mathbf{Lift}(A \to B) \quad := \quad \frac{Pr[A \cap B]}{Pr[A]\,Pr[B]} = \frac{Pr[B \mid A]}{Pr[B]}$$
$$= \quad \frac{\mathbf{Prec}(A \to B)}{Pr[B]}$$

The **Lift** is the multiplicative counterpart to the **Bias**, and is often more convenient in the scope of this work. Both metrics capture the value of "knowing" the prediction for estimating the probability of the target attribute. For independent events $A$ and $B$ we have $\mathbf{Lift}(A \to B) = 1$ and $\mathbf{Bias}(A \to B) = 0$, for positively correlated events $A$ and $B$ we have $\mathbf{Lift}(A \to B) > 1$ and $\mathbf{Bias}(A \to B) > 0$. In the latter case the conditional probability of $B$ given $A$ is higher than the default probability $Pr[B]$.

If $h$ characterises the subgroup of young drivers, $Y_+$ denotes the event of having an accident, and if $Pr[Y_+] = 1\%$ is the default probability of this event, then the above rule $h \to Y_+$ [5%] has a **Bias** of 4%, which is the absolute difference between conditional and default probability, and a **Lift** of 5, reflecting the increase in terms of a relative factor.

Knowing the **Lift** of rules allows to combine them in a simple way to predict the conditional probability of a target class. If a new attribute is defined according to the prediction of each rule, then predictions can be combined by means of classifier induction techniques. The underlying assumption of Naïve Bayes [11] is that all attributes are conditionally independent given the class. These classifiers work surprisingly well in practice, often even if the underlying assumption is known to be violated. When mining

rules iteratively, using the sampling technique proposed in section 4, the conditional independence assumption is not as unrealistic as one might expect. The reason is that all correlations "reported" by previously found patterns are "removed" from subsequently constructed samples.

Let $\{h_i \rightarrow Y_{+/-} \mid 1 \leq i \leq n\}$ denote a set of rules, $\langle h_1, \ldots, h_n \rangle(x)$ the corresponding vector $\hat{y} = \langle \hat{y}_1, \ldots, \hat{y}_n \rangle$, $\hat{y} \in \mathbf{Y}^n$, of predictions. Then for a given example $x \in \mathbf{X}$ and class $Y_c$ the Naïve Bayes classifier estimates

$$\begin{aligned} & Pr\left[Y_c \mid \langle h_1, \ldots h_n \rangle(x) = \hat{y}\right] \\ = \; & \frac{Pr\left[Y_c\right]}{Pr\left[\langle h_1, \ldots h_n \rangle(x) = \hat{y}\right]} \cdot Pr\left[\langle h_1, \ldots h_n \rangle(x) = \hat{y} \mid Y_c\right] \\ \approx \; & \frac{Pr\left[Y_c\right]}{Pr\left[\langle h_1, \ldots, h_n \rangle(x) = \hat{y}\right]} \cdot \prod_{1 \leq i \leq n} Pr\left[h_i(x) = \hat{y}_i \mid Y_c\right] \\ = \; & \frac{Pr\left[Y_c\right] \cdot \prod_i Pr\left[h_i(x) = \hat{y}_i\right]}{Pr\left[\langle h_1, \ldots, h_n \rangle(x) = \hat{y}\right]} \prod_{1 \leq i \leq n} \frac{Pr\left[Y_c \mid h_i(x) = \hat{y}_i\right]}{Pr\left[Y_c\right]} \end{aligned}$$

Defining

$$\alpha(x) := \frac{\prod_{1 \leq i \leq n} Pr\left[h_i(x) = \hat{y}_i\right]}{Pr\left[\langle h_1, \ldots, h_n \rangle(x) = \hat{y}\right]}$$

allows to rewrite this term to

$$\alpha(x) \cdot Pr\left[Y_c\right] \cdot \prod_{1 \leq i \leq n} \mathbf{Lift}((h_i(x) = \hat{y}_i) \rightarrow Y_c).$$

For boolean $\mathbf{Y}$ it is easier to consider the odds

$$\begin{aligned} \beta(x) \; &:= \; \frac{Pr\left[Y_+ \mid \langle h_1, \ldots, h_n \rangle(x) = \hat{y}\right]}{Pr\left[Y_- \mid \langle h_1, \ldots, h_n \rangle(x) = \hat{y}\right]} \\ &= \; \frac{Pr\left[Y_+\right]}{Pr\left[Y_-\right]} \prod_{1 \leq i \leq n} \frac{\mathbf{Lift}((h_i(x) = \hat{y}_i) \rightarrow Y_+)}{\mathbf{Lift}((h_i(x) = \hat{y}_i) \rightarrow Y_-)}, \quad (1) \end{aligned}$$

as $\alpha(x)$ cancels out, but it is still possible to recalculate

$$Pr\left[Y_+ \mid \langle h_1, \ldots, h_n \rangle = \hat{y}\right] = \frac{\beta(x)}{1 + \beta(x)} \quad (2)$$

based on eqn. (1). So following the conditional independence assumption it is possible to combine rules to predict class probabilities, just knowing their **Lift** and the class priors. Eqn. (1) has an intuitive interpretation. The first factor reflects the positive-negative ratio, which is 1/99 for the prior $Pr\left[Y_+\right] = 1\%$. The **Lift**-ratios specify how to update these positive-negative ratios in relative terms, if it is known whether a specific rule is applicable or not. If the driver under consideration is young, then there is a known factor associated to the corresponding rule, which allows to update the previously computed ratio, increasing it from 1/99 to 5/95. The same holds for further, subsequently discovered rules, each of which may be annotated by an empirically estimated **Lift**-ratio.

It is not necessary to restrict rules to the case in which they are applicable. Please note that

$$\mathbf{Lift}(h \rightarrow Y_+) > 1 \;\Rightarrow\; \mathbf{Lift}(\overline{h} \rightarrow Y_-) > 1,$$

but the precisions of both rules may differ. This is easily seen considering an example. If, compared to the average driver, young drivers have a higher risk of accidents, then the average risk of the remaining drivers *has* to be lower, since removing a subgroup with higher risk always decreases the overall risk. So each rule $(h \rightarrow Y_{+/-})$ should rather be considered to partition the instance space into $h$ and $\overline{h}$, making a prediction for both subsets. As a consequence any two rules overlap. Thus, for any known degree of overlap between a rule $R_1$ that is part of the prior knowledge and a rule candidate $R_2$ under consideration, there is an expectation for $\mathbf{Lift}(R_1)$ based on $\mathbf{Lift}(R_2)$. This expectation reflects the assumption that $R_2$ does not introduce a **Lift** of its own, but simply shares a biased subset with $R_1$. If this assumption is met, then the rule candidate is redundant and should be ranked low. The **Lift** of a rule should change in the presence of prior knowledge. A corresponding technique is introduced in section 4.

