Knowledge-Based Sampling for Subgroup Discovery

Martin Scholz[1]

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

Abstract. Subgroup discovery aims at finding interesting subsets of a classified example set that deviates from the overall distribution. The search is guided by a so-called utility function, trading the size of subsets (coverage) against their statistical unusualness. By choosing the utility function accordingly, subgroup discovery is well suited to find interesting rules with much smaller coverage and bias than possible with standard classifier induction algorithms. Smaller subsets can be considered local patterns, but this work uses yet another definition: According to this definition global patterns consist of all patterns reflecting the prior knowledge available to a learner, including all previously found patterns. All further unexpected regularities in the data are referred to as local patterns. To address local pattern mining in this scenario, an extension of subgroup discovery by the knowledge-based sampling approach to iterative model refinement is presented. It is a general, cheap way of incorporating prior probabilistic knowledge in arbitrary form into Data Mining algorithms addressing supervised learning tasks.

1 Introduction

The discipline of Knowledge Discovery in Databases (KDD) is about finding useful and novel patterns, hidden in huge amounts of real-world data. A common problem is that the applied Data Mining techniques primarily find "obvious" patterns which are already known to domain experts. In this work we distinguish between global and local patterns, paying special attention to mining the local ones.

The notion of patterns is central to KDD. This work assumes that for a given target variable the absence of any pattern is equivalent to its independence of all the other variables. If prior knowledge is available then the absence of further patterns means that the prior knowledge models the distribution of the target variable precisely. In turn, a pattern is defined as a regular deviation from the independence assumption or given prior model, respectively. Thus, it shows as a correlation between the given target attribute and the other variables that has not been reported yet.

As a first idea the reader might want to think of global patterns as those discovered easily, e.g. because of having a high correlation to the target attribute in a densely populated subset of the instance space. Whether these patterns reflect prior domain knowledge or are the result of an earlier application of a Data Mining technique, in any case we might be interested in finding further patterns. From a technical point of view, the presence of some patterns may increase necessary efforts to observe others. Given a sample of limited size a less frequent pattern showing little effect on the target attribute may easily be considered to be part of another pattern. Due to a lack of significance it may also be hard to distinguish such patterns from random noise. To this end a specific sampling technique is proposed, paying special attention to patterns of lower frequency in subsequent Data Mining iterations.

Defining local patterns is possible based on the learner's prior knowledge: deviation from expectation indicates the presence of patterns not yet discovered. These patterns are referred to as local patterns. For simplicity this work confines itself to probabilistic rules as the representation language for patterns. Guiding the discovery of patterns by unexpectedness is close to the idea of subgroup discovery, a learning task discussed in section 3 after some necessary definitions are given in section 2. As the main contribution of this work a generic sampling technique to incorporate prior knowledge into subgroup discovery is presented in section 4 and empirically evaluated in section 5.

2 Basic Definitions

This section embeds the problem of mining local patterns into a formal Data Mining framework. The notion of a pattern as used in this work is in-line with the definition given in [9]: A pattern is characterised as a subset of the instance space with an anomalously high local density of data points. Local patterns are defined in terms of a (global) background model. Probabilistic rules, for simplicity always predicting the value of a boolean target attribute, are considered to be our target representation language for local patterns. A definition of this formalism is given in subsection 2.2 after some more basic definitions.

2.1 Instance Space and Distribution

The two learning tasks discussed in this paper are subgroup discovery and classifier induction. Both tasks are supervised, so the learning step is performed based on a sample of classified examples. *Examples* are defined as elements of an *instance space* $\mathbf{X}$. Usually the instance space $\mathbf{X} = A_1 \times A_2 \times \ldots \times A_k$ is the Cartesian product of a fixed set of nominal and/or numerical attributes. A set of examples $E \subset \mathbf{X}$ can be considered to be the extension of a single table of a relational database. To simplify formal aspects, $\mathbf{X}$ is assumed to be finite in this work. All results are easily generalised to the case of continuous domains.

Examples are assumed to be sampled i.i.d with respect to a *distribution* $D : \mathbf{X} \rightarrow [0, 1]$. The probability to observe an instance $x \in \mathbf{X}$ under D is denoted as $P_{x\sim D}(x)$. The probability to observe an instance from a subset $W \subseteq \mathbf{X}$ is denoted as $P_D(W)$. If the underlying distribution is clear from the context we omit the subscripts.

Each example is assigned a label from the set $\mathbf{Y}$ of all possible labels by the *target function* $C : \mathbf{X} \to \mathbf{Y}$. We assume C to be fixed but unknown to the learner, whose task is to approximate it in a specified way. This work considers only supervised learning with a boolean target attribute $\mathbf{Y} = \{0, 1\}$.

2.2 Probabilistic Rules for Knowledge Representation

Encoding prior knowledge[1] is often done using any form of rules. For subgroup discovery Horn logic rules are the main representation language.

Definition 1. *A Horn logic rule consists of a body A, which is a conjunction of atoms over $A_1, \ldots, A_k$, 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 evaluates to true, also, it is called correct.*

More generally a rule can be considered as a function $h : \mathbf{X} \to \mathbf{Y}$, assigning a prediction to each $x \in \mathbf{X}$.

For now we assume any form of prior knowledge to be represented by rules of this form. In subsection 4.3 we will see that the presented approach can easily be extended to incorporate any form of prior knowledge predicting the conditional distribution of the target variable.