# 3. SUBGROUP DISCOVERY

Subgroup discovery aims at finding interesting subsets of the instance space that deviate from the overall distribution. Different search strategies led to several algorithms, which are briefly described in subsection 3.1. In all cases the search is guided by a *utility function*, a specific type of rule selection metric, which can be regarded as a parameter of the learning task itself. Choosing this function carefully allows to direct the search towards different kinds of interesting rules, e.g. rules having a lower bias but a higher coverage compared to those found via standard classifier induction. For a specific utility function some recently proposed subgroup discovery algorithms are limited to, there is a simple way to transform the corresponding formal Data Mining problem into a classifier induction problem. This transformation is presented in subsection 4.4 and it will be used later on to evaluate the proposed techniques empirically. The benefits of incorporating prior knowledge into subgroup discovery and some existing approaches are discussed in subsection 3.2, before the generic knowledge-based sampling approach is presented in section 4.

## 3.1 Existing Approaches

The common goal of all subgroup discovery strategies is to find interesting and novel patterns in datasets. To this end utility functions are used that formalise a trade-off between the size of the subgroup and its unusualness in terms of a target attribute's observed frequency. Each subgroup is represented by a Horn logic rule. A popular utility function is the *weighted relative accuracy* [13]. It is used for subgroup discovery since EXPLORA [12], and it is the default utility function of MIDOS [27].

*Definition 6.* The *weighted relative accuracy* (**WRAcc**) of a rule $A \rightarrow B$ multiplies coverage ($Pr\left[A\right]$) with bias:

$$\mathbf{WRAcc}(A \rightarrow B) := Pr\left[A\right] \cdot \mathbf{Bias}(A \rightarrow B)$$

Several other functions have been suggested in the literature [12], basically putting more emphasis on either coverage or bias. This work only makes use of the commonly used **WRAcc** metric for mainly two reasons. First of all it allows the underlying formal Data Mining problem to be tackled with approved rule induction algorithms (subsection 4.4), which simplifies the evaluation in practice. The second reason is that it is sufficient to compare the presented approach to the reweighting schemes that have recently been proposed in the scope of subgroup discovery [14].

There are two different strategies of searching for interesting rules: exhaustive and heuristic search. EXPLORA [12] and MIDOS [27] tackle subgroup discovery by exhaustively evaluating the set of rule candidates. The set of rules are

ordered by generality, which allows to prune large parts of the search space. The advantage of this strategy is that it allows to find the $n$ best subgroups reliably. Finding subgroups on subsamples of the original data is a straightforward method to speed up the search process. As shown in [22] most of the utility functions commonly used for subgroup discovery are well suited to be combined with adaptive sampling. This sampling technique reads examples sequentially, continuously updating upper bounds for the sample errors, based on the data read so far. In this way, the required sample size allowing to give a probabilistic guarantee of not missing any of the $n$ best subgroups can be reduced.

Heuristic search strategies are fast, but do not come with any guarantee to find the most interesting patterns. One recent example implementing a heuristic search is a variant of CN2. By adapting its rule selection metric to **WRAcc** the well known CN2 classifier has been turned into CN2-SD [14]. As a second modification the iterative cover approach of CN2 has been replaced by a heuristic weighting scheme. Example weights are either changed by a constant factor or by an additive term each time the example has been covered by a rule. In section 4 a new generic weighting scheme is proposed that allows to overcome some shortcomings of CN2-SD.

## 3.2 Missing Features

A drawback of classical subgroup discovery lies in a lack of expressiveness. Especially interesting exceptions to rules are hard to be detected using standard techniques, for mainly two reasons. First of all, due to the syntactical structure imposed by Horn logic it is often hard to exclude exceptions from rules, even if this improves the score assigned by the utility function. The syntactical bias is important, however, because results are required to be understandable and because it is the main reason for diversity within the $n$ best subgroups. The syntactical bias might not be sufficient to avoid sets of similar rules. Redundancy filters are a common technique to overcome this problem [12]. Overlapping patterns like exceptions to rules are not found reliably that way. Exceptions could still be represented by separate rules. This fails for the second reason, namely that utility functions evaluate rules globally. Interactions between rules do not affect their scores.

For the task of finding exception rules some efficient algorithms have been developed [25]. They focus on mining pairs of a strong rule and a corresponding exception, which is too specific for subgroup discovery in general. Subgroups do not necessarily have exceptions, and they may overlap in arbitrary ways.

As a strategy for pruning rulesets to cover different aspects, ROC analysis was suggested in [14]. According to the false positive and false negative rates all rules are plotted in ROC space [5]. Only rules lying on the convex hull are deemed relevant and may be turned into a single classifier by weighted majority vote. A major drawback of this filter is that it systematically discards one of two rules covering disjoint subsets and having almost the same performance. As soon as one of these rules is superior in both true positive and false negative rates the other rule is considered to be redundant. This is not desirable in descriptive scenarios, as the only rule covering a specific subset of the instance space should not easily be discarded, nor for predictive settings, as diversity of base classifiers is crucial for reaching high predictive accuracy. The latter has empirically been shown by the success of Random Forests [2] and similar ensemble methods.

A way to improve the interestingness and diversity of rulesets is to make use of previously found patterns and formalised prior knowledge during construction. Incorporating prior knowledge like Bayesian Networks into existing Data Mining techniques is an active field of research. Some approaches like [28, 20] try to utilise prior knowledge to compensate for a lack of data. In these scenarios the models are fitted to both, the prior knowledge and the dataset at hand. In contrast, the goal of subgroup discovery is to find rules that contradict expectation, as this is assumed to indicate interestingness. In such a scenario any available information that allows to compute estimates of the user's expectation may help to refine the metric for selecting interesting rules. A similar idea has recently been proposed in the scope of frequent itemset mining [10].

Following this idea, a subgroup pattern may be interesting relative to prior knowledge, only, as illustrated by the following example:

$$Pr\left[Y_+ \mid A\right] = Pr\left[Y_+\right] = 0.5 \quad \text{for a rule} \quad A \rightarrow Y_+.$$

$\mathbf{Y}$ is distributed in $A$ just as in the overall population, so this rule would not be deemed interesting by any reasonable utility function. Now assume that in the prior knowledge there is a statement about a superset of $A$:

$$B \rightarrow Y_+ \quad [0.9] \quad \text{with} \quad A \subset B.$$

This rule predicts a higher conditional probability of $Y_+$ given $B$. In this context the rule $A \rightarrow Y_+$ becomes interesting as an exception to the prior knowledge, because one would rather expect $Pr\left[Y_+ \mid A\right] = Pr\left[Y_+ \mid B\right]$. The reason is that the prediction for $B \subset X$ is more specific than the general class priors.

To the best of the author's knowledge the only approach towards incorporating available knowledge into subgroup discovery reported in the literature so far, is the ILP system RSD [15]. It uses background knowledge exclusively to propositionalise relational data, a step which is out of the scope of this work. For the learning step itself CN2-SD is used. The next section shows a generic technique to incorporate prior knowledge into subgroup discovery and similar supervised learning tasks.

# 4. KNOWLEDGE-BASED SAMPLING

This section introduces a sampling-based technique to incorporate prior knowledge into supervised Data Mining algorithms. Subsection 4.1 discusses the overall idea before some constraints for sampling are defined in subsection 4.2. These constraints define a unique distribution as shown in 4.3. The last subsection 4.4 shows how to use stratified sampling in order to solve specific subgroup discovery tasks by means of classifier induction algorithms.

## 4.1 Weighting examples by prior knowledge

The most crucial question in an iterative Data Mining framework is how to pre-process the training data so that subsequent learning steps do not yield previously found patterns again. The goal is to find uncorrelated new patterns, so that the resulting ruleset is compact, but still allows for a precise characterisation of the target attribute. The two example subgroups mentioned in the introduction, one

containing all young drivers, and the other containing all persons who recently acquired their driving license, illustrates how a stand-alone evaluation of each rule may result in highly overlapping rulesets. This bears the risk of almost redundant rules.

The algorithm proposed in this work allows to focus on previously undiscovered patterns by means of sampling. Target samples are constructed in a way that does not allow to rediscover the available prior knowledge, because the target attribute is made independent of the available predictions. At the same time it is taken care, that the remaining patterns part of the original data remain intact. Similar techniques are found in the boosting literature [7, 8, 21]. Please note, that boosting was first introduced in terms of altering an initial distribution function and a corresponding sampling technique [19].

The technique which is shown to be capable of sampling out prior knowledge in the following sections is called *rejection sampling* [16]. It allows to sample with respect to a distribution $D'$, given a procedure to sample from another distribution $D$: Assume that an example set of size $n$ at hand has been sampled from distribution $D^n$. Then each example $x$ is assigned a weight

$$w : \mathbf{X} \to \mathbb{R}^+, w(x) := \frac{Pr_{D'}(x)}{Pr_D(x)}$$

rather than sampling directly with respect to $D'$, which may be infeasible. A sample

$$x^m \sim D' \quad \text{with} \quad m := n \cdot \left[\sup_{x \in X} w(x)\right]^{-1}$$

may then be constructed by weight-proportionate resampling. Alternatively, the weights may be interpreted as factors of being over- or underrepresented by all subsequently applied algorithms.

Rejection sampling has also been approved as a generic way to incorporate costs into the Data Mining step. In [29] a proof is given, that this kind of sampling does not increase the sample complexity in the agnostic PAC learning framework for cost sensitive classification. As illustrated in the next subsections, rejection sampling even allows to generically incorporate a user-given or previously discovered probabilistic model into the Data Mining step. For this purpose a knowledge-based sampling scheme based on altering the original distribution underlying a dataset is introduced. Depending on the application, examples may be weighted or resampled.

## 4.2 Constraints for resampling

Before going into detail the idea of removing prior knowledge by means of sampling is formulated in terms of constraints. Formally, this step means to define a new distribution $D'$, as close to the original function $D$ as possible, but independent of the estimates produced by available prior knowledge. Switching from the initial distribution to the resampled data is a step of applying prior knowledge by means of sampling. As a result the previously discussed rule selection metrics – when applied to these kind of samples – are "blinded" regarding the parts of rules that could already be concluded from prior knowledge. All that is accounted for is the unexpected component of each rule.

The scenario is discussed along the simplified case of a single rule as available prior knowledge:

$$R : h \to Y_+$$

The distribution to be constructed should no longer support rule $R$, so $h$ and $Y_+$ should be independent events:

$$Pr_{D'}[Y_+ \mid h] = Pr_{D'}[Y_+] \tag{3}$$

If $R$ predicts a higher accident probability for young drivers, for example, then in the constructed sample this subgroup should share the default probability of accidents.

As further constraints the probabilities of events part of the rule should not change, since it is sufficient to remove their correlation. This means that the class priors and the probability of $R$ being applicable to a randomly drawn instance are equal for both distributions:

$$Pr_{D'}[h] = Pr_D[h] \tag{4}$$
$$Pr_{D'}[Y_+] = Pr_D[Y_+] \tag{5}$$

For the example rule the probability of accidents and the probability of seeing a young driver will not change from the original training set to the constructed sample.

Finally, within each partition sharing the same class and prediction of $R$ the new distribution is defined proportionally to the initial one. The simple reason is that having just $R$ as prior knowledge all instances within one partition are indistinguishable. Changing the conditional probabilities within one partition would mean to prefer some instance over others despite their equivalence with respect to the available prior knowledge. For the boolean rule $R$ these constraints translate into the following equalities:

$$Pr_{D'}(x \mid h \cap Y_+) = Pr_D(x \mid h \cap Y_+) \tag{6}$$
$$Pr_{D'}(x \mid h \cap Y_-) = Pr_D(x \mid h \cap Y_-) \tag{7}$$
$$Pr_{D'}(x \mid \overline{h} \cap Y_+) = Pr_D(x \mid \overline{h} \cap Y_+) \tag{8}$$
$$Pr_{D'}(x \mid \overline{h} \cap Y_-) = Pr_D(x \mid \overline{h} \cap Y_-) \tag{9}$$

Hence, if the probability of seeing a specific driver halves from the original to the new distribution, then the same will happen to all other drivers sharing both, the property of being young or not, and the property of having had an accident or not. All that changes are the marginal probabilities of the partitions.