Assuming $\mathbf{Y} = \{0, 1\}$, the following abbreviations are used:

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

Using this notation, the Horn logic rules predicting a boolean target are of the form $(h \to Y_+)$ and $(h \to Y_-)$. Unlike for any strictly logical interpretation, rules are not expected to match the data exactly. Often it is sufficient if they point to regularities in the data. The intended semantics of a *probabilistic rule* is that the conditional probability $P(Y_+ \mid h)$ (or $P(Y_- \mid h)$) is higher than the class prior $P(Y_+)$ (or $P(Y_-)$). Probabilistic rules are often annotated by their corresponding conditional probabilities:

$$h \to Y_+ \ \ [0.8] \qquad :\Leftrightarrow \qquad P_{x \sim D}(C(x) = 1 \mid h(x) = 1) = 0.8$$

2.3 Performance metrics

As a general task in supervised learning we want to estimate conditional probabilities of target attributes. Different performance metrics help to evaluate how useful and interesting single rules are. For the notion of interestingness different

[1] The term “prior knowledge” will be preferred to “background knowledge”, because the latter is associated with precise knowledge for inference, while *prior knowledge* suggests a more probabilistic view.

formalisations have been proposed in the literature (e.g.[22]). In this work interestingness is considered equal to unexpectedness. This subsection collects some 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. *For a rule* $(A \rightarrow B)$ *the* accuracy *is defined as*

$$\mathbf{Acc}(A \rightarrow B) := P(A \cap B) + P(\overline{A} \cap \overline{B})$$

Definition 3. *The* precision *of a rule reflects the conditional probability that it is correct, given that it is applicable:*

$$\mathbf{Prec}(A \rightarrow B) := P(B \mid A)$$

Subgroup discovery focuses on rules covering subsets that – compared to the overall distribution – are biased in the data. The following metric has been used to measure interest in the domain of frequent itemset mining [4]. In the supervised context it measures the change in the target attribute's frequency for the subset covered by a rule.

Definition 4. *For any rule* $(A \rightarrow B)$ *the* **Lift** *is defined as*

$$\mathbf{Lift}(A \rightarrow B) := \frac{P(A \cap B)}{P(A)P(B)} = \frac{P(B \mid A)}{P(B)} = \frac{\mathbf{Prec}(A \rightarrow B)}{P(B)}$$

The **Lift** of a rule captures the value of "knowing" the prediction for estimating the probability of the target attribute. $\mathbf{Lift}(A \rightarrow B) = 1$ indicates that A and B are independent events. With $\mathbf{Lift}(A \rightarrow B) > 1$ the conditional probability of B given A increases, with $\mathbf{Lift}(A \rightarrow B) < 1$ it decreases.

During subgroup discovery rules are evaluated by a utility function. A popular function is the following one, e.g. available in EXPLORA [11]:

Definition 5. *The* weighted relative accuracy *(***WRAcc***) of a rule* $(A \rightarrow B)$ *multiplies coverage* $P(A)$ *and bias* $P(B \mid A) - P(B)$*:*

$$\mathbf{WRAcc}(A \rightarrow B) := P(A) \cdot (P(B \mid A) - P(B))$$

The use of **WRAcc** as a measure for rule interestingness has been motivated elaborately in [14]. It is similar to the binomial test function, thus favours significant rules, but puts more emphasis on coverage [11]. Many other functions have been suggested in the literature [24, 11], basically putting more emphasis on either coverage or bias.

3 Subgroup Discovery

Subgroup discovery aims at finding interesting subsets of the instance space that deviate from the overall distribution. The search is guided by a *utility function* that allows to find interesting rules with much smaller coverage and bias than possible with standard classifier induction algorithms. Subsection 3.1 briefly describes related work in subgroup discovery. How interesting rules interact, how to recognise redundant rules, and how to build single predictors from rulesets is discussed in 3.2. In subsection 3.3 incorporation of prior knowledge as a means to improve utility and diversity of the discovered rulesets is motivated. Subsection 3.4 shows a generic way of addressing subgroup discovery tasks using classifier induction algorithms.

3.1 Existing Approaches

The goal of subgroup discovery is to find interesting and novel patterns in datasets. Utility functions formalise a trade-off between the size of the subgroup and the unusualness in terms of a target attribute's observed frequency. There are two different strategies of searching for interesting rules: exhaustive and heuristic search.

MIDOS [24] and EXPLORA [11] 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. For the special case of exception rules similar exhaustive search strategies exists [**?**]. Finding subgroups on subsamples of the original data is a straightforward method to speed up the search process. As shown in [20, 21] 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 evaluation measure for rule candidates to **WRAcc** the well known CN2 classifier has been turned into CN2-SD [13]. 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.

For pruning rulesets ROC analysis was suggested in [13]. 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 [3].

3.2 Combining rules

There are different methods to combine a set of rules predicting the conditional probability of a target class. The approach put forward in this work is useful for descriptive and predictive settings, and it can be used to combine arbitrary predictors, especially rules represented in Horn logic. If the prediction of each rule is used to define a new attribute, then predictions can be combined by means of classifier induction techniques. The underlying assumption of Naïve Bayes [10] 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 $\{h_i : \mathbf{X} \to \mathbf{Y} \mid 1 \leq i \leq n\}$ denote a set of rules. Then for any given example $x \in \mathbf{X}$, labels $y_1, \ldots, y_n \in \mathbf{Y}$, and $h_1(x) = y_1, \ldots, h_n(x) = y_n$, the Naïve Bayes classifier estimates