Given a database and pattern $R$ it is possible to apply any Data Mining technique after sampling with respect to $D'$. Further interesting patterns, even if they are overlapping with $R$, are still observable in the new sample. For instance subgroups that are subsets of $h$ and have an even significantly higher or much lower **Lift** than rule $R$ are just rescaled proportionally. For instance, if unexperienced persons driving a specific kind of car tend to be involved in accidents even more frequently than young drivers in general, then this more specific rule can be found in a subsequent step. As motivated in subsection 3.2, various exceptions to previously found rules and patterns overlapping in some other way can be found, analogously.

## 4.3 Constructing a new distribution function

In subsection 4.2 the idea of sampling with respect to an altered distribution function has been presented. Intuitively, prior knowledge and known patterns are "filtered out". This subsection proves that the proposed constraints (3)-(9) induce a unique target distribution.

*Definition 7.* The **Lift** *of an example* $x \in \mathbf{X}$ for a rule $h \to Y_+$ is defined as

$$\mathbf{Lift}(h \to Y_+, x) := \begin{cases} \mathbf{Lift}(h \to Y_+), \text{ for } x \in h \cap Y_+ \\ \mathbf{Lift}(h \to Y_-), \text{ for } x \in h \cap Y_- \\ \mathbf{Lift}(\overline{h} \to Y_+), \text{ for } x \in \overline{h} \cap Y_+ \\ \mathbf{Lift}(\overline{h} \to Y_-), \text{ for } x \in \overline{h} \cap Y_- \end{cases}$$

*Theorem 1.* For any initial distribution $D$ and given rule $R$ the constraints (3)-(9) are equivalent to

$$Pr_{x \sim D'}(x) = Pr_{x \sim D}(x) \cdot (\mathbf{Lift}_D(R, x))^{-1}.$$

Up to a constant factor they induce $D' : \mathbf{X} \to \mathbb{R}^+$ uniquely.

PROOF. The proof is exemplarily shown for the partition $(h \cap Y_+)$, in which the rule under consideration is both applicable and correct. Assuming that the constraints hold $D'$ can be rewritten in terms of $D$ and $\mathbf{Lift}(R, x)$:

$$\begin{aligned} (\forall x \in h \cap Y_+) \quad : \quad & Pr_{D'}(x) \\ = \quad & Pr_{D'}(x \mid h \cap Y_+) \cdot Pr_{D'}[h \cap Y_+] \\ = \quad & Pr_D(x \mid h \cap Y_+) \cdot Pr_{D'}[h] \cdot Pr_{D'}[Y_+] \\ = \quad & \frac{Pr_D(x)}{Pr_D[h \cap Y_+]} \cdot Pr_D[h] \cdot Pr_D[Y_+] \\ = \quad & Pr_D(x) \cdot (\mathbf{Lift}_D(h \to Y_+))^{-1} \end{aligned}$$

The other three partitions can be rewritten analogously. On the other hand, it can easily be validated that $D'$ as defined by theorem 1 is in fact a distribution satisfying the constraints:

$$\begin{aligned} Pr_{D'}[h \cap Y_+] \quad = \quad & Pr_D[h \cap Y_+] \cdot (\mathbf{Lift}_D(R, x))^{-1} \\ = \quad & Pr_D[h] \cdot Pr_D[Y_+] \end{aligned}$$

and analogously for the other partitions. This directly implies constraints (3)-(5) by marginalising out. Constraints (6)-(9) are met, because for all four partitions $D'$ is defined proportionally to $D$. □

Please recall, that the **Lift** simply reflects the factor by which a label is overrepresented in a considered subset, compared to the label's prior. For the example rule the **Lift** is 5, since the risk for young drivers is 5 times higher than for the average driver. Hence, the probability to see a specific young driver who had an accident in the target sample is reduced by a factor of 1/5 compared to the original data. Each of the four partitions that are defined by a combination of prediction and true label is rescaled in the same fashion.

Theorem 1 defines a new distribution to sample from, given a single rule $R$ as prior knowledge. The same strategy may be applied iteratively, defining a new distribution after each selected rule. Section 6 introduces an appropriate algorithm and evaluates it empirically.

Please note, that without any changes to theorem 1 more complex forms of prior knowledge may be incorporated. Ensembles of base learners like propositional rules are valid background theories, for example, as long as the predictions are discrete. Straightforward generalisations of constraints (3)-(9) allow to incorporate probabilistic predictions: Let the prior knowledge $\theta$ be associated to a function

$$\widehat{Pr}(x, y \mid \theta) = \widehat{Pr}(C(x) = y \mid x, \theta) \approx Pr[C(x) = y \mid x]$$

estimating the target distribution for each $\langle x, y \rangle \in \mathbf{X} \times \mathbf{Y}$. Assuming the class priors $Pr[C(x) = y]$ to be known for each $y \in \mathbf{Y}$ and applying the definition of the **Lift** the corresponding *estimated* **Lift** can easily be computed as

$$\widehat{\mathbf{Lift}}(x \to y \mid \theta) := \frac{\widehat{Pr}(x, y \mid \theta)}{Pr_{z \sim D}(C(z) = y)}$$

Given a procedure for sampling examples $x \sim D$ i.i.d., the following distribution that generalises theorem 1 can be used to weight examples:

$$D'(x) = D(x) \cdot \left(\widehat{\mathbf{Lift}}(x \to C(x) \mid \theta)\right)^{-1} \tag{10}$$

To remove prior probabilistic knowledge, for example from a data stream, it is sufficient to assign to each example $x$ a weight of $(\widehat{\mathbf{Lift}}(x \to C(x) \mid \theta))^{-1}$. This strategy is surprisingly simple and well suited to apply rejection sampling.

Up to here no assumptions about the utility function for evaluating rule candidates were made. In fact any utility function can be used in combination with knowledge-based sampling.

## 4.4 A Connection to Classifier Induction

This subsection shows a simple connection between subgroup discovery limited to the utility function **WRAcc** and the better known task of classifier induction.