$$
\begin{aligned}
& P(C(x) = y \mid h_1(x) = y_1, \ldots h_n(x) = y_n) \\
= & \frac{P(h_1(x) = y_1, \ldots, h_n(x) = y_n \mid C(x) = y) \cdot P(C(x) = y)}{P(h_1(x) = y_1, \ldots, h_n(x) = y_n)} \\
\approx & \frac{P(C(x) = y)}{P(h_1(x) = y_1, \ldots h_n(x) = y_n)} \prod_{1 \leq i \leq n} P(h_i(x) = y \mid C(x) = y) \\
= & \frac{P(C(x) = y) \prod_i P(h_i(x) = y_i)}{P(h_1(x) = y_1, \ldots, h_n(x) = y_n)} \prod_{1 \leq i \leq n} \frac{P(C(x) = y \mid h_i(x) = y)}{P(C(x) = y)} \\
= & \frac{P(C(x) = y) \prod_i P(h_i(x) = y_i)}{P(h_1(x) = y_1, \ldots, h_n(x) = y_n)} \prod_{1 \leq i \leq n} \mathbf{Lift}((h_i(x) = y_i) \to (C(x) = y))
\end{aligned}
$$

for each class $y \in \mathbf{Y}$. Especially for boolean $\mathbf{Y}$ it is easier to consider the ratios

$$
\begin{aligned}
\alpha(x) := & \frac{P(Y_+ \mid h_1(x) = y_1, \ldots, h_n(x) = y_n)}{P(Y_- \mid h_1(x) = y_1, \ldots, h_n(x) = y_n)} \\
= & \frac{P(Y_+)}{P(Y_-)} \prod_{1 \leq i \leq n} \frac{\mathbf{Lift}((h_i(x) = y_i) \to Y_+)}{\mathbf{Lift}((h_i(x) = y_i) \to Y_-)},
\end{aligned}
\tag{1}
$$

as most of the terms cancel out, but we can still recalculate

$$
P(Y_+ \mid h_1(x) = y_1, \ldots, h_n(x) = y_n) = \frac{\alpha(x)}{1 + \alpha(x)}
$$

based on formula (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. It is not necessary to restrict rules to the case in which the body evaluates to true. 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. 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, we have 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 each rule can be expressed relative to prior knowledge, e.g. of preceding rules. The following equation illustrates this idea for the simplified case of two rules and the subset $h_1 \cap h_2 \subset \mathbf{X}$:

$$\begin{aligned}\mathbf{Lift}((h_1, h_2) \rightarrow Y_+) &= \frac{P(h_1, h_2 \mid Y_+)}{P(h_1, h_2)} = \frac{P(h_1 \mid Y_+) \cdot P(h_2 \mid h_1, Y_+)}{P(h_1) \cdot P(h_2 \mid h_1)} \\ &= \mathbf{Lift}(h_1 \rightarrow Y_+) \cdot \underbrace{\frac{\mathbf{Lift}(h_2 \rightarrow (h_1, Y_+))}{\mathbf{Lift}(h_2 \rightarrow h_1)}}_{=:\mathbf{Lift}(h_2 \rightarrow Y_+ \mid h_1)}\end{aligned}$$

The term $\mathbf{Lift}(h_2 \rightarrow Y_+ \mid h_1)$ can be regarded as the *relative* **Lift** of the rule $h_2 \rightarrow Y_+$ with respect to prior knowledge. It replaces $\mathbf{Lift}(h_2 \rightarrow Y_+)$ when estimating $\alpha(x)$ in formula (1) given $h_1 \rightarrow Y_+$. Applying the sampling technique introduced in section 4, rules with high relative performance are favoured. This usually results in rulesets with low redundancy and high diversity.

3.3 Iterative Subgroup Discovery

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, although this would improve the score assigned by the utility function. The syntactical bias is important, however, because we want the results to be understandable, and because it is the main reason for *diversity* within the n best subgroups. Without any syntactical restrictions the second best subgroup would usually be the best one after adding or removing a single example. The syntactical bias might not be sufficient to avoid sets of similar rules. Redundancy filters are a common technique to overcome this problem. 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.

Formalised prior knowledge like previously found patterns could help to refine existing utility measures. Two different approaches to exploit prior knowledge in the scope of subgroup discovery have been suggested so far. The first one is to prune rules violating a redundancy constraint [11]. This is possible during search, or as a post-processing step to present only the most interesting rules. With the ILP system RSD [12] another way of incorporating background knowledge has been proposed. It uses background knowledge to propositionalise relational data. For the learning step itself CN2-SD is used.

One of the advantages of the approach presented here is that it allows to turn any algorithm for training classifiers in the presence of noise into one for subgroup discovery with utility function **WRAcc** that can exploit prior knowledge. The next subsection shows a generic way to transform subgroup discovery tasks into classifier induction tasks, before a generic way to incorporate prior knowledge into supervised Data Mining is introduced in section 4.

3.4 Subgroup Discovery by Classifier Induction

This subsection briefly discusses the relation between subgroup discovery with utility function **WRAcc** and the task of classifier induction.

The goal of classifier induction is to select a predictive model that separates positive and negative examples with high predictive accuracy. Many algorithms and implementation exists for this purpose [17, 23], basically differing in the set of models (hypothesis space H) and search strategies. Subgroup discovery is also a supervised learning task. Examples are classified with respect to a "property of interest". The overall goal is to find understandable and interesting rules, which is hard to be formalised. Thus, the process of model selection is guided by a utility function. In the following definition subgroup discovery is reduced to finding a single rule, only.

Definition 6. *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*

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

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

$$h^* := maxarg_{h \in H}\ q(h).$$

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 a class is observed with a frequency that is significantly higher than in the overall population. In cases of skewed class distributions the frequency in the covered subset might still be far below 50% for the most interesting rules. 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 7. *For* $D : \mathbf{X} \to [0,1]$, $C : \mathbf{X} \to \mathbf{Y}$ *let the* stratified random sample distribution $D\prime$ *of* D *(and* C*) be defined by*

$$P_{x\sim D\prime}(x) := \frac{P_{x\sim D}(x)}{|Y| \cdot P_{z\sim D}(C(z) = C(x))} = P_D(x) / \begin{cases} 2P_D(Y_+), \text{ for } C(x) = 1 \\ 2P_D(Y_-), \text{ for } C(x) = 0 \end{cases}$$

$D\prime$ is defined by rescaling D so that the class priors are equal.