The goal when inducing a classifier from data generally is to select a predictive model that separates positive and negative examples with high predictive accuracy. Many algorithms and implementations exist for this purpose [18, 26], basically differing in the set of models (hypothesis space $H$) and search strategies. Subgroup discovery demands the definition of a property of interest, which can be assumed to be present in the form of a target attribute. In this sense this task is also supervised. The process of model selection is guided by a utility function. In the following definition subgroup discovery is simplified to finding the most interesting rule.

*Definition 8.* Let $H$ denote the set of models (rules) valid as output and $D$ denote a distribution function over $\mathbf{X}$. The task of *classifier induction* is to find

$$h^* := \operatorname*{maxarg}_{h \in H} \mathbf{Acc}(h).$$

For a given utility function $q : H \to \mathbb{R}$ the task of *subgroup discovery* is to find

$$h^* := \operatorname*{maxarg}_{h \in H} q(h).$$

For boolean target attributes common classifier induction algorithms do not benefit from finding rules with a precision below 50%. In contrast, for subgroup discovery it is sufficient if the precision of a rule is higher than the corresponding class prior. Choosing the utility function **WRAcc** we can transform subgroup discovery as defined above into classifier induction by a simple sampling technique to overcome imbalanced class distributions.

*Definition 9.* For $D : \mathbf{X} \to \mathbb{R}^+$, $C : \mathbf{X} \to \mathbf{Y}$ the *stratified random sample distribution* $D'$ of $D$ (and $C$) is defined by

$$Pr_{x \sim D'}(x) := \frac{Pr_{x \sim D}(x)}{|Y| \cdot Pr_{z \sim D}[C(z) = C(x)]}$$

$D'$ is defined by rescaling $D$ so that the class priors are equal. This definition allows to state the following theorem.

*Input:*
- labelled example set $E = \langle x_1, y_1\rangle, \ldots, \langle x_m, y_m\rangle$
- integer $n$

*Output:*
- Set of $n$ (probabilistic) Horn logic rules

**KBS(E, n):**

1. Let $D_0$ denote the uniform distribution over $E$.
2. For each $c \in \mathbf{Y}$ define $\pi(c) := Pr_{x\sim D_0}(C(x) = c)$.
3. Let $D_1(x_i) := \pi(y_i)^{-1}$ for $i \in \{1, \ldots, n\}$.
4. For $k = 1$ to $n$ do
   (a) $r_k \leftarrow$ RuleInduction($D_k$, $E$)
   (b) Compute $\mathbf{Lift}_{D_k}(r_k, x_i)$ applying definition 7.
   (c) Let $D_{k+1}(x_i) := D_k(x_i) \cdot (\mathbf{Lift}_{D_k}(r_k, x_i))^{-1}$.
5. Output the set of rules $\{r_1, \ldots, r_n\}$ and their **Lift**s.

**Figure 1: Algorithm KBS**

*Theorem 2.* For every rule $h \rightarrow Y_+$ the following equalities hold if $D'$ is the stratified random sample distribution of $D$:

$$\begin{aligned}&\mathbf{Acc}_{D'}(h \rightarrow Y_+) = 2\mathbf{WRAcc}_{D'}(h \rightarrow Y_+) + 1/2\\ =\ &\mathbf{WRAcc}_D(h \rightarrow Y_+) \cdot \underbrace{\frac{1}{2Pr_D[Y_+] \cdot Pr_D[Y_-]}}_{\text{irrelevant for ranking rules}} + 1/2\end{aligned}$$

Proof. The first equality can be proved by rewriting accuracy in terms of **WRAcc**, exploiting $P_{D'}(C) = P_{D'}(\overline{C})$. The second equality follows by applying the definition of $D'$ to $\mathbf{WRAcc}_{D'}(h \rightarrow Y_+)$, reaching at a reformulation in terms of the original distribution $D$. □

For a full proof please refer to [23], for a broader discussion of rule selection metrics to [6, 9]. As a consequence of theorem 2 subgroup discovery tasks with utility function **WRAcc** can as well be solved by rule induction algorithms optimising predictive accuracy after stratified resampling. The induced rankings of rules are equivalent.

## 5. ALGORITHMS

This section discusses three subgroup discovery algorithms, which have been integrated into the learning environment YALE [17]. The first of these is the *knowledge-based sampling algorithm* (KBS) shown in figure 1. It applies sampling as presented in subsection 4.3. More precisely, the implementation allows to use example reweighting if the training data fits into main memory. As discussed in subsection 4.1 the probabilities computed by rejection sampling procedures may as well be interpreted as example weights.

The KBS algorithm iteratively selects a single rule corresponding to a subgroup with high **WRAcc**, updates the distribution according to theorem 1, and selects the next rule according to the updated distribution. Theorem 2 implies that high predictive accuracy on stratified samples directly translates into high **WRAcc** on the original data. To this end lines 1-3 define a distribution $D_1$ by reweighting the training examples $E$ with respect to definition 9. Each loop in line 4 induces a single rule with high predictive accuracy, characterising a new subgroup. The rule induction algorithm used in the experiments is ConjunctiveRule, part of the WEKA learning environment [26]. It iteratively constructs the body of rules comparing the information gain of each candidate literal, and it prunes rules applying the reduced error pruning heuristic.

As distribution updates are computed according to theorem 1 (line 4c) all constraints defined in subsection 4.2 hold. Constraint (5) implies that all subsequently defined distributions share the stratification property of $D_1$. The corresponding distributions without stratification are very similar to the distributions actually used. They can be reconstructed by rescaling with respect to the original class priors.

In a prediction scenario significance tests help to avoid overfitting, in a descriptive setting they avoid to report rules which could easily be valid just by chance. For simplicity significance tests are avoided in the experiments, but the mining step is restricted to $n$ iterations. Rules are selected by **WRAcc**, a metric proportional to coverage, so rules are not expected to overfit if $n$ is chosen small enough.

The selected rules annotated by the **Lift**s allow to compute predictions for the target attribute. If the property of interest is boolean[1], then all rules $r_i$ are of the form $h_i \rightarrow Y_{+/-}$. An application of the Naïve Bayes strategy for combining predictions (see section 2) yields

$$\beta(x) = \frac{Pr_{D_0}(Y_+)}{Pr_{D_0}(Y_-)} \cdot \prod_{1\le i<k} \underbrace{\frac{\mathbf{Lift}_{D_i}((h_i \rightarrow Y_+), x)}{\mathbf{Lift}_{D_i}((h_i \rightarrow Y_-), x)}}_{=:\beta_i(x)} \qquad (11)$$

for the odds (eqn. (2)). Estimating each $\beta_i$ with respect to $D_i$ prevents poor approximations in case of violated conditional independence. If the precision of a selected rule is 1 then eqn. (11) is not applicable, but the covered subset may simply be removed during training.

The reweighting performed by KBS is very similar to eqn. (10). Enumerator and denominator of the product in (11) approximate the **Lift**s for positives and negatives, respectively, e.g.:

$$\widehat{\mathbf{Lift}}(x \rightarrow Y_+ \mid \theta) \approx \prod_{1\le i<k} \mathbf{Lift}_{D_i}((h_i \rightarrow Y_+), x),$$

where $\theta$ denotes the set of probabilistic rules $\{r_1, \ldots r_k\}$. It can easily be seen that after stratification the algorithm weights examples inverse proportionally to these estimates.

In the next section KBS is compared to the only two other reweighting strategies reported in the subgroup discovery literature so far [14]. These strategies just affect the reweighting step of the algorithm shown in figure 1: After a positive example $e$ has been covered by $i$ rules its new weight is computed as either

**additive update:** $w_i(e) := \frac{1}{i+1}$ or

**multiplicative update:** $w_i(e) := \gamma^i$ for given $\gamma \in [0, 1]$

Accordingly, two versions of subgroup discovery ruleset induction (SDRI) have been implemented, which are similar

[1]This restriction is only made for simplicity, as it is always possible to estimate each class against all the others.

| Dataset | Examples | Discr. | Cont. | Minority |
|---|---|---|---|---|
| KDD Cup | 10.000 | – | 71 | 50.0% |
| Adult | 32.562 | 8 | 6 | 24.1% |
| Ionosphere | 351 | – | 34 | 35.8% |
| Credit Domain | 690 | 6 | 9 | 44.5% |
| Voting-Records | 435 | 16 | – | 38.6% |
| Mushrooms | 8.124 | 22 | – | 48.2% |

**Table 1: Data characteristics: size, number discrete/continuous attributes, frequency min. class**

to CN2-SD. The variant that applies ConjunctiveRule on stratified samples after additive updates is referred to as SDRI$^+$, the one with multiplicative updates as SDRI$^*$. Weights are updated after each iteration. The class explicitly predicted by a rule is defined to be the positive one, as fixing one of the classes as positive gave worse experimental results. Multiple occurrences of rules in the ruleset are allowed, reflecting the importance of patterns that are still observable after reweighting. The rulesets constructed by SDRI are combined as in CN2-SD rather than by applying Naïve Bayes: The predicted target class distributions of all applicable rules are averaged.

## 6. EXPERIMENTS

A primary goal of subgroup discovery is to find a small set of understandable rules that characterise a target variable. In more formal terms the probabilistic classifiers built from the rulesets should be accurate. This property is commonly measured by the area under the ROC curve metric (AUC).

The proposed idea of sequential sampling-based subgroup discovery has been evaluated on five datasets from the UCI Machine Learning Library [1] and a 10K sample taken from the KDD Cup 2004 Quantum Physics dataset[2]. All datasets have boolean target attributes. Further characteristics are listed in table 1.

Figure 2 to 7 show how the AUC performance changes with an increasing number of iterations. All values have been estimated by 10fold cross-validation[3]. The columns **n** and **auc** in table 2 list the average performances of rulesets from the cross-validation experiments for the empirically best choice of $n$. For the KDD Cup data and the adult dataset the number of iterations were limited. To further evaluate the differences between the algorithms another ruleset was induced for each variant, using the same value for parameter $n$. For evaluation the same set was used as for training, as common for descriptive learning tasks. Table 2 shows the resulting average coverages (**cov**) and average weighted relative accuracies (**wracc**). The ROC filter for rulesets discussed in subsection 3.2 was applied to both SDRI variants, denoted as RF in table 2.

The column **div** reflects the diversities of rulesets. The entropy of predictions is an appropriate measure for diversity of classifier ensembles in general [4]. Each rule can be considered to predict the conditional distribution of the target, given whether it is applicable or not. For a boolean target, a set of $n$ rules with $p_i(x) := Pr(Y_+ \mid h_i(x))$, and a set of $m$ examples the diversity was computed as

$$\frac{1}{m}\sum_{k=1}^{m}\frac{1}{n}\sum_{i=1}^{n} -p_i(x_k)\log(p_i(x_k)) - (1-p_i(x_k))\log(1-p_i(x_k))$$

after removing multiple occurrences of rules.

Throughout the figures 2 to 7 the KBS algorithm outperforms SDRI with both reweighting strategies, while none of the SDRI variants is clearly superior to the other one. In figure 2 all three algorithms manage to find useful rules repeatedly. SDRI$^+$ performs best for sets of 3 to 6 rules, but for larger rulesets and for any other dataset and number of iterations KBS is superior. In figures 3 to 5 KBS improves AUC much quicker than SDRI, although for the smallest dataset (fig. 4) it overfits after the 3rd iteration. For the credit domain data (fig. 5) the AUC values of the SDRI rulesets improve non-monotonically. Inspecting the rulesets reveals many duplicates. For the voting-records (figure 6) SDRI effectively finds just 2 useful rules with both reweighting strategies, improving AUC by about 1% compared to the first iteration. KBS selects 6 rules and improves AUC by about 4%. Finally, in the experiment shown in figure 7 KBS reaches 100% AUC with just 12 rules, while SDRI hardly improves over the performance of the first rule at all. For the smaller datasets the ROC filter basically just removes duplicates from the rulesets, which has a marginal impact on the performance metrics. For the large datasets the filter prunes the ruleset at the price of a reduced AUC performance and diversity.

Although the KBS rulesets often have a smaller coverage and **WRAcc** their predictions outperform those of the other algorithms. It is interesting to note that for all datasets the KBS rulesets have the highest diversity, but according to the standard deviation of the AUC performance (column "±") they are nevertheless most robust against minor changes to the data.