Theorem 1. *For every rule* $h \to Y_+$ *the following equalities hold if* $D\prime$ *is the stratified random sample distribution of* D*:*

$$\begin{aligned} \mathbf{Acc}_{D\prime}(h \to Y_+) &= 2\mathbf{WRAcc}_{D\prime}(h \to Y_+) + 1/2 \\ &= \mathbf{WRAcc}_D(h \to Y_+) \cdot \underbrace{\frac{1}{2P_D(Y_+) \cdot P_D(Y_-)}}_{\textit{irrelevant for ranking rules}} + 1/2 \end{aligned}$$

Theorem 1 indicates that subgroup discovery tasks with utility function **WRAcc** can as well be solved by rule induction algorithms optimising predictive accuracy after a step of stratified resampling. A proof is given in the appendix. Further interesting relations between performance metrics are proven in [8].

4 Knowledge-Based Sampling

Before introducing techniques for sampling with respect to prior knowledge the task is formalised by a set of constraints.

4.1 Constraints for resampling

After a first rough analysis has discovered global patterns we want to prepare a second iteration of Data Mining to find local patterns. The proposed idea is to construct samples that do not show the *biases* underlying previously discovered patterns, while taking care that all the remaining patterns remain intact.

Practically, for a given rule $R : h \to Y_+$ this means to consider a new distribution $D\prime$, as close to the original function D as possible. This is formalised by the following set of constraints. First of all, we want to remove the bias corresponding to R. In other words we want h and Y_+ to be independent:

$$P_{D\prime}(Y_+ \mid h) = P_{D\prime}(Y_+) \tag{2}$$

Next, we do not want the priors of h and Y_+ to change:

$$P_{D\prime}(h) = P_D(h) \tag{3}$$
$$P_{D\prime}(Y_+) = P_D(Y_+) \tag{4}$$

Finally, within each partition sharing the same class and prediction of R the new distribution is defined proportionally to the initial one:

$$P_{D'}(x \mid h \cap Y_+) = P_D(x \mid h \cap Y_+) \tag{5}$$
$$P_{D'}(x \mid h \cap Y_-) = P_D(x \mid h \cap Y_-) \tag{6}$$
$$P_{D'}(x \mid \overline{h} \cap Y_+) = P_D(x \mid \overline{h} \cap Y_+) \tag{7}$$
$$P_{D'}(x \mid \overline{h} \cap Y_-) = P_D(x \mid \overline{h} \cap Y_-) \tag{8}$$

Given a database and a global pattern R we can apply any Data Mining technique after sampling with respect to D'. This might ease the detection of further patterns. An advantage of mining the resampled data rather than a dataset without the covered examples shows, if there are further patterns within the covered subset. These patterns can still be observed after resampling, just rescaled proportionally. This helps to find exceptions to successful rules, as motivated in subsection 3.3, or patterns overlapping in some other way.

Please note, that a subgroup pattern showing in the new sample may be interesting relative to the prior knowledge, only. Let

$$P(Y_+ \mid A) = P(Y_+) = 0.5 \quad \text{for a rule} \quad A \to Y_+.$$

$\mathbf{Y}$ 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:

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

This rule predicts a higher conditional probability of Y_+ given B. In this context the rule $(A \to Y_+)$ becomes interesting as an exception to the prior knowledge, because we would rather expect $P(Y_+ \mid A) = P(Y_+ \mid B)$. The reason is that the prediction for $B \subset X$ is more specific than the general class priors. In general switching from the initial distribution to the resampled data is a step of applying prior knowledge by means of sampling. This step allows to find overlapping and nested patterns sequentially.

4.2 Constructing a new distribution function

In subsection 4.1 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 (2) to (8) induce a unique target distribution.

Definition 8. *The lift of an example $x \in \mathbf{X}$ for a rule $(h \to Y_+)$ is defined as*

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

Theorem 2. *For any initial distribution D and given rule R the probability distribution $D\prime$ is induced uniquely by the constraints (2) to (8) as follows:*

$$P_{D\prime}(x) := P_D(x) \cdot (\mathbf{Lift}_D(R, x))^{-1}$$

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

$$\begin{aligned}
(\forall x \in h \cap Y_+) : P_{D\prime}(x) &= P_{D\prime}(x \mid h \cap Y_+) \cdot P_{D\prime}(h \cap Y_+) \\
&= P_D(x \mid h \cap Y_+) \cdot P_{D\prime}(h) \cdot P_{D\prime}(Y_+) \\
&= \frac{P_D(x)}{P_D(h \cap Y_+)} \cdot P_D(h) \cdot P_D(Y_+) \\
&= P_D(x) \cdot (\mathbf{Lift}_D(h \to Y_+))^{-1}
\end{aligned}$$

The other three partitions can be rewritten analogously. On the other hand, it can easily be validated that $D\prime$ as defined by theorem 2 is in fact a distribution satisfying constraints (2) to (8):

$$P_{D\prime}(h \cap Y_+) = P_D(h \cap Y_+) \cdot (\mathbf{Lift}_D(R, x))^{-1} = P_D(h) \cdot P_D(Y_+)$$

and analogously for the other partitions. This directly implies constraints (2) to (4) by marginalising out. Constraints (5) to (8) are met, because for all four partitions $D\prime$ is defined proportionally to D. This implies that the conditional probabilities given the partitions are equivalent.