## 7. CONCLUSION

In this paper the idea of knowledge-based sampling has been presented, a generic technique of making rule selection metrics sensitive to prior knowledge. The interestingness of rules is often relative to a user's expectation or previously found patterns. A set of intuitive constraints has been proposed, that formalise how to construct samples which are independent of prior knowledge, so that subsequently applied rule induction techniques focus on novel patterns. The constraints have been shown to uniquely define a new distribution, which can easily be operationalised by either resampling or defining example weights.

Incorporating prior knowledge has been shown to be beneficial for the learning task of subgroup discovery. Evaluating rules globally results in overlapping patterns. To cover various aspects of a dataset it is more appropriate to construct sets of smaller rules, each of which captures a new pattern. Knowledge-based sampling is a way to shift the focus of subgroup discovery to undiscovered patterns, which allows to construct small sets of rules with high diversity. A new subgroup discovery technique based on stratification, iterative reweighting, and an arbitrary embedded rule learner has been presented. Experiments with six real world datasets indicate that the algorithm outperforms state of the art subgroup discovery algorithms which are based on alternative reweighting strategies.

[2]http://kodiak.cs.cornell.edu/kddcup/

[3]For SDRI$^*$ results are reported for the empirically best $\gamma$ from the candidate set $\{.1, .2, .3, .4, .5, .6, .7, .8, .9, .95\}$.

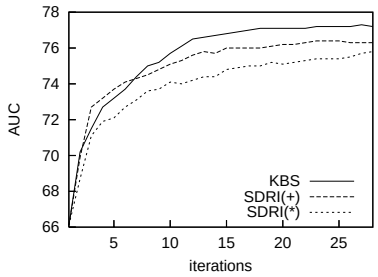


**Figure 2: KDD Cup**

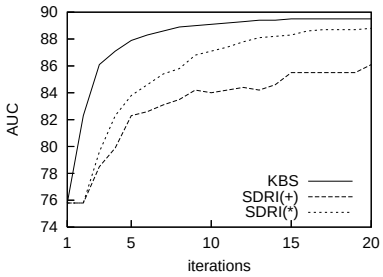


**Figure 3: Adult**

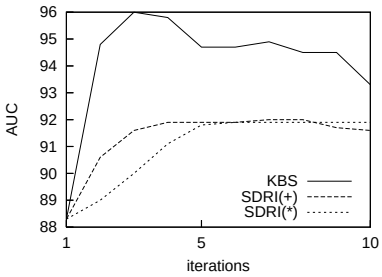


**Figure 4: Ionosphere**

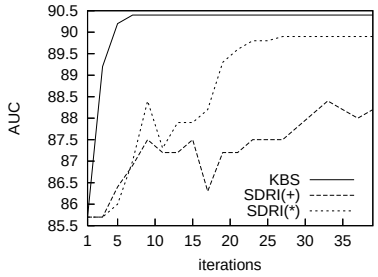


**Figure 5: Credit Domain**

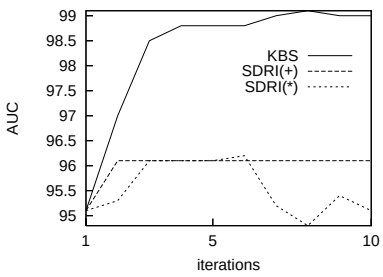


**Figure 6: Voting-Records**

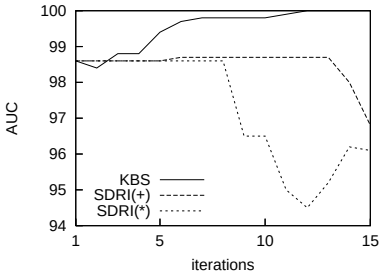


**Figure 7: Mushrooms**

| algorithm | n | auc | ± | cov | wracc | div |
|---|---|---|---|---|---|---|
| KBS | 15 | 76.8 | 1.2 | 38.6% | 0.023 | 0.972 |
| SDRI$^+$ | 15 | 76.0 | 1.9 | 50.5% | 0.054 | 0.932 |
| SDRI$^+$, RF | 12 | 74.3 | 2.0 | 50.0% | 0.056 | 0.928 |
| SDRI$^*$ | 15 | 74.8 | 2.1 | 42.7% | 0.071 | 0.917 |
| SDRI$^*$, RF | 8 | 74.2 | 2.1 | 44.7% | 0.074 | 0.914 |

KDD Cup

| algorithm | n | auc | ± | cov | wracc | div |
|---|---|---|---|---|---|---|
| KBS | 15 | 89.5 | 1.1 | 48.8% | 0.036 | 0.739 |
| SDRI$^+$ | 20 | 86.1 | 2.8 | 47.0% | 0.053 | 0.703 |
| SDRI$^+$, RF | 7 | 83.6 | 2.6 | 49.8% | 0.055 | 0.703 |
| SDRI$^*$ | 20 | 88.8 | 1.5 | 43.5% | 0.051 | 0.719 |
| SDRI$^*$, RF | 7 | 84.9 | 1.4 | 39.6% | 0.050 | 0.706 |

Adult

| algorithm | n | auc | ± | cov | wracc | div |
|---|---|---|---|---|---|---|
| KBS | 3 | 96.0 | 3.0 | 42.7% | 0.121 | 0.669 |
| SDRI$^+$ | 7 | 92.0 | 7.4 | 37.6% | 0.120 | 0.643 |
| SDRI$^+$, RF | 4 | 91.7 | 7.0 | 35.3% | 0.120 | 0.643 |
| SDRI$^*$ | 6 | 91.9 | 7.3 | 60.1% | 0.123 | 0.652 |
| SDRI$^*$, RF | 3 | 91.0 | 6.7 | 40.6% | 0.119 | 0.652 |

Ionosphere

| algorithm | n | auc | ± | cov | wracc | div |
|---|---|---|---|---|---|---|
| KBS | 7 | 90.4 | 3.4 | 42.2% | 0.057 | 0.893 |
| SDRI$^+$ | 31 | 88.4 | 4.2 | 56.8% | 0.156 | 0.796 |
| SDRI$^+$, RF | 3 | 87.0 | 5.3 | 66.9% | 0.139 | 0.668 |
| SDRI$^*$ | 27 | 89.9 | 4.0 | 55.8% | 0.164 | 0.739 |
| SDRI$^*$, RF | 2 | 85.7 | 5.3 | 66.9% | 0.139 | 0.668 |