4.3 Weighting examples using prior knowledge

In the last subsection it was discussed how to alter an initial distribution in the presence of prior knowledge. The goal is to construct samples not reflecting previously found patterns anymore. This idea stems from boosting classifiers, which was also first introduced in terms of altering an initial distribution function and a corresponding sampling technique [18]. The idea of boosting is to repeatedly apply a "weak" base learner and to combine the predictions. The probabilities of examples are adjusted in such a way that in later iterations the weak learner has to focus on the "hard" examples not yet covered sufficiently by the ensemble of base classifiers.

As a general alternative to resampling it is possible to assign weights to examples, reflecting a change in the underlying distribution. This method is common in boosting literature to avoid resampling [6, **?**,19]. It can be understood in terms of importance sampling [15]: The example set is assumed to be drawn independently from an initial distribution D. Then each example x is assigned the weight $D\prime(x)/D(x)$ rather than sampling directly with respect to $D\prime$, which may be infeasible.

For subgroup discovery the use of weighted examples may be less appropriate, as even uniformly distributed subsets may be represented as a single example with high weight. On the other hand, for given example weights resampling can easily be performed by a Monte Carlo technique called rejection sampling [15]. A straight-forward implementation of this technique has successfully been applied to cost-sensitive learning [25], which is very similar from a technical point of view. In this subsection a knowledge-based weighting scheme is introduced. It can replace resampling if all subsequently applied algorithms are capable of using example weights, and if it meets the requirements of the learning task. In other cases it can still be used as a basis for rejection sampling.

Theorem 2 defines a new distribution to sample from, given a single rule R as prior knowledge. The following strategy for weighting examples is more general. First of all the number of classes $|\mathbf{Y}|$ is not restricted to two. As a second generalisation the prior knowledge θ may be of arbitrary form. It is assumed to be associated to a function

$$\hat{P}(x, y, \theta) = \hat{P}(C(x) = y \mid x, \theta) \approx P(C(x) = y \mid x)$$

estimating probabilities for each $\langle x, y\rangle \in \mathbf{X} \times \mathbf{Y}$. Assuming the class priors $P(C(x) = y)$ 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

$$\widehat{\mathbf{Lift}}(x, \theta) := \frac{\hat{P}(x, C(x), \theta)}{P_{z \sim D}(C(z) = C(x))}$$

Given a procedure for sampling examples $x \sim D$ independently, the following distribution generalising theorem 2 can be used for weighting each example:

$$P_{D'}(x) := P_D(x) \cdot (\widehat{\mathbf{Lift}}(x, \theta))^{-1} \tag{9}$$

To remove prior probabilistic knowledge from a data stream applying formula (9) it is sufficient to assign each example x from the stream a weight of $\widehat{\mathbf{Lift}}(x, \theta)^{-1}$, as the factor $D(x)$ is already accounted for by sampling with respect to D.

5 Experiments

The proposed idea of subgroup discovery utilising all forms of previously discovered patterns has been evaluated on three datasets from the UCI Machine Learning Library [2] and a sample of the KDD Cup 2004 Quantum Physics dataset[2]. For simplicity attributes with missing values have been discarded. All datasets have boolean target attributes. Further characteristics are listed in table 1.

Three subgroup discovery algorithms have been integrated into the learning environment YALE [16]. For mining subgroup rules from samples the embedded WEKA [23] rule induction algorithm has been applied to stratified samples,

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

Dataset	Examples	# Nominal Attr.	# Numerical Attr.	Minority class
Quantum Physics	10.000	–	71	50.0%
Ionosphere	351	–	34	35.8%
Credit Domain	690	6	9	44.5%
Mushrooms	8.124	22	–	48.2%

Table 1. Datasets used for experimental evaluation.

which is valid due to theorem 1. The algorithm ConjunctiveRule heuristically selects a single Horn logic rule with high predictive accuracy, which translates into high **WRAcc**. It is applied repeatedly by the subgroup discovery algorithms. The *knowledge-based sampling algorithm* (KBS) applies sampling as presented in section 4. Rules are combined as discussed in subsection 3.2, similar to the Naïve Bayes method. KBS is compared to two other reweighting strategies reported in the subgroup discovery literature [12]. After a positive example e has been covered by i rules its new weight is computed as

$$w_i(e) := \frac{1}{i+1} \text{ (additive), or } w_i(e) := \gamma^i \text{ for given } \gamma \in [0, 1] \text{ (multiplicative).}$$

Accordingly, two versions of subgroup discovery ruleset induction (SDRI) have been implemented, which are similar to CN2-SD. The variant that applies ConjunctiveRule on stratified samples after *additive* reweighting is referred to as SDRI^+, the one with *multiplicative* reweighting as SDRI^*. Reweighting is performed iteratively. 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. The rulesets constructed by SDRI are combined to a single probabilistic prediction as in CN2-SD: The predicted target class distributions of all applicable rules are averaged.

The goal of subgroup discovery in this setting is to find a small set of (understandable) rules, giving a good picture of the data. In more formal terms the probabilistic classifiers built from the rulesets should be accurate. This property is measured by the area under the ROC curve metric (AUC) [5].

Figure 1 to 4 show how the AUC metric changes with an increasing number of rules. All values have been estimated by 10fold cross-validation. The default for the parameter γ of SDRI^* was set to 0.9 as suggested in [12]. For all but the mushrooms dataset this value gave best results[3]. For mushrooms the results for the better value $\gamma = 0.7$ are reported. For a higher value of γ it generally took more iterations to reach a similar AUC performance, for lower values the algorithm converged more quickly, but reached worse results.

In all figures the KBS algorithm outperforms SDRI with both reweighting strategies, while none of the SDRI variants is clearly superior to the other one. In figure 1 all three algorithms manage to find useful rules repeatedly. SDRI^+ performs best for sets of 3 to 6 rules. For larger rulesets KBS is superior. SDRI^* performs worst. Figure 2 shows the performance for a smaller

[3] The parameter was empirically decreased in steps of 0.1 and increased to 0.95.

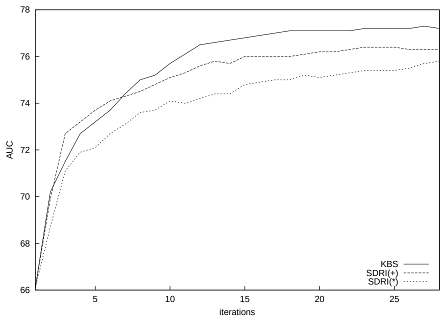

Fig. 1. Quantum Physics Data

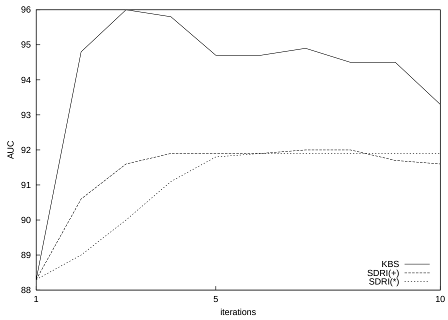

Fig. 2. Ionosphere

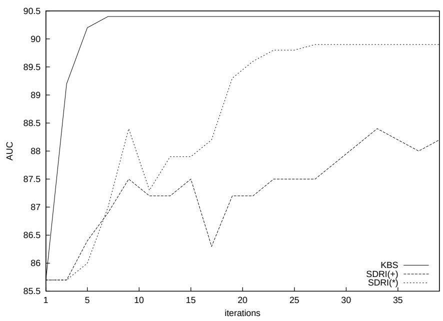

Fig. 3. Credit Domain

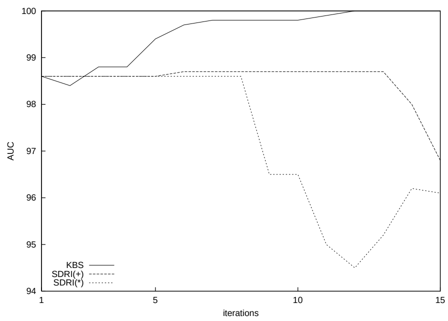

Fig. 4. Mushrooms

dataset. Again KBS performs best, although it overfits after the 3rd iteration. The SDRI variants reach their maxima later. This delay is even more significant for the credit domain data, illustrated in figure 3. After iteration 7 the predictions of KBS remain constant, while the AUC values of the SDRI rulesets improve non-monotonically and are still significantly worse after 40 iterations. Finally, in the experiment shown in figure 4 KBS reaches 100% AUC with just 12 rules, while SDRI does not manage to improve over the performance of the first rule at all. As a further experiment on this dataset AdaBoost has been run on top of ConjunctiveRule. After 15 iterations it still has an error rate of about 2.5%.

Table 2 lists the average performance of rulesets. For the ionosphere and credit domain dataset the number of rules with best performance regarding AUC was chosen. For the KDD Cup data (Quantum Physics) the number of rules was set to 15. The ROC filter for rulesets discussed in subsection 3.2 was applied to both SDRI variants, denoted as RF in table 2. As mentioned in subsection 4.1 some patterns are interesting relative to prior knowledge, only. The columns **AvgCov** and **AvgWRAcc** in table 2 demonstrate that absolute values of performance metrics may be misleading regarding how well rules are suited to predict a target class. **AvgCov** denotes the average coverage of rules,

Dataset	**Algorithm**	**# Rules**	**AUC**	**AvgCov**	**AvgWRAcc**
Ionosphere	KBS	3	96.0 ($\pm$ 3.0)	42.7%	0.121
Ionosphere	SDRI$^+$	7	92.0 ($\pm$ 7.4)	37.6%	0.120
Ionosphere	SDRI$^+$, RF	4	91.7 ($\pm$ 7.0)	35.3%	0.120
Ionosphere	SDRI*	6	91.9 ($\pm$ 7.3)	60.1%	0.123
Ionosphere	SDRI*, RF	3	91.0 ($\pm$ 6.7)	40.6%	0.119
Credit Domain	KBS	7	90.4 ($\pm$ 3.4)	42.2%	0.057
Credit Domain	SDRI$^+$	31	88.4 ($\pm$ 4.2)	56.8%	0.156
Credit Domain	SDRI$^+$, RF	3	87.0 ($\pm$ 5.3)	66.9%	0.139
Credit Domain	SDRI*	27	89.9 ($\pm$ 4.0)	55.8%	0.164
Credit Domain	SDRI*, RF	2	85.7 ($\pm$ 5.3)	66.9%	0.139
Quantum Physics	KBS	15	76.8 ($\pm$ 1.2)	38.6%	0.023
Quantum Physics	SDRI$^+$	15	76.0 ($\pm$ 1.9)	50.5%	0.054
Quantum Physics	SDRI$^+$, RF	12	74.3 ($\pm$ 2.0)	50.0%	0.056
Quantum Physics	SDRI*	15	74.8 ($\pm$ 2.1)	42.7%	0.071
Quantum Physics	SDRI*, RF	8	74.2 ($\pm$ 2.1)	44.7%	0.074

Table 2. Performance values for different subgroup algorithms.

AvgWRAcc the average weighted relative accuracy. Global evaluation rewards overlapping rules for reporting the same pattern multiple times, while rules capturing smaller patterns not covered by any other rule may perform worse if evaluated stand-alone. This explains why both the average absolute coverage and absolute **WRAcc** of KBS is lower for two of the three datasets than the corresponding values of SDRI, but the AUC values are still higher. The ROC filter generally seems to neither improve the AUC score nor the average global utility function. In most cases it prunes the ruleset at the price of a reduced performance. Increasing coverage is comparably trivial.