Credit Domain

| algorithm | n | auc | ± | cov | wracc | div |
|---|---|---|---|---|---|---|
| KBS | 8 | 99.1 | 1.0 | 46.2% | 0.142 | 0.615 |
| SDRI$^+$ | 2 | 96.1 | 2.0 | 50.0% | 0.215 | 0.244 |
| SDRI$^+$, RF | 2 | 96.1 | 2.0 | 49.8% | 0.215 | 0.244 |
| SDRI$^*$ | 6 | 96.2 | 2.0 | 50.3% | 0.214 | 0.244 |
| SDRI$^*$, RF | 2 | 96.4 | 1.9 | 51.4% | 0.216 | 0.254 |

Voting-Records

| algorithm | n | auc | ± | cov | wracc | div |
|---|---|---|---|---|---|---|
| KBS | 12 | 100 | 0.0 | 69.5% | 0.086 | 0.843 |
| SDRI$^+$ | 6 | 98.7 | 0.2 | 43.4% | 0.195 | 0.470 |
| SDRI$^+$, RF | 1 | 98.6 | 0.1 | 43.4% | 0.195 | 0.470 |
| SDRI$^*$ | 1 | 98.6 | 0.1 | 43.4% | 0.195 | 0.470 |
| SDRI$^*$, RF | 1 | 98.6 | 0.1 | 43.4% | 0.195 | 0.470 |

Mushrooms

**Table 2: Performance of different subgroup discovery algorithms.**

## 8. ACKNOWLEDGEMENTS

Thanks to Timm Euler for carefully proof-reading some drafts, and to the anonymous reviewers for several useful hints.

## 9. REFERENCES

[1] C. Blake and C. Merz. UCI repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[3] S. Brin, R. Motwani, J. Ullman, and S. Tsur. Dynamic Itemset Counting and Implication Rules for Market Basket Data. In *Proceedings of ACM SIGMOD Conference on Management of Data (SIGMOD '97)*, pages 255–264, Tucson, AZ., 1997. ACM.

[4] P. Cunningham and J. Carney. Diversity versus Quality in Classification Ensembles Based on Feature Selection. In *Proceedings of the 11th European Conference on Machine Learning (ECML 2000)*, pages 109 – 116. Springer Verlag Berlin, Barcelona, Spain, 2000.

[5] T. Fawcett. ROC Graphs: Notes and Practical Considerations for Researchers, 2004. Submitted to Machine Learning.

[6] P. A. Flach. The Geometry of ROC Space: Understanding Machine Learning Metrics through ROC Isometrics. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. Morgen Kaufman, 2003.

[7] Y. Freund and R. R. Schapire. A decision–theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997.

[8] J. H. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, (28):337–374, 2000.

[9] J. Fürnkranz and P. Flach. ROC 'n' Rule Learning – Towards a Better Understanding of Covering Algorithms. *Machine Learning*, 58(1):39–77, 2005.

[10] S. Jaroszewicz and D. A. Simovici. Interestingness of Frequent Itemsets Using Bayesian Networks as Background Knowledge. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (KDD-2004)*. AAAI Press, August 2004.

[11] G. H. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann, 1995.

[12] W. Klösgen. Explora: A Multipattern and Multistrategy Discovery Assistant. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 3, pages 249–272. AAAI Press/The MIT Press, Menlo Park, California, 1996.

[13] N. Lavrac, P. Flach, and B. Zupan. Rule Evaluation Measures: A Unifying View. In *9th International Workshop on Inductive Logic Programming*, Lecture Notes in Computer Science. Springer, 1999.

[14] N. Lavrac, B. Kavsek, P. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, Feb 2004.

[15] N. Lavrac, F. Zelezny, and P. Flach. RSD: Relational subgroup discovery through first-order feature construction. In *12th International Conference on Inductive Logic Programming*. Springer, 2002.

[16] D. Mackay. Introduction To Monte Carlo Methods. In *Learning in Graphical Models*, pages 175–204. 1998.

[17] I. Mierswa, R. Klinkberg, S. Fischer, and O. Ritthoff. A Flexible Platform for Knowledge Discovery Experiments: YALE – Yet Another Learning Environment. In *LLWA 03 - Tagungsband der GI-Workshop-Woche Lernen - Lehren - Wissen - Adaptivität*, 2003.

[18] T. M. Mitchell. *Machine Learning*. McGraw Hill, New York, 1997.

[19] R. E. Schapire. The Strength of Weak Learnability. *Machine Learning*, 5:197–227, 1990.

[20] R. E. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating Prior Knowledge into Boosting. In *Proc. of the 19th International Conference on Machine Learning (ICML-02)*, 2002.

[21] R. E. Schapire and Y. Singer. Improved Boosting Using Confidence-rated Predictions. *Machine Learning*, 37(3):297–336, 1999.

[22] T. Scheffer and S. Wrobel. Finding the Most Interesting Patterns in a Database Quickly by Using Sequential Sampling. *Journal of Machine Learning Research*, 3:833–862, 2002.

[23] M. Scholz. Knowledge-Based Sampling for Subgroup Discovery. In K. Morik, J.-F. Boulicaut, and A. Siebes, editors, *Proc. of the Workshop on Detecting Local Patterns*, Lecture Notes in Computer Science. Springer, 2005. To appear.

[24] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, dec 1996.

[25] E. Suzuki. Discovering Interesting Exception Rules with Rule Pair. In *ECML/PKDD 2004 Workshop, Advances in Inductive Rule Learning*, 2004.

[26] I. Witten and E. Frank. *Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 2000.

[27] S. Wrobel. An Algorithm for Multi–relational Discovery of Subgroups. In J. Komorowski and J. Zytkow, editors, *Principles of Data Mining and Knowledge Discovery: First European Symposium (PKDD 97)*, pages 78–87, Berlin, New York, 1997. Springer.

[28] X. Wu and R. Srihari. Incorporating Prior Knowledge with Weighted Margin Support Vector Machines. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (KDD-2004)*. AAAI Press, August 2004.

[29] B. Zadrozny, J. Langford, and A. Naoki. Cost–Sensitive Learning by Cost–Proportionate Example Weighting. In *Proceedings of the 2003 IEEE International Conference on Data Mining (ICDM'03)*, 2003.