As an overall result the experiments show that knowledge-based sampling helps to shift the focus of subgroup discovery to yet undiscovered patterns, allowing to find a small number of rules that help to build accurate probabilistic classifiers. Rulesets with higher average values of utility functions that were not constructed to maximise diversity turn out to be less accurate.

6 Conclusion

In this work local pattern mining was defined in terms of prior knowledge available to a learner. Subgroup discovery was identified as a matching learning task, but the available algorithms do not incorporate previously discovered patterns and prior domain knowledge into their utility functions. In section 4 a generic way of incorporating prior knowledge by means of sampling was presented. The selected samples do no longer reflect the prior knowledge and can be used to mine further local patterns. Applying the utility function to such a sample means not to reward rules for overlapping with previously known biased subsets, but to rank rules by their new own contribution. This helps to focus on rulesets that are almost orthogonal, thus the conditional independence assumption is not as unrealistic as in general. As a consequence, rules predicting the conditional probabilities of a target attribute can well be combined by the Naïve Bayes strategy. The simplicity of the reweighting scheme allows to interpret the found patterns either globally or in their specific context, based on the intuitive **Lift** measure. To simplify subgroup discovery it was shown how to address pattern mining with utility function **WRAcc** with common rule induction algorithms. The developed subgroup discovery algorithm has been validated experimentally and shown to outperform existing reweighting and rule combination strategies in the scope of subgroup discovery.

References

1. http://kodiak.cs.cornell.edu/kddcup/
2. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
3. L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
4. 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, 1997.
5. T. Fawcett. ROC Graphs: Notes and Practical Considerations for Researchers, 2004. submitted to Machine Learning.
6. Y. Freund and 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.
7. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. Technical report, Departement of Statistics, Stanford University, Stanford, California 94305, July, 23 1998.
8. J. Fürnkranz and P. Flach. An Analysis of Rule Evaluation Metrics. In *Proc. of the 20th International Conference on Machine Learning*. Morgan Kaufman, 2003.
9. D. Hand. Pattern detection and discovery. In D. Hand, N. Adams, and R. Bolton, editors, *Pattern Detection and Discovery*. Springer, 2002.
10. G. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *Proc. of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1995.
11. W. Klösgen. Explora: A multipattern and multistrategy discovery assistant. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 3. AAAI/MIT Press, 1996.
12. 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.
13. N. Lavrac, P. Flach, B. Kavsek, and L. Todorovski. Rule Induction for Subgroup Discovery with CN2-SD. In D. Mladenic M. Bohanec and N. Lavrac, editors, *2nd Int. Workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and MetaLearning*, August 2002.
14. 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.
15. D.J.C. Mackay. Introduction To Monte Carlo Methods. In *Learning in Graphical Models*, pages 175–204. 1998.

16. I. Mierswa, R. Klinkenberg, 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.
17. T. M. Mitchell. *Machine Learning*. McGraw Hill, New York, 1997.
18. R. E. Schapire. The Strength of Weak Learnability. *Machine Learning*, 5:197–227, 1990.
19. R. E. Schapire and Y. Singer. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.
20. T. Scheffer and S. Wrobel. A Sequential Sampling Algorithm for a General Class of Utility Criteria. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2000.
21. 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.
22. 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.
23. I. Witten and E. Frank. *Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 2000.
24. 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, 1997. Springer.
25. 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.

APPENDIX

We repeat the definition of the two tasks, substituting C for Y_+ (or Y_-) and $\overline{C}$ for Y_- (or Y_+). H denotes a set of valid Horn logic rules with head C.

Classification Find an $h \in H$ maximising *predictive accuracy*:

$$\mathbf{Acc}(h \to C) = P(h \cap C) + P(\overline{h} \cap \overline{C})$$

Subgroup Discovery with WRAcc Find an $h \in H$ maximising

$$\mathbf{WRAcc}(h \to C) = P(h) \cdot (P(C|h) - P(C))$$

The correctness of the theorem is shown using two lemmas.

Lemma 1. *The two tasks are equivalent, if and only if the priors of both class labels are equal:*

$$P(C) = P(\overline{C}) = 1/2$$

Proof. First we rewrite predictive accuracy:

$$\begin{aligned} \mathbf{Acc}(h \to C) &= P(h \cap C) + P(\overline{h} \cap \overline{C}) = P(h \cap C) + \left(P(\overline{h}) - P(\overline{h} \cap C)\right) \\ &= P(h \cap C) + P(\overline{h}) - (P(C) - P(h \cap C)) = 2P(h \cap C) + P(\overline{h}) - P(C) \\ &= 2P(C|h)P(h) + 1 - P(h) - P(C) = 2P(h)\,(P(C|h) - 1/2) + P(\overline{C}) \end{aligned} \tag{10}$$

The order of rules according to this metric does not change if we drop the constant additive terms $P(\overline{C})$ and the constant factor of 2 in formula (10), so

$$argmax_{h\in H}\mathbf{Acc}(h \to C) = argmax_{h\in H}\left(P(h)\cdot(P(C|h)-1/2)\right))$$

Obviously the second term is equivalent to **WRAcc** if and only if $P(C) = 1/2$. In this case **Acc** and **WRAcc** induce the same ranking of rules.

If the condition of lemma 1 is violated for the original distribution D we can perform stratified sampling using definition 7:

$$P_{x\sim D\prime}(x) := \frac{P_{x\sim D}(x)}{2P_{z\sim D}(C(z) = C(x))} \tag{11}$$

Considering a sample from $D\prime$ as defined by (11) we expect $P_{D\prime}(h)$ and $P_{D\prime}(C|h)$ to differ from $P_D(h)$ and $P_D(C|h)$, respectively. As the following lemma states such samples are nevertheless appropriate for rule selection.

Lemma 2. *The order of a ruleset H induced by the WRAcc metric is equivalent for any two distributions D and $D\prime$, as long as formula (11) holds.*

Proof. Let us first rewrite $P_{D\prime}(h)$ in terms of D:

$$\begin{aligned}P_{D\prime}(h) &= \frac{P_D(h\cap C)}{2P_D(C)} + \frac{P_D(h\cap \overline{C})}{2P_D(\overline{C})} = \frac{P_D(h)}{2}\left(\frac{P_D(h\cap C)}{P_D(h)P_D(C)} + \frac{P_D(h\cap \overline{C})}{P_D(h)P_D(\overline{C})}\right)\\ &= P_D(h)\cdot\underbrace{\frac{1}{2}\left(\mathbf{Lift}_D(h\to C)+\mathbf{Lift}_D(h\to\overline{C})\right)}_{=:\alpha}\end{aligned} \tag{12}$$

Having $P_{D\prime}(h) = P_D(h)\cdot\alpha$ allows to reformulate $\mathbf{WRAcc}_{D\prime}$ like this:

$$\begin{aligned}&\mathbf{WRAcc}_{D\prime}(h\to C) = P_{D\prime}(h)\cdot(P_{D\prime}(C|h) - P_{D\prime}(C))\\ &= P_{D\prime}(h)\cdot\left(\frac{P_{D\prime}(C\cap h)}{P_{D\prime}(h)} - 1/2\right) = P_D(h)\cdot\alpha\cdot\left(\frac{\frac{P_D(C\cap h)}{2P_D(C)}}{P_D(h)\cdot\alpha} - 1/2\right)\\ &= P_D(h)\cdot\alpha\cdot\left(\frac{1}{2}\frac{P_D(C\cap h)}{P_D(C)\cdot P_D(h)\cdot\alpha} - 1/2\right)\\ &= \frac{1}{2}P_D(h)\left(\mathbf{Lift}_D(h\to C) - \alpha\right)\end{aligned} \tag{13}$$

Formula (13) can be simplified by rewriting α, exploiting that

$$\mathbf{Lift}_D(h\to\overline{C}) = \frac{1-P_D(C|h)}{P_D(\overline{C})} = \frac{1}{P_D(\overline{C})} - \frac{P_D(C)}{P_D(\overline{C})}\cdot\mathbf{Lift}_D(h\to C) \tag{14}$$

After plugging (14) into α we receive

$$\alpha = 1/2\cdot\left(\mathbf{Lift}_D(h\to C) + \frac{1}{P_D(\overline{C})} - \frac{P_D(C)}{P_D(\overline{C})}\cdot\mathbf{Lift}_D(h\to C)\right)$$

$$= 1/2 \cdot \left(\left(1 - \frac{P_D(C)}{P_D(\overline{C})} \right) \mathbf{Lift}_D(h \to C) + \frac{1}{P_D(\overline{C})} \right)$$
$$= \frac{1}{2P_D(\overline{C})} \cdot \big(\big(P_D(\overline{C}) - P_D(C)\big) \mathbf{Lift}_D(h \to C) + 1 \big)$$
$$= \frac{1}{2P_D(\overline{C})} \cdot ((1 - 2P_D(C)) \mathbf{Lift}_D(h \to C) + 1)$$

which can now be substituted into (13):

$$\frac{1}{2} P_D(h) \cdot (\mathbf{Lift}_D(h \to C) - \alpha)$$
$$= \frac{1}{2} P_D(h) \cdot \left(\mathbf{Lift}_D(h \to C) - \frac{(1 - 2P_D(C)) \mathbf{Lift}_D(h \to C) + 1}{2P_D(\overline{C})} \right)$$
$$= \frac{1}{2} P_D(h) \cdot \left(\mathbf{Lift}_D(h \to C) \left(1 - \frac{1 - 2P_D(C)}{2 - 2P_D(C)} \right) - \frac{1}{2P_D(\overline{C})} \right)$$
$$= \frac{1}{2} P_D(h) \cdot \left(\mathbf{Lift}_D(h \to C) \frac{1}{2 - 2P_D(C)} - \frac{1}{2P_D(\overline{C})} \right)$$
$$= \frac{1}{4P_D(\overline{C})} \cdot P_D(h) \cdot (\mathbf{Lift}_D(h \to C) - 1)$$
$$= \frac{1}{4P_D(\overline{C}) \cdot P_D(C)} \cdot P_D(h) \cdot (P_D(C|h) - P_D(C))$$
$$= \underbrace{\frac{1}{4P_D(\overline{C}) \cdot P_D(C)}}_{\text{irrelevant}} \cdot \mathbf{WRAcc}_D(h \to C) \tag{15}$$

The constant factor on the left hand side does not change the ranking of rulesets. We may drop it and end up with the definition of the **WRAcc** metric for D, which completes the proof of lemma 2.

Putting together formulas (15) and (10) we receive

$$\mathbf{Acc}_{D'}(h \to C) = 2P_{D'}(h) \left(P_{D'}(C|h) - 1/2\right) + P_{D'}(\overline{C})$$
$$= 2P_{D'}(h) \left(P_{D'}(C|h) - P_{D'}(C)\right) + 1/2 = 2\mathbf{WRAcc}_{D'}(h \to C) + 1/2$$
$$= \frac{1}{2P_D(\overline{C}) \cdot P_D(C)} \cdot \mathbf{WRAcc}_D(h \to C) + 1/2,$$

which proves theorem 1.